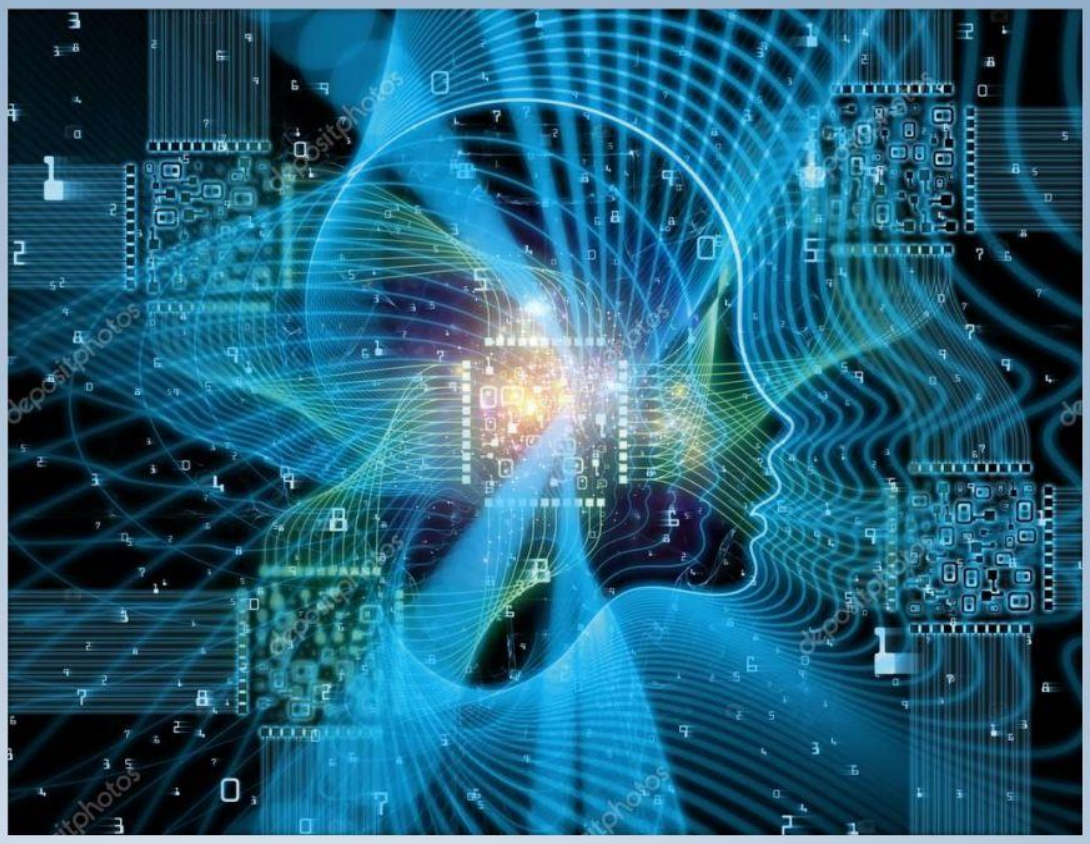
16+



Современное программирование

Материалы IV Международной научно-практической конференции

Нижневартовск, 8 декабря 2021 года

Нижневартовск
2022

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Нижневартовский государственный университет»

СОВРЕМЕННОЕ ПРОГРАММИРОВАНИЕ

Материалы IV Международной научно-практической конференции

Нижневартовск,
08 декабря 2021 года

Под общей редакцией Т.Б. Казиахмедова

Нижневартовск
НВГУ
2022

ББК 32.973я43 **16+**
С 56

Печатается по решению Ученого совета
ФГБОУ ВО «Нижневартовский государственный университет»
(приказ № 236-О от 29.11.2021 г.)

https://doi.org/10.36906/AP-2022

Нучные редакторы:

Туфик Багаутдинович Казиахмедов, канд. пед. наук, доцент, зав. кафедрой информатики и методики преподавания информатики Нижневартовского государственного университета
Александр Александрович Русаков, канд. физ.-мат. наук, д-р пед. наук, президент Академии информатизации образования
Георгий Юрьевич Яламов, канд. физ.-мат. наук, доцент, вед. научн. сотрудник Института управления образованием РАО

Ответственный редактор:

Туфик Багаутдинович Казиахмедов, канд. пед. наук, доцент, зав. кафедрой информатики и методики преподавания информатики Нижневартовского государственного университета

С 56 **Современное программирование:** материалы IV Международной научно-практической конференции (г. Нижневартовск, 08 декабря 2021 года) / отв. ред. Т.Б. Казиахмедов. Нижневартовск: Нижневартовский государственный университет, 2022. 483 с.

ISBN 978–5–00047–631-4

Авторами конференции рассмотрены вопросы эффективности алгоритмов, применяемых при решении различных задач использования параллельных, генетических алгоритмов и нейронных сетей в задачах распознавания, а также методы оптимизации в таких современных направлениях информатики, как телемедицина, искусственный интеллект, большие данные (Big Data), интеллектуальный анализ больших данных, интернет вещей, математические методы и компьютерные технологии для решения экологических проблем современности, компьютерные технологии прогнозирования природных катаклизмов и др. Освещены темы формирования профессиональных компетенций бакалавров и магистров направлений 09.03.01 Информатика и вычислительная техника, 09.04.01 Информатика и вычислительная техника.

Издание адресовано специалистам-практикам, педагогическим работникам, научным сотрудникам, аспирантам и студентам.

ББК 32.973я43

Тип лицензии CC, поддерживаемый журналом: Attribution 4.0 International (CC BY 4.0).

© НВГУ, 2022

ISBN 978-5-00047-631-4
9 785000 476314

РЕЗОЛЮЦИЯ

IV Международная научно-практическая конференция
Современное программирование
(г. Нижневартовск, 08 декабря 2021 года)

Секция 1
Среды, инструменты, методы программирования

Секция «Среды, инструменты, методы программирования» работала на базе кафедры информатики и методики преподавания информатики факультета информационных технологий и математики НВГУ. В связи с ограничениями по присутствию в помещениях из-за рисков распространения новой коронавирусной инфекции работа конференции была организована в онлайн-режиме средствами Skype.

В работе секции приняли участие более 20 человек. Было заслушано 14 докладов. Принимали участие ученые и магистранты разных университетов, такие как: Сибирский федеральный университет (г. Красноярск), Орловский государственный университет имени И.С. Тургенева (г. Орел), Нижневартовский государственный университет, Чувашский государственный университет им. И.Н. Ульянова (г. Чебоксары), Военная академия связи имени Маршала Советского Союза С.М. Буденного (г. Санкт-Петербург), Донской государственный технический университет (г. Ростов-на-Дону), Евразийский национальный университет им. Л.Н. Гумилева (г. Нур-Султан, Казахстан), Кузбасский гуманитарно-педагогический институт Кемеровского государственного университета (г. Новокузнецк).

Широкая география участников подтверждает актуальность темы конференции и рассматриваемых в её рамках вопросов.

Масштаб и актуальность проблем развития информационных технологий в полной мере отразили доклады, признанные наиболее интересными:

1. «Внутренние представления в системе оптимизации функционально-потоковых параллельных программ» Васильев Владимир Сергеевич.

2. «Технология объектно-реляционного отображения в построении современных приложений», Короткий Александр Владимирович.

3. «Обзор протоколов интернета вещей» Королев Руслан Игоревич.

4. «Об эффективности некоторых подходов к решению задачи коммивояжёра» Димитриев Александр Петрович.

5. «Современные инструменты разработки программных приложений» Земцов Алексей Владимирович.

По итогам работы секции предлагается принять следующие решения:

1. Признать актуальность темы, заявленной в конференции и рассматриваемых в рамках секции среды, инструменты, методы программирования.

CC BY

2. Поддержать необходимость проведения регулярных конференций по заявленной тематике с периодичностью не реже 1 раза в год.

3. Признать, что результаты работы секции могут быть применены в сферах образования, промышленности и городском хозяйстве.

4. Считать необходимым продолжение апробации разработанных положений, а также расширение спектра проектных разработок за счет включения новых, не охваченных на данный момент аспектов информационных технологий.

5. Рекомендовать к рассмотрению учебно-методическим отделом возможности формализации учебной нагрузки, представляемой на кафедры.

6. Активизировать использование современных инструментов и методов программирования в реализации различных видов деятельности.

7. Выразить глубокую признательность и объявить благодарность руководителям организаций и их подразделений, делегировавших своих представителей для участия в конференции.

Секция 2–3

Математическое моделирование в сложных системах. Компьютерное моделирование в прикладных науках (физика, химия, биология и др.)

Секция работала на базе факультета информационных технологий и математики НВГУ; для выступления в очной форме было заявлено 17 докладов. Работа конференции была организована в онлайн-режиме средствами Skype.

Большой интерес вызвали доклады Гладких Т.Д., Кожевникова А.Н., Доржиевой Э.С., Полковой Е.В., Баскакова С.А., Харитонова Л.С., по которым состоялись содержательные дискуссии.

В результате открытого диалога участники конференции приняли следующие решения:

1. Рассмотреть возможность формирования секции конференции по математическим пакетам моделирования, с возможностью обмена опытом между участниками.

2. Необходимо в дальнейшем обсудить вопросы получения данных для моделирования с профильными организациями.

3. Необходимо продолжить работу конференции в данном направлении для обмена опытом между учёными из разных прикладных областей.

4. Работу секции одобрить; рекомендовать участникам опубликовать результаты исследований, представленные в докладах, в сборнике материалов конференции.

CC BY

Секция 4–5

Управление разработкой программных комплексов и вычислительными процессами. Интеллектуальные информационные системы. Робототехника и программирование

8 декабря 2021 года в ФГБОУ ВО «Нижневартовский государственный университет» состоялась IV Международная научно-практическая конференция «Современное программирование». В секционных заседаниях 4-5 «Управление разработкой программных комплексов и вычислительными процессами. Интеллектуальные информационные системы. Робототехника и программирование» было заявлено 29 докладов. Работа секций была организована в онлайн режиме средствами Skype.

Проводимая ежегодно конференция имеет своей целью объединение сил международного научного сообщества в обсуждении современных достижений в области программирования.

Одним из итогов работы секции конференции стало принятие ее резолюции.

Участники секции обсудили широкий спектр актуальных научных проблем: отечественный и зарубежный опыт в робототехнике и использовании программного обеспечения; различные аспекты машинного обучения и нейронных сетей; автоматизация бизнес-процессов. Отмечен профессиональный научный уровень представленных докладов.

В результате открытого диалога участниками конференции приняты следующие решения:

1. Значимость современного программирования в развитии экономики не подлежит сомнению. Залогом ее успешного развития являются научные разработки и соответствующее качественное образование. В связи с этим необходимо поддержать решение ректората НВГУ и Оргкомитета конференции о ее проведении.

2. Актуальность представленных к обсуждению результатов научных исследований обусловлена современными тенденциями образовательного процесса и подтверждается значимостью математических и технических наук в социально-экономическом развитии стран.

3. Работу секции одобрить. Продолжить практику публикации статей в изданиях, индексируемых в системе РИНЦ и международных базах данных.

4. Отметить очевидные потребности междисциплинарной интеграции в рамках подготовки бакалавров, инженерных и специальных дисциплин.

5. Отметить значимость участия в научной конференции молодых ученых и учащихся магистратуры. Продолжить обеспечивающую преемственность в науке практику совместного обсуждения вопросов и проблем.

6. Продолжить совершенствование системы информационного обмена и укрепления научных связей по всем актуальным вопросам современного программирования.

CC BY

Секция 6

Вопросы методологии формирования профессиональных компетенций выпускников вузов по IT-направлениям и взаимодействия с общеобразовательными школами

Секция «Вопросы методологии формирования профессиональных компетенций выпускников вузов по IT-направлениям и взаимодействия с общеобразовательными школами» работала на базе кафедры информатики и методики преподавания информатики факультета информационных технологий и математики НВГУ. В связи с ограничениями по присутствию в помещениях из-за рисков распространения новой коронавирусной инфекции работа конференции была организована в онлайн-режиме средствами ZOOM.

В работе секции приняли участие более 25 человек. Было заслушано 21 докладов. Принимали участие ученые и магистранты разных университетов, такие как:

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, ГБОУВ РК КИПУ имени Февзи Якубова, Бухарский государственный университет, Колледж ЭБП КУК, Казахстан, Нижневартовский государственный университет, Московский государственный технический университет гражданской авиации, Приднестровский государственный университет им. Т.Г. Шевченко, Магнитогорский государственный технический университет им. Г.И.Носова, ФГБОУ ВО «Казанский государственный энергетический университет», Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российский государственный педагогический университет им. А. И. Герцена»,ФГБОУ ВО «Тихоокеанский государственный университет», Евразийский национальный университет им Л.Н.Гумилева, Дагестанский государственный университет народного хозяйства, Тюменский Индустриальный Университет, Тюменский индустриальный университет филиал в Нижневартовске

Широкая география участников подтверждает актуальность темы конференции и рассматриваемых в её рамках вопросов.

Масштаб и актуальность проблем развития информационных технологий в полной мере отразили доклады, признанные наиболее интересными:

1. «Вариант взаимодействия выпускников вузов по IT-направлениям со школами в процессе цифровой трансформации образования» Миронова Людмила Ивановна.

2. «Междисциплинарный подход при изучении типов и структур данных в языках программирования» Симурзина Екатерина Александровна.

3. «О содержательном и технологическом аспекте преподавания дисциплины высшая математика» Басов Вадим Александрович.

4. «Формирование профессиональных компетенций студентов образовательных организаций СПО на основе организации самостоятельной работы и производственных практик» Алексеева Виктория Николаевна.

5. «Процесс создания фильтра дополненной реальности на примере «Spark AR studio» для продвижения компании в социальных сетях» Махмутова Марина Владимировна.

CC BY

6. «Исследовательская и проектная активности студентов в цифровых продуктах учебной деятельности: цели и ценности в развитии информационной компетенции» Табачук Наталья Петровна.

7. «Система поддержки принятия решения для формирования профессиональных компетенций выпускников вузов по it-направлениям» Валеева Диана Рустамовна.

По итогам работы секции предлагается принять следующие решения:

1. Признать актуальность темы, заявленной в конференции и рассматриваемых в рамках секции «Вопросы методологии формирования профессиональных компетенций выпускников вузов по IT-направлениям и взаимодействия с общеобразовательными школами».

2. Поддержать необходимость проведения регулярных конференций по заявленной тематике с периодичностью не реже 1 раза в год именно в декабре месяца.

3. Признать, что результаты работы секции могут быть применены в сферах образования, промышленности и городском хозяйстве.

4. Считать необходимым продолжение апробации разработанных положений, а также расширение спектра проектных разработок за счет включения новых, не охваченных на данный момент аспектов информационных технологий.

5. Активизировать использование современных инструментов и методов программирования в реализации различных видов деятельности.

6. Выразить глубокую признательность и объявить благодарность руководителям организаций и их подразделений, делегировавших своих представителей для участия в конференции.

Председатель организационного комитета,
Заведующий кафедрой Информатики
и методики преподавания информатики Т.Б. Казиахмедов

08 декабря 2021 г.

CC BY

СРЕДЫ, ИНСТРУМЕНТЫ, МЕТОДЫ ПРОГРАММИРОВАНИЯ

УДК 004.4'23
https://doi.org/10.36906/AP-2022/01

Васильев В.С.
ORCID: 00000-0002-3340-6678
Непомнящий О.В.
ORCID: 0000-0002-2459-6414, канд. техн. наук
Сибирский федеральный университет
г. Красноярск, Россия

ВНУТРЕННИЕ ПРЕДСТАВЛЕНИЯ В СИСТЕМЕ ОПТИМИЗАЦИИ ФУНКЦИОНАЛЬНО-ПОТОКОВЫХ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ

Аннотация. Рассмотрена проблема создания инструментальных средств оптимизации параллельных программ. Определено, что задача оптимизации кода имеет высокую вычислительную сложность и поэтому чаще всего выполняется не над исходным кодом программы, а над его промежуточными представлениями. Показано, что при оптимизации функционально-потоковых параллельных программ данная задача обладает рядом специфических особенностей. В частности, явным выделением в виде графов информационных и управляющих зависимостей для каждой функции. Предложены структуры данных, обеспечивающих повышение эффективности системы автоматической оптимизации кода с учётом особенностей функционально-потоковой парадигмы параллельного программирования. Приведены результаты анализа форм внутреннего представления программ, обеспечивающих выполнение ключевых функций при оптимизации кода.

Ключевые слова: оптимизация кода; параллельное программирование; алгоритм; функционально-потоковая парадигма; граф.

CC BY

Vasilev V.S.
ORCID: 00000-0002-3340-6678
Nepomnuashy O.V.
ORCID: 0000-0002-2459-6414, Ph.D.
Siberian Federal University
Krasnoyarsk, Russia

INTERNAL REPRESENTATIONS IN THE OPTIMIZATION SYSTEM OF FUNCTIONAL DATAFLOW PARALLEL PROGRAMS

Abstract. The problem of creating tools for optimizing parallel programs is considered. It has been determined that the problem of optimizing the code has a high computational complexity and, therefore, most often it is performed not over the source code of the program, but over its intermediate representations. It is shown that when optimizing dataflow parallel programs, this problem has a number of specific features. In particular, an explicit assignment in the form of graphs of information and control dependencies for each function. Data structures are proposed that provide an increase in the efficiency of the automatic code optimization system taking into account the features of the dataflow paradigm of parallel programming. The results of the analysis of the forms of the internal representation of programs that ensure the execution of key functions in the optimization of the code are presented.

Keywords: code optimization; parallel programming; algorithm; dataflow programming; graph.

На любом языке программирования может быть написана не оптимальная программа, то есть программа, характеристики которой можно улучшить. Оптимизация представляет собой процесс эквивалентного преобразования, в результате которого улучшаются требуемые характеристики. Часть характеристик может быть улучшена с помощью автоматизированных программных средств - оптимизаторов, которые выполняют определённый объем работ. Это позволяет снизить временные затраты на разработку.

Оптимизирующее преобразование программы сводится к удалению, перемещению или замене её фрагментов. Перед выполнением преобразования система оптимизации определяет в автоматическом режиме допустимость преобразования того или иного фрагмента, а также, осуществляет предварительную оценку эффективности полученных результатов [1, с. 705].

Для обеспечения требуемого быстродействия оптимизатора эффективные решения могут быть найдены при работе не непосредственно над исходным кодом программы, а над его промежуточными представлениями. В данном контексте рассмотрим выбор формы внутреннего представления функционально-потоковой [2, с. 71] программы в системе оптимизации кода. Существует ряд широко известных форм промежуточных представлений подобных программ, отражающих различные типы зависимостей [3, с. 90]. Функционально потоковая парадигма параллельного программирования (ФПП) обладает специфическими

CC BY

особенностями [4], которые необходимо учитывать при оптимизации кода, следовательно, с их учётом должны формироваться внутренние представления программ.

В ФПП парадигме программа представляет собой набор графов – реверсивного информационного (РИГ) и управляющего (УГ) [4, с. 18]. Узлы РИГ задают данные и операции над данными, его дуги отражают зависимости по данным между узлами. Узлы УГ соответствуют узлам РИГ, а его дуги задают зависимости по управлению, то есть управляют порядком выполнения программы. Исходный код каждой функции является формой представления РИГ, то есть явно задаёт информационные зависимости. Управляющий граф строится по информационному с учётом особенностей вычислительной архитектуры, на которой планируется исполнение программы.

Динамика выполнения операторов ФПП программы задаётся механизмом продвижения сигналов готовности по дугам УГ. Некоторые вершины, например константы, являются изначально готовыми и сразу формируют сигнал готовности на своём выходе.

Разные виды оптимизаций требуют учёта различных зависимостей в программе. В связи с этим во время оптимизации одновременно может использоваться несколько промежуточных представлений [5, с. 47]. Известен ряд широко используемых форм представления программ:

– граф зависимостей по данным (Data Dependence Graph, DDG) отражает информационные зависимости в программе, он эквивалентен РИГ;

– граф зависимостей по управлению (Control-flow graph, CFG) отражает управляющие зависимости, эквивалентен УГ;

– форма единственного статического присваивания (Static single assignment form, SSA). В данном случае выстраиваются зависимости между состояниями (версиями) переменных, при этом на каждое присваивание значения переменной создаётся новая версия. В ФПП языках отсутствует оператор разрушающего присваивания, поэтому любая программа изначально находится в SSA форме;

– цепочки определение-использование (def-use chains) и граф программных зависимостей (Program dependence graph, PDG), отражающий def-use, use-def и def-def зависимости. Так, если оператор X использует результат работы оператора Y, то в PDG должна существовать дуга X→Y типа def-use и обратная дуга типа use-def.

Выбор необходимой формы представления зависит от особенностей конкретного алгоритма анализа и преобразования. Однако, значительную часть преобразований можно выполнить на небольшом наборе форм представления программы. В связи с этим, для выбора внутренней формы представления функционально-потоковой программы для оптимизатора, необходимо:

– выполнить анализ особенностей ФПП парадигмы, влияющих на процесс оптимизации кода, по результатам которого разработать форму представления программы;

– с учётом операций, часто выполняемых в ходе оптимизации, разработать способ реализации этой формы представления в памяти ЭВМ.

Анализ операций ФПП языков показал, что эффективность их выполнения должна обеспечить структура внутреннего представления. Поэтому необходимо организовать

CC BY

хранение use-def и def-use зависимостей. Такие зависимости задаются графом зависимостей по данным. При этом для ФПП характерно отсутствие def-def зависимостей поскольку в этих языках отсутствует оператор разрушающего присваивания.

Граф может задаваться отношениями смежности или инцидентности. При использовании отношений инцидентности явно выделяются дуги графа. Такой способ может быть применён если дуги графа несут какую-либо дополнительную информацию.

Любое представление графа описывает наборы вершин и дуг, которые могут задаваться в памяти ЭВМ с помощью массивов (матриц) или связных списков. Матричное представление не эффективно в случаях динамического изменения графа процессе обработки. Кроме того, если граф не является сильно связным, то в матрицах будет храниться значительное количество нулей.

Таким образом, наиболее подходящей формой представления графа в системе оптимизации кода являются списки смежности или инцидентности.

В графе зависимостей по данным выделяется два типа дуг, поэтому для каждой вершины необходимо хранить раздельные списки, задающие def-use и use-def зависимости.

При задании информационного графа программы в виде списков смежности, узел графа может иметь следующую структуру:

Node = inode(type, id, value, delay, Inputs, Outputs), где Inputs и Ouputs идентифицируют узлы, от которых текущий узел принимает данные или является их источником. Элемент такого списка соответствует дуге графа. Он может представлять собой либо пару (type, id), однозначно задающую узел, либо ссылку на узел (адрес узла). Рассмотрим оба варианта.

В первом случае граф состоит из N узлов и M дуг. Так как обычно число дуг, инцидентных вершине много меньше M, обозначим его за K (K<< M). При задании графа списками инцидентности и хранении в структуре дуги ссылки на инцидентные ей вершины, асимптотические оценки сложности будут такими же. Для получения вершины потребуется лишь один лишний переход по ссылке, имеющий вычислительную сложность O(1).

При оценке трудоёмкости операции удаления узла, подразумевалось, что необходимо удалить также все входящие и исходящие дуги (не только в текущем, но и в смежных узлах). Под полным обходом графа подразумевается его обход от аргумента функции до возвратной вершины. Обходу графа на L уровней в ширину соответствует, например, поиск всех узлов, зависимых по данным от некоторой вершины.

При добавлении узла в граф, ему должен быть присвоен номер, для эффективной организации отдельно сохраняется наибольший номер узла. Тогда: номера узлов графа, формируемого в результате оптимизации, должны идти строго друг за другом. Вычислительная сложность операции приведения номеров в корректное состояние после удаления узла составляет O(N). Выполнять её после каждого удаления узла нецелесообразно, поэтому данная операция выполняется после завершения всех оптимизаций.

Узлы графа могут быть упорядочены по ключу (type, id) или любому другому полю, такой метод позволит улучшить асимптотические оценки вычислительной сложности.

CC BY

Узлы графа помещаются в хэш-таблицу, это позволяет улучшить оценку трудоёмкости алгоритмов в среднем случае, так как поиск, вставка и удаление выполняются за O(1), однако в худшем случае (большом числе коллизий) оценка этих операций O(N).

Хранение узлов в дереве поиска или хэш таблице позволяет более эффективно выполнять поиск, но лишь по некоторым полям. На самом деле, поиск по паре (type, id) не требуется ни в одном из рассмотренных алгоритмов. Таким образом, из структуры узла, используемой внутри оптимизатора, изначально убираются номера узлов и добавляются они после проведения всех оптимизирующих преобразований.

Хранение узлов упорядоченными по значению или номеру задержанного списка не изменят асимптотическую оценку, так как эти значения многократно повторяются для различных узлов графа. Можно сказать, что в графе из N узлов может находиться порядка N узлов со значением «(---)» (узел формирования списка данных). При этом хэш таблица или дерево поиска усложняет структуру внутреннего представления графа и даст некоторые дополнительные расходы.

Наиболее подходящим для реализации из рассмотренных вариантов является хранение информационного графа в виде матрицы смежности, связи в которой задаются ссылками, а список вершин не упорядочен. Из структуры узла на время оптимизации целесообразно убрать номер (id), таким образом узлы информационного графа во внутреннем представлении оптимизатора могут задаваться следующими структурами: inode(type, value, delay, DefUseReferences, UseDefReferences).

Схематично структура внутреннего представления информационного графа показана на рисунке 1, при этом сплошные линии отражают зависимости между узлами функции (def-use и use-def), а штриховые линии – зависимости между узлами списка. Зависимости def-use и use-def хранятся в списках, при этом для def-use зависимостей позиция ссылки в списке определяет номер входа (на приведённой схеме соответствующие номера подписаны на дугах).

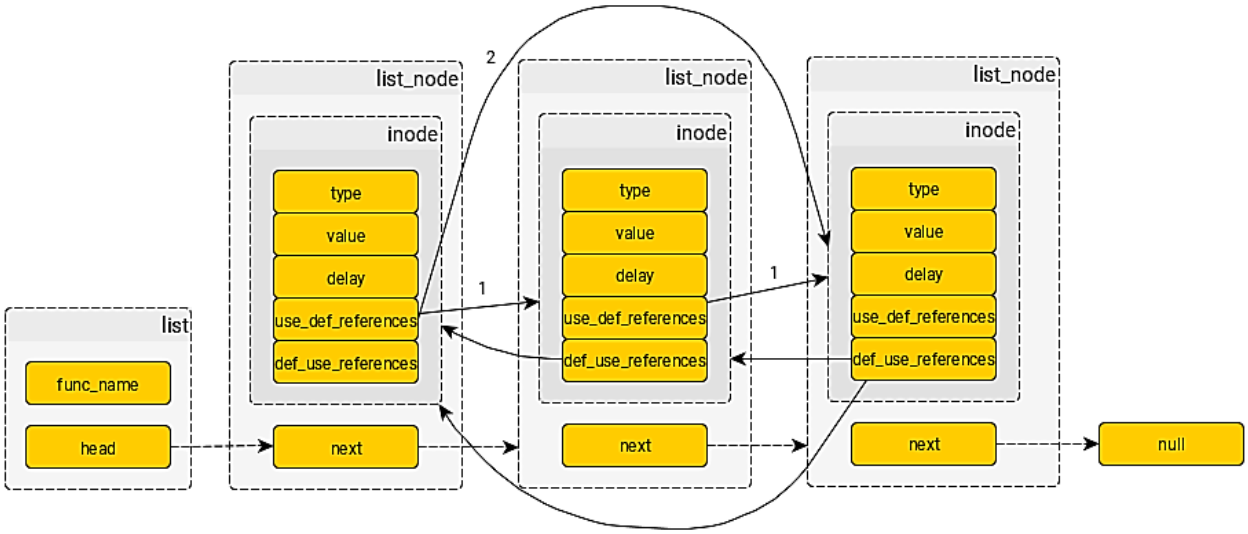


Рис. 1. Схема внутреннего представления информационного графа программы

Дуга РИГ, помимо ссылок на узлы источника и приёмника данных, несёт информацию о номере информационного входа. Этот номер является свойством дуги, а не узла так как один

и тот же узел графа может быть источником данных для нескольких других узлов. Во внутреннем представлении эта информация кодируется в виде индекса ссылки на узел в use_def_references для каждого узла. Поэтому use-def зависимости узла хранятся в массиве, а def-use зависимости - в связном списке.

Для управляющего графа справедливы все рассуждения, приведённые при анализе вариантов реализации информационного графа. Однако, узел управляющего графа «привязывается» к узлу информационного по идентификатору узла. В связи с тем, что оптимизацию РИГ и УГ можно проводить раздельно, а в процессе оптимизации УГ этот идентификатор используется только для генерации корректного выходного представления графа, хранить его целесообразно в виде целого числа.

Для представления УГ в оптимизаторе используются списки инцидентности, так как необходимо разделять зависимости узлов УГ и сигналов.

Так как номер информационного входа явно хранится в структуре дуги, то нет необходимости кодировать его индексом массива, а значит use-def зависимости и def-use можно хранить в связных списках.

Схематично структура внутреннего представления управляющего графа показана на рисунке 2. Жирными линиями обозначены зависимости от сигналов.

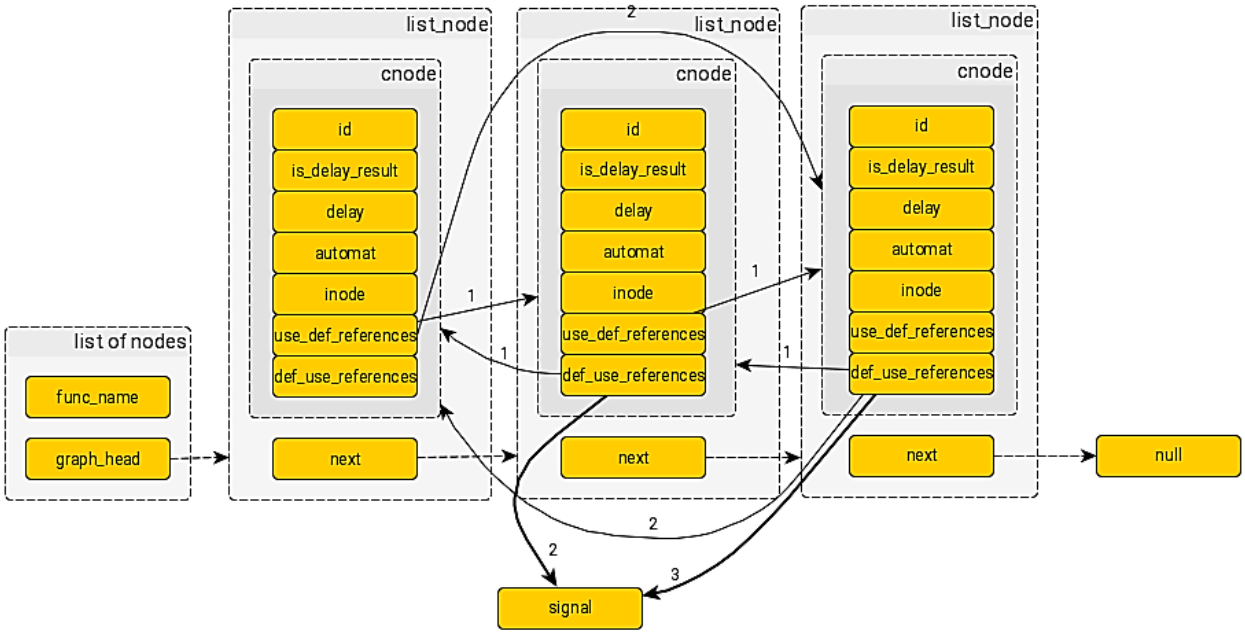


Рис. 2. Схема внутреннего представления управляющего графа программы

Показано, что в качестве основного внутреннего представления функционально-потоковой параллельной программы при оптимизации целесообразно хранить графы программных зависимостей, при этом отдельно хранить информационные и управляющие зависимости.

Для обеспечения наилучшего быстродействия системы оптимизации, информационные зависимости следует задавать в памяти ЭВМ в виде списков смежности, а управляющие – в виде списков инцидентности.

CC BY

Описанные формы представления программ являются основными не только в системе оптимизации функционально-потоковых программ, но и системе преобразования таких программ в императивную форму [6, с. 198]. Эти формы, без использования вспомогательных структур данных, позволяют эффективно реализовать ряд методов оптимизации, например, оптимизации как обнаружение и замена хвостовой рекурсии циклом, удаление неиспользуемого кода и вынос инварианта из повторяющихся вычислений.

Литература

1. Ахо А.В., Лам М.С, Сети Р., Ульман Дж.Д. Компиляторы: принципы, технологии и инструментарий. М.: Издательский дом «Вильямс», 2008. 1184 с.

2. Легалов А.И. Функциональный язык для создания архитектурно-независимых параллельных программ // Вычислительные технологии. 2005. Т. 10. № 1. С. 71-89.

3. Штейнберг Б.Я., Макошенко Д.В., Черданцев Д.Н., Шульженко А.М. Внутреннее представление в Открытой распараллеливающей системе // Искусственный интеллект. 2003. № 4. С. 89-97.

4. Legalov A.I., Vasilyev V.S., Matkovskii I.V., Ushakova M.S. (2018) A Toolkit for the Development of Data-Driven Functional Parallel Programmes // Parallel Computational Technologies. PCT 2018. Communications in Computer and Information Science. 2018. Vol. 910. Springer, Cham. https://doi.org/10.1007/978-3-319-99673-8_2

5. Click C., Paleczny M. A simple graph-based intermediate representation // ACM Sigplan Notices. 1995. Vol. 30. Issue 3. Pp. 35-49.

6. Васильев В.С., Легалов А.И, Зыков С.В., Трансформация функционально-потоковых параллельных программ в императивные // Моделирование и анализ информационных систем. 2021. Т. 28. № 2. С. 198-214. https://doi.org/10.18255/1818-1015-2021-2-198-214

© Васильев В.С., Непомнящий О.В., 2022

УДК 004.415.2
https://doi.org/10.36906/AP-2022/02

Горелик А.А., Овчинников А.В., Алимбеков Д.Р.
Огрызков А.В.
ORCID: 0000-0001-6072-2716
Оренбургский государственный университет
г. Оренбург, Россия

ИСПОЛЬЗОВАНИЕ СРЕДСТВ И ТЕХНОЛОГИЙ МОБИЛЬНОЙ РАЗРАБОТКИ ДЛЯ СОЗДАНИЯ НАВИГАЦИОННОГО ПРИЛОЖЕНИЯ

Аннотация. В статье рассмотрены возможности среды разработки Android Studio для создания мобильного приложения. Приводятся достоинства и недостатки фреймворка Flutter. Описывается процесс разработки мобильного приложения, работающего с онлайн-картами. Это приложение подсказывает велосипедисту, как доехать до ближайшей точки продажи разливной воды.

Ключевые слова: мобильное приложение; Flutter; геопозиция; онлайн-карта.

Gorelik A.A., Ovchinnikov A.V., Alimbekov D.R.
Ogryzkov A.V.
ORCID: 0000-0001-6072-2716
Orenburg State University
Orenburg, Russia

USAGE OF INSTRUMENTS AND TECHNOLOGIES OF MOBILE APP DEVELOPMENT FOR GIS APPLICATIONS

Abstract. In this article, we look at functionality of development environment Android Studio for mobile application development. We look at advantages and disadvantages of a Flutter framework. A process of a development of a specific mobile geo app is described. This app guides cyclist to the nearest water station.

Keywords: mobile application; Flutter; geolocation; gis.

Мобильные приложения играют важную роль в современном мире. Многие уже не могут себе представить жизнь без них. Мобильное приложение всегда с собой и может помочь во многих ситуациях. В данной статье описан процесс разработки программы, которая показывает маршрут до ближайшей точки продажи разливной воды (водяной башни). Приложение предназначено для велосипедиста, у которого закончилась вода в личной бутылке.

CC BY

В качестве среды разработки использовалась Android Studio [1]. Из запомнившихся отличительных особенностей среды стоит выделить такие:

• эмулятор различных телефонов, удобный для тестирования на компьютере;

• возможность подключать физическое устройство через usb для тестирования и отладки приложения;

• консоль с логами запуска и работы приложения, удобная для отладки и выявления ошибок;

• запуск приложения в эмуляторе в горячем режиме, позволяющий экономить время при запуске и тестировании на устройстве.

Также одним из участников проекта использовалась среда IntelliJ IDEA, которая является прародительницей Android Studio и имеет свои преимущества и недостатки:

• поддерживает три дополнительных языка Kotlin, Scala и Groovy;

• поддерживает разработку не только под Android;

• не имеет совместимости с некоторыми плагинами для Android-разработки, например, Andriod APK Support;

•есть платная версия.

Для разработки были выбраны язык Dart и фреймворк Flutter [2], как наиболее передовые и современные. В качестве альтернативы рассматривались более старые Kotlin, React Native на JavaScript, Xamarin Studio для C#, оказавшийся платным и весьма дорогим.

Приведём некоторые черты Flutter, которые склонили выбор в его пользу:

• Flutter компилирует весь код Dart в нативный Java/Swift код, поэтому он демонстрирует лучшую производительность по сравнению с React Native;

• поддержка кроссплатформенности – готовое приложение можно запустить, как на Android телефоне, так и на iPhone. Это позволяет снизить время разработки в два раза, хотя приложение, написанное для iPhone на Swift, будет работать быстрее;

• Dart и Flutter созданы в Google, разрабатывающей и операционную систему Android;

• одна из особенностей языка Dart – заимствование синтаксиса из известных и популярных языков программирования, что позволяет писать код на нём практически сразу.

К недостаткам Flutter стоит отнести:

• не такое обширное сообщество разработчиков, как на React Native, и поэтому меньшее количество освещённых в интернете возникающих при разработке проблем.

В качестве архитектуры приложения выбрана одна из разновидностей архитектуры «модель-представление-контроллер» – BLoC (Business Logic Component), которая отделяет бизнес-логику приложения от пользовательского интерфейса. BLoC работает с асинхронными потоками событий и данных. Каждый блок использует потоки состояний и событий, которые позволяют выстроить взаимодействие с UI и данными.

Для разработки и адаптации графических элементов и логотипа использовались следующие программы: онлайн сервис FlutterIcons, Adobe Photoshop, Adobe Illustrator. Современный графический дизайн в мобильной разработке не придерживается

CC BY

скевоморфизма (имитация реальных теней, бликов, объёма объектов), поэтому дизайн выполнялся в стиле «flat»: плоские, ненагруженные формы, ограниченное количество цветов.

Координаты расположения водяных башен было решено хранить не в приложении, а на удалённом сервере, чтобы иметь возможность их добавления и удаления. Программная часть сервера написана на языке php, данные хранятся на MySQL сервере. Общение приложения с сервером реализовано посредством GET запросов и JSON ответов, дополнительно применены авторизация по ключу и шифрование.

Для упрощения работы с базой данных на сервере была использована библиотека PHP классов «Database», использующая расширение PHP mysqli и имитацию подготовленных запросов: https://github.com/Vasiliy-Makogon/Database/. Преимущества библиотека таковы:

- избавляет от многословности: вместо 3 (и более) строк кода для исполнения одного запроса пишется 1 строка;
- надёжная защита от SQL-инъекций: самостоятельно экранируются все параметры, идущие в тело запроса, согласно указанному типу заполнителей;
- работает без замещения функциональности mysqli адаптера, а дополняет его;
- в отличие от класса-прослойки PHP DataObjects (PDO) используемая библиотека позволяет просматривать полноценно сформированный SQL-запрос, что очень удобно для поиска ошибок в нём.

Для работы навигационного приложения необходимы карты с улицами и зданиями. Карты Яндекс и Google распространяются на коммерческой основе, поэтому в качестве альтернативы были выбраны карты Open Street Map, которые весьма точны благодаря использованию принципа «вики», когда любой человек может внести уточняющие изменения в карту.

Для работы приложения нужно не только отображать карты, но и строить маршруты передвижения. Для этого тоже можно использовать, как коммерческие сервисы от Google, Яндекс, Microsoft, в которых цена за построение маршрута колеблется от 20 до 38 копеек, так и условно бесплатные. Был выбран сервис graphhopper.com, который предоставляет 500 бесплатных построений маршрута в день для некоммерческих проектов.

Опишем логику проекта со стороны пользователя. При запуске приложения пользователь видит заставку, во время которой происходит загрузка всех необходимых компонентов программы и подгрузка информации с сервера. Следующий экран – непосредственно отвечает за главный функционал приложения:

а) показать, где пользователь находится на карте,

б) показать, где относительно пользователя находятся ближайшие три башни по продаже воды,

в) показать маршрут до ближайшей башни в двух видах: по следованию автомобильных дорог и по прямой. Маршрут по прямой нужен, так как часто велосипедист знает, как срезать путь, не придерживаясь строго автомобильных дорог, и для этого ему важнее направление движения, которое и показывает маршрут по прямой.

CC BY

На этом же главном экране есть кнопки: приблизить, отдалить, простроить маршрут до следующей ближайшей точки на экране, кнопка возвращения к собственному местоположению, кнопка меню. При нажатии на изображение водяной башни открывается окно с рейтингом вкуса и названием фирмы-поставщика воды.

Кнопка меню открывает дополнительное окно сбоку, в котором содержится:

а) добавление новой водяной башни пользователем,

б) отображать или не отображать прямой маршрут (по умолчанию включён).

По мере приближения к выбранной башне пройдённый участок маршрута исчезает. По мере изменения расстояния между велосипедистом и башней изменяется масштаб карты – происходит приближение. Когда пользователь уже находится в нескольких метрах от водяной башни, появляется окно, которое одновременно служит трём целям:

а) желанию пользователя закрыть приложение: закрытие приложения происходит автоматически, когда пользователь ставит оценку вкуса. Это сокращает количество физических действий пользователя для закрытия приложения в два раза;

б) получить от пользователя оценку вкуса воды в данной башне;

в) в случае отсутствия башни или её нерабочего состояния, сообщить об этом.

Благодаря обратной связи с пользователем (добавление башен и оповещение о неработающих) на сервере осуществляется корректировка базы данных силами сообщества велосипедистов, то есть разработчики следуют принципу «вики».

Опишем логику работы приложения со стороны разработчика. При запуске приложения опрашивается геопозиция пользователя. Затем проверяется есть ли у приложения данные, сохранённые в кэше. В кэше могут храниться координаты и характеристики воды (вкус, фирма-поставщик), которые были загружены с сервера при прошлых запусках программы.

Кэш реализован на быстрейшей технологии баз данных для Flutter – Hive. По сравнению с SQLite и S.Prefs, Hive работает с нулевой задержкой благодаря отсутствию проверок хранимых данных на корректность. Для работы Hive требуется указать формат, в котором будет храниться информация, в противном случае информация для хранения должна быть приведена к поддерживаемым простым типам данных.

Геопозициии точек продажи разливной воды нескольких фирм, работающих на территории России, были получены путём парсинга общедоступных карт. Парсинг осуществлялся средствами языка Python при помощи библиотеки BeautifulSoup 4.

Карты Open Street Maps подключаются к виджету Flutter Map при инициализации слоя с изображениями карты. Эти квадраты, из которых складывается карта, называются tiles (tile). Формат запроса к API Open Street Maps для получения нужного тайла:

https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png, где s – буква сервера, z – масштаб тайла, x и y – широта и долгота тайла на карте Земли.

Взаимодействие непосредственно с API Open Street Map происходит в недрах библиотеки Flutter Map, поэтому после инициализации карты её подгрузка и отрисовка происходят автоматически.

CC BY

Если в кэше найдены подходящие по удалённости от велосипедиста башни, то запрос к серверу не происходит и подходящие башни отображаются на экране.

Если кэш пуст или подходящих башен нет, то на сервер передаётся get-запрос, содержащий геопозицию велосипедиста и радиус поиска башен. В ответ получаем JSON, содержащий координаты, рейтинг вкуса, названия фирм-поставщиков воды для каждой башни.

Формат JSON для точек:

```
{ "status" : 200,
  "points" : [
        {       "latitude" :<широта>,
                "longitude" :<долгота>,
                "rating" :<рейтинг>,
                "name" :<название фирмы>…
```

Для доступа к API сервера необходим ключ, который заранее сгенерирован для приложения и который указывается во всех запросах к серверу. При отсутствии правильного ключа в запросе сервер возвращает ошибку 403. Шифрование данных происходит на уровне стандартных криптографических протоколов TLS/SSL.

Полученные точки сохраняются в кэше. Затем рассчитывается расстояние по прямой до ближайших точек в небольшом радиусе от велосипедиста через формулу гаверсинуса. Выбираются три ближайшие точки, которые выводятся на экран. Так как мы ограничены количеством простраиваемых маршрутов (500 в день), то маршрут запрашивается у graphhopper.com только до одной, ближайшей по прямой, точки. Простраивание маршрута до альтернативной водяной башни (второй и третьей на экране) производится по запросу пользователя – нажатием на кнопку со стрелочкой “следующий маршрут”.

В процессе разработки была проблема с передачей данных внутри программы. По своему поведению это напомнило невозможность прямого обращения к данным, находящимся в разных окнах в Windows Forms C#. Так и здесь мы не могли обратиться к нужным данным из любого места. Чтобы преодолеть эту проблему, были использованы Blocs, что в дальнейшем привело к другой проблеме, связанной с асинхронным способом их работы.

Логика работы конкретного блока прописывается в файле описания блока; состояния, которые может принимать поток, в файле состояний блока; события, передаваемые блоку, прописываются в файле событий блока. Можно напрямую вмешиваться в state, минуя event. На изменения в потоке можно подписаться и реагировать из любой части программы при условии подключения блоков по ссылке.

В main создаются экземпляры блоков:

```
MapGetPointBlocpointsBloc = MapGetPointBloc();
```

Потом необходимо предоставить доступ к данным блока всем желающим, потоки блоков помещаются в context:

```
BlocProvider<MapGetPointBloc>(create: (context) {
        return pointsBloc..add(MapGetPointsEvent());
```

CC BY

В нужном месте в другом файле создаются блоки, являющиеся ссылками на уже созданные, данные извлекаются из context:

```
pointsBloc = BlocProvider.of<MapGetPointBloc>(context);
```

Кроме сквозной передачи данных, блоки обеспечивают разделение бизнес-логики от дизайнерской части. Главенствующую роль в динамическом отображении контента в приложениях, созданных на Flutter, играют виджеты (Widgets). Виджеты – ядро механизма макетов Flutter. Изображения, значки, текст, невидимые элементы (строки, сетки, столбцы), которые упорядочивают видимые виджеты, тоже описываются в качестве виджетов. На практике разработчик создает макет путем их компоновки. На основном экране элементы располагаются в макете сеточной верстки («gridlayout»). С помощью элементов типа BlocBuilder, вложенных друг в друга, в виджеты подгружаются данные из блоков:

```
Widgetbuild (BuildContextcontext) {
RouteBlocrouteBloc = BlocProvider.of<RouteBloc>(context);
return BlocBuilder<MapGetPointBloc, List<WaterPoint>>(
    builder: (c1, points) {
     return BlocBuilder<RouteBloc, AllRouteInfo>(
        builder: (c2, route) {  …
routeBloc.add(ChangeRouteEvent(_geoRepository.currPos));
points.add(WaterPoint(succAndLatLng.newPoint, '000', 100, false));
```

Сюда же вложен виджет карты, использующий данные из блоков в отображаемых слоях:

```
return FlutterMap(….
/*видимые слои*/      layers: [
/*слой карты*/  tileLayerOptions (“https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png”)
/*слой с отрисованным маршрутом:*/
polyOptions(c3, route, line, numberOfLayer, onOffStraightRoute),
/*слой с водяными башнями:*/
markerOptions(c3, points, numberOfLayer, succAndLatLng)]
```

Дизайн для всех элементов нижней панели был создан вручную. Растровая графика создавалась с помощью графического планшета в Photoshop, векторная графика – в Adobe Illustrator. Чтобы Flutter поддерживал созданную графику, был использован сервис-конвертер FlutterIcon, генерирующий набор значков и иконок в формат специальном формате ttf. Для второстепенных частей дизайна использовалась встроенная библиотека «MaterialDesign».

Для загрузки готового приложения в Play Market используется онлайн сервис Google Play Console. Бесконечный доступ к нему стоит около 2 тысяч рублей и был нами получен. К сожалению, релиз-доступ к магазину приложений на iOS распространяется по подписке и стоит около 6 тысяч рублей ежегодно. Поэтому в рамках некоммерческого проекта мы ограничились публикацией на GooglePlay Market. Отметим, что Play Console используется не только для релиза приложений, но и для их тестирования, поэтому сюда загружаются тестовые версии, не имеющие явных ошибок компиляции. Здесь же имеется возможность

CC BY

разграничения доступа между пользователями, приглашение тестировщиков для поиска ошибок в загружаемых приложениях.

Таким образом, силами студенческой команды из трёх человек, было создано полноценное навигационное приложение для Android и iOS. Приложение позволяет велосипедисту, у которого закончилась вода в личной бутылочке, построить маршрут до ближайшей точки продажи разливной воды.

Литература

1. Филлипс Б., Стюарт К., Марсикано К., Гарднер Б. Android. Программирование для профессионалов. СПб.: Питер, 2021. 704 с.

2. Заметти Ф. Flutter на практике: Прокачиваем навыки мобильной разработки с помощью открытого фреймворка от Google. М.: ДМК Пресс, 2020. 328 с.

© Горелик А.А., Овчинников А.В., Алимбеков Д.Р., Огрызков А.В., 2022

УДК 519.168
https://doi.org/10.36906/AP-2022/03

Димитриев А.П.
ORCID: 0000-0002-7345-9790, канд. техн. наук
Чувашский государственный университет им. И.Н. Ульянова
г. Чебоксары, Россия

ОБ ЭФФЕКТИВНОСТИ НЕКОТОРЫХ ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ КОММИВОЯЖЁРА

Аннотация. Исследуются возможности программного решения задачи коммивояжёра без использования библиотеки Concorde, которая включает более 700 функций. Предложен ряд способов сокращения маршрута: устранение пересечений, локальная оптимизация методом ближайшего соседа и др. Разработан комплекс программ, реализующий эти способы и выводящий графическое изображение маршрута. Используются общедоступные файлы с данными о городах в формате TSPLIB. Полученный маршрут для 423 городов на 5% длиннее оптимального маршрута.

Ключевые слова: задача коммивояжёра; глобальный оптимум; Concorde.

Dimitriev A.P.
ORCID: 0000-0002-7345-9790, Ph.D.
Chuvash State University named after I.N. Ulyanov
Cheboksary, Russia

ON THE EFFICIENCY OF SOME APPROACHES TO SOLUTION OF THE TRAVELING SALESMAN'S PROBLEM

Abstract. Possibilities of software solution of the traveling salesman problem without using the Concorde library, which includes more than 700 functions, are investigated. A number of ways to shorten the route are proposed: elimination of intersections, local optimization by the nearest neighbor method, etc. A complex of programs has been developed that implements these methods and displays a graphical image of the route. Publicly available city data files in TSPLIB format are used. The resulting route for 423 cities is 5% longer than the optimal route.

Keywords: traveling salesman problem; global optimum; Concorde.

Задача коммивояжёра была сформулирована в 1934 году [3, с. 61] и заключается в том, что заданы расстояния между городами, и требуется посетить каждый город ровно по одному разу и вернуться в начало маршрута, так чтобы общая длина маршрута была минимальной. Эта задача стала широко известной. В частности, по адресу cs.rhul.ac.uk/home/zvero/GTSPLIB/ находится известная библиотека тестовых экземпляров для этой задачи, GTSPLIB, доступных

CC BY

для свободной загрузки [4, с. 269]. По адресу math.uwaterloo.ca/tsp/vlsi/index.html существует похожая библиотека VLSI Data Sets. По адресу https://clck.ru/bNFFq существует web-сервис NEOS, который позволяет за короткое время решать такие задачи с пользовательскими данными в формате TSPLIB и пользовательскими настройками, например, используя алгоритм Concorde либо эвристику Lin-Kernighan [5, с. 500]. Так, задача коммивояжёра для данных в файле pbn423.tsp из библиотеки VLSI Data Sets с помощью данного web-сервиса решается за 1,58 с; при этом находится глобальный оптимум. В этом файле представлены 423 «города», координаты которых – целые числа в декартовой плоскости, сгенерированные компьютером.

Однако время вычислений не всегда такое короткое. Помимо вышеназванных библиотек, в домене факультета математики университета Ватерлоо существует сайт math.uwaterloo.ca/tsp/world/index.html, где находится файл world.tsp с 1904711 городами мира, для которого время от времени учёными по всему миру изредка, иногда раз в несколько лет, находятся лучшие маршруты, чем ранее. На сайте math.uwaterloo.ca/tsp/index.html существуют и другие файлы, например, файл nl57912.xy, сообщающий координаты «широта-долгота» 57912 национальных памятников в Голландии. Нахождение оптимального маршрута для этого файла с помощью метода ветвей и границ заняло у исследователей университета Ватерлоо (Канада) совместно с исследователями из других стран в общей сложности 96,9 лет компьютерного времени. Поэтому разработка более быстрых алгоритмов нахождения глобального оптимума для задачи коммивояжёра остаётся актуальной.

Библиотека Concorde включает более 700 функций и свободно доступна по адресу https://clck.ru/bNFGT для использования в академических исследованиях. Общий размер файлов версии 2003 г. 4,2 Мбайт; имеются разные версии. Однако некоторые особенности решения задачи с помощью функций этой библиотеки сразу не очевидны.

Целью работы является исследование возможностей решения задачи коммивояжёра без использования библиотеки Concorde, то есть, как можно решать эту задачу очевидными способами, разрабатывая программные средства.

Для достижения цели работы разработан комплекс программ на Delphi, чтобы гарантировать невозможность импорта из библиотеки Concorde, так как та использует язык C++, а Delphi – Object Pascal. Программа при компиляции с максимальным числом городов 100000 при работе в Windows 7 Professional x64 использует 17,8 Мбайт оперативной памяти. Возможна компиляция и для 2 млн городов, однако время работы в таком случае становится неприемлемо длительным, а памяти требуется больше, поэтому установлено максимальное число городов 100000. Текст программы содержит 917 строк. Объем пакета программ с исходным кодом 1 Мб.

Реализуемыми приёмами решения рассматриваемой задачи являются следующие.

1. Составление первоначального маршрута с помощью метода ближайшего соседа.
2. Использование алгоритма оптимизации последовательности отбора.
3. Локальная оптимизация методом Монте-Карло.
4. Устранение «бахромы».

CC BY

5. Циклический сдвиг маршрута на задаваемое число городов.

6. Аннексия «полуостровов».

7. Устранение пресечений.

8. Локальная оптимизация методом ближайшего соседа.

Рассмотрим применение перечисленных приёмов.

1. Составление первоначального маршрута с помощью метода ближайшего соседа

Наиболее очевидно прокладывать маршрут, каждый раз выбирая в качестве последующего города тот, который ближе всех остальных к текущему городу. Получаемое решение далеко от оптимального. Так, для pbn423.tsp оно составляет 1702–1817, что на 20–25% больше оптимального, равного 1365. Время работы этого способа при 423 городах около одной секунды, однако для файла world.tsp оценивается приблизительно как сутки, поскольку временная сложность этого метода $O(n^2)$, где n – число городов (здесь и далее используется один процессор персонального компьютера). Когда это время невелико, можно начинать построение не только с первого города, а с разных городов. Число таких городов задаётся пользователем программы. Из результатов выбирается наилучший. Однако и в этом случае решение далеко от оптимального.

2. Использование алгоритма оптимизации последовательности отбора

Этот приём используется после составления первоначального маршрута методом ближайшего соседа и был эффективен, когда городов было 40 [2]. Суть метода состоит в применении метода ближайшего соседа, однако каждый раз выбирается не самый близкий сосед, а некоторый по порядку из наиболее близких. При этом оптимизируется вектор выбора. Для файла pbn423.tsp в результате длина маршрута составляет обычно 1648 и более, т. е. на 17% больше оптимального. Пользователь может задать различные параметры, оказывающие влияние на время вычислений и значение результата. Этот метод улучшает результат метода ближайшего соседа, но недостаточно. Поэтому задаются параметры, при которых время вычислений не превышает нескольких секунд, и затем к полученному маршруту применяются последующие приёмы.

3. Локальная оптимизация методом Монте-Карло

После применения второго приёма многократно случайным образом выбирается отрезок маршрута с задаваемым числом городов, и производятся многочисленные перестановки городов на этом участке. При улучшении результат фиксируется. Количества перестановок и отрезков задаются и оказывают влияние на время вычислений. Обычно эти параметры задаются так, чтобы время составляло до 1 минуты. Эффективность такой оптимизации незначительна (единицы для pbn423.tsp), но все же существует. Этот метод можно использовать несколько раз подряд, пока результат не перестанет улучшаться.

4. Устранение «бахромы»

Под «бахромой» понимаются города, образующие углы, например, менее 45 градусов в маршруте, причём величину угла задаёт пользователь. Если соседние по маршруту два города для такого города далеко, а вблизи расположены два других города, то такой город помещается в маршруте между двумя более близкими городами. Например, на рисунке 1,

CC BY

а) показана часть маршрута, построенного с помощью разработанной программы, и выделена тройка соседствующих городов, образующих «бахрому». Программа поместила центральный город в этой тройке в маршруте между более близкими к нему городами, как выделено на рисунке 1, б). Этот метод иногда уменьшает длину маршрута, а иногда увеличивает, но ненамного. Время работы данного приёма незначительно, так как его временная сложность $O(n^2)$: для каждого города просматривается весь маршрут для поиска близких городов. Словосочетание «устранение бахромы» встречается в области распознавания образов [1, с. 67] и по смыслу несколько сходно с использованием в данной статье (при распознавании укорачиваются линии, а здесь – далеко выступающие части маршрута).

5. Циклический сдвиг маршрута на задаваемое число городов

Это действие само по себе не приводит к изменению длины маршрута. Однако некоторые другие используемые приёмы не учитывают цикличность маршрута для упрощения программного кода, и крайние несколько городов в маршруте ими не затрагиваются. Применение данного действия позволяет при повторном использовании других приёмов иногда получать незначительное уменьшение длины маршрута. В программе номера городов не изменяются; изменяются только значения указателей в соответствующем массиве.

6. Аннексия «полуостровов»

Некоторые участки маршрута могут выглядеть как полуострова. Если рядом с таким «полуостровом» расположены два последовательных города, то можно попытаться соединить его с этими двумя городами и так сократить маршрут. Города, от которых отсоединился «полуостров», соединяются между собой, а к которым присоединился, напротив, друг от друга отсоединяются. Если это укорачивает маршрут, изменение фиксируется. Например, «полуостров», выделенный овалом на рисунке 1, в), аннексировался вправо, как выделено овалом на рисунке 1, г). Этот метод нередко приводит к небольшому укорочению маршрута и может применяться несколько раз подряд с различными значениями параметра, называемого «длина перешейка». Время работы этого приема для nl57912.xy около 1 минуты, а для pbn423.tsp незначительно.

7. Устранение пресечений

В любом выпуклом четырёхугольнике сумма длин диагоналей больше сумм длин любых двух противоположных сторон. Пересечение диагоналей представляет собой самопересечение в маршруте, поэтому от последнего нужно избавляться. Такой приём несколько укорачивает маршрут, если в нем были пересечения. Например, на рисунке 1, г) имеются самопересечения маршрута, выделенные кругами. На рисунке 1, д) они устранены. Время работы этого метода для nl57912.xy около 1 минуты, а для pbn423.tsp незначительно.

CC BY

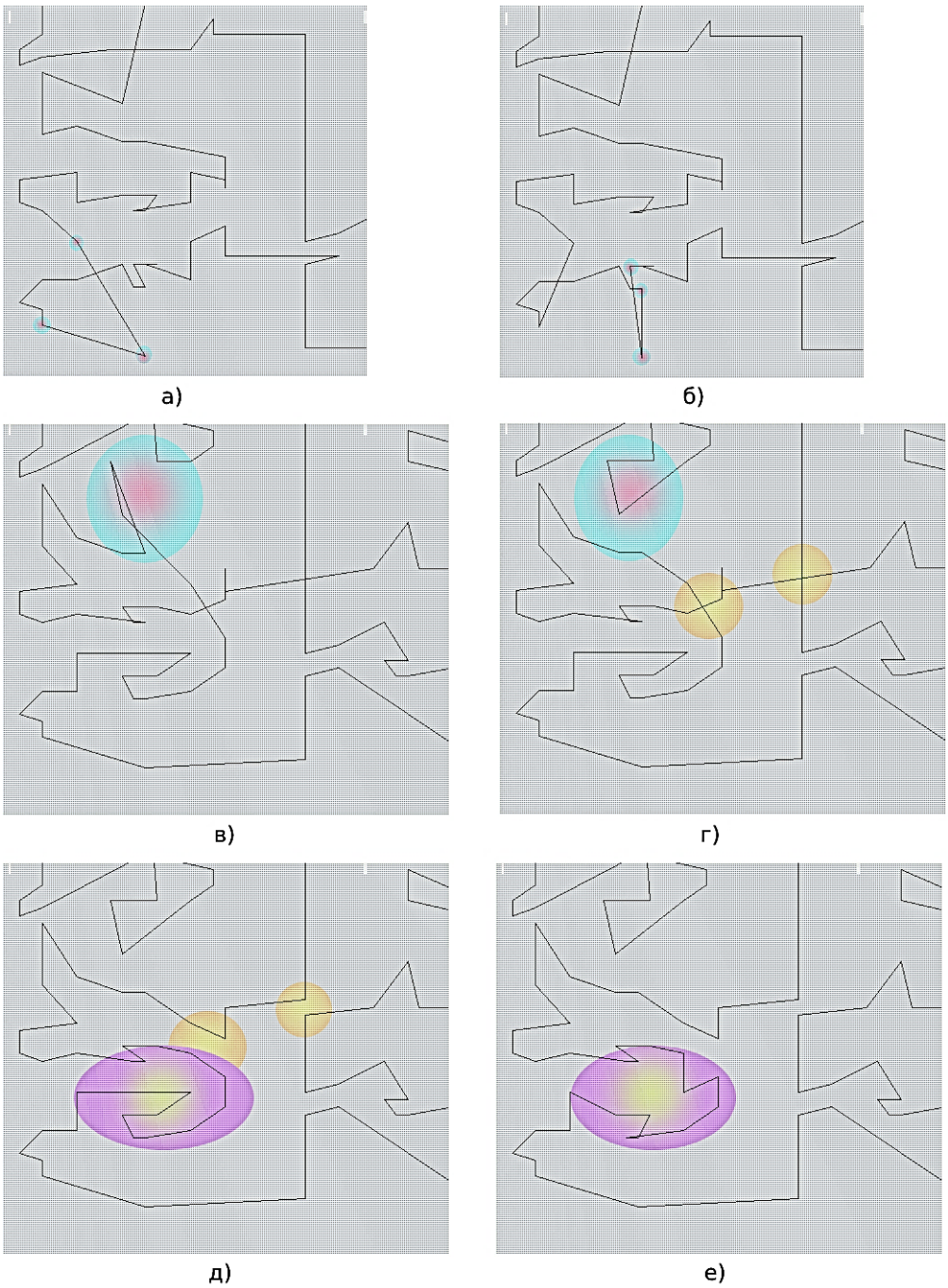


Рис. 1. Часть маршрута для файла pbn423.tsp:
а), б) – до и после устранения «бахромы»;
в), г) – до и после аннексии «полуострова»;
д) – после устранения пересечений;
е) – после локальной оптимизации методом ближайшего соседа

8. Локальная оптимизация методом ближайшего соседа

Приём напоминает приём 3, но в, отличие от него, рассматривает все отрезки заданной длины и вместо хаотичных перестановок строит локальный маршрут для данного отрезка аналогично первому методу. Например, на рисунке 1, е) представлен результат такой оптимизации, применённой к маршруту на рисунке 1, д) Как видно, часть маршрута, выделенная овалом, приобрела другой вид. Время работы этого метода приемлемо как для pbn423.tsp, так и для nl57912.xy. Приём нередко незначительно улучшает маршрут.

CC BY

Совокупность всех перечисленных приёмов, чередуемых по усмотрению пользователя с помощью нажатий на соответствующие кнопки, для файла pbn423.tsp приводит к маршрутам длиной от 1423 до 1456 за несколько минут. Это в среднем на 5% хуже, чем для оптимального маршрута, а время работы примерно в 200 раз дольше, чем при использовании Concorde.

На рисунке 2 изображено наложение с небольшим сдвигом итогового маршрута, построенного с помощью разработанного программного комплекса (зелёный цвет), и оптимального маршрута, данные которого получены по адресу math.uwaterloo.ca/tsp/vlsi/pbn423.tour.html (розовый цвет). Сравнение этих двух маршрутов показывает, что некоторые их участки совпадают, но многие не совпадают. Более никакой информации из визуального сравнения этих двух рисунков не получено, поэтому других очевидных приёмов оптимизации не найдено.

Рис. 2. Наложение экранных копий маршрута для файла pbn423.tsp длиной 1423 после применения всех приёмов и оптимального маршрута

Необходимо признать, что использование только очевидных приёмов решения задачи коммивояжёра является менее эффективным по сравнению с возможностями, предоставляемыми библиотекой Concorde.

Литература

1. Бутаков Е.А., Островский В.И., Фадеев И.Л. Обработка изображений на ЭВМ. М.: Радио и связь, 1987. 240 с.

2. Димитриев А.П. Модификация алгоритма оптимизации последовательности отбора для решения задачи коммивояжёра // Современные проблемы науки и образования. 2015. № 2-1. http://www.science-education.ru/122-20628

3. Flood M.M. The Traveling-Salesman Problem // Operations Research. 1956. Vol. 4. Issue 1. Pp. 61-75. https://doi.org/10.1287/opre.4.1.61

CC BY

4. Helsgaun K. Solving the equality generalized traveling salesman problem using the Lin-Kernighan-Helsgaun algorithm // Mathematical Programming Computation. 2015. Vol. 7. Issue 3. Pp. 269-287. https://doi.org/10.1007/s12532-015-0080-8

5. Lin S., Kernighan B.W. An Effective Heuristic Algorithm for the Traveling Salesman Problem // Operations Research. 1973. Vol. 21. Issue 2. Pp. 498-516. https://doi.org/10.1287/opre.21.2.498

© Димитриев А.П., 2022

УДК 004.4'2
https://doi.org/10.36906/AP-2022/04

Земцов А.В.
Богомолов А.В.
канд. пед. наук
Чувашский государственный педагогический университет им. И.Я. Яковлева
г. Чебоксары, Россия

СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ

Аннотация. Данная статья содержит основную информацию о прикладных и системных инструментах разработки программных приложений. Проведён предварительный анализ об увеличении производительности IT-подразделений в зависимости от финансирования современных инструментов разработки программных приложений.

Ключевые слова: разработка; программные приложения; программные инструменты; производительность труда; IT.

Zemtsov A.V.
Bogomolov A.V.
Ph.D.
Chuvash State Pedagogical University named after I.Y. Yakovlev
Cheboksary, Russia

MODERN SOFTWARE APPLICATION DEVELOPMENT TOOLS

Abstract. This article contains basic information about application and system tools for developing software applications. A preliminary analysis was carried out on the increase in the productivity of IT departments depending on the financing of modern software application development tools.

Keywords: development; software applications; software tools; labor productivity; IT.

Данная тема, на сегодняшний день весьма актуальна. Разработка программных приложений стремительно развивается и программная среда, инструменты, методы тоже. И это не с проста. Для коммерческих организацией одной из приоритетных задач является увеличение производительности сотрудников организации. Совершенствование инструментов разработки программного обеспечения является ключевым решением.

Но, прежде чем рассматривать данную тему, необходимо вникнуть в саму интерпретацию такого понятия как «инструменты разработки программных приложений» с

CC BY

точки зрения различных авторов. Мы изучили много научных публикаций на эту тему, но так и не нашли подходящую трактовку данной дефиниции.

С нашей точки зрения «инструменты разработки программных приложений» это набор системных и прикладных средств, которые упрощают разработку программных приложений.

Такие системные и прикладные средства делятся на следующие виды:

1. Среда разработки. Это самый главный инструмент для разработки программных приложений. Среда разработки позволяет программисту не только писать код, но и выполнять множество различных функций по форматированию кода. К таким функциям можно отнести поиск, замену, выделение, глубокий анализ кода, умное автодополнение кода, языковые вставки, диаграммы, рефакторинг в масштабе всего проекта.

Немаловажным фактором является отладка кода в реальном времени. Порой нет возможности или времени заниматься установкой программного приложения в соответствующее окружение. Данная функция является отличным решением для этого.

Среди универсальных вариантов среды разработки можно выделить следующие: Geany, NetBeans, CodeBlocks, Eclipse, Qt Creator. У них есть свои плюсы и минусы. Для выбора оптимальной среды разработки нужно руководствоваться именно необходимыми характеристиками программного средства.

2. Система контроля версий. В процессе разработки зачастую приходится возвращаться на «прошлый» этап разработки. Это обусловлено разными причинами, но в силу роста проекта, в силу роста количества сотрудников, в силу роста сложности проекта система контроля версий превращается в необходимость.

Когда у вас код умещается в 100–200 строк, возможно, вам и не понадобится такой инструмент. Но когда кода тысячи и тысячи строк, плюс над ним работают несколько программистов, то без этого инструмента не обойтись.

Самые известные системы контроля версий: Git, Subversion, Mercurial. Перечисленные системы контроля версий работают, в основном на компьютерах программиста, но есть и исключения. К ним можно отнести: GitHub, Bitbucket. Они устанавливаются на удаленном сервере [2].

3. Редактор интерфейсов. Для многих GUI разработчиков это является одним из самых важных инструментов разработки программных приложений. Суть функционала заключается в отказе от написание html или xml файлов. Теперь этим занимается редактор интерфейсов. Разработчику нужно лишь перетащить необходимые элементы на форму.

Когда проект разрастается с геометрической прогрессией, редактор интерфейсов превращается в необходимость. В более продвинутых средах разработки уже доступны отдельные плагины, редактор интерфейсов уже встроен туда.

4. Редактор баз данных. Взаимодействие с СУБД, как правило, ставит новых сотрудников организации в не ловкое положение. Без полноты знаний о написании SQL скриптов, трудно заниматься полноценной разработкой программных приложений. Редакторы баз данных облегчают этот процесс. Их основная задача это помощь в управлении данными программных приложений [1].

CC BY

5. Инструменты программиста для тестирования ПО. Это не неотъемлемый пункт который многие авторы не упоминают, хотя он очень важен. Тестирование помогает понять, выполняет ли Ваша программа тот функционал, для которого она создавалась. Об этом можно почитать подробнее в TDD подходе к разработке программных приложений [3].

6. Фреймворки. На сегодняшний день, среди работодателей требуются не столь глубокие знания языков программирования, нежели знания фрейморков. Они представляют из себя набор готовых решений для программирования приложений. Фреймворки позволяют создавать программные приложения как с нуля, так и подключить их к готовому решению.

Как упоминалось ранее, тема перспективная. Многие компании проводят исследования по поводу влияния современных инструментов разработки на производительность сотрудников. За основу практической части нашего исследования мы брали данные (https://clck.ru/eRfye) представленные на Diasoft Special Conf 6 октября 2021 г. Данные исследования приведены в таблице.

Таблица

Увеличение производительности IT подразделений в зависимости
от финансирования развития современных инструментов разработки программных приложений

	2019 год	2020 год	2021 год	2022 год
Увеличение бюджета на использование современной среды разработки	1 млн. руб.	3 млн. руб.	5 млн. руб.	8 млн. руб.
Разработка собственного редактора интерфесов	0 млн. руб.	0,3 млн. руб.	0,5 млн. руб.	0,8 млн. руб.
Совершенствование инструментов программиста для тестирования ПО	8 млн. руб.	10 млн. руб.	12 млн. руб.	10 млн. руб.
Разработка собственных фреймворков	13 млн. руб.	15 млн. руб.	19 млн. руб.	16 млн. руб.
Количество трудозатрат на разработку 1 программного компонента	80 чел. / часов	70 чел. / часов	60 чел. / часов	40 чел. / часов

Результаты исследования, проведённого компанией Diasoft, отражают прямую взаимосвязь между увеличением производительности IT подразделений и финансированием развития современных инструментов разработки программных приложений. Один из спикеров конференции, главный архитектор Сергей Ломака отметил, что финансировать современные технологии не достаточно. Нужно оптимизировать существующие производственные процессы и организовывать работу сотрудников.

На основе результатов исследования компании Diasoft удалось сократить количество трудозатрат на разработку в 4 раза. На это повлияли покупка современных инструментов разработки, такие как PhpStorm, Microsoft Visual Studio, PyCharm, IntelliJ IDEA. Также стоит отметить строку с разработкой собственных фреймворков. На текущий момент в компании разработаны шаблоны по созданию программных компонентов, генераторы кода по техническому заданию, различны инструменты и утилиты по работе с базами данных.

В заключении хочется отметить, что на совершенствование инструментов разработки программного обеспечения влияют множество факторов. К таким факторам относятся: увеличение бюджета на использование современной среды разработки, разработка

CC BY

собственного редактора интерфейсов, совершенствование инструментов программиста для тестирования ПО, разработка собственных фреймворков. Но ключевым фактором всегда остаётся организация работы в компании и оптимизация существующих производственных процессов.

Литература

1. Земцов А.В. Анализ оптимального выбора системы управления базами данных для осуществления гибкого функционирования современных веб-приложений // Актуальные проблемы прикладной и школьной информатики: сб. научных статей. Чебоксары, 2020. С. 311-316.

2. Земцов А.В. Современные системы непрерывной интеграции веб-приложений // Актуальные проблемы прикладной и школьной информатики: сб. научных статей. Чебоксары, 2020. С. 316-321.

3. Земцов А.В., Богомолов А.В. Современные аспекты микросервисной архитектуры с целью реконструкции современного будущего // Интернет-технологии в образовании: сб. материалов Всероссийской научно-практической конференции (г. Чебоксары, 01-30 мая 2020 года). Чебоксары, 2020. С. 24-31.

© Земцов А.В., Богомолов А.В., 2022

УДК 004.051
https://doi.org/10.36906/AP-2022/05

Ибраев Т.Б.
ORCID: 0000-0002-0008-0739
Касымова А.Х.
ORCID: 0000-0002-4614-4021, канд. пед. наук
Западно-Казахстанский аграрно-технический университет им. Жангир хана
г. Уральск, Казахстан

АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ НАГРУЗКИ ПРЕПОДАВАТЕЛЯ

Аннотация. В статье определена суть понятия информационной системы, которая используется университетом в современных условиях. На основании теоретического материала и практических наработок современных исследователей в данной сфере авторами проанализировано создание качественной архитектуры информационной системы оптимального распределения нагрузки преподавателя. Совместное использование и управление отличающихся по содержанию и форме состовляющих информационного ресурса систем управления являетя важным вопросом, требующим внедрения различных подходов и инструментов к его решению.

Ключевые слова: информационная система; информационные технологии; архитектура информационной системы (ИС); клиент-сервер.

Ibraev T.B.
ORCID: 0000-0002-0008-0739
Kasymova A.Kh.
ORCID: 0000-0002-4614-4021, Ph.D.
West Kazakhstan Agrarian Technical University named after Zhangir Khan
Uralsk, Kazakhstan

ARCHITECTURE OF THE INFORMATION SYSTEM FOR THE OPTIMAL DISTRIBUTION OF THE TEACHER LOAD

Abstract. The article defines the essence of the concept of an information system, which is used by the university in modern conditions. On the basis of theoretical material and practical developments of modern researchers in this area, the author analyzed the creation of a high-quality architecture of an information system for optimal distribution of the teacher's workload. Joint use and management of control systems that differ in content and form, constituting the information resource is an urgent issue that requires the introduction of various approaches and tools to its solution.

Keywords: information system; information Technology; IS architecture; client-server.

CC BY

Разработка информационной системы предполагает прохождение ряда определенных процедур. Их конкретный состав зависит от того, какую задачу предстоит решить. Потребности заказчика определяют структуру ИС, вид операций, выполняемых с информацией: внесение, хранение, пересылка, корректировка. Также процесс работы над ИС определяется технологиями, применяемыми при разработке структуры и ее практической реализации [1].

Если необходимо разработать программное обеспечение вне зависимости от его характера и назначения, на первом этапе необходимо разработать его архитектуру, определиться со структурой. В рамках данной работы предположим, что оптимальной будет архитектура типа «клиент-сервер». На рисунке приведена модель архитектуры информационной системы, разрабатываемой в рамках данной работы.

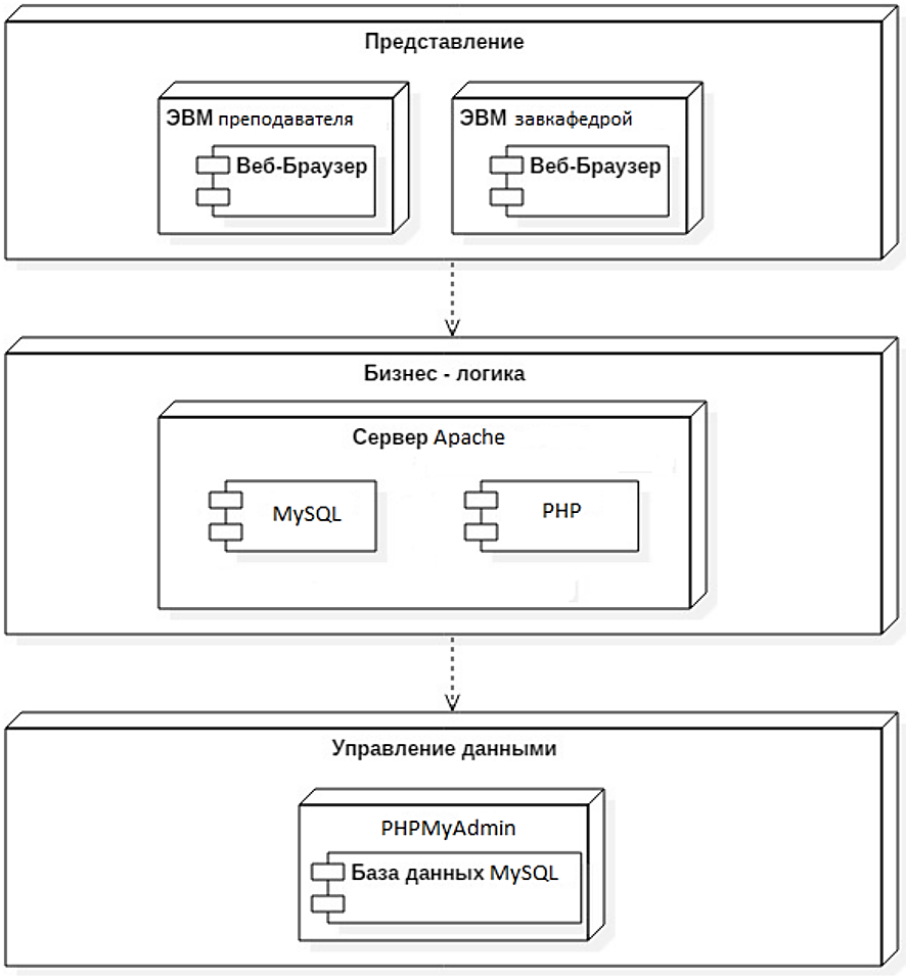


Рис. Архитектурное представление информационной системы

Необходимо объяснить, почему выбран именно этот тип архитектуры. В аудиториях учебного заведения работают компьютеры, отличающиеся объемом памяти и производительностью. Соответственно, рациональным решением будет перенос основной нагрузки в части работы вычислений, обработки запросов от пользователей на сервер.

Современные разработчики в работе над ПО могут использовать самые разные инструменты, позволяющие автоматизировать написание приложений или отдельные модули [2]. И эти инструменты характеризуются различными параметрами, которые нужно учитывать при выборе.

На данный момент времени разработчикам ПО доступны инструменты, при помощи которых можно:

• разрабатывать интерфейс. С их помощью пользователь формирует запросы к ИС;

• активировать необходимые процессы и передавать им функции управления при необходимости;

• разрабатывать БД и средства доступа к ним;

• модернизировать уже имеющиеся программные комплексы. Для этого нужно проанализировать и устранить возможные ошибки.

Современные средства разработки ПО обеспечивают пользователю поддержку следующих технологий и инструментов [3; 4]:

• объектно-ориентированное программирование;

• CASE-технологии. Их используют для разработки модели и проекта отдельных модулей информационных систем и баз данных с последующей поддержкой;

• алгоритмы реляционной алгебры. Их задействуют, если необходимо организовать управление базами данных реляционного типа;

• синхронизация работы отдельных элементов информационной системы или иного программного обеспечения.

Указанные ранее инструменты и возможности реализованы в современных языках программирования, например: PHP, Ruby, Java и т. д.

Разработчик программного обеспечения выбирает для работы определенные инструменты – программные средства. При их выборе нужно исходить из следующих факторов:

• процессы разработки занимают минимальное время;

• есть возможность оперативно вносить изменения в создаваемое программное обеспечение;

• есть возможность редактировать содержание баз данных с помощью инструментов интерфейса.

Есть еще несколько критериев, которые нужно учитывать при выборе средств для разработки программного обеспечения:

• наличие необходимых знаний для работы с конкретным ПО;

• минимальные требования к объему памяти, иным аппаратным ресурсам;

• возможность использовать ПО на разных платформах;

• предоставление пользователю разнообразных инструментов при работе с интерфейсом;

• возможность использовать самые разные средства при разработке ПО;

• практичность. Например, на обработку запроса или задачи должно тратиться минимум времени;

CC BY

• наличие средств и инструментов для выявления и устранения критических ошибок;

• возможность разрабатывать программный продукт с минимальными временными затратами.

Чтобы выбрать оптимальные программные средства, необходимые для разработки ИС, использован метод вариантных обоснований. Он позволяет выбрать лучший вариант из нескольких. Для реализации данного метода на практике необходимо выполнить следующие этапы:

– Выбор критериев, на основании которых будет проходить сравнение. Критерии распределяются в соответствии с их приоритетом.

– Анализ всех имеющихся вариантов. Для каждого нужно определить, присущи ли ему обозначенные критерии.

– Для всех выбранных вариантов проводится подсчет выставленных оценок.

– Лучшим признается вариант, получивший максимальное количество положительных оценок.

В рамках данной работы возможные программные средства оценивались на основе критериев, перечисленных ранее. При этом за основу были взяты требования разработчика, а не статистика, находящаяся в общем доступе. Результаты анализа сведены в таблицу и график.

Таблица

Результаты сравнительного анализа для языков программирования

№	Характеристика средств разработки	PHP	Ruby	Java
1.	Низкое потребление аппаратных ресурсов	12,5%	12,5%	0%
2.	Мультиплатформенность	12,5%	12,5%	12,5%
3.	Показательность при разработке интерфейса	0%	0%	12,5%
4.	Широкий спектр средств для разработки баз данных	12,5%	12,5%	12,5%
5.	Минимальное время обработки задач	0%	0%	12,5%
6.	Наличие средств отладки критических ошибок	12,5%	12,5%	12,5%
7.	Минимальное время разработки программного обеспечения	12,5%	0%	0%
8.	Эргономичность	12,5%	0%	0%

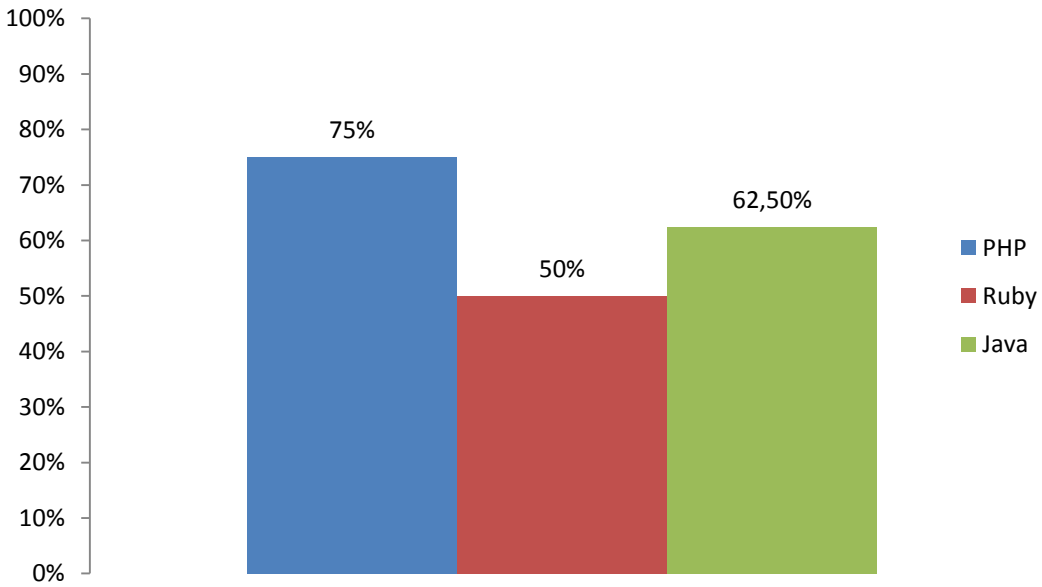


График. Результат сравнительного анализа

Далее можно исследовать инструменты и решения, применяемые для управления БД. Но для начала необходимо решить, как БД будет организована.

Информационная база может быть одного из следующих типов:

o Локальная БД. Этот способ предполагает создание отдельной базы данных для каждой решаемой задачи или совокупности задач.

o o Интегрированная БД. Такая база данных – неотъемлемый компонент определенного приложения. Данные можно хранить на одном носителе. Но при этом в каждой базе можно хранить данные, относящиеся только к конкретному приложению.

Помимо этого необходимо отметить, что базы данных делятся на локальные и распределенные. Файлы локальной БД хранят на одном сервере или рабочей станции. Для хранения файлов распределенной БД можно использовать, например, все компьютеры локальной вычислительной сети.

В рамках данной работы планируется работать с интегрированной БД.

БД в зависимости от способа организации связей между данными может иметь один из следующих типов логической структуры:

• иерархическая;

• сетевая;

• реляционная.

Иерархическая модель предполагает, что каждый модуль (сегмент) системы (за исключением корневого) имеет один исходный элемент. Между корневым и порожденными элементами устанавливается связь. И она может быть только единственной. Но порожденный элемент при этом может быть связан с определенным количеством элементов, находящихся на более низком уровне. Такую структуру можно назвать «один ко многим». Если нужно найти какой-либо элемент иерархии, необходимо двигаться от корневого элемента к элементам следующих уровней. Но пользователь не может пропустить элементы одного из уровней.

Базы данных иерархического типа имеют ряд существенных недостатков:

• сложно реализовать связи типа «многие ко многим»;

• если база состоит из множества уровней, на поиск нужного элемента может потребоваться значительное время;

• запросы к подобной БД должны иметь определенную структуру.

Со временем иерархические БД были доработаны. Их продолжением стали сетевые БД. Их можно изобразить в виде графа. Основное отличие от иерархической БД заключается в том, что порожденный элемент может соединяться с неограниченным количеством элементов более высокого уровня.

БД сетевого типа также имеют ряд недостатков:

• схема БД получается достаточно сложной;

• сложно организовать доступ к элементам БД.

Базы данных реляционного типа можно представить в виде совокупности двумерных таблиц.

CC BY

Строки таблиц получили название «записи», столбцы – «поля». На пересечении записей и полей размещаются некоторые данные. Для получения доступа к ним нужно указать строку и столбец.

Реляционные базы данных имеют как преимущества, так и недостатки. К первым можно отнести то, что на получение нужных данных тратится больше времени в сравнении с БД иерархического или сетевого типов. К тому же создание подобной базы – достаточно трудоемкий процесс.

К плюсам реляционных БД можно отнести следующие моменты:

о Простота восприятия данных.

о Наличие теоретического обоснования. Все операции, выполняемые над данными, можно описать при помощи существующего математического аппарата.

о Независимость данных. Если необходимо скорректировать данные в таблицах, не приходится менять непосредственно информационную систему.

В данной работе для разработки ИС решено использовать БД реляционного типа. Выбор в пользу данной модели сделан из-за ее простоты и распространенности в сравнении с иными существующими моделями.

Модель для базы данных выбрана. Теперь осталось определиться со способом управления БД (СУБД).

Разработчики ИС для управления базами данных часто используют MySQL. В частности, эту систему используют при разработке веб-сайтов. К преимуществам СУБД можно отнести то, что для работы потребляется минимум аппаратных ресурсов. Еще один плюс – отказоустойчивость системы.

MySQL – не единственная доступная СУБД. Если предстоит разработка веб-приложения, можно использовать возможности таких СУБД, как Oracle, Microsoft SQL Server, PostgreSQL. Чтобы выбрать лучший вариант, необходимо сравнить все имеющиеся, провести сравнительный анализ, как это было сделано ранее для языков программирования. Но сначала нужно определиться с критериями, по которым будет проводиться сравнение. На основании полученных данных предстоит составить таблицу.

Оценивать СУБД предстоит оценивать в соответствии с критериями, перечисленными ниже:

- наличие у пользователя необходимых знаний для работы с СУБД;
- минимальное потребление ресурсов;
- возможность работать на разных платформах;
- функционал пользовательского интерфейса;
- время, затрачиваемое на обработку пользовательского запроса;
- возможность выявлять и устранять критические ошибки;
- время, необходимое на проектирование БД;
- эргономичность СУБД.

В рамках данной работы проанализированы возможности трех языков программирования. Для каждого выставлены оценки по ряду факторов. Далее оценки

CC BY

просуммированы. На основании этих данных сделан вывод, что разработка информационной системы будет вестись с использованием возможностей языка программирования PHP. Он содержит инструменты, необходимые для реализации данной задачи. В ходе анализа проведено сравнение данных четырех СУБД. По итогам всех процедур сравнения выбор сделан в пользу СУБД MySQL. Именно эта система управления базами данных будет использована в данной работе при разработке ИС. Эта СУБД максимально соответствует всем предъявляемым требованиям.

Также выбор сделан в пользу веб-сервера Apache. Он поддерживает интерпретатор языка программирования PHP. Дополнительно здесь реализованы инструменты, позволяющие работать с СУБД MySQL.

Литература

1. Заботина Н.Н. Проектирование информационных систем. М.: Инфра-М, 2013. 336 с.

2. Затонский А.В. Информационные технологии. Разработка информационных моделей и систем. М.: РИОР, Инфра-М, 2014. 344 с.

3. Braun R., Esswein W. Towards Multi-perspective Modeling with BPMN // Enterprise Engineering Working Conference.. Springer, Cham. 2015. Pp. 67-81. https://doi.org/10.1007/978-3-319-19297-0_5

4. Ritter D. Using the Business Process Model and Notation for Modeling Enterprise Integration Patterns // arXiv preprint arXiv: 1403.4053. 2014.

© Ибраев Т.Б., Касымова А.Х., 2022

УДК 004.057.4
https://doi.org/10.36906/AP-2022/06

Никонова Е.З.
ORCID: 0000-0003-4416-2629, канд. пед. наук
Королев Р.И.
ORCID: 0000-0002-8906-1963
Нижневартовский государственный университет
г. Нижневартовск, Россия

ОБЗОР ПРОТОКОЛОВ ИНТЕРНЕТА ВЕЩЕЙ

Аннотация. В статье рассматриваются способы подключения различных устройств к сети, организация их взаимодействия с помощью специализированных стандартов и протоколов. Анализируются распространённые протоколы интернета вещей и их соответствие стандартным сетевым протоколам. Актуальность работы обусловлена стремительно увеличивающимся количеством устройств интернета вещей и необходимостью выбора оптимального протокола.

Ключевые слова: Интернет вещей; протоколы взаимодействия; стандарты.

Nikonova E.Z.
ORCID:0000-0003-4416-2629, Ph.D.
Korolev R.I.
ORCID:0000-0002-8906-1963
Nizhnevartovsk State University
Nizhnevartovsk, Russia

OVERVIEW OF THE INTERNET OF THINGS PROTOCOLS

Abstract. The article discusses ways of connecting various devices to the network, organizing their interaction using specialized standards and protocols. Common IoT protocols and their compliance with standard network protocols are analyzed. The relevance of the work is due to the rapidly increasing number of IoT devices and the need to choose the optimal protocol.

Keywords: Internet of things; communication protocols; standards.

Сегодня технологии Интернета вещей (IoT, Internet of Things) активно внедряются во все сферы жизни общества, позволяя использовать различные устройства, не обязательно физические, для создания конкретных решений, способных облегчить жизнь человечества. Устройства становятся способными слышать, видеть, думать, в некоторых случаях действовать.

CC BY

Согласно исследованиям, проведённым АО «ЦНИИ «Электроника» совместно с организацией «Цифровая экономика» и журналом «Прикладная информатика» «…общий объем рынка Интернета вещей в 2018 году в России составил порядка 3,7 млрд долл. США, что составляет примерно 0,5% от общемирового» [https://www.instel.ru]. В качестве препятствий развитию IoT в России были отмечены «…нерешённые вопросы обеспечения безопасности производства (58%), отсутствие промышленных стандартов (46%), общего понимания концепции Интернета вещей (46%), а также нормативной базы (46%)».

На наш взгляд, дополнительную трудность для руководителей при внедрении технологии IoT представляет выбор среди огромного количества поставщиков, платформ и устройств.

Для правильной и эффективной работы устройства должны корректно общаться и координировать свои действия с другими для того, чтобы принимать решения, которые могут быть столь критичны, как спасение жизней или зданий. Технологии распределённых вычислений, встроенные датчики, современные беспроводные технологии позволяют Интернету вещей выполнять поставленные задачи. Однако принимая во внимание сегодняшнее разнообразие данных технологий и устройств, огромное количество производителей, возникает множество проблем их взаимодействия и необходимость в осознанном выборе специализированных стандартов и протоколов Интернета вещей, оптимизированных для различных сценариев использования.

Следует уточнить способ подключения устройств к сети, предполагающий использование шлюзов или встроенных функций. С помощью шлюзов можно не только подключать устройства, не имеющие прямого доступа к интернету, но и снизить задержку передаваемых данных, а также обеспечить дополнительный уровень безопасности.

Определенный протокол Интернета вещей работает на одном из следующих уровней:

– устройство ↔ устройство;

– устройство ↔ шлюз;

– шлюз ↔ облако (центр обработки данных).

Выбор типа протокола Интернета вещей определяется уровнем архитектуры системы передачи данных в сети. За основу классификации протоколов IoT берётся базовая модель сетевого взаимодействия OSI.

– Прикладной уровень

Помимо стандартных для этого уровня протоколов, таки как HTTP, могут использоваться специальные протоколы IoT: MQTT, AMQP, CoAP, DDS, XMPP, JMS и др., отличающиеся принципами работы:

- Протокол для систем реального времени **DDS (**Data Distribution Service**),** реализующий прямую связь между устройствами по шине методом «запрос-ответ»**;**
- Протоколы, передающие сообщения через специальный сервер-брокер, расположенный либо на облаке, либо в дата-центре. К протоколам этой группы относятся **MQTT**, **XMPP**, **AMQP** и **JMS,** среди которых наиболее популярным в настоящее время

CC BY

является протокол MQTT (Message Queuing Telemetry Transport) благодаря отсутствию требования постоянного Интернет-соединения.

Основной целью MQTT является удалённый мониторинг данных, собираемых из большого количества устройств, и их телеметрия в IT инфраструктуру. Протокол нацелен на большую сеть небольших устройств, которые необходимо контролировать или управлять из облака. Также он предназначен для «многоадресной передачи» данных для многих приёмников. MQTT чрезвычайно прост, предлагая несколько вариантов управления. Примером работы является мониторинг нефтепровода на наличие утечек или вандализма. Это информация от тысячи датчиков, которая должны быть сконцентрирована в одном месте для анализа. Когда система обнаруживает проблему, она может принять меры, чтобы исправить эту проблему.

- Протокол Modbus, принцип работы которого основан на концепции «ведущий, передающий запросы подчинённым устройствам - ведомым». Данный протокол используется для мониторинга, сбора данных с датчиков, управления контроллерами и поддерживается почти всеми производителями промышленного оборудования.
- Протокол CoAP (Constrained Application Protocol), поддерживающий многоадресную рассылку по принципу «точка-точка» для устройств с ограниченными ресурсами. Основой разработки данного протокола послужил протокол HTTP, использующий TCP и являющийся текстовым. В отличие от предшественника CoAP – бинарный протокол, работающий на основе UDP, что позволяет значительно уменьшить размер передаваемых данных.

– ***Транспортный уровень***

- TCP (Transmission Control Protocol) – один из основных протоколов передачи данных интернета. Предназначен для управления передачей данных интернета. Пакеты в TCP называются сегментами. В стеке протоколов TCP/IP выполняет функции транспортного уровня модели OSI. Механизм TCP предоставляет поток данных с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым (в отличие от UDP) целостность передаваемых данных и уведомление отправителя о результатах передачи. Реализации TCP обычно встроены в ядра ОС. Существуют реализации TCP, работающие в пространстве пользователя. Когда осуществляется передача от компьютера к компьютеру через Интернет, TCP работает на верхнем уровне между двумя конечными системами, например, браузером и веб-сервером. TCP осуществляет надёжную передачу потока байтов от одного процесса к другому.
- UDP (User Datagram Protocol) – один из ключевых элементов набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году и официально определён в RFC 768. UDP использует простую модель передачи, без неявных «рукопожатий» для

CC BY

обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени. При необходимости исправления ошибок на сетевом уровне интерфейса приложение может задействовать TCP или SCTP, разработанные для этой цели. Природа UDP как протокола без сохранения состояния также полезна для серверов, отвечающих на небольшие запросы от огромного числа клиентов, например DNS и потоковые мультимедийные приложения вроде IPTV, Voice over IP, протоколы туннелирования IP и многие онлайн-игры.

–***Сетевой уровень***

- IPv4 – четвёртая версия интернет-протокола IP. Она создана в 1981 году и актуальна по настоящее время. IPv4 является основным протоколом коммуникации в Интернете. Он состоит из 4 частей, каждая из которых разделена точкой, например, 127.101.102.103. Диапазон каждой части - от 0 до 255. Это сделано для лёгкости чтения и запоминания. IPv4 использует 32-битное адресное пространство, его размер 4 байта. Общее количество IP в Сети - около 4,3 миллиарда. Уже после внедрения IPv4 стало очевидно, что распределение адресного пространства происходит значительно более быстрыми темпами, чем было заложено в архитектуру IPv4. Это привело к разработке нового протокола IPv6. 25 ноября 2019 года были распределены последние свободные IPv4-адреса в Европе, странах бывшего СССР и на Ближнем Востоке. Теперь получить IPv4-адрес можно будет только, если его освободит текущий владелец, например, закроется компания или какая-либо сеть освободит ненужный ей адресный ресурс.
- IPv6 – шестая версия интернет-протокола IP. Протокол был создан в декабре 1995 года. Он был разработан Инженерным советом Интернета (IETF) и является самой последней версией интернет-протокола. IPv6 более продвинутый, чем IPv4, и предоставляет лучшую функциональность. Как было обозначено выше, каждому устройству в Интернете назначается определенный уникальный IP-адрес. Новый протокол может предоставить практически бесконечное количество адресов для устройств и заменяет прошлую версию для обслуживания растущего числа трафика по всему миру и решает проблему нехватки IP-адресов. Количество адресов в IPv6 составляет 5x1028 (около 79 228 162 514 264 337 593 543 950 336 октиллионов). Это означает, что протокол обеспечит возможность использования более 300 млн IP-адресов на каждого жителя Земли. В отличие от IPv4, типичный адрес IPv6 состоит из 128 бит. Он включает в себя восемь групп, каждая из которых состоит из четырех шестнадцатеричных цифр, разделённых ":". Например: 3005: 0db6: 82a5: 0000: 00 00: 7a1e: 1460: 5334. В 2014 году доля IPv6 в интернет-трафике составляла около 5%. На 2021 год, согласно данным Google, эта доля составляет более 32%.
- 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) – стандарт, который позволяет перенести распространённые IP-протоколы, в частности IPv6, на

CC BY

сенсорные сети (IEEE 802.15.4) для согласованного соединения с уже существующими IP-сетями, то есть 6LoWPAN-сеть совместима с любыми сетевыми подключениями, применяет популярные стандарты (HTTP, UDP, TCP, COAP) и устанавливает между всеми узлами IP-адресацию. Архитектура 6LoWPAN включает маршрутизатор, роутеры и хосты. Маршрутизатор объединяет сети и подключает их к Интернету, роутеры передают данные между элементами сети, а хосты (конечные устройства) длительное время находятся в спящем режиме, обеспечивая энергосбережение. 6LoWPAN – mesh-сеть, топология которой гарантирует надёжность, самовосстановление и гибкость при настройке.

–***Канальный уровень данных***

- BLE (Bluetooth Low Energy) - это стандарт, характерной чертой которого является низкое потребление энергии. Он работает на частоте 2.4 ГГц и обеспечивает подключение устройств на расстоянии более 100 м. Широкая известность Bluetooth позволила ему проникнуть во все уголки мира и во многие устройства – смартфоны, наушники, медицинские приборы, бытовую технику, ПК и ноутбуки, автомобили и многие другие. BLE с успехом применяется в системах «умный дом». Также он лёг в основу iBeacon-маяков. Последний стандарт Bluetooth 5, разработанный в 2016 году, ориентирован на Internet of Things.
- LoRaWAN (Long Range WAN) - технология передачи данных для Интернета вещей IoT. Принцип действия заключается в том, что «умные» smart-датчики в онлайн-режиме осуществляют мониторинг показателей, анализируют их и передают данные на сервер. Через приложение данные с датчиков отображаются на мониторе. В целом система способна самостоятельно оценивать сотни показателей, что позволяет, к примеру, разгрузить человека на производстве. Неквалифицированные рутинные операции теперь спокойно можно доверить машине, оставив сотрудникам больше времени для интеллектуальной работы. Все датчики работают автономно без электросети до 10 лет без проводов. Это особенно актуально для объектов, на которых нет электричества, а также там, где электричество может являться источником потенциальной опасности – возгорания. Отсутствие проводов также делает процесс внедрения технологии максимально простым и удобным. При этом стоимость остаётся доступной для массового применения. Протокол LoRaWAN был разработан компаниями Semtech и IBM Research. Он на канальном уровне модели ISO/OSI , в то время как на физическом применяется модуляция LoRa, основанная на расширении спектра. Сеть LoRaWAN имеет топологию «звезда», то есть множество устройств (датчиков, сенсоров) по беспроводному соединению передают данные на шлюзы, а шлюзы перенаправляют информацию на сервер.
- LTE-M представляет собой стандарт радиотехнологии маломощной глобальной сети (LPWAN), разработанный 3GPP для поддержки широкого спектра сотовых устройств. и услуги (в частности, для приложений межмашинного взаимодействия и Интернета вещей). Спецификация для eMTC (LTE Cat-M1) была заморожена в 3GPP версии 13 (LTE Advanced Pro) в июне 2016 года. Другие технологии 3GPP IoT включают NB-IoT.и EC-GSM-IoT. Преимущество LTE-M перед NB-IoT заключается в его сравнительно более высокой скорости

передачи данных, мобильности и передаче голоса по сети, но он требует большей полосы пропускания, является более дорогостоящим и пока не может быть помещён в полосу частот защитной полосы. По сравнению с модемом LTE Release 12 Cat-0, модель LTE-M, как утверждается, на 80% дешевле (с точки зрения стоимости материалов), поддерживает лучшее покрытие до 18 дБ и срок службы батареи, чем может. длиться до нескольких лет. В марте 2019 года Глобальная ассоциация поставщиков мобильной связи сообщила, что более 100 операторов развернули, / запустили сети NB-IoT или LTE-M.

- Lora – это технология модуляции сигнала физического уровня (PHY), ориентированная на устройства с питанием от батареи и передачу сигнала на большие расстояния. Технология LoRa разработана компанией Semtech и использует запатентованный метод модуляции сигнала с расширением спектра, где данные кодируются широкополосными импульсами с изменяемой частотой. Применение такого метода имеет следующие преимущества: значительно повышается чувствительность приёмника (до -148дБм) и снижается критичность к расстройке по частоте между приёмником и передатчиком. Технология LoRa обеспечивает связь на большие расстояния (до 20 км прямой видимости), низкое энергопотребление, высокую чувствительность приёмника, низкие скорости передачи данных и защищённую передачу с шифрованием.
- SigFox - технология беспроводной низкоскоростной связи устройств в сетях с низким потреблением энергии. Для передачи данных SigFox использует ультра-узкую полосу частот (Ultra-Narrow Band, UNB) с двоично-фазовой манипуляцией (BPSK), а для кодирования данных меняет фазу несущей радиоволны. Это позволяет уменьшить уровень шума на принимающей стороне, следовательно, сделать принимающие устройства более дешёвыми. Характеристики: радиус действия: 30-50 км (3-10 км в зашумлённых и труднодоступных районах); срок службы устройств без замены батареи: 20 лет от 2-х батареек типа AA; используемые частоты: 868 МГц (Европа) и 902 МГц (США); топология сети: звезда (базовая станция, к которой подключаются конечные точки). Существующий стандарт SigFox определяет максимальное количество сообщений от базовой станции до конечного устройства в день: 140 сообщений, при этом каждое сообщение должно быть размером не более 12 байт (исключая заголовок сообщения и информацию о передаче). И также количество сообщений, исходящих от конечного устройства: 4 сообщений в день с полезной нагрузкой 8 байт.

Таким образом при выборе протоколов интернета вещей следует исходить из таких параметров как количество подключаемых устройств, потребление ресурсов, объем передаваемых данных и расстояние, на которое их нужно передать.

© Никонова Е.З., Королев Р.И., 2022

УДК 004.652
https://doi.org/10.36906/AP-2022/07

Ужаринский А.Ю.
ORCID: 0000-0002-0499-4781, канд. техн. наук
Стычук А.А.
ORCID: 0000-0001-8424-5264, канд. техн. наук
Короткий А.В., Аршинова А.А.
Орловский государственный университет им. И.С. Тургенева
г. Орёл, Россия

ТЕХНОЛОГИЯ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ В ПОСТРОЕНИИ СОВРЕМЕННЫХ ПРИЛОЖЕНИЙ

Аннотация. В статье рассмотрены вопросы построения приложения, взаимодействующего с базой данных. Предложена технология объектно-реляционного отображения при построении приложения. Рассмотрены альтернативные подходы, используемые в качестве интерфейса между основным приложением и уровнем данных.

Ключевые слова: приложение; база данных; технология объектно-реляционного отображения; ORM; нативный запрос; конструктор запросов.

Uzharinskiy A.U.
ORCID: 0000-0002-0499-4781, Ph.D.
Stychuk A.A.
ORCID: 0000-0001-8424-5264, Ph.D.
Korotkiy A.V., Arshinova A.A.
Orel State University named after I.S. Turgenev
Orel, Russia

OBJECT-RELATIVE MAPPING TECHNOLOGY FOR BUILDING MODERN APPLICATIONS

Abstract. The article discusses the issues of building an application that interacts with a database. The technology of object-relational mapping when building an application is proposed. Alternative approaches used as an interface between the main application and the data layer are considered.

Keywords: application; database; object-relational mapping technology; ORM; native query; query builder.

Современные приложения работают с большим количеством информации, для хранения которой разработчики прибегают к использованию баз данных. Эта технология позволяет

CC BY

управлять информацией приложения, обеспечивая целостность данных и их постоянство после окончания срока службы отдельного экземпляра приложения.

Однако весьма непросто работать напрямую с базами данных из приложения, поскольку часто возникают такие проблемы, как различия в способах представления структур данных и сложность в выражении тонкостей отношений между различными объектами. Для решения этих проблем было создано множество различных инструментов, которые помогают выступать в качестве посредника между приложением и базой данных.

Технология объектно-реляционного отображения в web-приложениях

У технологии объектно-реляционного отображения (англ. ORM – Object-Relational Mapping) существует несколько различных определений, но все они схожи и говорят, что объектно-реляционное отображение – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

ORM используется для упрощения процесса исполнения CRUD-операций (https://clck.ru/bkv6U). При этом сама ORM налаживает связь между базой данных и объектно-ориентированными моделями. В большинстве случаев используется некий шаблон, который отображает класс в таблицу в базе данных, а атрибуты объекта этого класса в поля этой же таблицы. ORM создаёт «виртуальную объектную базу данных» в памяти и через неё осуществляет управление данными в режиме объектной ориентированности. Информация, полученная этим отображением, используется для преобразования данных из формы объекта в базу и наоборот, а также для создания SQL-запросов для реализации CRUD-операций. В такой технологии объектам ничего не нужно знать о структуре базы данных, а ей в свою очередь об организации приложения.

ORM служит уровнем абстракции, призванным помочь работать с базами данных без изменения объектно-ориентированной парадигмы в отличие от реляционной модели данных.

В технологии объектно-реляционного отображения используются разные способы связи между базой данных и объектно-ориентированным представлением данных в коде приложения: активный шаблон записи (англ. ActiveRecord) [1, с. 182-187] и преобразователь данных (англ. Data Mapper) [1, с.187-192].

В активной записи модели происходит инкапсуляция данных в пределы структуры объектов в коде. Объекты содержат методы для сохранения, обновления или удаления из базы данных, а изменения в объектах должны отражаться в базе данных. Реализация активных записей позволяет управлять базой данных, создавая классы и экземпляры классов непосредственно в коде.

Преобразователь данных является другой реализацией ORM. Преобразователь действует как связующее звено, посредник, между кодом и данными в базе данных. Вместо инкапсуляции данных в структуре объектов в коде, он концентрируется на разъединении и трансляции данных в объекты, позволяя каждому элементу существовать независимо. Это отделяет бизнес-логику от деталей реализации базы данных.

Технология объектно-реляционного отображения популярна по многим причинам:

CC BY

1. Она помогает абстрагировать область данных до уровня контекста разрабатываемого приложения. ORM обеспечивает работу с базой данных, как продолжение работы с кодом приложения, что помогает концентрировать внимание на реализации основной бизнес-логики и не тратить время на нюансах серверной части хранилища.

2. Также одним из преимуществ ORM является возможность автоматического создания запросов. Она избавляет от необходимости знания и использования DDL (англ. Data Definition Language) (https://clck.ru/Nsr8C) и DML (Data Manipulation Language) (https://clck.ru/Nsr8C) при разработке базы данных.

3. ORM часто поставляются с инструментами миграции, которые помогают управлять изменениями схемы базы данных на основе изменений, внесённых в код. Не обязательно заранее выяснять идеальную схему базы данных, если ORM может помочь управлять изменениями в структуре базы данных. Изменения в приложении и базе данных часто являются одним и тем же или тесно связаны, что помогает отслеживать изменения в базе данных по мере того, как вносятся изменения в код.

ORM не лишено недостатков. Во многих случаях они возникают из тех же решений, которые делают ORM полезным:

1. Одна из фундаментальных проблем с ORM – попытка скрыть детали серверной части базы данных. Это упрощает работу с ORM в простых случаях или в небольших временных масштабах, но часто приводит к проблемам в дальнейшем по мере роста сложности. Абстракция никогда не бывает полной и попытки использовать ORM без понимания основного языка запросов или структуры базы данных часто приводят к сомнительным предположениям. Это может затруднить или сделать невозможным отладку и настройку производительности.

2. ORM обычно медленнее, чем альтернативные подходы. ORM часто генерируют сложные запросы для относительно простых операций с базой данных, потому что они используют общие шаблоны, которые должны быть достаточно гибкими для обработки других случаев.

3. Возможно, самая известная проблема работы с ORM – трудность перевода между объектно-ориентированным программированием и реляционной парадигмой, используемой в реляционных базах данных. Несовместимость моделей данных, используемых этими двумя категориями технологий, означает, что с каждым увеличением сложности необходима дополнительная несовершенная абстракция. Данная проблема приводит к увеличению сложности со временем и ситуациям, когда пути к успеху или изменению курса трудны или невозможны.

Альтернативные подходы к управлению данными

Теперь рассмотрим два других подхода к управлению данными: использование встроенных в базу данных языков запросов и конструктор запросов.

Некоторые приложения взаимодействуют напрямую с базой данных, записывая и выполняя запросы на языке, поддерживаемом ядром базы данных. Отправка запросов, написанных на поддерживаемом языке базы данных, выполняется через подключение к ней.

CC BY

В свою очередь, база данных предоставит результаты запроса в одном из своих собственных форматов.

Одним из преимуществ данного подхода является производительность. Написанный нативный запрос, основанный на предметной области, позволяет избежать ненужных условий, перегруженных конструкций и других ошибок, приводящих к медленным операциям с базой данных и трудностям отладки.

Ещё одна причина использования нативных запросов к базе данных - гибкость. Более высокие уровни абстракции пытаются преодолеть разрыв между двумя разными парадигмами, что может ограничивать типы операций, которые они могут выражать. А при написании нативных запросов можно воспользоваться всеми функциями механизма базы данных и выражать более сложные запросы.

Из недостатков нативных запросов можно выделить то, что на программисте лежит ответственность за перевод между типами данных и структурами, которые использует приложение, и конструкциями, доступными в системе базы данных.

Ещё одна вещь, о которой следует помнить при работе с нативными запросами, заключается в том, что программист сам должен управлять безопасностью своего ввода. Например, когда специально созданный ввод - SQL-инъекция [2, с. 25-32] может повлиять на состояние базы данных.

Иной подход к управлению данными заключается в использование инструмента или библиотеки, называемых конструктором запросов, для взаимодействия с базой данных.

Конструктором запросов добавляет уровень абстракции над нативными языками запросов для баз данных. Они делают это путём формализации шаблонов запросов и предоставления методов или функций, которые автоматически экранируют элементы для упрощения интеграции в приложения.

Обычно конструкторы запросов предоставляют интерфейс, который использует методы или функции для добавления условия в запрос. Объединив методы вместе, разработчики могут составлять полные запросы к базе данных из этих отдельных частей.

Поскольку конструкторы запросов используют те же методы или функции, что и остальная часть приложения, разработчикам легче управлять ими в долгосрочной перспективе, чем нативными запросами к базе данных, записанными в виде строк.

Конструкторы запросов также поддерживают несколько серверов данных, например, абстрагируя некоторые тонкие различия в различных реляционных базах данных. Это позволяет использовать одни и те же инструменты для проектов, использующих разные базы данных.

Одним из критических недостатков конструктора запросов заключается в том, что они требуют от разработчика понимания и учёта структур и возможностей базы данных, разбираться в нативных запросах в дополнение к специфическому синтаксису и возможностям самого конструктора запросов.

CC BY

Кроме того, конструкторы запросов требуют определения того, как извлекаемые данные соотносятся с данными приложения: нет автоматической синхронизации между объектами в памяти и объектами в базе данных.

Хотя конструкторы запросов имитируют язык запросов, для работы с которым они предназначены, дополнительный уровень абстракции может означать, что иногда определенные операции невозможны с использованием предоставленных методов.

Результаты и выводы

При построении приложений, взаимодействующих с базой данных разработчик сталкивается с проблемой управления данными. Для её решения возможно использование различных инструментов, таких, как технология объектно-реляционного отображения, нативные запросы и конструктор запросов.

Для большой производительности и гибкости при взаимодействии приложения с базой данных лучше всего использовать нативный запрос, но при этом необходимо учитывать, что на программисте лежит ответственность за безопасность ввода и совместимости типов данных и структур приложения с конструкциями из базы данных.

При необходимости увеличения уровня абстракции над нативными запросами для облегчения управления ими в долгосрочной перспективе и сглаживания тонких различии в разных реляционных базах данных лучше использовать конструктор запросов, но от разработчика требуется умение разбираться в нативных запросах в дополнение к специфическому синтаксису и возможностям самого конструктора запросов. У конструктора запросов нет автоматической синхронизации между объектами в памяти и объектами в базе данных.

Технология объектно-реляционного отображения позволяет сконцентрировать внимание на реализации основной логике приложения, а не тратить время на особенности реализации базы данных. А также ORM предоставляет возможность автоматического создания запросов. Проблемой данного подхода является генерация сложных запросов для относительно простых операций с базой данных.

В современных реалиях разработки приложений, где требуется учитывать множество особенностей реализации основной логики приложения, наилучшим решением будет, то которое снимает дополнительную нагрузку с разработчика. Таким решением может быть технология объектно-реляционного отображения – ORM.

Литература

1. Фаулер М. Шаблоны корпоративных приложений. М.: И.Д. Диалектика-Вильямс, 2019, 546 с.

2. Аникин Е.А. SQL инъекция и как защититься от несанкционированного доступа // Continuum. Математика. Информатика. Образование. 2016. № 4(4). С. 25-32.

© Ужаринский А.Ю., Стычук А.А., Короткий А.В., Аршинова А.А., 2022

УДК 519.687.6
https://doi.org/10.36906/AP-2022/08

Слива М.В.
ORCID: 0000-0001-7915-2282, канд. пед. наук
Левкин Л.В.
Нижневартовский государственный университет
г. Нижневартовск, Россия

КОНТЕЙНЕРИЗАЦИЯ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ПЛАТФОРМЫ DOCKER

Аннотация. В статье рассматривается принцип контейнеризации приложений с помощью платформы Docker. Сравнивается развёртывание приложений на виртуальных машинах. Рассматривается принцип работы контейнеров, образов, создание слоев и упаковка необходимых зависимостей. Конфигурирование файла сборки Dockerfile.

Ключевые слова: Docker; Docker Hub; Виртуальная машина; Хост; Контейнер; Образ; Docker Registry; Dockerfile; Docker console.

Sliva M.V.
ORCID: 0000-0001-7915-2282, Ph.D.
Levkin L.V.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

APPLICATIONS CONTAINERIZATION VIA THE DOCKER PLATFORM

Abstract. This article discusses the principle of application containerization using the Docker platform. Compares the deployment of applications to virtual machines. The principle of operation of containers, images, the creation of layers and the packaging of the necessary dependencies are considered. An example of configuring the Dockerfile assembly file is given.

Keywords: Docker; Docker Hub; Virtual machine; Host; Container; Image; Docker Registry; Dockerfile; Docker console.

Достаточно большое количество времени стандартным средством для запуска серверных приложений являлись виртуальные машины. Виртуальная машина – программа, задачей которой является имитация физического оборудования, необходимого для запуска операционной системы, а реальный физический сервер с настоящим процессором, оперативной памятью и дисками для хранения данных именуется хостом. На сервере может стоять любая операционная система семейств: **Windows**, **Linux** или **MacOs**.

Гипервизор – программа, отвечающая за имитацию оборудования и управления виртуальными машинами, позволяющая сделать из одного сервера несколько, разделив его

CC BY

ресурсы в любой пропорции. Например, кому-то отдать больше процессорного времени, а кому-то оперативной памяти или дискового пространства. Таким образом можно реализовать множество виртуальных серверов на одной хост машине (рис. 1).

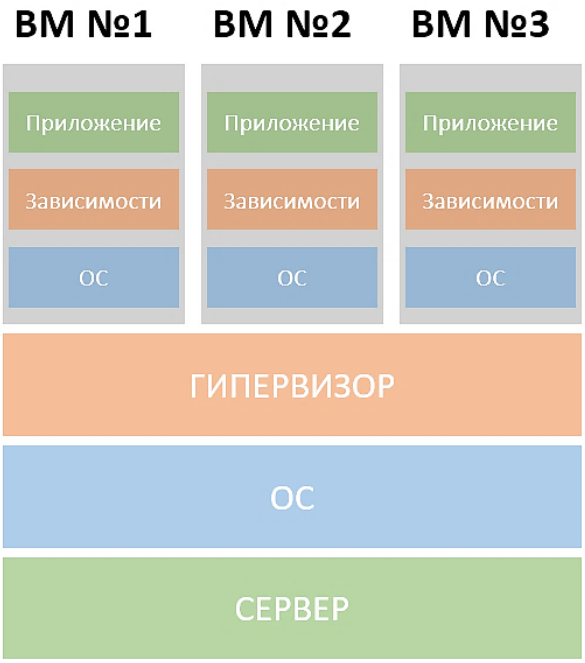


Рис. 1. Схема хост машины

Виртуальные машины изолированы друг от друга, поэтому на каждый из виртуальных серверов необходимо установить свою операционную систему, которую называют гостевой операционной системой. Перед тем как приложение будет развёрнуто на виртуальной машине, следует убедиться в наличии необходимых драйверов, библиотек или любых других зависимостей для корректной работы вашей программы.

Приложение внутри виртуальной машины полностью изолировано как от приложений в других виртуальных машинах, так и от самой хост системы. Именно изоляция является ключевым преимуществом. Если перенести виртуальную машину с одного физического сервера на другой, то приложения, развёрнутые внутри виртуальной машины, не заметят каких-либо изменений. Не потребуется их повторная установка или настройка.

Контейнеризация решает фактически ту же самую задачу, что и виртуальный сервер, создавая изолированную виртуальную среду, внутри которой запускается готовое приложение. Первоначально контейнер – это файл, содержащий приложение и все необходимые ему зависимости, а также инструкции для его запуска. В начале выполнения операции сборки создаётся образ, из которого в будущем можно реализовать слоёную структуру образов, где каждый новый слой содержит изменения только относительно предыдущего (рис. 2).

CC BY

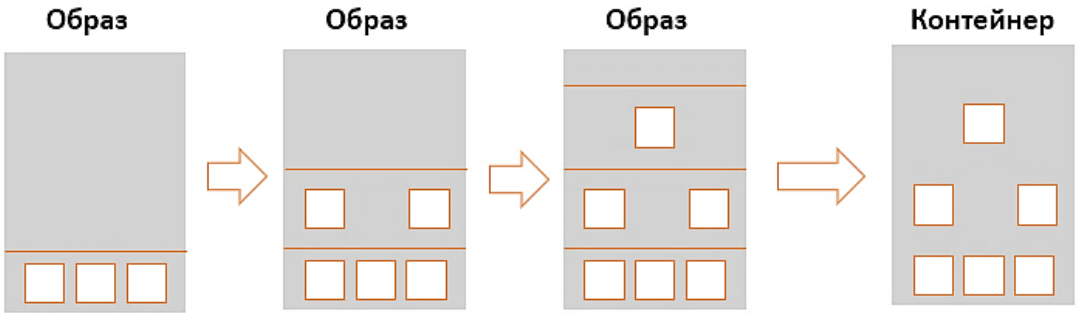


Рис. 2. Схема структуры образов

При запуске платформа контейнеризации **Docker** распаковывает образ контейнера с нужными файлами и метаданными, затем инициализирует сеть и файловую систему, и запускает наше приложение в соответствии с инструкциями, которые есть в образе. В конечном итоге получается обычный процесс операционной системы. **Docker Engine** состоит из трёх основных компонентов. Первое – ядро докера, представляет из себя обычную программу, умеющею создавать образы или скачивать готовые из сети. Позволяет запускать из образов контейнеры, организовать между ними сетевое взаимодействие и управлять томами, предоставляя контейнеру доступ на чтение или запись к файловой системе. Вторым компонентом является **REST API – HTTP** интерфейс для передачи команд докеру. Последний компонент – консольной клиент, вызываемый с помощью команды **docker**. Консольный клиент взаимодействует с ядром докера через **REST API**.

Для хранения образов используют репозитории. Докер предоставляет свой собственный публичный репозиторий **Docker Hub**, содержащий огромное количество образов, которые вы можете использовать для контейнеризации собственных приложений. Помимо этого, вы можете загружать в репозиторий свои собственные образы. В основе **Docker Hub** лежит проект с открытым исходным кодом **Docker Registry**.

В качестве основы для создания собственного образа часто используют базовый образ той или иной операционной системы. Важным преимуществом докер контейнера по сравнению с виртуальным сервером является то, что для запуска исполнения контейнеров используется одно и тоже ядро операционной системы. Это позволяет избежать накладных расходов, связанных с моделированием оборудования, а также необходимостью запуска отдельно операционной системы для каждого контейнера, как это происходит в случае виртуального сервера. Также к преимуществу можно отнести более эффективное управление ресурсами вследствие того, что контейнер – просто процесс, память которому выделяется динамически, а в случае виртуального сервера мы должны заранее зарезервировать память с запасом. Потенциально на одном и том же физическом сервере может поместиться гораздо большее количество контейнеров чем виртуальных машин. Образ контейнера является более легковесным по сравнению с образом виртуальной машины. Контейнер легче перенести и развернуть новым окружением или масштабировать путём дублирования. Но теоретически контейнеры менее безопасны чем виртуальные машины. Связано это с тем, что контейнеры запускаются на одном ядре операционной системы, следовательно имеют более слабую

CC BY

изоляцию чем виртуальной машины, где мы имеем дополнительный слой в виде виртуального оборудования.

Чтобы собрать образ нашего приложения для начала потребуется создать текстовый файл **Dockerfile** в каталоге программы, который служит конфигурацией к сборке, запуску и другим параметрам будущего контейнера. Рассмотрим создание конфигурационного файла на примере проекта веб приложения, написанного на языке программирования **Kotlin** с использованием библиотеки **Ktor**. Для сборки и установки приложения со всеми необходимыми зависимостями используется система сборки **Gradle** (рис. 3).

Рис. 3. Команда сборки приложения в Gradle

Для запуска и работы приложения на хост машине потребуется установленный пакет **openjdk:8-jdk**. С помощью **Docker** мы можем загрузить базовый образ **openjdk:8-jdk** из репозитория **Docker Hub** и на основе него собрать свой образ, добавив слой с нашим приложением.

В **Dockerfile** прописываем директиву сборки **FROM**, которая служит пометкой, какой существующий образ взять за основу. В нашем случае **FROM openjdk:8-jdk**. Для веб приложения понадобится открыть порт и сопоставить его с портом контейнера с помощью директивы **EXPOSE 80:80**. Теперь будущий контейнер будет прослушивать 80 порт виртуальной машины и перенаправлять на свой внутри контейнера. Используя директиву **RUN**, можно прописать различные консольные команды, например создание директории в слое: **RUN mkdir /app**. Теперь нужно добавить файлы приложения из локального контекста сборки в рабочую директорию образа: **COPY ./build/install/NvguNorify/ /app/**. Так как исполняемый файл запуска приложения находится в каталоге /bin, мы можем прописать рабочую директорию контейнера в каталоге **WORKDIR /app/bin**. Это нужно для следующей команды **CMD**, которая выполняется всегда при запуске контейнера. В нашем примере мы запускаем свое приложение: **CMD ["./NvguNorify"]**. Исходный файл **Dockerfile** показан на рисунке 4.

```
FROM openjdk:8-jdk
EXPOSE 80:80
RUN mkdir /app
COPY ./build/install/NvguNorify/ /app/
WORKDIR /app/bin
CMD ["./NvguNorify"]
```

Рис. 4. Готовый Dockerfile приложения

CC BY

Чтобы собрать приложение, нужно перейти в каталог, где находится **Dockerfile,** и выполнить команду сборки:

docker build -t lev:notify .

Флаг **-t** позволяет задать свой тег для образа**: lev:notify**, а точка в конце команды говорит докеру искать файл **Dockerfile** в каталоге приложения. Образ **openjdk:8-jdk**, к которому мы добавляем свой слой с приложением, будет загружен автоматически из репозитория **Docker Hub** и на его основе будет собран новый образ. Затем будут последовательно выполнены все директивы, описанные в **Dockerfile** (рис. 5).

Рис. 5. Выполнение сборки

После выполнения команды **docker images** будет выведен список всех образов, находящихся на локальной машине (рис. 6). Обратите внимание, что образ **openjdk:8-jdk** был загружен и добавлен в список автоматически. Его можно также смонтировать отдельно в контейнер.

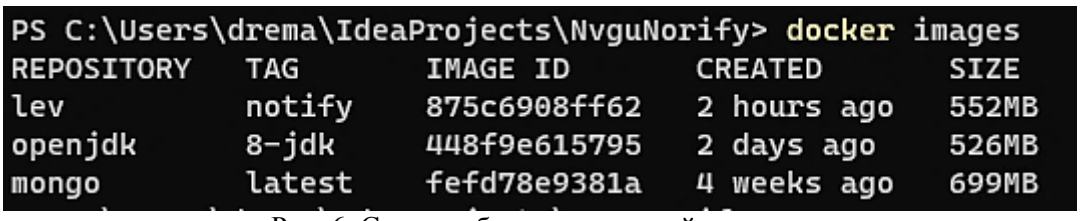


Рис. 6. Список образов локальной машины

Наконец для запуска нашего приложения в виде контейнера выполним команду: **docker run -d -p 80:80 –-name notify-app-container lev:notify.**

И отобразим все контейнеры с помощью команды: **docker ps -a** (рис. 7).

```
PS C:\Users\drema> docker run -d -p 80:80 --name notify-app-container lev:notify
9e8d34009558a79abfea89aec1ea39b1d7e28fb8ac46f01ad92d0e6e03dc7c5
PS C:\Users\drema> docker ps -a
CONTAINER ID   IMAGE        COMMAND                  CREATED          STATUS         PORTS                     NAMES
9e8d34009558   lev:notify   "./NvguNorify"           10 seconds ago   Up 8 seconds   0.0.0.0:80->80/tcp        notify-a
pp-container
9f4e43725863   mongo        "docker-entrypoint.s…"   3 minutes ago    Up 3 minutes   0.0.0.0:27017->27017/tcp  mongo
```

Рис. 7. Список контейнеров локальной машины

Для запуска контейнера в фоне используется флаг **-d**. Указание переадресации портов между локальной машиной и контейнером **-p xx:xx**. Задать имя контейнеру можно через флаг **–name**. Если убрать **-a**, то отобразятся только контейнеры, работающие в данный момент.

Чтобы остановить работу контейнера, введите команду **docker stop** с наименованием тега контейнера или его уникального индикационного номера. Например **docker stop lev:notify** или **docker stop 9e8d34009558.** Для удаления используются команды **docker rm** для контейнеров и **docker rmi** для образов аналогичным способом по наименованию или идентификатору (рис. 8).

```
PS C:\Users\drema> docker stop 9e8d34009558
9e8d34009558
PS C:\Users\drema> docker rm 9e8d34009558
9e8d34009558
PS C:\Users\drema> docker rmi lev:notify
Untagged: lev:notify
Deleted: sha256:875c6908ff625516129417ba13aaccc3a38130bfceb9bd40fa6fe5ad4f29b1fc
```

Рис. 8. Остановка, удаление контейнера. Удаление образа

© Слива М.В., Левкин Л.В., 2022

УДК 004.652
https://doi.org/10.36906/AP-2022/09

Мамедли Р.Э.
канд. физ.-мат. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия

ВОССТАНОВЛЕНИЕ БАЗЫ ДАННЫХ ИЗ НЕСОВМЕСТИМОГО СОСТОЯНИЯ В РАНЕЕ СОГЛАСОВАННОЕ СОСТОЯНИЕ

Аннотация. В работе рассматриваются различные методы восстановления базы данных из заданного состояния (обычно несовместимого) в ранее согласованное состояние. Обоснована необходимость настройки производительности SQL и настройки производительности СУБД. Приведены примеры восстановления данных для принятия важных решений по повышению эффективности обработки запросов.

Ключевые слова: транзакция; восстановление базы данных; аппаратные сбои; программные сбои; техногенные инциденты; буферы базы данных; контрольные точки; журналы транзакций.

Mammadli R.E.
Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

RESTORING A DATABASE FROM AN INCOMPATIBLE STATE TO A PREVIOUSLY CONSISTENT STATE

Abstract. The paper discusses various methods for restoring a database from a given state (usually incompatible) to a previously consistent state. The necessity of SQL performance tuning and DBMS performance tuning is substantiated. Examples of data recovery are given for making important decisions to improve the efficiency of query processing.

Keywords: transaction; database recovery; hardware failures; software failures; man-made incidents; database buffers; checkpoints; transaction logs.

Методы восстановления основаны на атомарном свойстве транзакции: все части транзакции должны рассматриваться как единая логическая единица работы, в которой все операции применяются и выполняются для создания согласованной базы данных. Если операция по какой-либо причине не может быть завершена, транзакция должна быть прервана, а любые изменения в базе данных должны быть отменены. Короче говоря, восстановление транзакции отменяет все изменения, которые транзакция внесла в базу данных до того, как транзакция была прервана.

CC BY

Методы восстановления также применяются к базе данных и системе после возникновения критических ошибок определенного типа. Критические события могут привести к прекращению работы базы данных и нарушению целостности данных. Примеры критических событий:

• Аппаратные/программные сбои. Отказ такого типа может быть отказом жёсткого диска, неисправным конденсатором на материнской плате или неисправным банком памяти. Другие причины ошибок в этой категории включают ошибки прикладной программы или операционной системы, которые приводят к перезаписи, удалению или потере данных. Некоторые администраторы баз данных утверждают, что это один из наиболее распространённых источников проблем с базами данных.

• Техногенные инциденты. Событие такого типа можно отнести к категории непреднамеренных или преднамеренных.

– Непреднамеренный сбой вызван неосторожным конечным пользователем. К таким ошибкам относятся удаление неправильных строк из таблицы, нажатие неправильной клавиши на клавиатуре или аварийное завершение работы основного сервера базы данных.

– Преднамеренные события имеют более серьёзный характер и обычно указывают на то, что данные компании подвергаются серьёзному риску. В эту категорию входят угрозы безопасности, вызванные хакерами, пытающимися получить несанкционированный доступ к ресурсам данных, и вирусные атаки, вызванные недовольными сотрудниками, пытающимися скомпрометировать работу базы данных и нанести ущерб компании.

• Стихийные бедствия. К этой категории относятся пожары, землетрясения, наводнения и перебои в подаче электроэнергии.

Восстановление транзакций базы данных использует данные в журнале транзакций для восстановления базы данных из несовместимого состояния в согласованное состояние.

Прежде чем продолжить, необходимо изучить четыре важных понятия, которые влияют на процесс восстановления:

• Протокол лог с опережением записи гарантирует, что журналы транзакций всегда записываются до того, как какие-либо данные базы данных будут фактически обновлены. Этот протокол гарантирует, что в случае сбоя база данных может быть впоследствии восстановлена до согласованного состояния с использованием данных в журнале транзакций.

• Избыточные журналы транзакций (несколько копий журнала транзакций) гарантируют, что сбой физического диска не повлияет на способность СУБД восстанавливать данные.

• Буферы базы данных – временные области в основной памяти, используемые для ускорения дисковых операций. Для сокращения времени обработки программное обеспечение СУБД считывает данные с физического диска и сохраняет их копию в «буфере» в памяти. Когда транзакция обновляет данные, она фактически обновляет копию данных в буфере, потому что этот процесс намного быстрее, чем каждый раз доступ к физическому диску. Буферы, которые содержат обновлённые данные, записываются на физический диск во время одной операции, тем самым экономя значительное время обработки.

CC BY

• Контрольные точки базы данных - это операции, в которых СУБД записывает все свои обновлённые буферы в память (также известные как грязные буферы) на диск. Пока это происходит, СУБД не выполняет никаких других запросов. Операция контрольной точки также регистрируется в журнале транзакций. В результате этой операции физическая база данных и журнал транзакций будут синхронизированы. Эта синхронизация необходима, потому что операции обновления обновляют копию данных в буферах, а не в физической базе данных. Контрольные точки автоматически и периодически выполняются СУБД в соответствии с определенными операционными параметрами (такими как верхний водяной знак для размера журнала транзакций или объёма незавершённых транзакций), но также могут выполняться явно (как часть оператора транзакции базы данных) или неявно (как часть операции резервного копирования базы данных). Конечно, слишком частые контрольные точки могут повлиять на производительность транзакций; слишком редкие контрольные точки могут повлиять на производительность восстановления базы данных. В любом случае, контрольно-пропускные пункты выполняют очень практическую функцию. Как вы увидите далее, контрольные точки также играют важную роль в восстановлении транзакций.

Процесс восстановления базы данных включает в себя приведение базы данных в согласованное состояние после сбоя. Процедуры восстановления транзакций обычно используют методы отложенной записи и сквозной записи.

Когда процедура восстановления использует метод отложенной записи (также называемый отложенным обновлением), операции транзакции не сразу обновляют физическую базу данных. Вместо этого обновляется только журнал транзакций. База данных физически обновляется только данными из зафиксированных транзакций, используя информацию из журнала транзакций. Если транзакция прерывается до того, как достигнет точки фиксации, в базу данных не нужно вносить никаких изменений (без ROLLBACK), поскольку она никогда не обновлялась. Процесс восстановления для всех запущенных и подтверждённых транзакций (до сбоя) состоит из следующих шагов:

1. Определить последнюю контрольную точку в журнале транзакций. Это последний раз, когда данные транзакции были физически сохранены на диск.

2. Для транзакции, которая началась и была зафиксирована до последней контрольной точки, ничего делать не нужно, поскольку данные уже сохранены.

3. Для транзакции, которая выполнила операцию фиксации после последней контрольной точки, СУБД использует записи журнала транзакций, чтобы повторить транзакцию и обновить базу данных, используя значения «после» в журнале транзакций. Изменения вносятся в порядке возрастания, от самого старого до самого нового.

4. Для любой транзакции, в которой выполнялась операция ROLLBACK после последней контрольной точки или которая оставалась активной (без COMMIT или ROLLBACK) до возникновения ошибки, ничего не нужно делать, поскольку база данных никогда не обновлялась.

Когда процедура восстановления использует метод сквозной записи (также называемый немедленным обновлением), база данных немедленно обновляется операциями транзакции во

CC BY

время выполнения транзакции, даже до того, как транзакция достигает своей точки фиксации. Если транзакция прерывается до того, как достигнет точки фиксации, необходимо выполнить операцию ROLLBACK, чтобы восстановить базу данных в согласованное состояние. В этом случае операция ROLLBACK будет использовать значения «до» журнала транзакций. Процесс восстановления состоит из следующих шагов:

1. Определите последнюю контрольную точку в журнале транзакций. Это последний раз, когда данные транзакции были физически сохранены на диск.

2. Для транзакции, которая началась и была зафиксирована до последней контрольной точки, ничего делать не нужно, поскольку данные уже сохранены.

3. Для транзакции, которая была зафиксирована после последней контрольной точки, СУБД повторно выполняет транзакцию, используя значения «после» журнала транзакций. Изменения применяются в порядке возрастания, от самого старого до самого нового.

4. Для любой транзакции, в которой выполнялась операция ROLLBACK после последней контрольной точки или которая оставалась активной (без COMMIT или ROLLBACK) до возникновения ошибки, СУБД использует записи журнала транзакций для ROLLBACK или отмены операций, используя «до» значения в журнале транзакций. Изменения применяются в обратном порядке, от самых новых до самых старых.

Восстановление базы данных восстанавливает базу данных из заданного состояния в предыдущее согласованное состояние. Восстановление базы данных запускается при возникновении критического события, такого как аппаратная ошибка или ошибка приложения.

© Мамедли Р.Э., 2022

УДК 004.94
https://doi.org/10.36906/AP-2022/10

Позов Р.А., Рассадкин Е.Ю.
Донской государственный технический университет
г. Ростов-на-Дону, Россия

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПЛАТФОРМ ДЛЯ СОЗДАНИЯ САЙТОВ

Аннотация. В статье рассматривается актуальность платформ для создания сайтов. Для конференции мы решили выбрать шесть самых распространённых платформ для создания веб-сайтов, такие как WordPress, WIX, uCoz, Nethouse, Joomla и uKit. В данной статье мы совершаем полный анализ данных платформ и проводим сравнительную характеристику.

Ключевые слова: платформа; характеристика; WordPress; WIX; uCoz; Nethouse; Joomla; uKit.

Pozov R.A., Rassadkin E.Y.
Don State Technical University
Rostov-on-Don, Russia

COMPARATIVE CHARACTERISTICS OF WEBSITE CREATION PLATFORMS

Abstract. The article discusses the relevance of platforms for creating websites. For the conference we decided to choose the six most common platforms for creating websites, such as WordPress WIX, uCoz, Nethouse, Joomla and uKit. In this article we make a complete analysis of these platforms and conduct a comparative characteristic.

Keywords: platform; characteristics; WordPress; WIX; uCoz; Nethouse; Joomla; uKit.

В настоящее время для создания сайтов все чаще используются платформы, которые позволяют воплощать многофункциональные интернет-ресурсы без знания специализированных языков программирования. Конструкторы защищены от вирусов и атак благодаря высоким стандартам защиты, есть множество вариантов оформления страниц, административные панели интуитивно понятны и просты, что привлекает пользователей.

У каждого сайта есть свои технические требования и особенности. Поэтому конструкторы сайтов должны полностью перекрывать их. Для интернет-магазина необходимо подключение платёжных систем, служб доставки товара. Для сайта, с размещённой на нем информацией о компании, нужны удобные инструменты вёрстки. Существующие платформы условно делятся на две группы: узкоспециализированные и универсальные.

Первая группа предполагает возможность создания только блога, интернет-магазина и т. д. Вторая – запуск различных сайтов, начиная с одностраничных версий, заканчивая сложными проектами. При этом наличие избыточного функционала не всегда идёт на пользу,

CC BY

производительность может упасть, интерфейсы становятся сложнее для понимания. Специализированные платформы дают возможность выбора набора необходимых инструментов, которые можно легко подключать и отключать.

Современные конструкторы или платформы для создания сайтов обладают рядом преимуществ:

1. Простота в использовании. Даже новичок способен разобраться в функционировании платформы и создать сайт без использования кодов;

2. Для смены дизайна нужно нажать одну кнопку, контент при этом не исчезнет, поскольку хранится в отдельном месте;

3. Есть множество шаблонов оформления сайта, возможность подключения платных модулей;

4. Добавление контента, фото и видео файлов интуитивно понятно;

5. Возможность внесения изменений в готовый текст и мгновенная публикация;

6. Высокая степень безопасности. Все данные, введённые пользователем, сохраняются на сервере платформы, который защищён от вирусов и взломов.

Полностью бесплатных конструкторов нет. Но в большинстве случаев функционала без доплат хватает для создания качественного ресурса. Если пользователю необходимы какие-то расширения, их можно дополнительно оплатить.

Для выбора платформы необходимо ориентироваться на такие факторы, как рейтинг и статистика конструктора. При этом рекомендуется учесть:

– простоту работы;

– функциональность;

– качество работы технической поддержки;

– тарифный план.

У каждой платформы есть свои особенности, которые выделяют их в рейтинге конструкторов. Рассмотрим наиболее популярные.

WordPress разрабатывался для создания блогов, но со временем перерос в инструмент, позволяющий создать любой интернет-ресурс. Главной особенностью системы конструктора является управление контентом с открытым исходным кодом. Панель администрирования понятна и проста, наполнять и редактировать сайт легко. Есть множество бесплатных шаблонов, возможность подключения платных опций.

Особенности платформы:

1. Можно править код движка, исходя из своих потребностей;

2. Множество плагинов, шаблонов;

3. Совместимость с большинством хостингов.

Основной код WordPress используется бесплатно. Стоимость подключения отдельных модулей – от 20 долларов.

Платформой пользуются такие известные компании, такие как: Reuters (новости), Vogue (женский журнал), Open Nasa, Sony Music.

CC BY

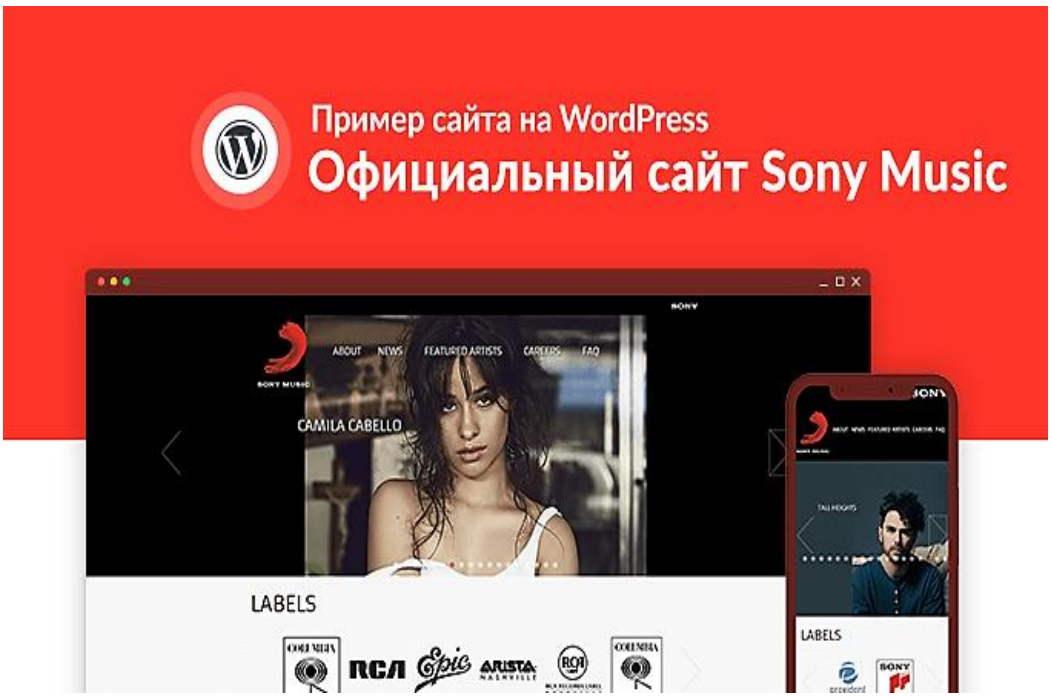


Рис. 1. Сайт на WordPress. Sonymusic.com «Sony Music»

На втором месте по популярности конструктор с мировым охватом, функционирующий почти в 200 странах и на разных языках – Wix. Всего насчитывается более 170 млн пользователей платформы. Популярность сервис получил благодаря особым способам создания дизайнов и шаблонов для сайтов.

Кроме адаптированных заготовок блоков есть сервис, созданный на основе искусственного интеллекта, способный генерировать дизайн с учётом предпочтений пользователя. Для расширения базового функционала существует возможность активации платных приложений. Особенности платформы:

1. Функциональность онлайн-редактора;
2. Большой выбор тематических шаблонов без подписок;
3. Все тарифы подразумевают использование бесплатных SSL-сертификатов;
4. Интеграция популярных сервисов;
5. Возможность создания сайтов любой сложности.

Для тестирования платформы есть бесплатный тариф с рядом технических ограничений. Подключение премиум-опций – от 80 рублей в месяц, модулей для торговли – от 350 руб., отключение рекламы – от 120 руб.

CC BY

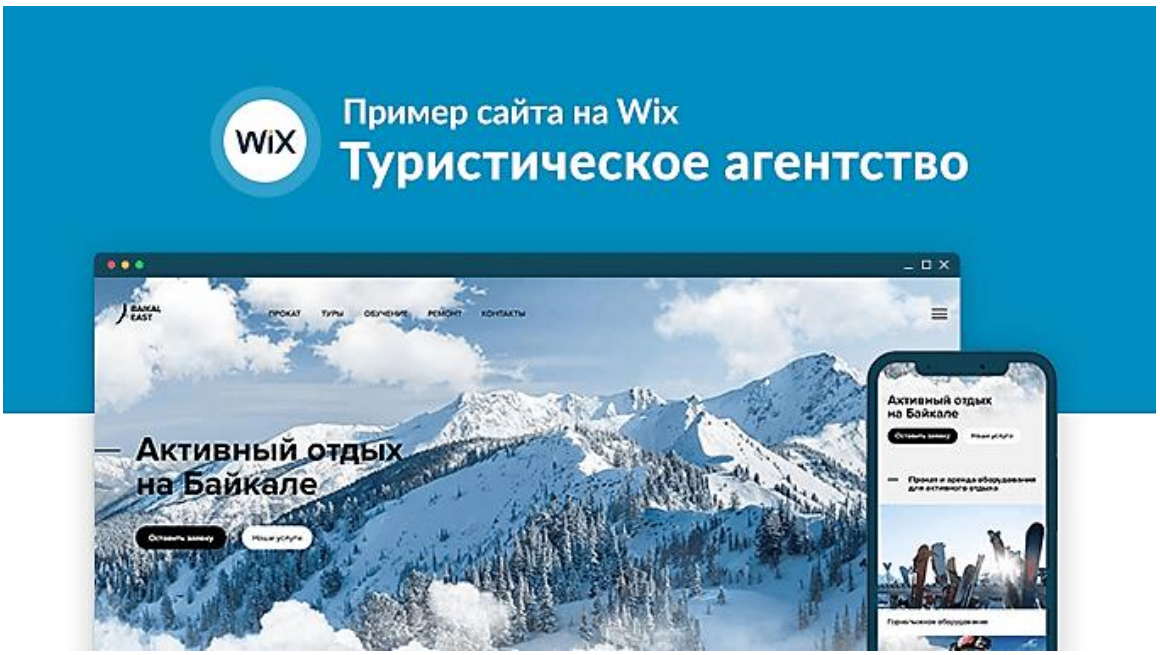


Рис. 2. Сайт на WIX. baikaleast.net/ «Baikal East»

Согласно предпочтениям пользователей, третье место занимает uCoz. Онлайн-конструктор подходит как для создания небольших сайтов, так и при реализации сложных и многофункциональных ресурсов. В сети есть вся необходимая информация с пошаговыми уроками для новичков.

Особенности платформы:

1. Есть платные и бесплатные шаблоны, можно загрузить свой вариант;
2. При выборе бесплатного тарифа можно использовать свой домен второго уровня;
3. Есть модули для создания магазинов;
4. Имеется API;
5. Есть возможность подключения своих PHP-файлов;
6. Интеграция с распространёнными внешними сервисами.

Кроме функционального бесплатного тарифа есть возможность использования следующих услуг:

1. Премиум-тарифы – от 2,09 USD;
2. Магазин – от 6,99 USD;
3. Отключение рекламы – от 4 USD.

Возможности тестирования модулей онлайн-магазина нет. Они доступны после оплаты тарифа. На данной платформе создано множество сайтов с тематическим уклоном, разработками для проведения презентаций, торрентами для скачивания и пр.

CC BY

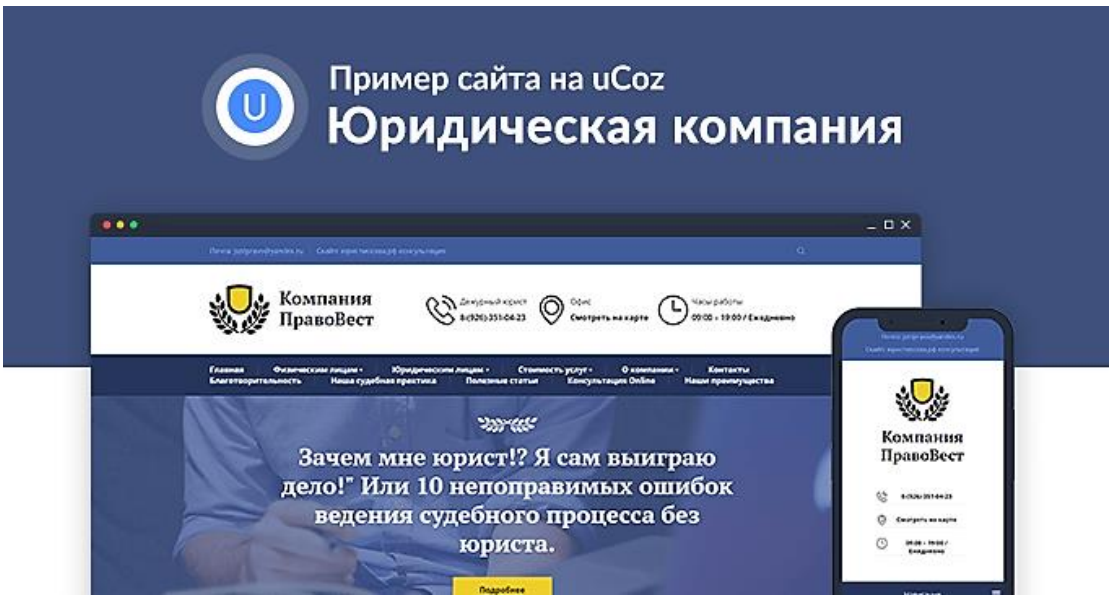


Рис. 3. Сайт на uCoz. Юристмосква.рф «Сайт юридической компании»

На четвёртом месте Nethouse – платформа для создания интернет-ресурсов для малого бизнеса. Конструктор позволяет регистрировать домены, поскольку является официальным регистратором в зоне РФ/ RU. Для создания сайта любой сложности можно воспользоваться готовыми шаблонами, заказать продвижение ресурса профессионалам.

Особенности платформы:

1. Множество готовых шаблонов;
2. Понятный и удобный интерфейс;
3. Встроенная CRM-система;
4. Бесплатные SSL-сертификаты.

Тестирование конструктора – бесплатное, не ограниченное по времени, но с жатым функционалом. Стоимость премиум-подписки – 210 рублей в месяц. Это минимальная цена.

На платформе размещены следующие сайты (компании): интернет-магазин Волкошапок (volkoshapka.ru), интернет-магазин цветов и подарков (blumenbum.ru).

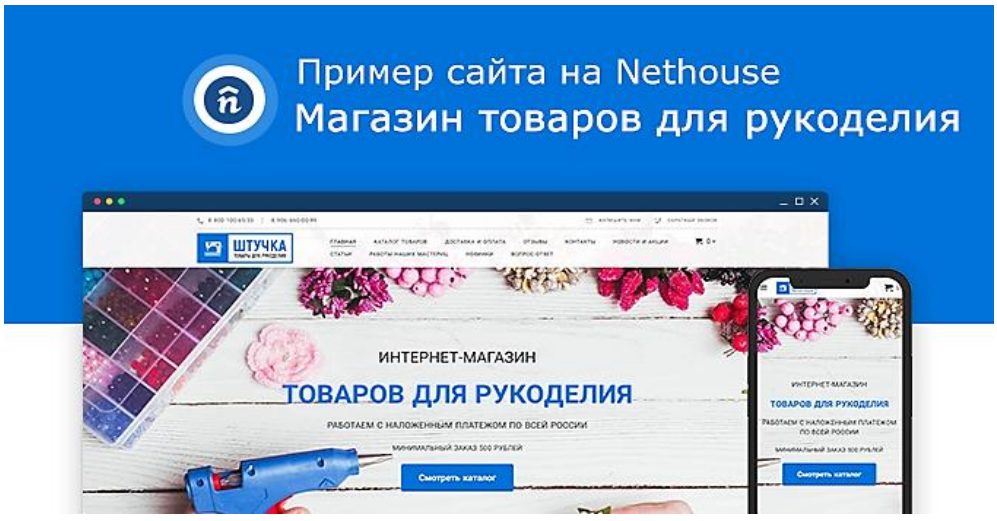


Рис. 4. Сайт на Nethouse. B1ser.ru «Интернет-магазин для рукоделия»

CC BY

Ещё одна платформа с открытым кодом, выдерживающая высокие нагрузки – Joomla. Конструктор рассчитан на людей, обладающих опытом в создании сайтов, поскольку админ-панель имеет сложную структуру.

Особенности платформы:

1. Высокий функционал;
2. Кроме бесплатных шаблонов есть множество платных расширений;
3. Высокая безопасность.

Распространяется Joomla бесплатно. При необходимости можно оплатить шаблоны и дополнительные модули.

На платформе разработаны следующие популярные сайты: Site&Seo, Rizon и SeoLand.

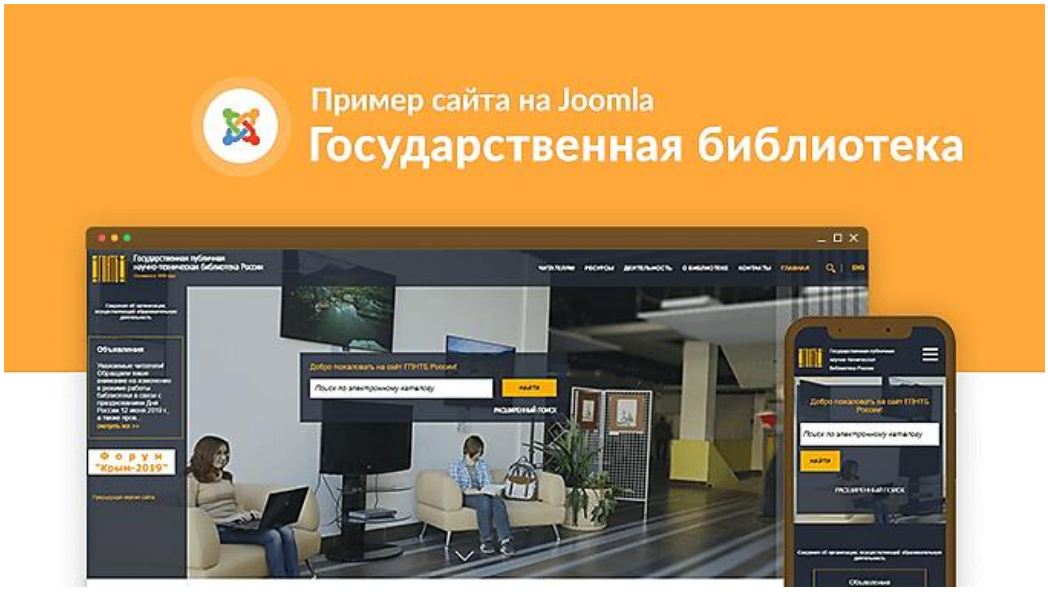


Рис 5. Сайт на Joomla. Gpntb.ru «Государственной библиотеки РФ»

Основной аудиторией платформы uKit являются предприниматели, бизнесмены разного уровня, сфера услуг, частные мастера. Данный конструктор возможность создания небольшого, но функционального сайта. Для старта есть готовые тематические шаблоны, модуль интернет-магазина с базовыми возможностями торговли в онлайн.

Особенности платформы:

1. Бесплатные шаблоны;
2. Понятный интерфейс блочного редактора;
3. Интеграция с платёжными системами и другими востребованными сервисами;
4. Отсутствие ограничений по объёму трафика, места на диске;
5. Автоматическое создание резервных копий сайта;
6. Безлимитные SMS-уведомления, приходящие с сайта.

Тестовый период конструктора составляет 15 дней, минимальная подписка стоит 2,5 доллара в месяц. Бесплатный тариф отсутствует.

На платформе работают: фотограф Евгений Хмелев (http://eugine-khmelev.com/), флорист-декоратор Оксана Шальнева (http://oksana-florist.ru/) и др.

CC BY

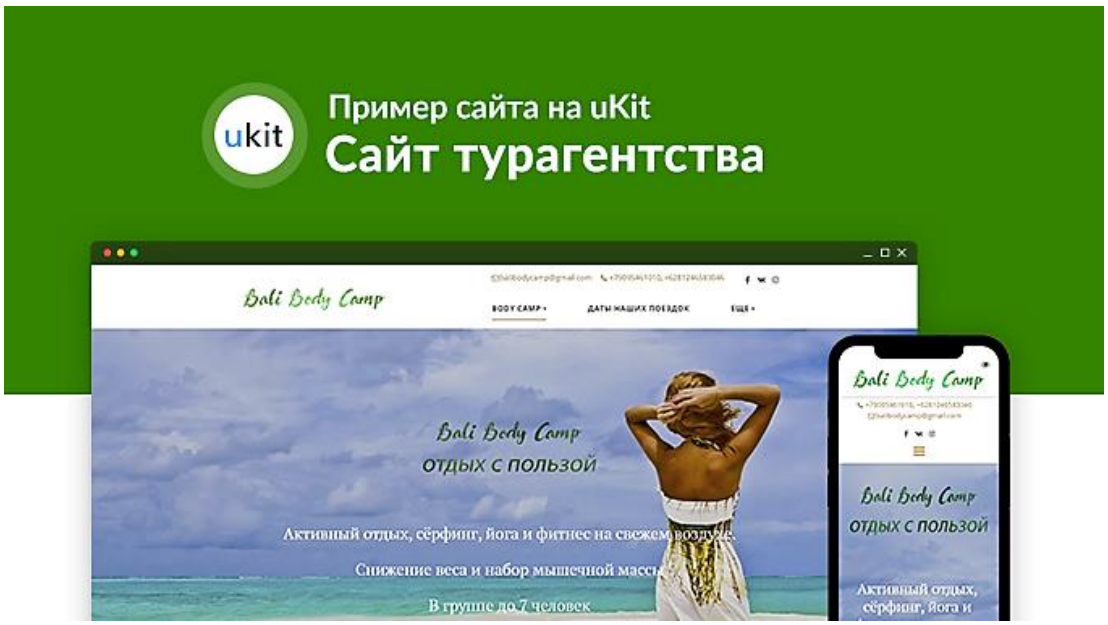


Рис. 6. Сайт на uKit. Bailbodycamp.com «Сайт турагентства»

Сравним платформы, занимающие лидирующие позиции: WordPress и WIX, при этом важно отметить функционал и удобство использования. Оба конструктора не уступают друг другу в количестве готовых решений для создания сайтов. Они просты в использовании и доступны для новичков. Стоимость платформ также мало отличается.

Главное отличие состоит в том, что на WordPress пользователь может самостоятельно править код сайта под свои нужды. Сделать это на платформе-конкуренте нельзя. Но у WIX есть сервис, который учитывает пожелания пользователя и генерирует под них шаблоны.

У WIX нет представительства в России. Все возникшие проблемы необходимо решать через головной офис, расположенный в Нью-Йорке. Но на WordPress для безопасной работы требуется подключение надёжного хостинга. В плане безопасности WIX превосходит конкурента.

Платформы для создания сайтов – возможность попробовать себя в сфере web-разработок, не имея необходимого опыта и знаний. Поэтому конструкторами пользуются не только владельцы мини-блогов. Среди клиентов есть известные предприниматели, популярные фирмы и интернет-магазины. Разнообразие конструкторов позволяет подобрать инструмент, способный полностью реализовать задумки разработчика.

© Позов Р.А., Рассадкин Е.Ю., 2022

УДК 004.4'22
https://doi.org/10.36906/AP-2022/11

Волков В.Н.
канд. техн. наук
Новиков С.В.
ORCID: 0000-0003-1055-0113, канд. техн. наук
Герасимова Н.М.
ORCID: 0000-0002-5065-5078
Полонская Т.С.
ORCID: 0000-0003-0184-3889
Орловский государственный университет им. И.С. Тургенева
г. Орел, Россия

ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ФРЕЙМВОРКА VUE.JS ДЛЯ СОЗДАНИЯ ГИБКИХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Аннотация. В статье рассмотрены преимущества и недостатки использования популярного фреймворка Vue.js для создания пользовательских интерфейсов. Выполнен сравнительный анализ с фреймворком React по популярности. Также представлен график популярности по статистике запросов, которую предоставляет GoogleTrends. Дополнительно, приведены активно развивающиеся библиотеки интерфейсов «BootstrapVue» и «Vuetify», их популярность на основе запросов. Проанализирован способ создания приложений с помощью различных библиотек.

Ключевые слова: JS-фреймворк; javascript; фреймворк; компоненты; мобильные адаптивные сайты; пользовательские интерфейсы.

Volkov V.N.
Ph.D.
Novikov S.V.
ORCID: 0000-0003-1055-011, Ph.D.
Gerasimova N.M.
ORCID: 0000-0002-5065-5078
Polonskaya T.S.
ORCID: 0000-0003-0184-3889
Orel State University named after I.S. Turgenev
Orel, Russia

PROSPECTS FOR USING THE VUE FRAMEWORK.JS FOR CREATING FLEXIBLE USER INTERFACES

CC BY

Abstract. The article discusses the advantages and disadvantages of using the popular Vue framework.js for creating user interfaces. A comparative analysis was performed with the React framework by popularity. There is also a graph of popularity based on query statistics provided by Google Trends. Additionally, the actively developing libraries of the "Bootstrapvue" and "Vuetify" interfaces, their popularity based on queries, are given. The method of creating applications using various libraries is analyzed.

Keywords: JS framework; javascript; framework; components; mobile adaptive sites; user interfaces.

Выбирая фреймворк для создания пользовательских интерфейсов, стоит рассмотреть основные их виды, оценить все возможные положение и отрицательные стороны. Изначально можно оценить уровень интереса других разработчиков к фреймворкам. Согласно опросам на «The State of JavaScript», результаты представлены на рисунке 1, лидирующие позиции занимает «Svelte», «Vue.js» и «React» [1, с. 70]. Большинство специалистов считает, что освоить «Vue» намного легче чем другие популярные JS-фреймворки. Документация – еще одна сильная сторона «Vue» – так считают более 60% опрошенных разработчиков [2, с. 132]. Аналогичное число респондентов (56%) отмечает производительность данного решения как одну из его самых сильных сторон. Разберемся подробнее, что же такое Vue.js?

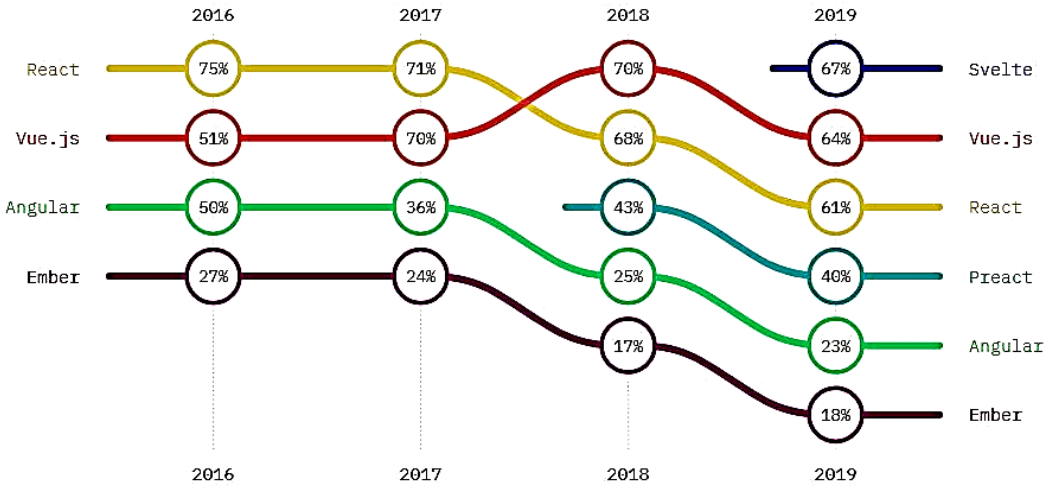


Рис. 1. Результаты опроса «The State of JavaScript»

Касаемо популярности данных фреймворков в России складывается ситуация представленная на рисунке 2, согласно статистике из «Google Trends» (код доступа: https://clck.ru/YF8BE). Следовательно, основными конкурентами при выборе фреймворка для разработки остаются «React» и «Vue.js».

Основываясь на статистике выше, можно отметить, что большинство разработчиков отдают предпочтение «React.js», однако это не фреймворк, а библиотека с широким функционалом, где инструмент можно использовать без дополнительной интеграции.

CC BY

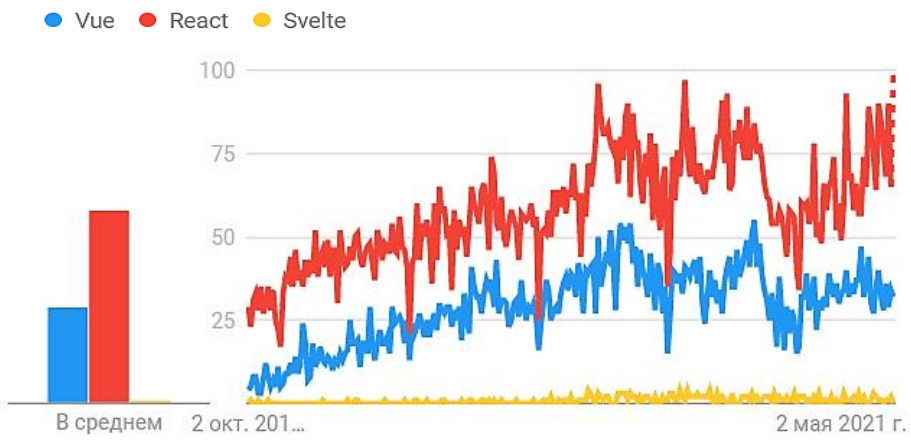


Рис. 2. Статистика «Google Trends» по запросам «Svelte», «Vue.js» и «React»

В свою очередь «Vue.js» – это именно JS-фреймворк. Если сравнивать «React» и «Vue.js» по производительности, то «Vue.js» отличается более высокой производительностью и стабильностью, в силу использования виртуальной DOM.

Одним из основных критериев сравнения выступает размерность. Относительно этого показателя «React» (около 100КБ) уступает «Vue.js» (около 80КБ). Хотя следует отметить, что для «React» необходимо использование сторонних библиотек, а «Vue.js» же – проект с обширным набором стандартных возможностей.

Говоря о гибкости, можно отметить, что ядро «React» дает только основной функционал, что позволяет разработчику выбрать дополнительный инструмент самостоятельно, «Vue.js» же дает большой набор стандартных инструментов, которые позволяют решать задачи, где при использовании «React» необходимо было бы подключение дополнительных библиотек.

«Vue.js» является исключительным инструментом для любого разработчика, так как все его функции находятся в свободном доступе. Для удобства использования разработчики могут просто назвать функцию так, как они хотят. Каждая деталь может иметь дискретную функцию, что значительно упрощает модификацию веб-приложения в соответствии с конкретными требованиями [3, с. 210].

«Vue.js» – гибкая технология, которую можно использовать при разработке приложений без использования многочисленных ресурсов. «Vue.js» имеет гибкий дизайн, который подчеркивает состав компонентов и декларативный рендеринг.

Архитектурный стиль фреймворка определяется моделью MVVM (Модель-Представление-Модель представления). Прогрессивные характеристики, необходимые в многогранных приложениях: управление состоянием, маршрутизация - представлены через пакеты и библиотеки, где «Nuxt.js» стал наиболее популярным решением.

Основными концепциями «Vue.js» являются, компоненты, конструкторы, директивы и переходы. Изначально необходимо создать новый экземпляр «Vue.js», где указываются элемент для рендеринга, данные, вычисляемые свойства и методы [4, с. 49].

При разработке следует придерживаться идеи разбиения сложных вещей на части (компоненты) и связывать их наиболее удобным образом. Таким образом, компоненты помогают расширить основные html элементы и повторно использовать часть UI. Во время

CC BY

проектирования, приложение разбивается на независимые части, которые в дальнейшем формируют древовидную структуру компонентов.

Применение директив даёт возможность дополнительных функциональности html-элементам. С помощью префиксов «v-» можно добавлять обработчики событий, управлять отображением элементов и передавать данные. Так, появляется возможность использовать условия и циклы следующих видов, представленных на рисунке 3.

```
<p v-if="x > 5 ">X больше 5</p>
<p v-else> X меньше 5 </p>

<div>
  <span v-for="n in 10">{{ n }} </span>
</div>
```

Рис. 3. Разработка с использованием префиксов «v-»

Так же при обновлении объектов DOM и их добавлении или удалении «Vue.js» даёт возможность сопровождать этот процесс анимацией.

В связи со стремительным ростом популярности «Vue.js», появляются эффективные для создания интерфейсов библиотеки, как, например, «BootstrapVue», «Vuetify», «Vue Material» и «Vux», статистика Google Trends» по запросам которых представлена на рисунке 3. Самой популярной является «Vuetify», так как она нацелена на создание чистых, семантических компонентов с использованием принципов «Google Material Design». В состав входит более 80 компонентов, которые построены на простых и чётких именах свойств для более простого запоминания.

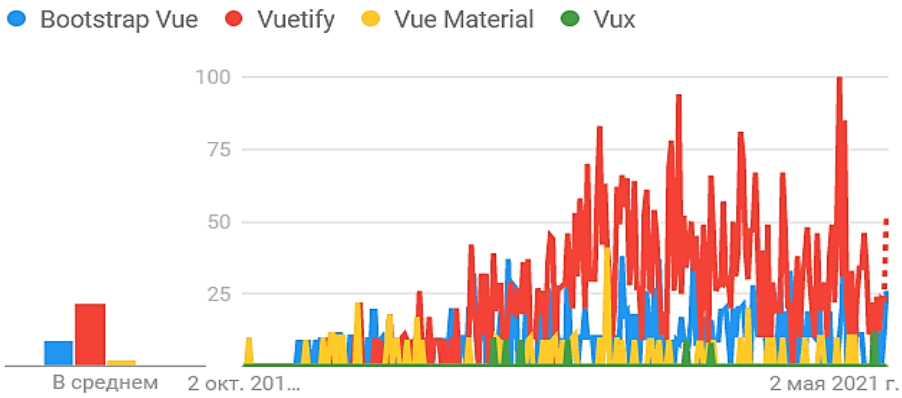


Рис. 4. Статистика «Google Trends» по запросам «Bootstrap Vue», «Vuetify», «Vue Material» и «Vux»

Вторым по популярности стал «Bootstrap Vue», так как «Bootstrap» стал популярным фреймворком HTML / CSS для создания мобильных адаптивных сайтов. Однако для своих основных функций он, как правило, использует «jQuery», а также обширный список компонентов, таких как оповещения и модальные элементы. Поэтому использование «Bootstrap» с «Vue.js», даёт возможность удалить необходимость в «jQuery» (код доступа: https://2019.stateofjs.com/front-end-frameworks). Обладая более чем 85 компонентами, более чем 45 доступными плагинами, несколькими директивами и более 1200 значками, «BootstrapVue» предоставляет одну из наиболее полных реализаций компонентов «Bootstrap v4» и системы сетки.

CC BY

Для сравнения их преимуществ рассмотрим создание одинаковых компонентов «Card» с применением «Bootstrap Vue» и «Vuetify» соответственно. Использование «Bootstrap Vue» представлено на рисунке 5. Пример использования «Vuetify» представлен на рисунке 6.

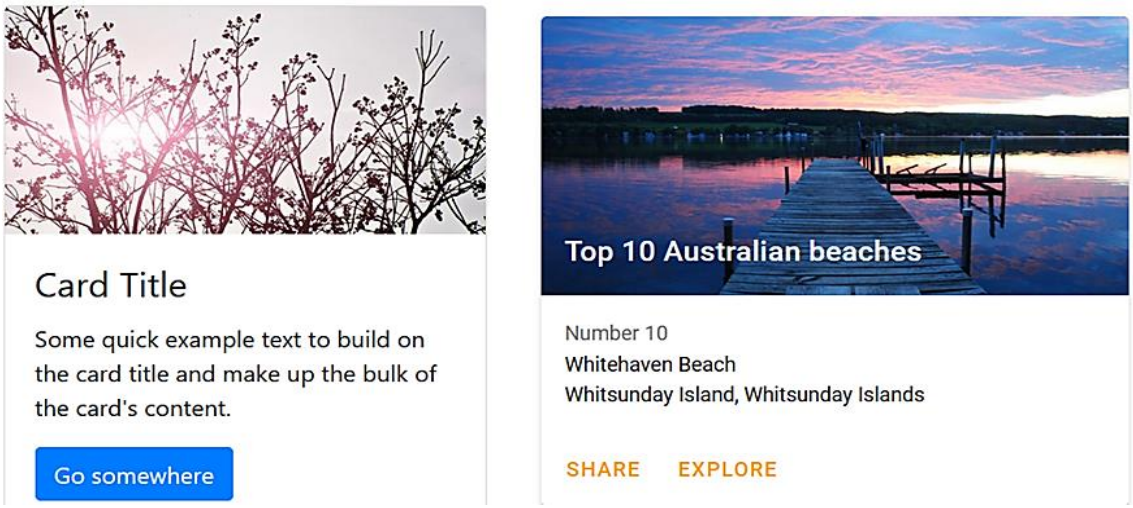
```
<b-card
    title="Card Title"
    img-src="https://picsum.photos/600/300/?image=25"
    img-alt="Image"
    img-top
    tag="article"
    style="max-width: 20rem;"
    class="mb-2">
    <b-card-text>
      Some quick example text to build on the card title and make up the bulk of the card's content.
    </b-card-text>
    <b-button href="#" variant="primary">Go somewhere</b-button>
  </b-card>
```

Рис. 5. Пример разработки с использованием «Bootstrap Vue»

```
<v-card class="mx-auto" max-width="400">
  <v-img class="white--text align-end" height="200px" src="https://cdn.vuetifyjs.com/images/cards/docks.jpg"/>
  <v-card-title>Top 10 Australian beaches</v-card-title>
  </v-img>
  <v-card-subtitle class="pb-0">Number 10</v-card-subtitle>
  <v-card-text class="text--primary">
    <div>Whitehaven Beach</div>
    <div>Whitsunday Island, Whitsunday Islands</div>
  </v-card-text>
  <v-card-actions>
    <v-btn color="orange" text>Share</v-btn>
    <v-btn color="orange" text>Explore</v-btn>
  </v-card-actions>
</v-card>
```

Рис. 6. Пример разработки с использованием «Vuetify»

Ниже представлены результаты разработки – визуальное представление компонентов «Card» на рисунке 7 соответственно.



а) б)

Рис. 7. «Card»: а) «Bootstrap Vue»; б) «Vuetify»

Как видно из рисунка 7, визуально компоненты имеют много общего, у каждого есть изображение, заголовок, некая информация и кнопки для переходов на дальнейший контент. Если сравнивать скрипты, то в целом их структура и названия элементов схожа.

Таким образом, выбор библиотеки не сильно влияет на отображение элементов на странице, однако выбор фреймворка может сильно повлиять на процесс создания гибких пользовательских интерфейсов, в некоторых случаях усложнив его, а в некоторых способен помочь разработчику сократить время работы над проектом.

Литература

1. Nelson B. Getting to Know Vue.js: Learn to Build Single Page Applications in Vue from Scratch. Minnesota: Apress, 2018. 280 p. https://doi.org/10.1007/978-1-4842-3781-6

2. Macrae C. Vue.js: Up and Running: Building Accessible and Performant Web Apps. O'Reilly Media, Inc., 2018. 174 p.

3. Hanchett E., Listwon B. Vue.js in Action. Manning, 2018. 375 p.

4. Шкарбан, Ф.В, Халилова З.Э., Абдуллаев А.Н. Javascript-фреймворки и библиотеки // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. № 3(17). С. 48-52.

© Волков В.Н., Новиков С.В., Герасимова Н.М., Полонская Т.С., 2022

УДК 004.94
https://doi.org/10.36906/AP-2022/12

Рачев А.А., Харсеев Н.А.
Донской государственный технический университет
г. Ростов-на-Дону, Россия

ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ БИБЛИОТЕЧНЫХ КАТАЛОГОВ

Аннотация. В статье рассматривается актуальность внедрения программных средств и информационных технологий в отрасль развития библиотек. Исследуется необходимость в разработке автоматизированных библиотечных каталогов для оптимизации деятельности библиотек. Описывается процесс построения диаграммы бизнес–процессов библиотеки с помощью программного средства All Fusion Modeller и диаграммы декомпозиции. Строится структура базы данных и информационная система для автоматизации библиотечных каталогов на языке C#.

Ключевые слова: информационные системы; автоматизация; программирование; библиотека; C#.

Rachev A.A., Kharseev N.A.
Don State Technical University
Rostov-on-Don, Russia

LIBRARY SYSTEM FOR AUTOMATING LIBRARY CATALOGS

Abstract. The article discusses the relevance of the implementation of software and information technology in the development of libraries. The need for the development of automated library catalogs to optimize the activities of libraries is investigated. Describes the process of building a diagram of the library's business processes using the All Fusion Modeller software and a decomposition diagram. The structure of the database and the information system for the automation of library catalogs in the C # language are being built.

Keywords: information systems; automation; programming; library; C#.

В данный момент, основное направление в отрасли развития библиотек занимает разработка автоматизированных библиотечных каталогов. Наиболее явным преимуществом автоматизации библиотек является возможность предоставления наиболее полных и достоверных данных о состоянии библиотечного фонда, а также быстрого поиска необходимой информации.

Внедрение автоматизированных информационных систем в отрасль деятельности библиотек позволяет существенно оптимизировать работу сотрудников библиотеки,

CC BY

сократить время обслуживания читателей и позволит сделать более удобным процесс поиска более наглядным и эффективным.

Главная задача в разработке информационных библиотечных систем это автоматизация основных процессов, протекающих в библиотеке: организация, ведение, учёт и систематизация библиотечного фонда, регистрация читателей и пр. Подобные системы позволяют небольшим библиотекам использовать современные технологии без больших затрат [1, с. 857]. Основу автоматизированных библиотечных систем составляет система электронных каталогов, в которых содержится полная информация об изданиях, которые хранятся в библиотечных фондах.

Внедрение автоматизированной системы в любую отрасль, как правило, приводит к изменению существующих организационных структур и методов управления, требует более чёткого регулирования рабочего процесса и процедур управления, оптимизации использования и создания новых стандартов [2, с. 182].

В настоящее время, на высококонкурентном рынке буквально ни одно предприятие не в состоянии вести успешный бизнес, не используя современными информационными технологиями для своей отрасли экономики. Если рассматривать информационную систему в целом, то она представляет собой организационно–структурированный, взаимосвязанный набор средств и методов информационных технологий для хранения, обработки и выдачи информации, с помощью которых можно добиться поставленной цели [3, с. 84-85]. В современном понимании информационной системы предполагается, что персональный компьютер используется как основное техническое средство обработки информации.

Автоматизированные информационные системы обеспечивают усовершенствование таких функций как:

1. запись данных и их хранение;
2. обработку информации;
3. поиск и фильтрацию нужной информации.

Главной целью деятельности библиотеки является предоставление населению бесплатного доступа к литературе разных видов и жанров. На рисунке 1 представлена контекстная диаграмма, отражающая процессы деятельности библиотеки.

На вход в систему поступают книги, которые приобретает библиотека для последующей поставки читателям, а также книги, которые читатели возвращают в библиотеку. За обеспечением работы самой системы следит персонал библиотеки.

Пользователи системы к ней с запросами для того, чтобы получить необходимую им информацию о наличии определённой книги или чтобы вернуть уже прочитанную книгу.

Деятельность библиотечной системы регламентирована определёнными правилами. Их регулируют нормативные акты, а также устав самой библиотеки, которые определяют порядок выдачи, учёта, отчётности и документирования, правила поведения персонала в различного рода ситуациях.

CC BY

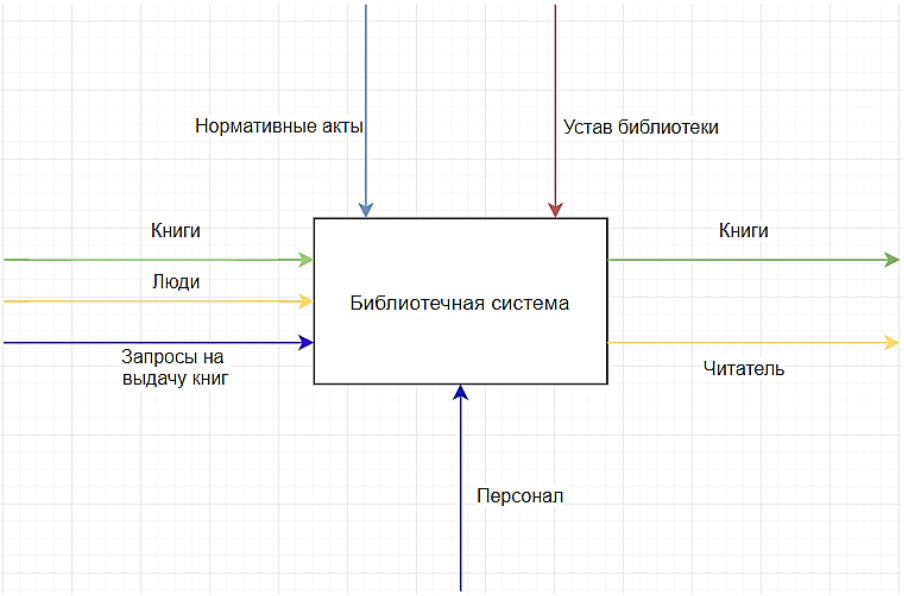


Рис. 1. Диаграмма деятельности библиотеки

После построения контекстной диаграммы строиться диаграмма декомпозиции деятельности библиотеки, представленная на рисунке 2.

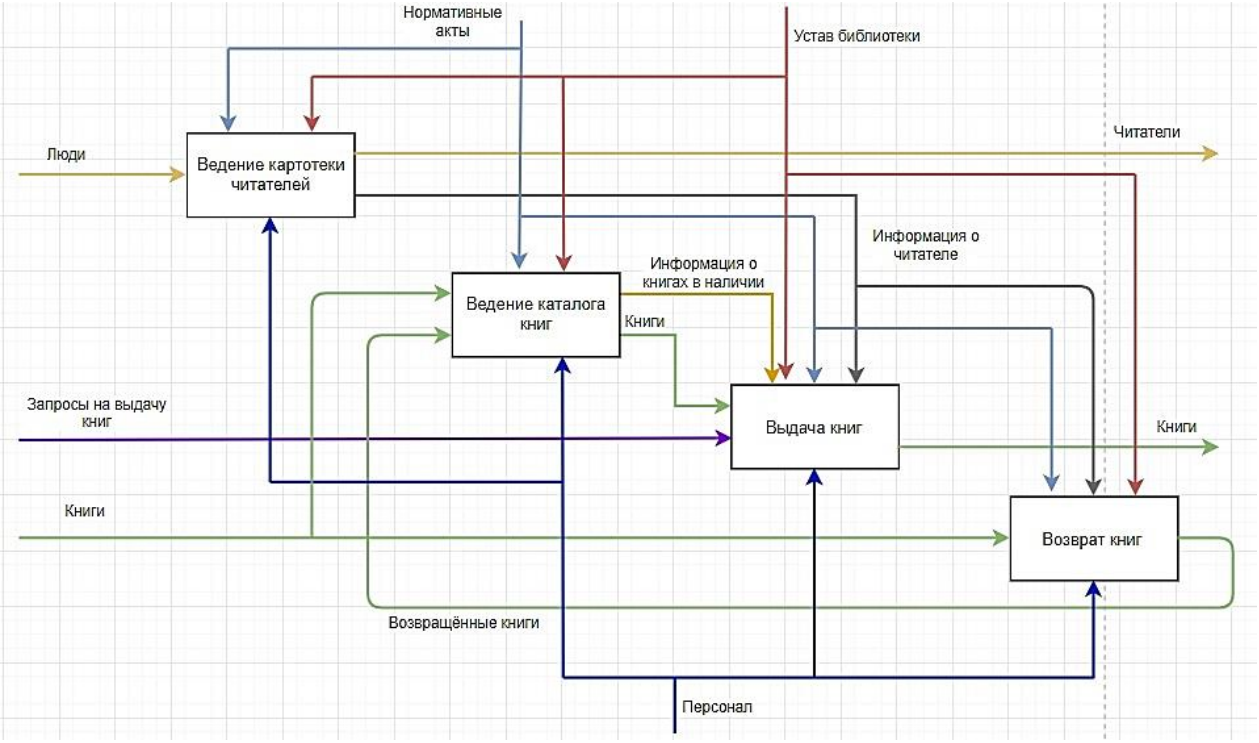


Рис. 2. Диаграмма декомпозиции деятельности библиотеки

Процесс «Ведение каталогов книг» регулируется нормативными актами и уставом библиотеки на основе законов РФ. За добавление удаление и редактирование информации о книгах в базе данных отвечает библиотекарь. Именно там происходит организация каталогов, учёт книг, как выданных на руки читателям, так и находящихся в библиотечном фонде.

CC BY

Процесс «Ведение картотеки читателей» регулируется нормативными актами и уставом библиотеки на основе законов РФ. Добавление, удаление, редактирование данных читателей в базе данных осуществляется библиотекарем.

На основе диаграммы декомпозиции для создания информационной системы автоматизации деятельности библиотеки нами разработана база данных, представленная на рисунке 3.

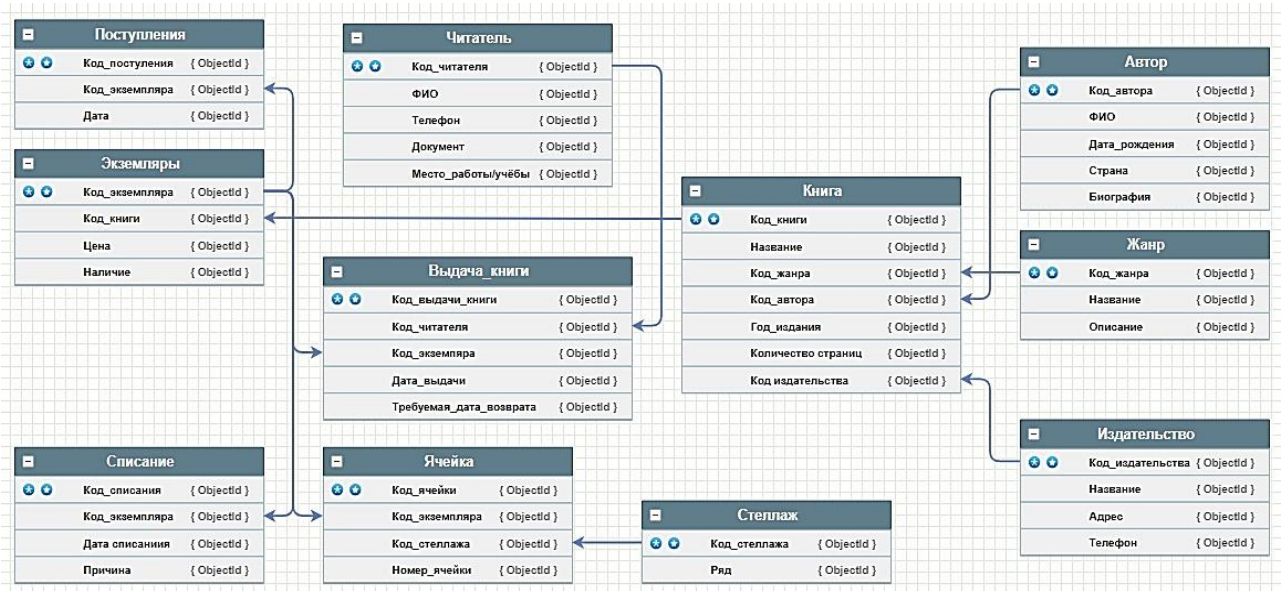


Рис. 3. Автоматизация библиотечных каталогов

Исходя из того, что пользователями разрабатываемой системы будет являться персонал библиотеки в ней должный быть реализованы следующие функции:

1. Возможность добавления, удаления и редактирования данных о читателях и книгах;
2. Фильтрация и поиск необходимой информации;
3. Система формирования отчётов.

На рисунке 4 представлен фрагмент кода для получения информации из базы данных.

```
string connectionString = System.Configuration.ConfigurationManager.
    ConnectionStrings["Bibl_Catalog.Properties.Settings.Библиотечный_каталогConnectionString"].ConnectionString;

string sqlExpression = "SELECT Автор.ФИО as Автор, Книга.Название as Книга, Книга.Год_издания as Год, Выдача_книги.Дата_выдачи"
                     + " FROM ((Автор INNER JOIN Книга ON Автор.[Код автора] = Книга.Код_автора)"
                     + "INNER JOIN Экземпляр ON Книга.Код_книги = Экземпляр.Код_книги)"
                     + "INNER JOIN (Читатель INNER JOIN Выдача_книги ON Читатель.Код_читателя = "
                     + "Выдача_книги.Код_читателя) ON Экземпляр.[Код_экземпляра] = Выдача_книги.Код_экземпляра"
                     + " WHERE (((Выдача_книги.Возвращена)=0) AND ((Читатель.Код_читателя)="
                     + читательDataGridView.Rows[читательDataGridView.CurrentCell.RowIndex].Cells[0].Value.ToString() + "));";

using (SqlConnection conn = new SqlConnection(connectionString))
{
    using (SqlDataAdapter da = new SqlDataAdapter())
    {
        da.SelectCommand = new SqlCommand(sqlExpression, conn);
        DataSet ds = new DataSet();
        da.Fill(ds, "result_name");
        DataTable dt = new DataTable();
        dt = ds.Tables["result_name"];
        dataGridView1.DataSource = dt;
    }
}
```

Рис. 4. Фрагмент кода программы

CC BY

Данный фрагмент кода отвечает за получение информации из базы данных о книгах, имеющихся сейчас в наличие в библиотеке.

На рисунке 5 представлено основное окно приложения. Оно содержит главное меню приложения и кнопки, которые позволяют осуществлять процессы работника библиотеки. Здесь реализованы такие функции каталога библиотеки как: «Выдача книг», «Читатели», «Списание книг», «Поступление книг», а также такие функции системы как: «Справочники», «Произведения», «Отчёты», «Выход».

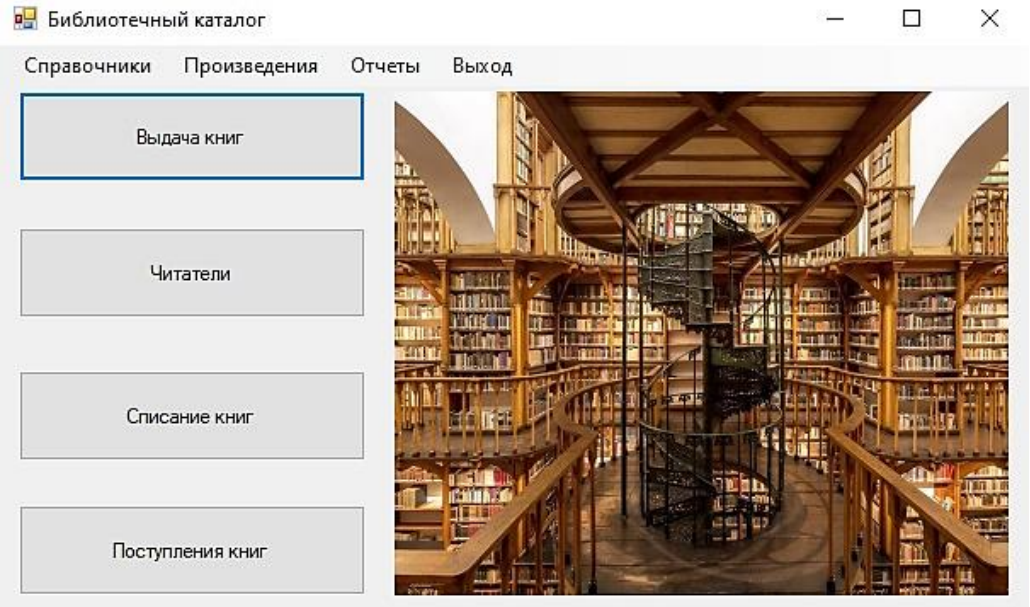


Рис. 5. Пользовательский интерфейс

Представим каждый из этих модулей:

Модуль «Выдача книг» реализована соответствующим модулем. Она предназначена для хранения информации о выдаваемых книгах читателям. Содержит такую информацию, как: имя, фамилия и отчество читателя, какую книгу он взял, дата выдачи книги, дата возврата книги, а также статус выданных книг.

Модуль «Читатели» нужен для регистрации читателей в библиотеке.

Модуль «Списание книг» нужен для того, чтобы помечать экземпляры книг, по каким–то причинам уже непригодными для чтения.

В модуле «Поступление книг» описываются все поступающие в библиотечный фонд книги, а именно их уникальный идентификатор и дата поступления в библиотеку.

Основным назначением разработанной информационной системы является организация библиотечного каталога, а также дальнейшая его работа.

Особенностью, реализованной автоматизированной библиотечной системы является возможность предоставление читателю наиболее полной и достоверной информации о состоянии библиотечного фонда.

Разработанная система имеет следующие основные функции: создание и редактирование списка читателей и книг, ознакомление с биографией различных авторов, вывод различного рода отчётов о книгах и читателях. Основные разработанные элементы и функции включают:

CC BY

разработку главной страницы, меню навигации, страницы работы с оформлением книг, функцию поиска.

Литература

1. Севрюкова А.А. Автоматизированные библиотечные системы // Молодой учёный. 2015. № 13(93). С. 857-859.

2. Летуновская О.В. Использование автоматизированной библиотечно-информационной системы в научной библиотеке вуза // Проблемы и перспективы развития образования в России. 2010. № 5-1. С. 180-184.

3. Егорова А.А., Козлов С.А. Информационные системы: методы и средства проектирования // Научный Вестник Московского государственного технического университета гражданской авиации. 2006. № 105. С. 84-93

© Рачев А.А., Харсеев Н.А., 2022

УДК 519.6
https://doi.org/10.36906/AP-2022/13

Русина Л.Г.
ORCID: 0000-0001-6115-8328, канд. техн. наук
Военная орденов Жукова и Ленина Краснознаменная академия связи
им. Маршала Советского Союза С.М. Буденного
г. Санкт-Петербург, Россия

О МЕТОДАХ РЕШЕНИЯ «ЖЁСТКИХ» СИСТЕМ УРАВНЕНИЙ

Аннотация. В статье рассматриваются методы решения «жёстких» задач. Представленная «жёсткая» система уравнений исследуется на жёсткость и решается различными численными методами. Численное решение системы обыкновенных дифференциальных уравнений кинетики химических реакций при начальных условиях позволило получить значения концентраций всех компонентов как функций времени на всем протяжении реакции. Развитое программное обеспечение современных компьютеров позволяет проводить вычислительные эксперименты в программе Mathcad и решать «жёсткие» задачи.

Ключевые слова: «жёсткая» система уравнений; концентрация; численные методы; шаг интегрирования; сходимость; программа Mathcad.

L.G. Rusina
ORCID: 0000-0001-6115-8328, Ph.D.
Marshal Budyonny Military Signal Academy
Saint-Petersburg, Russia

ON METHODS FOR SOLVING RIGID SYSTEMS OF EQUATIONS

Abstract. The paper considers methods for solving “rigid” problems. The presented “rigid” system of equations is investigated for rigidity and solved by various numerical methods. Numerical solution of the system of ordinary differential equations of kinetics of chemical reactions under initial conditions allowed to obtain values of concentrations of all components as functions of time throughout the reaction. Developed software of modern computers allows to carry out computational experiments in Mathcad program and to solve “hard” problems.

Keywords: “rigid” system of equations; concentration; numerical methods; integration step; convergence; Mathcad.

«Жёсткие» системы дифференциальных уравнений особенно часто появляются при моделировании явлений, где разброс временных характеристик заложен в самой физической природе. Примерами могут служить задачи электротехники, химической кинетики [1, с. 33].

CC BY

«Жёсткие» системы появляются при описании процессов в таких областях, как медицина, биология, социология и др. [2, с. 437].

Пусть химические реакции описываются системой обыкновенных дифференциальных уравнений (ОДУ) следующего вида:

$$\begin{cases} \frac{dc_1}{dt} = -0{,}013c_1 + 10^3 c_1 \cdot c_3 \\ \frac{dc_2}{dt} = 2{,}5 \cdot 10^3 \cdot c_2 \cdot c_3 \\ \frac{dc_3}{dt} = 0{,}013c_1 - 10^3 \cdot c_1 \cdot c_3 - 2{,}5 \cdot 10^3 \cdot c_2 \cdot c_3 \end{cases}, \qquad (1)$$

с начальными условиями:

$$c_1(0) = c_1^0,\ c_2(0) = c_2^0,\ c_3(0) = c_3^0, \qquad (2)$$

где $c_1(t),\ c_2(t),\ c_3(t)$ – концентрации химических (реагирующих) веществ, $c_1(0), c_2(0), c_3(0)$ - концентрации химических веществ при $t = 0$. Значения начальных концентраций для решения системы (1) - (2) приведены в таблице 1.

Таблица 1

Значения начальных концентраций

c_1^0	0 моль/л
c_2^0	0 моль/л
c_3^0	1 моль/л

Допустим, что система дифференциальных уравнений (1) с начальными условиями (2) описывает термодинамические гомогенные реакции в закрытой системе.

Покажем, что система (1) с начальными условиями (2) действительно «жёсткая». Для этого построим матрицу Якоби, найдём корни характеристического уравнения, используя начальные данные в таблице 1:

$$|A - \lambda E| = \begin{vmatrix} -0.013 + 10^3 c_3 - \lambda & 0 & 10^3 c_1 \\ 0 & 2.5 \cdot 10^3 c_3 - \lambda & 2.5 \cdot 10^3 c_2 \\ 0.013 - 10^3 c_3 & -2.5 \cdot 10^3 c_3 & -10^3 c_1 - 2.5 \cdot 10^3 c_2 - \lambda \end{vmatrix} = 0 \qquad (3)$$

Расписывая по элементам последней строки определителя и приводя подобные члены, получим:

$$-\lambda^3 - 3500\lambda^2 - 2500{,}013\lambda = 0 \qquad (4)$$

Находим корни уравнения (4). Получаем $\lambda_1 = 0$ и оставшиеся два корня определяем из квадратного уравнения. Нетрудно вычислить, что $\lambda_2 = 3500{,}7141,\ \lambda_3 = 0{,}7141.$

Определяем коэффициент «жесткости» по формуле [2, с.439]:

$$\mu(t) = \frac{\max\limits_{i=1,m} \mathrm{Re}(-\lambda_i)}{\min\limits_{i=1,m} \mathrm{Re}(-\lambda_i)} = \frac{(-3499{,}2856)}{(-0{,}7144)} \approx 4898{,}2161 \succ\succ 1.$$

Таким образом, установили, что полученная система ОДУ (1)-(2) – «жёсткая».

CC BY

Найти аналитическое решение задачи (1)-(2) сложно, поэтому основным инструментом исследования указанной системы является численный эксперимент. Численное интегрирование системы (1)-(2) было проведено разными методами:

– явным методом Эйлера (I);

– неявным методом Адамса-Моултона (II);

– методом Гира (III) [3, с. 172];

– алгоритмом Булирша-Штера в программе Mathcad (IV).

Отметим, что явный метод Эйлера (условно устойчивый) имеет первый порядок точности, а устойчивый метод Адамса-Моултона с итерациями дает на каждом шаге погрешность аппроксимации дифференциального уравнения разностным методом порядка $O(\tau^2)$, где $\tau = \frac{t}{N}$, N – число узлов на сетке, t – время протекания химической реакции ($t \in [0,\ 1]$), τ – шаг интегрирования.

Решение задачи (1)–(2) с целью выбора наиболее оптимального численного метода из перечисленных, проводилось с различными шагами ($\tau = 0{,}00005$; $\tau = 0{,}0005$; $\tau = 0{,}001$; $\tau = 0{,}01$; $\tau = 0{,}1$) и точностью ($\varepsilon = 0{,}00001$; $\varepsilon = 0{,}0001$).

Использование классических методов (I), (II) первого и второго порядка точности накладывает жёсткие ограничения на шаг интегрирования ($\tau \le 0{,}001$). Действительно, попытка увеличить шаг интегрирования ($\tau > 0{,}001$) приводит к значительным изменениям в численном решении системы ОДУ, к «взрыву погрешностей». Поэтому наряду с методами (I), (II) для решения системы дифференциальных уравнений (1) с начальными условиями (2) следует использовать метод Гира с автоматическим контролем погрешности вычислений и шага интегрирования [3, с. 172] или алгоритм Булирша-Штера в математической системе Mathcad. Известно, что выбор величины шага имеет существенное значение, т. к. от него зависит общий объем вычислений.

Отметим, что решение «жёстких» систем дифференциальных уравнений в среде Mathcad осуществляется с помощью встроенных функций:

1) $Radau(y_0, t_0, t_1, N, F)$;

2) $Stiffb(y_0, t_0, t_1, M, F, J)$ – алгоритм Булирша-Штера;

3) $Stiffr(y_0, t_0, t_1, N, F, J)$ – алгоритм Розенброка:

y_0 – вектор начальных значений в точке t_0 (t_0 – начальная точка расчёта);

t_1 –конечная точка расчёта;

M, N – число шагов численного метода;

F – векторная функция $F(t, y)$, задающая систему обыкновенных дифференциальных уравнений;

J – матричная функция $J(t, y)$, составленная из вектора производных функции $F(t, y)$ по t (правый столбец) и её якобиана (N– левых столбцов).

Преимущество опции *Radau* в том, что не нужно знать якобиан системы ОДУ. Однако это преимущество может стать недостатком, поскольку методы, использующие якобиан могут быть более точными.

Алгоритм решения данной системы (1)–(2) при помощи функции $Stiffb(y_0, t_0, t_1, M, F, J)$ в среде Mathcad представлен на рисунке 1.

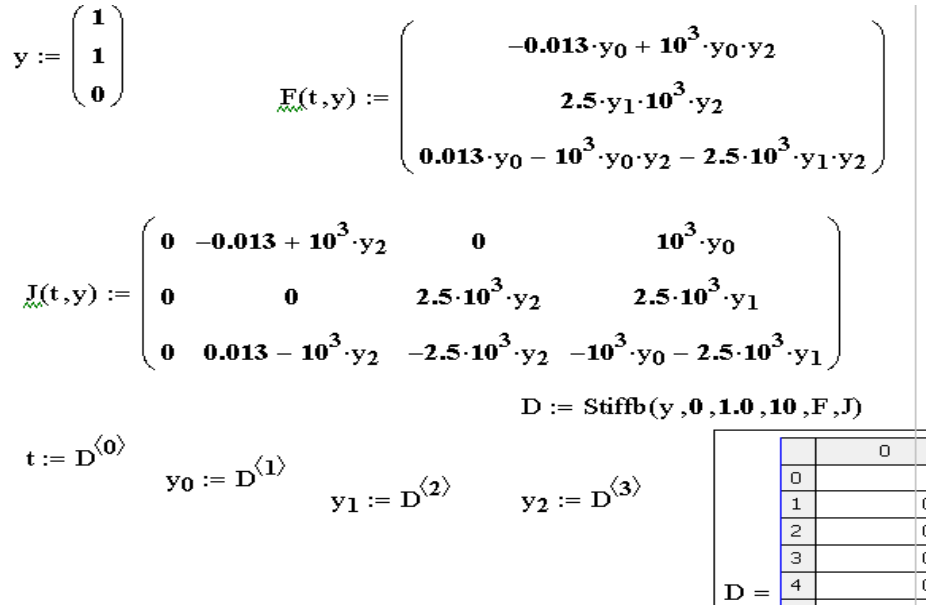


Рис. 1. Применение функции $Stiffb(y_0, t_0, t_1, M, F, J)$

Результаты численных исследований задачи (1)-(2) в различные моменты времени представлены в таблицах 2, 3.

Таблица 2

Численное решение задачи (1)-(2) методом (III)

τ – шаг ε -точность	$c_1(t)$, $c_2(t)$, $c_3(t)$	III	t-время
$\tau = 0{,}00005$ $\varepsilon = 0{,}00001$	$c_1(t)$	$1{,}00000 \cdot 10^{0}$	0
	$c_2(t)$	$1{,}00000 \cdot 10^{0}$	
	$c_3(t)$	$3{,}60401 \cdot 10^{-6}$	
$\tau = 0{,}00005$ $\varepsilon = 0{,}00001$	$c_1(t)$	$9{,}99998 \cdot 10^{-1}$	$0{,}69 \cdot 10^{-2}$
	$c_2(t)$	$1{,}00500 \cdot 10^{0}$	
	$c_3(t)$	$3{,}71401 \cdot 10^{-6}$	
$\tau = 0{,}00005$ $\varepsilon = 0{,}00001$	$c_1(t)$	$9{,}99998 \cdot 10^{-1}$	0,406
	$c_2(t)$	$1{,}01205 \cdot 10^{0}$	
	$c_3(t)$	$3{,}69472 \cdot 10^{-6}$	
$\tau = 0{,}00005$ $\varepsilon = 0{,}00001$	$c_1(t)$	$9{,}99998 \cdot 10^{-1}$	1
	$c_2(t)$	$1{,}02401 \cdot 10^{0}$	
	$c_3(t)$	$3{,}66531 \cdot 10^{-6}$	

CC BY

Таблица 3

Численное решение задачи (1)-(2) методом (IV)

τ – шаг, ε – точность, N – число узлов	$c_1(t)$, $c_2(t)$, $c_3(t)$	IV	t – время
$\tau = 0{,}001$ $\varepsilon = 0{,}00001$ $N = 1000$	$c_1(t)$	$9.9999 \cdot 10^{-1}$	0,001
	$c_2(t)$	$1.0000 \cdot 10^{0}$	
	$c_3(t)$	$3.6104 \cdot 10^{-6}$	
$\tau = 0{,}001$ $\varepsilon = 0{,}00001$ $N = 1000$	$c_1(t)$	$9.9998 \cdot 10^{-1}$	0,002
	$c_2(t)$	$1.0000 \cdot 10^{0}$	
	$c_3(t)$	$3.7113 \cdot 10^{-6}$	
$\tau = 0{,}001$ $\varepsilon = 0{,}00001$ $N = 1000$	$c_1(t)$	$9.9997 \cdot 10^{-1}$	0,004
	$c_2(t)$	$1.0000 \cdot 10^{0}$	
	$c_3(t)$	$3.7141 \cdot 10^{-6}$	
$\tau = 0{,}001$ $\varepsilon = 0{,}00001$ $N = 1000$	$c_1(t)$	$9.9073 \cdot 10^{-1}$	1,0
	$c_2(t)$	$1.0093 \cdot 10^{0}$	
	$c_3(t)$	$3.6653 \cdot 10^{-6}$	
$\tau = 0{,}0005$ $\varepsilon = 0{,}00001$ $N = 2000$	$c_1(t)$	$9.9999 \cdot 10^{-1}$	0,0005
	$c_2(t)$	$1.0000 \cdot 10^{0}$	
	$c_3(t)$	$3.0725 \cdot 10^{-6}$	
$\tau = 0{,}0005$ $\varepsilon = 0{,}00001$ $N = 2000$	$c_1(t)$	$9.9073 \cdot 10^{-1}$	1,0
	$c_2(t)$	$1.0093 \cdot 10^{0}$	
	$c_3(t)$	$3.6653 \cdot 10^{-6}$	

Проведено сравнение между собой полученных решений в различные моменты времени и на рисунках 2–4 представлены кривые изменения концентраций с течением времени.

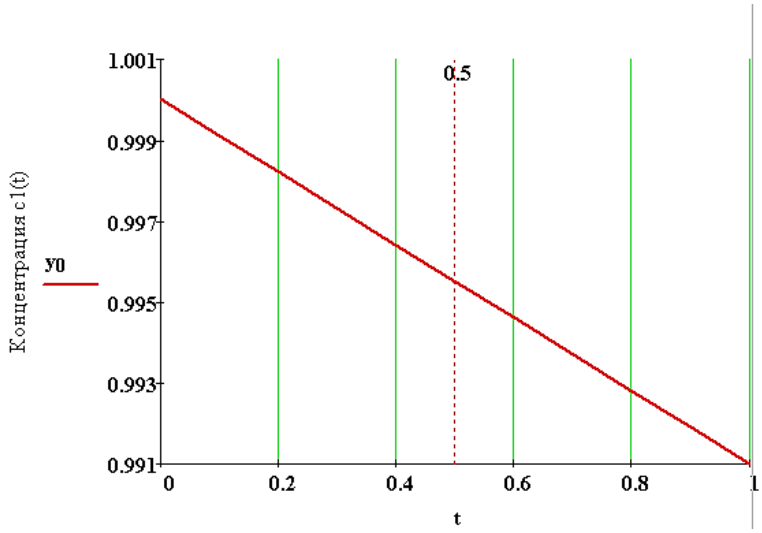


Рис. 2. Изменение концентрации $c_1(t)$

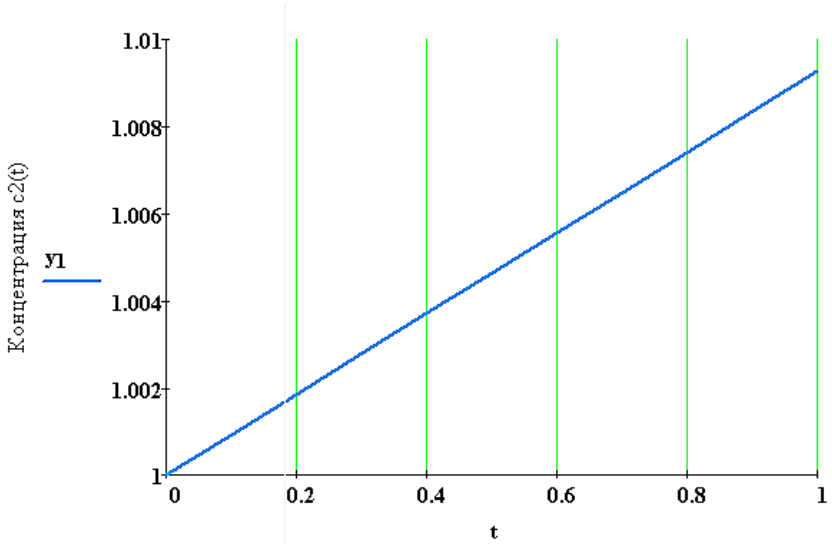


Рис. 3. Изменение концентрации $c_2(t)$

Рис. 4. Изменение концентрации $c_3(t)$

Сравнивая результаты, полученные по методам I, II, III, IV, имеем похожую картину изменения концентрации с течением времени t: (c_1 – убывает и сохраняет постоянное значение равное $9{,}9355\cdot10^{-1}$; c_2 – монотонно возрастает и в $t=1$ равно $1{,}0065$; в c_3 – наблюдается скачок: функция возрастает от $t=0$ до $t=0{,}067$ и затем до $t=1$ принимает значение равное $3{,}6653\cdot10^{-6}$). Заметим, что в решении c_3 наблюдается наличие пограничного слоя. Его продолжительность составляет примерно 0,001 (на шаг интегрирования накладывается условие $\tau \le 0{,}001$). Такое же неравенство получается из условия устойчивости, т.е. $\tau \le \dfrac{2}{|\lambda_{\max}|} \approx 0{,}001$. Следовательно, только при $\tau \le 0{,}001$ наблюдается сходимость решения к одним и тем же предельным значениям при $t=1$.

Присутствие пограничного слоя [4, с. 8] в решении c_3 говорит о том, что эта компонента представляет оптимальную концентрацию активного вещества, являющегося промежуточным

в ходе химической реакции, а c_1, c_2 – концентрации менее активных веществ. Поэтому существенный вклад в решение вносит третья компонента c_3.

Таким образом, проведено исследование системы (1)-(2) на «жёсткость».

Изучены методы решения «жёстких» систем. Задача (1)-(2) была решена следующими численными методами: явным методом Эйлера, неявным методом Адамса-Моултона, методом Гира и с помощью алгоритма Булирша-Штера в программе Mathcad.

Полученные результаты позволяют сделать заключение о том, что применение метода Гира с автоматическим контролем погрешности вычислений и шага интегрирования или алгоритма Булирша-Штера при интегрировании «жёстких» систем является наиболее рациональным при изучении поведения концентраций химических веществ. В данной химической реакции участвуют как химически активное вещество, так и менее активные вещества.

Литература

1. Русина Л.Г., Грибкова Ю.В. Математическая модель процесса полимеризации // Вестник Череповецкого государственного университета. Череповец. 2016. № 4(73). С. 33-39.

2. Иванов В.В. Методы вычислений на ЭВМ. Киев: Наукова думка. 1986. 584 с.

3. Арушанян О.Б., Залёткин С.Ф. Численное решение дифференциальных уравнений на фортране. М.: МГУ. 1990. 336 с.

4. Ракитский Ю.В., Устинов С.М., Черноруцкий И.Г. Численные методы решения жёстких систем. М.: Наука, 1979. 208 с.

© Русина Л.Г., 2022

УДК 004.432
https://doi.org/10.36906/AP-2022/14

Смирнова Т.Н.
ORCID: 0000-0001-6687-9415, канд. физ.-мат. наук
Румянцев А.А.
ORCID: 0000-0002-9795-3427
Антонов Б.В.
ORCID: 0000-0002-9878-1874
Наумов К.Д.
ORCID: 0000-0002-4577-104X
Чувашский государственный университет им. И.Н. Ульянова
г. Чебоксары, Россия

РАЗРАБОТКА ПРОГРАММНОГО ПРИЛОЖЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ НАД МНОЖЕСТВАМИ

Аннотация. В данной статье приводится пример выполнения задания «Разработка программного приложения для выполнения операций над множествами» в условиях студенческого научного кружка. Рассматриваются основные этапы разработки программы, а также факторы, влияющие на работу программы и её функционал. Приведено описание программы, предназначенной для выполнения пяти операций над двумя пользовательскими множествами, а также пользовательского интерфейса.

Ключевые слова: разработка программа; функционал; операции; множества; C++.

Smirnova T.N.
ORCID: 0000-0001-6687-9415, Ph.D.
Rumantsev A.A.
ORCID: 0000-0002-9795-3427
Antonov B.V.
ORCID: 0000-0002-9878-1874
Naumov K.D.
ORCID: 0000-0002-4577-104X
I.N. Ulianov Chuvash State University
Cheboksary, Russia

DEVELOPMENT OF A SOFTWARE APPLICATION FOR PERFORMING OPERATIONS ON SETS

Abstract. This article provides an example of the task "Development of a software application for performing operations on sets" in the conditions of a student scientific circle. The main stages of

CC BY

program development are considered, as well as factors affecting the operation of the program and its functionality. The description of a program designed to perform five operations on two user sets, as well as a user interface, is given.

Keywords: program development; functionality; operations; sets; C++.

Одной из форм активизации научно-исследовательской работы обучающихся является кружковая деятельность, в рамках которой студенты закрепляют и расширяют знания, умения и навыки по дисциплинам учебного плана [1]. В данной статье приведён пример на основе опыта работы студенческого кружка «Современные информационные технологии в математике» [2]. В ходе изучения темы «Элементы теории множеств» студентами была выбрана тема «Разработка программного приложения для выполнения операций над множествами».

Процесс разработки программы состоит из нескольких этапов. Дадим им краткую характеристику.

I. Этап определения проблемы

Прежде чем приступить к программированию, необходимо чётко сформулировать проблему, которую должна решать наша будущая программа. Потому что, не имея чёткого определения проблемы, мы можем потратить много времени и усилий на решение неправильной проблемы. На этом этапе осуществляется простая формулировка сути проблемы без упоминания её возможных решений, при этом она должна быть сформулирована на понятном пользователю языке, т.е. описываться с точки зрения пользователя.

Основываясь на решении поставленной задачи, можно выявить проблему – это выполнение реализуемых операций над множествами, значения которых вводит пользователь.

II. Выработка требований

Требования к программе – подробное описание всех функций программы и действий, которые программа должна выполнять. Такие требования иногда также называют «функциональными спецификациями» или просто «спецификациями».

К функциям разработанной нами программы относятся: удобный ввод-вывод данных (ввод элементов множеств и вывод результатов после операций над ними), правильное выполнение операций над заданными пользователем множеств, представление пользователю программы с удобным и понятным интерфейсом.

III. Кодирование и отладка

Если все вышеперечисленные шаги выполнены, данный шаг является «механической» работой, то есть кодированием. В этом случае программисту не нужно ничего изобретать и разрабатывать самостоятельно, достаточно просто написать код, реализующий этот алгоритм, который подробно описан в проекте.

После написания кода разработчику необходимо отладить этот код, чтобы избежать ошибок.

CC BY

Код программы для выполнения операций над множествами основан на языке программирования C++ [3].

Основной выполнения операция над множествами будет Array. Array – структура данных, которая хранит набор значений, идентифицированных индексом или набором индексов, которые принимают целочисленные значения из указанного непрерывного диапазона.

Массив – простейший тип агрегата. Моделирует набор элементов одного типа, расположенных в ряд в непрерывном блоке памяти. Массивы в той или иной форме поддерживаются почти всеми языками программирования, и неудивительно, что они появились в ранних версиях C, а затем стали частью C ++.

Массивы легки в понимании и имеют большой функционал в хранении данных. Существует два вида массивов: статический и динамический.

Размер статического массива не может изменяться во время работы в коде. Т.е. при создании массива разработчик указывает фиксированное значение размера массива (количество его элементов), которое пользователь изменить не сможет, что и ставит в тупик концепт программы. Пользователь не сможет ввести больше элементов, чем фиксированное значение, в этом и проблема. Обратимся к примеру («Код 1»):

int a[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

Очевидно, что создаваемый массив имеет ограниченное количество элементов, а именно, 10. Задать элементов больше чем 10, пользователь не сможет, т.к. размер статического массива задаётся константой.

И тут приходит на помощь динамический массив. Так размер динамического массива может изменяться в зависимости от задачи пользователя. Всё что нужно, это получить на вход число, которое будет представлять количество элементов для динамического массива. Это в данном случае позволяет реализовать идею разработки с неограниченным количеством элементов массива (элемент массива равен элементу множества).

Рассмотрим следующий пример («Код 2»):

int * array = new int[length]().

Здесь «array» – название массива, который мы хотим создать, «int» – целочисленный тип данных, «length» – размер массива, переменная, которая задаётся числом. В данном типе массива уже можно реализовать подборку размера множества путём присвоения «length» значения через ввод пользователя нужного ему размера массива (количество элементов пользовательского множества).

После полного кодирования и назначения всех функций программы идёт полная отладка и сборка проекта, что в дальнейшем позволит запускать программу вне среды её программирования. В нашем случае средой программирования выступает программное обеспечение Visual Studio 2019.

Понятный и удобный интерфейс позволяет программисту легко освоиться и приступить к программированию [4]. Поддерживает множество языков программирования, в том числе, C++, C#, JS (Java Script), Basic, F#.

CC BY

IV. Тестирование всей системы

На этом этапе тестируется система с учётом интеграции всех компонентов, имеется возможность выявить и исправить проблемы взаимодействия компонентов. Важной темой тестирования является безопасность, производительность, утечка ресурсов и другие проблемы, которые нельзя проверить на более низких уровнях тестирования.

После выполнения всех этапов разработанная программа готова к выполнению своей задачи.

Наша программа называется OoS (Operations on set). Ссылка для скачивания: https://disk.yandex.ru/d/ISNmO0xeByz9hQ.

Инструкция

1. Введите количество элементов во множествах A и B. Для этого в соответствующие поля ввода «Кол-во элементов» (верхнее и нижнее) введите количество элементов ваших множеств. После введения данных нажмите кнопку со знаком «+». Поля ввода и соответствующая кнопка показаны на рисунке 1.

Рис. 1. Окно ввода количества элементов

2. Поочерёдно вводите элементы множеств. Т. е. в поле ввода (верхнее) «Элемент» (рис. 2) ввести первый элемент множества A, нажать кнопку со знаком «+», ввести второй элемент множества A, нажать кнопку со знаком «+». Продолжать, пока не введёте все элементы вашего множества A.

Рис. 2. Окно ввода элементов

Аналогично выполняем действия с элементами множества B. Ввод осуществляется в поле ввода (нижнее).

Каждый новый элемент, который будет вами введён, автоматически будет появляться в поле вывода данных множеств (рис. 3).

CC BY

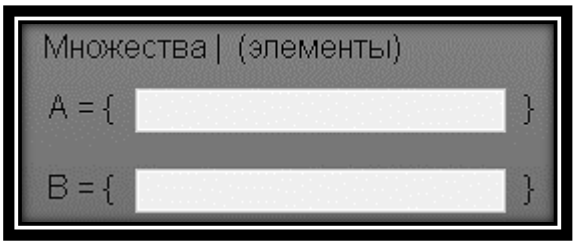


Рис. 3. Окно «Множества»

При этом вы не сможете ввести в множество больше элементов, чем было указано в поле ввода «Кол-во элементов».

3. После ввода количества элементов и всех элементов ваших множеств A и B, можете выполнить операции над множествами.

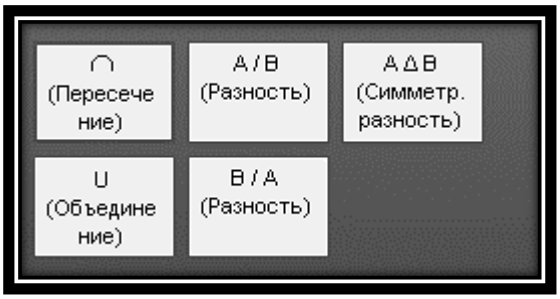


Рис. 4. Окно с операциями над множествами

Программа предназначена для выполнения следующих операций над множествами:

1) пересечение;

2) объединение;

3) разность (множества B из множества A);

4) разность (множества A из множества B);

5) симметрическая разность.

Для выполнения определённой операции нажмите на соответствующую кнопку. После чего в поле вывода «Ответ» вы получите результат (рис. 5).

Рис. 5. Окно для вывода результата

Пример работы с приложением

1. Ввод количества элементов в множестве A (верхнее поле ввода) – «5». Ввод количества элементов в множестве B (нижнее поле ввода) – «4». Нажимаем кнопку со знаком «+».

2. Ввод элементов множества A в поле ввода (верхнее) «Элемент», ввод «1», нажатие кнопки со знаком «+»», ввод «2», нажатие кнопки со знаком «+»», ввод «3», нажатие кнопки со знаком «+»», ввод «4», нажатие кнопки со знаком «+»», ввод «5», нажатие кнопки со знаком «+»». Аналогично вводим элементы множества B в поле ввода (нижнее) «Элемент».

3. Для примера рассмотрим операцию «Симметрическая разность». В результате выполнения этой операции является новое множество, включающее все элементы исходных множеств, не принадлежащие одновременно обоим исходным множествам.

Для выполнения операции, нажимаем на кнопку «A Δ B (Симметр. разность)».

Пусть A = {1, 2, 3, 4, 5}, B = {1, 3, 5, 7}.

Симметрическая разность будет равна {2, 4, 7}.

На рисунке 6 в поле вывода «Ответ» показан результат вычисления симметрической разности множеств A и B.

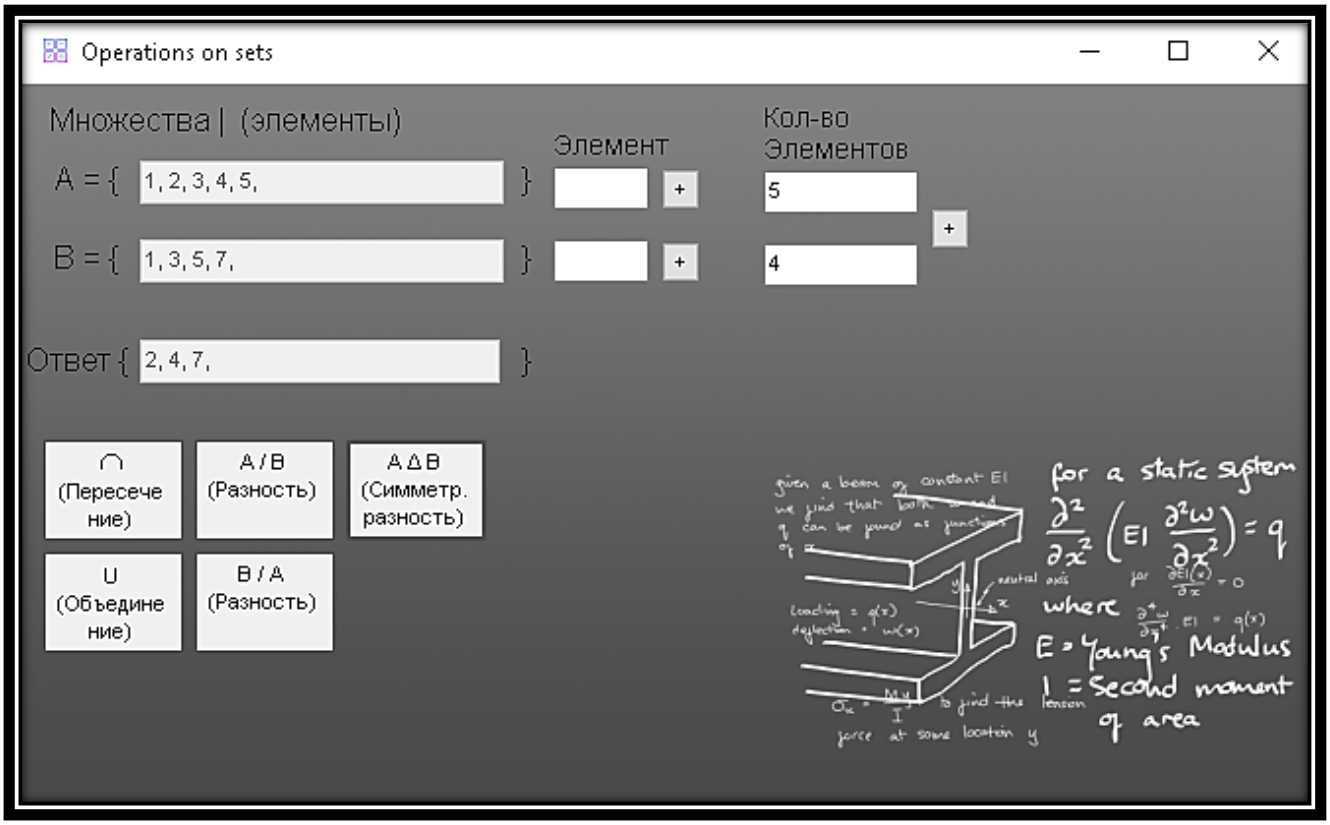


Рис. 6. Результат вычисления симметрической разности

Аналогично выполняются все остальные операции, представленные в данной программе.

Таким образом, в ходе выполнения задания обучающимися было разработано программное приложение для выполнения операций над множествами.

Литература

1. Волик М.В., Копышева Т.Н., Митрофанова Т.В., Смирнова Т.Н. Применение онлайн-компиляторов для обучения программированию студентов ИТ-специальностей вуза // Новые

компетенции цифровой реальности: теория и практика их развития у обучающихся: сб. докладов и научных статетй II Всероссийской научно-практичекой конференции (г. Чебоксары, 21 марта 2021 года). Чебоксары, 2021. С. 168-173.

2. Копышева Т.Н., Митрофанова Т.В., Дмитриева Л.А. Анализ игровых платформ для обучения программированию // Актуальные проблемы прикладной и школьной информатики: сб. научых статей. Чебоксары, 2020. С. 64-68.

3. Копышева Т.Н., Митрофанова Т.В., Дмитриева Л.А. Образовательный хакатон по программированию для школьников как интерактивный метод обучения // Инженер-созидатель материального мира будущего: сб. статей III (VIII). Всероссийской очно-заочной научно-практической конференции с международным участием (г. Санкт-Петербург, 25 марта 2020 года). СПб., 2020. С. 230-233.

4. Копышева Т.Н., Митрофанова Т.В., Дмитриева Л.А. О способах включения программирования во внеурочную деятельность младших школьников // Инженерное образование как ответ на вызовы общества – Формирование престижа профессии инженера у современных школьников: сб. статей IX Всероссийской очно-заочной научно-практичекой конференции с международным участием (г. Санкт-Петербург, 23 марта 2021 года). СПб., 2021. С. 277-279.

© Смирнова Т.Н., Румянцев А.А., Антонов Б.В., Наумов К.Д., 2022

УДК 004.6
https://doi.org/10.36906/AP-2022/15

Ташатов Н.Н.
Абылхан С.Б.
ORCID: 0000-0002-3271-2163, канд. физ.-мат. наук
Евразийский национальный университет им. Л.Н. Гумилева
г. Нур-Султан, Казахстан

ХРАНЕНИЕ ДАННЫХ В ОБЛАЧНЫХ ХРАНИЛИЩАХ

Аннотация. В данной статье описываются уникальные функции, операции и приложения облачного хранилища данных, которое становится все более важным с развитием различных интернет-сервисов, позволяющих хранить и обрабатывать данные в «облаке». Облачное хранилище данных – модель онлайн-хранилища, в которой данные хранятся на нескольких серверах, распределённых по сети и предлагаемых клиентам, в основном третьими сторонами.

Ключевые слова: облачные сервисы; облачное хранилище; хранение данных; интернет сервис; обработка данных.

Tashatov N.N.
Abylkhan S.B.
ORCID: 0000-0002-3271-2163, Ph. D.
L.N. Gumilyov Eurasian National University
Nur-Sultan, Kazakhstan

DATA STORAGE ON A CLOUD SERVICES

Abstract. This article describes the unique features, operations and applications of cloud storage, which is becoming increasingly important with the development of various Internet services that allow you to store and process data in the “cloud”. Cloud data storage is an online storage model in which data is stored on multiple servers distributed over a network and offered to customers, mainly by third parties.

Keywords: cloud services; cloud storage; data storage; internet service; data processing.

Облачное хранилище – модель онлайн-хранилища, в которой данные хранятся на нескольких серверах, распределённых по сети. Серверы обычно предоставляются заказчику третьей стороной. Информация о количестве серверов и элементах их внутренней структуры обычно не видна заказчику, в отличие от хранения данных на выделенных серверах, принадлежащих компании, которые приобретаются или арендуются для этой цели [1].

В модели облачного хранилища данные хранятся и обрабатываются в облаке.

CC BY

С точки зрения заказчика «облако» – большой виртуальный сервер. Однако эти серверы могут быть географически отделены друг от друга.

В современном мире, где на первый план выходят стремительное развитие информационных технологий и их использование в бизнесе, тема построения облачных сервисов привлекает большое внимание [2].

Формулировка цели статьи: обеспечить углублённый анализ подходов к разработке облачных сервисов для хранения данных, которые позволяют автоматизировать современные бизнес-процессы в широком спектре отраслей.

Методы исследования: Процесс облачных вычислений подробно описан в книге Дж. Риса. Анализ научных работ показывает, что использование облачного хранилища возможно не только в бизнесе, но и в других сферах деятельности современных пользователей Интернета. Исследователи в стране и за рубежом подчёркивают необходимость внедрения облачных сервисов в медицину и образование.

Ряд аналитических компаний изучали развитие рынка, создание облачных сервисов и их практическое применение. В частности, Forrester Research оценила текущую динамику внедрения облачных хранилищ и пришла к выводу, что к 2020 году объем рынка облачных вычислений достигнет 241 миллиарда долларов [2].

Аналитики из Gartner Group («Gartner Group») провели детальный анализ и определили многообещающие возможности развития облачных вычислений. Результаты показывают, что облачные вычисления являются наиболее многообещающей областью для улучшения, поскольку значительная часть существующих информационных систем переходит в облако только в ближайшие пять-семь лет [3].

Обзор литературы показывает, что разработка облачных сервисов является относительно новой областью и что необходимы дополнительные исследования для расширения возможностей создания облачных хранилищ и их успешного практического применения в бизнесе, а также в не менее важных областях человеческой деятельности.

Результаты исследования:

Облачные вычисления - это современная технология согласованной и распределённой обработки данных, которая предоставляет пользователям функциональные возможности вычислительных ресурсов в виде интернет-сервисов. Клиент-серверная технология с облачными сервисами - это особая структура, которая позволяет клиентам использовать необходимые им ресурсы (программное обеспечение, память, контроллеры, сетевые каналы и т. д.). У заказчика есть целая группа сетевых серверов в виде виртуального сервера, но при этом в большинстве случаев пользователь имеет право изменять количество потребляемых ресурсов.

В настоящее время вычислительная техника стала популярной с появлением облачных вычислений, чтобы лучше представить себе масштабы его использования.

Чтобы лучше проиллюстрировать использование облачных технологий, выпуск TAdviser показывает размер мирового рынка облачных вычислений в миллиардах долларов США.

CC BY

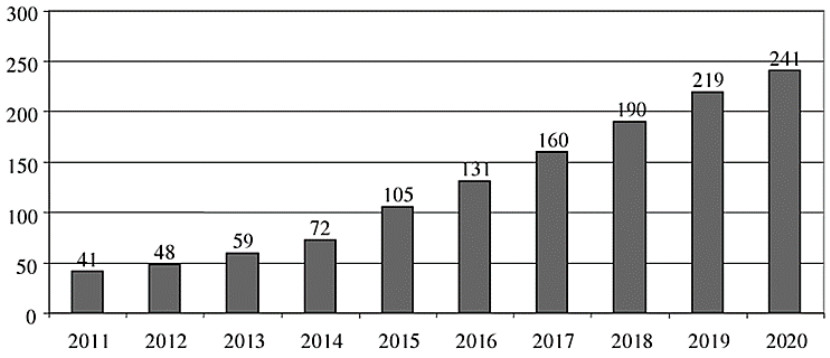


Рис. 1. Объём мирового рынка облачных вычислений

Из приведённых выше данных видно, что мировой рынок облачных приложений и сервисов с каждым годом становится все более популярным [5].

В этом разделе мы сравним, как работают облачные технологии. Облачные технологии условно можно разделить на три категории:

1. «Инфраструктура как услуга» (IaaS). Пользователь имеет доступ к «пустым» виртуальным серверам, а также может приобретать базовые вычислительные ресурсы, такие как процессоры и устройства хранения. Это позволяет вам создавать свою собственную операционную систему и приложения, так как программный интерфейс открыт. Преимущество такого подхода в том, что конфигурацию можно изменять. К недостаткам можно отнести стоимость и сложность настройки.

2. Платформа как услуга (PaaS). На этом уровне пользователь решает установить приложение на платформе, предоставляемой поставщиком услуг. Пользователь вносит лишь минимальные изменения в конфигурацию и не имеет прямого доступа к предоставленным виртуальным машинам. Преимуществами этого подхода (по сравнению с IaaS) являются лёгкая структура конфигурации и низкие затраты. Пользователь имеет право выбрать из списка услуг, предлагаемых облачной службой, те, которые необходимы для рассматриваемой задачи. Недостатки: ограниченная возможность вносить изменения в конфигурацию и потенциально более высокие риски безопасности системы.

3. «Программное обеспечение как услуга» (SaaS). Здесь пользователь работает с собственной CRM или CMS. У пользователя нет доступа к веб-серверу, СУБД или даже к конфигурации операционной системы. На этом уровне данные и связанные с ними приложения хранятся в облаке, и у пользователя практически нет возможности изменять конфигурацию или параметры безопасности службы, поэтому они могут работать только с веб-браузером. Причина этого в том, что пользователям не нужно покупать дорогое программное обеспечение, а только временно использовать программное обеспечение для тех целей, для которых оно необходимо [4].

Для успеха бизнес-процессов большая часть программного обеспечения воплощена в комбинации моделей SaaS, наиболее популярными из которых в настоящее время являются веб-службы Amazon, Google App Engine и Microsoft Windows Azure, и с их помощью следует отметить, что это даёт вам право пользователям спектр услуг, от инструментов, необходимых

CC BY

для поддержки различных типов связи, до работы с офисными приложениями, такими как электронная почта и электронные таблицы [5].

На основе приведённых выше характеристик абстракция облачных сервисов показана на рисунке 2.

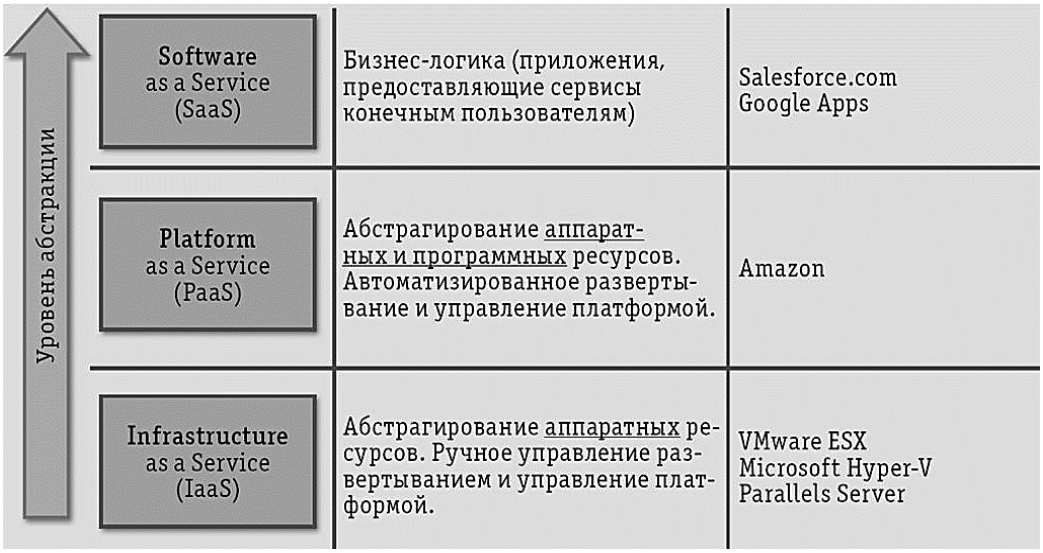


Рис. 2. Модели облачных сервисов

Широкое распространение высокоскоростной связи через Интернет сделало возможным обмен данными между компьютерами централизованно в одном месте – это «облако».

Чтобы объяснить современный подход к построению облака, принято различать три типа частных облаков, которые обычно используются одной организацией и допускают нескольких пользователей; Public – предназначены для свободного доступа к ним; Гибридные – уникальные сами по себе, связанные стандартными или специализированными технологиями передачи данных и объектными приложениями, связанными стандартами или специализированными технологиями для передачи данных и приложений; Общедоступный – предназначен для использования определенной группой.

Вот некоторые из текущих инициатив в индустрии облачных инструментов разработки Amazon Web Services (AWS).

Это облако относится к категории публичных облаков, и его основными услугами являются хостинг виртуальных машин на основе Xen (Amazon EC2) и хостинг данных (S3). В дополнение к этим основным сервисам AWS предлагает пользователям: Amazon EBS, хранилище данных для EC2; Amazon CloudFront, CDN для распределённой подготовки объектов S3; Amazon MapReduce, готовая к использованию установка Hadoop; и «Amazon MapReduce», нереляционной распределённой базе данных «Amazon SimpleDB» и очереди сообщений «Amazon Simple Queue Service».

Платформа, функционально похожая на Amazon Web Services, – Windows Azure, которая используется вычислительными центрами Microsoft Windows 2008 с Hyper-V и SQL Azure. Помимо стандартных сервисов, Azure предоставляет пользователям следующие инструменты: Azure Table – нереляционная база данных с характеристиками, аналогичными Amazon

CC BY

SimpleDB; Azure Blob - оболочка для облачного хранения данных, аналогичная Amazon EBS; Azure Queue - установка Microsoft Message Queuing, Azure Drive – виртуальный жёсткий диск, поддерживающий манипуляции с файлами, и многие другие службы, которые обеспечивают основу для создания приложений на платформе Windows Azure, которая обеспечивает основу для создания приложений на платформе Windows Azure.

Не менее популярна Force.com, платформа облачной разработки. Основное преимущество Salesforce Force.com – возможность быстро создавать SaaS-решения. Это означает, что Sales Cloud и Service Cloud построены на этой платформе, и пользователи могут разрабатывать приложения и расширения для CRM-системы Salesforce.com во время работы. Чтобы добиться успеха на этой платформе, приложения должны разрабатываться практически с нуля в среде Oracle Apex и Visualforce.

Разработчики ценят то, что они могут использовать общие технологии для создания необходимых им приложений, но считают, что эта платформа связывает разработчиков со службами, созданными на уровне кода, и позволяет им использовать только определенные инструменты. Хотя вышеуказанные решения могут быть идеальными для пользователей, которые хотят расширить функциональность облачных сервисов, другие варианты менее удобны.

Нам не нужно буквально понимать, что вам нужно оборудование для использования облачных технологий. Это компьютер, который делает всю работу.

Разработчикам не нужно настраивать службы для пользователей, чтобы запускать их код, потому что платформа, которая им нужна, управляется там. Бессерверный код почти всегда строится с использованием традиционных архитектур (например, микросервисов) [2].

Также можно создавать и размещать веб-приложения с помощью Google App Engine (GAE); как и в случае с платформой Force.com, интеграция происходит на первом уровне исходного кода, в среде распространения. Языками программирования, которые могут быть реализованы, являются Python, Java, все языки JVM (Groovy, JRuby, Jython и т. д.) и Go. GAE предоставляет пользователям нереляционную базу данных, доступ к которой разработчики могут получить с помощью языка, подобного SQL. GQL. Google App Engine.

Несложно заметить, что Google App Engine предоставляет возможность организовать удобство работы от построения масштабируемых приложений. Однако следует отметить, что результат зависит от GAE, а это значит, что приложение разработано исключительно для этой платформы [3].

В нашей оценке облачных сервисов хранения данных такие платформы, как «Google Диск», «MEGA», «Яндекс.Диск», «4shared» и «OneDrive» предлагают много места для хранения, но не все эти сервисы могут быть расширены для бесплатно. Не все эти услуги можно бесплатно расширять. Обычно разработчики увеличивают ёмкость хранилища, приглашая друзей и знакомых.

В перечисленных облаках пользователи могут обмениваться файлами и редактировать их по мере необходимости. Облачное хранилище имеет большое преимущество в том, что оно

CC BY

доступно из любого места с подключением к Интернету, а также предлагает практическое преимущество мобильного приложения.

Давайте взглянем на некоторые ключевые аспекты облачных вычислений, которым посвящена эта статья. Независимые компании, разрабатывающие программное обеспечение для облачных сервисов, находятся в процессе организации своей работы с пользователями, и проблема заключается в организации каналов, необходимых для предоставления услуг, которые они создают. Есть два способа предложить услуги облачного хранения данных: индивидуально для конкретных пользователей в компании (корпоративное облако) или через поставщиков, которые продают создаваемые ими облачные приложения. Такое решение объясняется тем, что не всегда удаётся напрямую предложить пользователям нужные им услуги. В этом случае используются услуги поставщика услуг, т.е. необходим «посредник» [2].

Предположим, перед разработчиком стоит задача доставить готовую услугу конечному пользователю. На данном этапе перед компанией-разработчиком стоит задача внедрения своего программного продукта в инфраструктуру вендора. Эту проблему можно решить путём стандартизации и автоматизации процесса развёртывания и обновления новых приложений на объекте поставщика или путём предоставления функции выставления счетов, которая позволяет компании-разработчику определять и управлять стоимостью предоставления услуг облачных приложений своим клиентам.

В настоящее время подход к внедрению новых сервисов в облачную среду в значительной степени ограничен одним путём. Стандарт APS для упаковки приложений очень полезен, поскольку позволяет поставщикам автоматически доставлять приложения, упакованные в этом формате.

Из этого видно, что существует потребность в автоматизированном процессе при организации доставки приложений в рамках организации-разработчика с аналогичными процессами в репозитории общедоступного облака. Самая большая проблема для разработчиков программного обеспечения - сформировать соответствующий программный слой на основе требуемой модели.

Нет существенной разницы в иерархии бизнес-процессов между коммерческими и частными облаками, поскольку порядок облачных процессов одинаков: 1) выбор услуги, 2) выделение ресурсов для услуги, 3) предоставление и установка услуги, 4) выставление счетов. услуги; 5) способность пользователей самостоятельно переупорядочивать услугу; 6) отказ от услуги и высвобождение вычислительных ресурсов. В связи с этим на массовом рынке популярны стандарты, разработанные специально для коммерческого предоставления облачных сервисов.

В этой статье были исследованы основные подходы к разработке сервисов облачного хранения с использованием новейших методов бессерверной технологии. Детальный анализ научных результатов в этой области позволяет систематизировать последние открытия. Преимущества облачного хранилища – большое пространство для хранения, возможность бесплатно расширять хранилище и возможность редактировать документы. по мере

CC BY

необходимости. Технологические разработки, такие как улучшение работы сетевого оборудования, снижение энергопотребления и увеличение вычислительной мощности, привели к снижению стоимости предоставляемых услуг.

Естественно, что количество облачных сервисов и решений будет расти по мере того, как облачные вычисления становятся все более популярными.

Литература

1. Архипенков С., Голубев Д., Максименко О. Хранилища данных. От концепции до внедрения. М.: Диалог-МИФИ, 2016. 528 с.

2. Хрусталев Е.Ю. Облачные технологии «1С:Предприятия». 1С-Паб, 2016. 218 с.

3. Леонов В. Google Docs, Windows Live и другие облачные технологии. М.: Эксмо, 2017. 304 с.

4. Том Л., Камаль А., Пийюм З., Эрик Ф. Облачные архитектуры: разработка устойчивых и экономичных облачных приложений. СПб.: Питер, 2018. 320 с.

5. Вишняков А.С., Макаров А.Е., Уткин А.В., Зажогин С.Д., Бобров А.В. Современные подходы разработки облачных сервисов хранения данных // Наука и образование сегодня, 2019. № 7(42). С. 13-19.

© Ташатов Н.Н., Абылхан С.Б., 2022

УДК 004.4'236
https://doi.org/10.36906/AP-2022/16

Каледин В.О.
д-р техн. наук
Решетникова Е.В.
канд. техн. наук
Ульянов А.Д.
канд. техн. наук
Паульзен А.Е.
Кузбасский гуманитарно-педагогический институт
Кемеровского государственного университета
г. Новокузнецк, Россия

СРЕДСТВА РЕАЛИЗАЦИИ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ В СРЕДЕ ФУНКЦИОНАЛЬНО-ОБЪЕКТНОГО ПРОГРАММИРОВАНИЯ

Аннотация. В статье описывается реализация среды визуального программирования. Разработанная среда позволяет задавать выполняемые алгоритмы в виде схемы функциональных зависимостей вычисляемых конечных автоматов (алгоматов), а также представление и управляющий алгоритм для интерактивного интерфейса пользователя. Представление интерфейса пользователя, кроме функционально-объектных схем, определяется структурированным текстом, заданным во время разработки прикладной программы как свойство алгомата специального вида. Управляющие алгоритмы для событий объектов интерактивных форм формируются в виде сетей алгоматов и скриптов, а связываются при помощи модели «штекер-гнездо». Использование разработанных средств позволяет средствами функционально-объектного программирования формировать насыщенные интерфейсы для прикладных вычислительных программ.

Ключевые слова: комплекс программ; программа; алгоритм; алгомат; интерфейс.

Kaledin V.O.
Ph.D.
Reshetnikova E.V.
Ph.D.
Ulyanov A.D.
Ph.D.
Paulsen A.E.
Kuzbass Humanitarian and Pedagogical Institute of Kemerovo State University
Novokuznetsk, Russia

CC BY

MEANS OF IMPLEMENTATION A USER INTERFACE IN THE FUNCTIONAL-OBJECT PROGRAMMING ENVIRONMENT

Abstract. The implementation of the visual programming environment is described. The developed environment makes it possible to specify executable algorithms as a functional dependencies diagram of computable finite automat (algomat). The considered environment provides tools for making interactive visual interface and setting up a control algorithm for its events. The presentation of the user interface is determined by structured text specified during the development of an application program as a special type algomat property. Control algorithms for interactive forms events are formed by networks of algomats and scripts, and are connected using the “plug-hollow” model. The use of the developed tools makes it possible by means of functional-object programming to build rich interfaces for applied computing programs.

Keywords: complex of programs; program; algorithm; algomat; interface.

Среда функционально-объектного программирования «Алгозит» [1; 2] регулярно используется для разработки расчётного программного обеспечения для решения частных физических задач [3; 4]. В ходе вычислительного эксперимента последовательно принимаются, анализируются и уточняются рабочие гипотезы, что приводит к необходимости модификации вычислительных программ. Существующие универсальные программные комплексы [5-8], (https://clck.ru/eYGyf; https://clck.ru/eYGrA; https://clck.ru/eYGsz) содержат заложенные разработчиками модели физических процессов. Эти комплексы, как правило, закрыты для модификации и разработки расчётных алгоритмов пользователем. Поэтому решение узконаправленных задач может оказаться для пользователя недоступным или неоправданно трудоёмким.

Благодаря графическому представлению алгоритмов в среде «Алгозит» обеспечивается высокая гибкость при разработке и модификации. При этом открытость исходного кода алгоматов и встроенная поддержка интерпретируемого языка «Ядро» [2] позволяют модифицировать или добавлять новые расчётные алгоритмы, используя существующие.

Известные подходы в области наглядного представления алгоритмов в основном базируются на двух парадигмах: объектно-ориентированного и функционального программирования [9; 10]. Однако исходные коды программ, включающие функциональные зависимости объектов, в текстовом представлении трудно читаемы. Поэтому актуально визуальное задание функциональных зависимостей вычисляемых объектов (алгоматов) в виде орграфа.

Код программы в среде «Алгозит» состоит из двух частей: функциональной, представленной орграфом алгоматов в визуальном виде, и объектной, написанной на одном из традиционных объектно-ориентированных языков, например, C++. Такой вариант парадигмы программирования логично назвать функционально-объектным программированием (ФОП) с визуальным представлением. Трансляцию в последовательность операций с алгоматами и их выполнение обеспечивает среда программирования «Алгозит».

CC BY

Эта среда позволяет строить технологии вычислительного эксперимента, обеспечивающие быструю модификацию вычислительных программ при проведении вычислительного и натурного эксперимента. Подробное описание среды и её возможностей приводится в [1].

Функционально-объектное программирование (ФОП) – развитие парадигмы объектно-ориентированного программирования с добавлением концепций функционального программирования. Алгоритмы в ФОП строятся путём установки зависимостей между алгоматами с заданными атрибутами. **Алгомат** – конечный автомат, смена состояний которого определена поведением. Он реализован при помощи средств объектно-ориентированного программирования, как экземпляр полиморфного класса. Методы интерфейса доступны интерпретатору и вызываются при выполнении операций с алгоматами интерпретатором. При этом их реализация определена конечным классом алгомата (функционального класса) или скриптами. Более полное описание рассматриваемой парадигмы и среды содержится в [1].

На рисунке 1 приведён пример использования среды для разработки алгоритма решения простейшей задачи в виде **функционально-объектной схемы**.

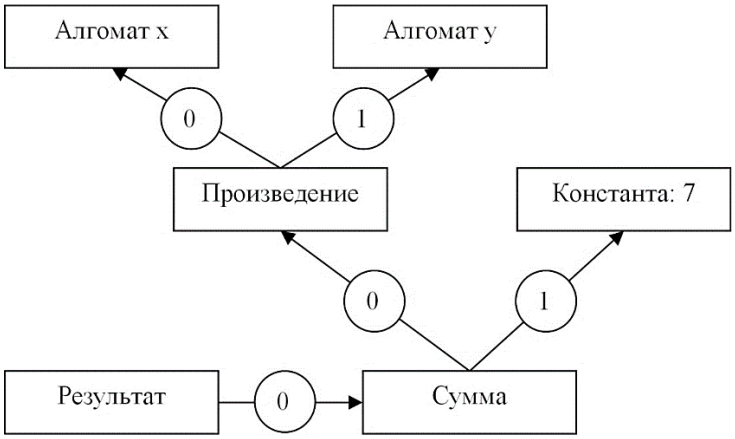


Рис. 1. Пример ФОС для вычисления арифметического выражения: «x×y+7»

ФОС является нециклическим ориентированным графом с одним входом (результирующий объект) и несколькими выходами (исходные данные). Вход ФОС законченного алгоритма – алгомат специального класса «Результат» (рис. 1). Выходы сети – алгоматы, не требующие аргументов для вычисления (например, осуществляющие ввод данных из файла или имеющие значения-константы).

Для реализации интерфейса пользователя в сеть включаются алгоматы интерактивных форм расчётного приложения, как показано на рисунке 2.

CC BY

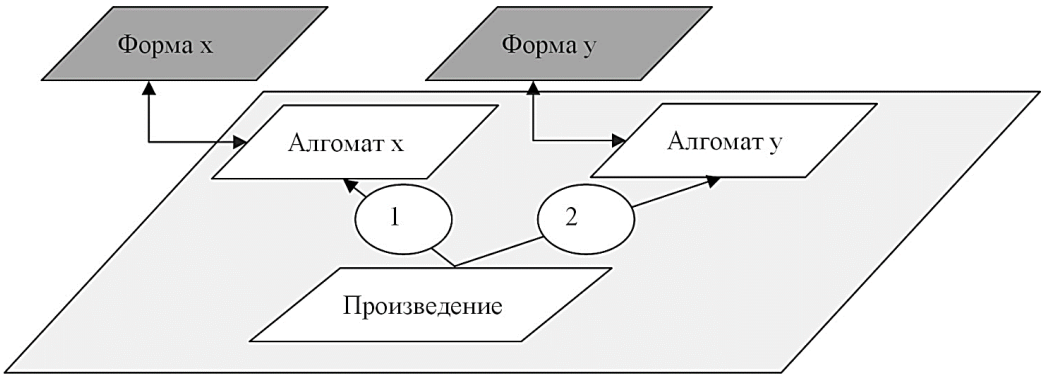


Рис. 2. Пример настроенных алгоматов, включающих форму

Для создания интерактивных форм используется платформенно-независимый подход с использованием шаблона «мост» [11]. Абстракцией в этом случае является **глиф**. Глифы формируют иерархию, сходную с иерархией объектов платформы интерфейса, при этом не используют программные средства платформы. Абстрактные глифы хранят только основные параметры (имя, глиф-владелец, положение на форме) и указатель на промежуточный объект – абстрактную реализацию (в иерархии промежуточного объекта определены ветви, использующие конкретную платформу). Для каждого объекта формы, включая корневой объект-окно, диаграмма классов имеет принципиально одинаковый вид (рис. 3). Классы «глиф», «абстрактная реализация», «реализация для платформы» и «объект платформы» образуют собственные иерархии.

Рис. 3. Диаграмма классов для объекта формы

При программировании визуального представления форм пользователем среды используется структурированное текстовое описание, установленное в специальный атрибут алгомата формы. Пример представлен ниже:

```
object tDialogForm: TAlgoGlypheWindow
  Caption = 'Диалог из ресурса'
  object MainFrame: TAlgoGlypheFrame
    object Edit1: TAlgoGlypheEdit
      Text = 'Ввод x'
      Width = 100
    end
    object Button1: TAlgoGlypheButton
      Caption = 'Готово'
      Width = 100
      Align = alLeft
```

CC BY

```
   end
  end
end.
```

На рис.4 приведен конечный вид окна, соответствующего коду из примера.

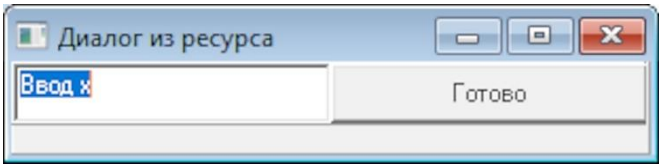


Рис. 4. Интерактивная форма ввода значения алгомата

При выполнении интерпретатором команды вычисления алгомата текстовое представление передаётся построителю форм для создания необходимых объектов и установки функций-обработчиков в гнезда. После этого поток выполнения интерпретатора приостанавливается в ожидании завершения взаимодействия с пользователем.

Обработчики событий элементов формы могут быть связаны с атрибутами алгомата, которые включают текст скрипта на встроенном языке «Ядро». Скрипты являются управляющим кодом элементов формы, который выполняется в её потоке, и могут использовать готовые значения аргументов алгомата интерфейса или инициировать повторное вычисление аргументов алгомата интерфейса с передачей параметров.

Связывание «**гнёзд**» глифов формы с атрибутами алгомата, включающими скрипты, осуществляется с использованием модели «штекер-гнездо». «**Штекером»** является структура, включающая имя допустимого класса-владельца «гнезда», представление прототипа функции-штекера, указатель на функцию-штекер указанного прототипа и атрибут ограничения потоков. «**Гнездо»** – объект отдельного класса, предназначенного для проверки допустимости присвоения и хранении указателя на штекерную функцию. «Гнездо» входит в композицию объекта-владельца. После корректной установки «штекера» в «гнездо» становится возможным вызов штекерной функции из объекта формы. Например, при событии нажатия на кнопку формы «готово» скрипт запишет введённое в редактор значение как значение алгомата и закроет форму. При закрытии активной формы выполнение интерпретатора возобновится.

Таким образом, разработаны средства создания интерактивных форм для среды функционально-объектного программирования «Алгозит». Для обеспечения независимости средств построения интерфейса от платформы был использован шаблон «мост», а для связывания событий формы с пользовательским кодом – модель «штекер-гнездо». Благодаря внедрённым средствам в среду «Алгозит» включена поддержка интерактивного пользовательского интерфейса ФОП.

Литература

1. Каледин В.О., Галдин Д.А., Паульзен А.Е., Ульянов А.Д. Вариант парадигмы программирования для реализации численных методов решения связанных задач

термомеханики гетерогенных сред // Механика композиционных материалов и конструкций, сложных гетерогенных сред: Сб. трудов 10-й Всероссийской научной конференции с международным участием (г. Москва, 17-19 ноября 2020 года). М., 2020. С. 348-356. https://doi.org/10.33113/conf.mkmk.ras.2020.348_356.55

2. Kaledin V.O., Paulzen A.E., Ulyanov A.D. Automated visualization of calculation results in the Algozit programming environment // OIP Conference Series: Materials Science and Engineering (Novokuznetsk, 28-20 November 2019). 2020. Vol. 856. Issue 1. 012019.

3. Паульзен А.Е. Разработка и апробация программного пакета для задач расчёта динамического деформирования многослойных преград при ударе // Инженерный вестник Дона. 2019. № 6(57). http://www.ivdon.ru/ru

4. Вячкин Е.С., Каледин В.О., Решетникова Е.В., Вячкина Е.В., Гилёва А.Е. Разработка математической модели статического деформирования слоистых конструкций с несжимаемыми слоями // Вестник Томского государственного университета. Математика и механика. 2018. № 55. С. 72-83. https://doi.org/10.17223/19988621/55/7

5. Морозов Е.М., Муйземнек А.Ю., Шадский А.С. ANSYS в руках инженера: Механика разрушения. М.: ЛЕНАНД, 2010. 456 с.

6. MSC. Software Corp., MSC Doc. Library, 2005. 363 p.

7. Abaqus. Применение комплекса в инженерных задачах. М.: ТЕСИС, 2008. 99 с.

8. SIMULIA Abaqus: CAE User’s Manual v 6.7. 2008. 773 p.

9. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. М.: ООО «И.Д. Вильямс», 2008. 720 с.

10. Роганова, Н.А. Функциональное программирование. М.: ГИНФО, 2002. 260 с.

11. Гамма Э., Хелм Р., Джонсон Р., Влиссиденс Дж., Приёмы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001. 368 с.

© Каледин В.О., Решетникова Е.В., Ульянов А.Д., Паульзен А.Е., 2022

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В СЛОЖНЫХ СИСТЕМАХ

УДК 004.942
https://doi.org/10.36906/AP-2022/17

Баскаков С.А., Ногтев С.С., Антонов И.М.
Чемоданов В.Б.
канд. техн. наук
Московский авиационный институт
г. Москва, Россия

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА ДЛЯ МОБИЛЬНОГО ПИЛОТАЖНОГО КОМПЛЕКСА

Аннотация. В настоящей статье рассмотрена модель беспилотного летательного аппарата схемы «летающее крыло» и особенности его разработки для мобильного пилотажного комплекса. Описаны принятые решения и удачные подходы.

Ключевые слова: беспилотный летательный аппарат; схема «летающее крыло».

Baskakov S.A., Nogtev S.S., Antonov I.M.
Suitcases V.B.
Ph.D.
Moscow Aviation Institute
Moscow, Russia

SOFTWARE FOR MOBILE FLIGHT SIMULATOR AND FEATURES OF ITS DEVELOPMENT

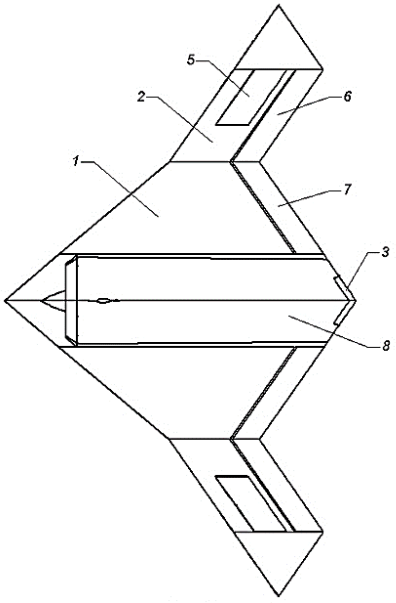
Abstract. This article discusses the unmanned aerial vehicle model of the “flying wing" scheme and the features of its development for a mobile aerobatic complex. The decisions taken and successful approaches are described.

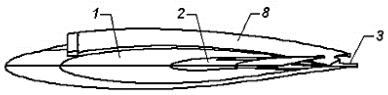
Keywords: unmanned aerial vehicle; the “flying wing” scheme.

При выполнении работы была поставлена задача создать модель беспилотного летательного аппарата (БПЛА) для внедрения модели в мобильный пилотажный стенд для отработки алгоритмов в области управления летательным аппаратом (ЛА).

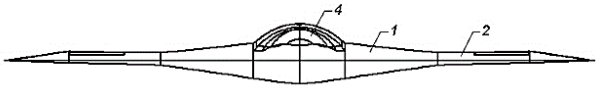
CC BY

В рамках поставленной задачи была выбрана аэродинамическая схема, геометрия крыла, аэродинамические профили и далее, соответственно, разработана модель.





Фиг. 2



Фиг. 3

Рис. 1. БПЛА в трёх видах

Как видно из рисунка 1, БПЛА обладает схемой «летающее крыло»:

1. Центроплан;

2. Консоль крыла;

3. Плоское сопло (или просто – сопло, сопло двигателя, сопло ТРДД и др.);

4. S-образный воздухозаборник (или просто – воздухозаборник, воздухозаборник двигателя, воздухозаборник ТРДД и др.)

5. Интерцептор

6. Внешний элевон

7. Внутренний элевон

8. Часть крыла, содержащая систему: воздухозаборник, двигатель, сопло.

Преимущества схемы летающее крыло перед другими аэродинамическими схемами:

1. Вся поверхность ЛА работает как одно крыло, что повышает аэродинамические характеристики, относительно остальных схем. Данное превосходство также позволяет при тех же параметрах размаха крыльев, значительно уменьшить массу самого ЛА и повысить полезную нагрузку. По сравнению с классической схемой при одинаковом размахе обоих ЛА, для ЛА со схемой «летающее крыло» понадобится двигатель на 33% слабее по мощности, чем на классической схеме.

2. Также схема «летающее крыло» позволяет снизить радиолокационную заметность, за счёт уменьшения количества поверхностей рассеяния лучей, по сравнению с остальными схемами.

3. Плюсом также является меньшее количество стыковых узлов, нежели чем в остальных схемах.

Геометрические параметры:

Размах крыла – 2 метра.

CC BY

Аэродинамические параметры:

Центроплан – Lockheed C-5A, 50 градусов.

Консоли – Lockheed C-5A переходит в NACA 0009, 35 градусов.

Законцовки - NACA 0009, 35 градусов.

Создав аэродинамические обводы по данным профилям мы также сделали S-образный воздухозаборник для дозвукового ЛА и продули модель в размахе 1500 мм в программе Solid FloEFD. Результаты. На модели общая отрицательная крутка составляет 6 градусов: 3 градуса на центроплане и 3 градуса на консоли. Центр тяжести взят на 25% САХ.

Условия продувки:

Размах крыла: 1500 мм;

Скорость набегающего потока воздуха: 28 м/с;

Высота над уровнем моря: 100.

На основании результатов расчёта схемы, продувки и моделировании полёта на мобильном пилотажном стенде было принято создать модель в размахе 10 метров. Был сделан выбор двигателя, которым оказался двигатель РД-1700

Характеристики двигателя представлены на рисунке 2.

Масса сухая, кг	297,5
Тяга максимальная, кгс	От 1700 до 2000
Удельный расход топлива, кг/кгс*ч	От 0,5 до 0,7
Степень повышения давления в компрессоре	14,3
Степень двухконтурности	0,78
Приемистость	4,6
Общий технический ресурс, ч	Около 6000

Рис. 2. Характеристики двигателя

Для двигателя (рис. 3) и для дозвуковой скорости наша команда сделала S-образный воздухозаборник с узким соплом (рис. 4). S-образный воздухозаборник позволяет скрыть компрессор ТРДД, улучшается РЛС незаметность. Пластина в воздухозаборнике необходима для ликвидации зон рассеяния и формирования равномерного набегающего потока воздуха. Плоское сопло позволяет быстрее рассеивать тепло реактивной струи, что улучшает незаметность в инфракрасном спектре.

CC BY

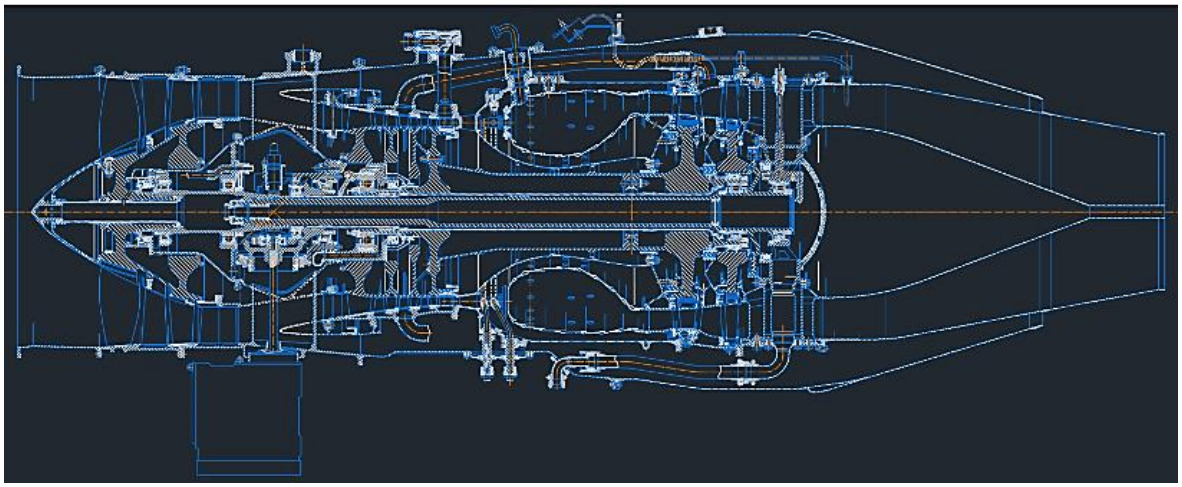


Рис. 3. Чертёж двигателя

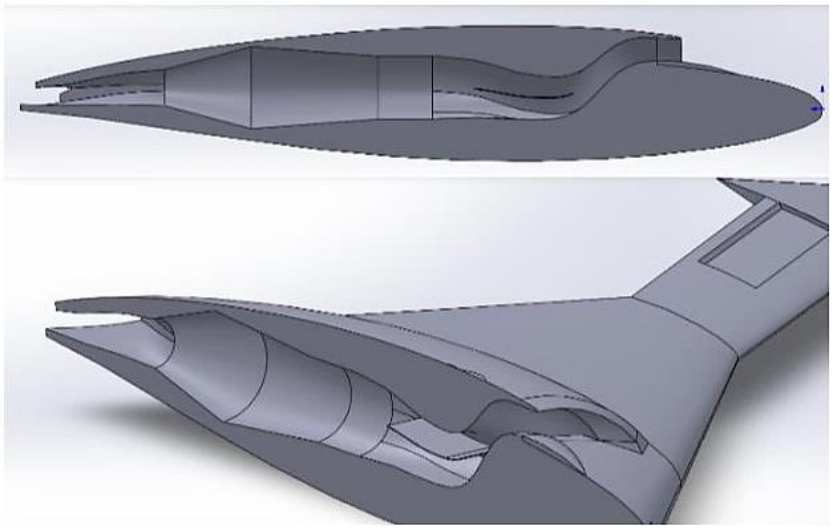


Рис. 4. S-образный воздухозаборник

Далее начался расчёт масс летательного аппарата:

1) m пн=800 кг (масса полезной нагрузки)

2) m сн=65 кг (масса узловых соединений)

3) m оу=300 кг (масса оборудования)

4) m су=600 кг (РД-1700 2000 кгс + топливная система)

5) m т = 1500 кг (масса топлива)

6) m к = 800 кг

Мобщ = 4130 кг

В гражданской сфере БПЛА может использоваться в таких средах как:

1 Спасательная – доставка оборудования, провизии, медикаментов и прочего в участки спасательных действий.

2 Пожарная – БПЛА может нести разведку горящего участка, также учитывая массу полезной нагрузки БПЛА может нести определенной объем тушащего раствора, который

CC BY

можно будет сбросить на участок пожара. У аппарата есть возможность установки резервуаров с раствором наружу.

В военной сфере БПЛА может использоваться:

1 Для разведки

2 Уничтожение наземных целей. (Данный ЛА сможет поддерживать войска обстрелом противника ракетами «Вихрь» класса «воздух-поверхность» (масса ракеты 40 кг, масса боевой части – 12 кг, следовательно Беспилотный комплекс способен нести на себе минимум 8 ракет).

3 Поддержка мотострелковых войск доставкой боеприпасов, провизии, медикаментов и др. в горячие точки.

На рисунке 5 можно увидеть боевой радиус БПЛА:

Рис. 5. Боевой радиус БПЛА

Модель БПЛА была внедрена в мобильный пилотажный комплекс. Подробно о нём рассказано в [1]. После серии тестов модели на комплексе в программе Flightgear была подтверждена адекватность математической модели. На рисунке 6 можно увидеть сам комплекс:

CC BY

Рис. 6. Мобильный пилотажный комплекс

Литература

1. Антонов И.М., Мелюков С.А., Иванов Б.А., Лямкин Ф.К. Мобильный пилотажный комплекс // Современные технологии в задачах управления, автоматики и обработки информации: Сб. тезисов докладов XXX Международной научно-технической конференции (г. Алушта, 14-20 сентября 2021 года). Алушта, 2021.

© Баскаков С.А., Ногтев С.С., Антонов И.М., Чемоданов В.Б., 2022

УДК 004.942
https://doi.org/10.36906/AP-2022/18

Сулейменова С.Т.
Баяхметова К.Е.
Евразийский национальный университет им. Л.Н. Гумилева
г. Нур-Султан, Казахстан

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ЧАСТОТНОГО РЕГУЛИРОВАНИЯ ПРИВОДА КАСКАДНОЙ НАСОСНОЙ СТАНЦИИ ПРОИЗВОЛЬНОЙ КОНФИГУРАЦИИ

Аннотация. Разработана система автоматического регулирования частоты каскадной насосной станции. Проведено математическое моделирование разработанной системы. Рассчитаны параметры передаточной функции системы управления. Реализовано имитационное моделирование системы без и с возмущающими воздействиями и с использованием пропорционально-плюсинтегрального закона управления. Универсальный функциональный блок был разработан в программе управления. Этот блок позволит управлять насосами случайного числа и со случайной схемой питания. Использование данной системы управления позволяет повысить надежность сети водоснабжения, улучшить выходные характеристики системы для потребителя, снизить износ насосного оборудования и потребление электроэнергии.

Ключевые слова: насосная станция; регулирование частоты; преобразователь частоты; математическое моделирование; система управления.

Suleimenova S.T.
Bayakhmetova K.E.
L.N. Gumilyov Eurasian National University
Nur-Sultan, Kazakhstan

AUTOMATED FREQUENCY CONTROL SYSTEM FOR THE DRIVE OF A CASCADE PUMPING STATION OF ARBITRARY CONFIGURATION

Abstract. A system of automatic frequency control of a cascade pumping station has been developed. Mathematical modeling of the developed system is carried out. The parameters of the transfer function of the control system are calculated. Simulation modeling of the system without and with disturbing influences and using proportional-plusintegral control law is implemented. The universal functional unit was developed in the management program. This unit will allow you to control pumps of a random number and with a random power scheme. The use of this control system makes it possible to increase the reliability of the water supply network, improve the output

CC BY

characteristics of the system for the consumer, reduce the wear of pumping equipment and electricity consumption.

Keywords: pumping station; frequency control; frequency converter; mathematical modeling; control system.

Решение задачи полной автоматизации объекта при работе с минимально необходимым потреблением энергии и одновременном повышении качества водоснабжения является основной задачей, стоящей перед предприятиями жилищно-коммунального хозяйства и производителями оборудования. В настоящее время центробежные насосы являются наиболее распространенными в сфере водоснабжения. Сепараторные асинхронные двигатели используются в основном для привода насосов. Насос в совокупности с электроприводом и трансмиссией образует насосный агрегат. Для обеспечения работы одного или нескольких агрегатов в требуемом режиме необходим целый комплекс оборудования: трубопроводы, запорная и регулирующая арматура, контрольно-измерительное оборудование, а также контрольно-защитное оборудование. Насосный агрегат с необходимым оборудованием образует насосную установку. Для создания требуемого напора и обеспечения водоснабжения часто применяются несколько установок, работающих на общую сеть. Сооружение, состоящее из одной или нескольких установок, а также вспомогательных систем и оборудования, бытовых отсеков и производственных помещений, обеспечивающих работоспособность в целом, называется насосной станцией. В зависимости от требуемой комбинированной характеристики насосные установки соединяются между собой параллельно, последовательно или смешанным способом. Установки чаще всего работают параллельно и перекачивают воду в один напорный трубопровод в области бытового водоснабжения. Режим работы насосной станции существенно зависит от изменения режима водопотребления.

Современные насосные станции широко используют автоматические электроприводы. В основном применяются два способа регулирования производительности на насосных станциях: каскадный и частотный. Каскадное регулирование заключается в включении и выключении параллельно установленных насосов. Частотное регулирование позволяет регулировать производительность насосной станции за счет изменения частоты вращения насоса с помощью преобразователей частоты. Сочетание каскадного и частотного регулирования производительности группы насосных установок является наиболее эффективным инструментом с точки зрения технико-экономических показателей. Поэтому мы проводим математическое моделирование и имитационное моделирование стандартной насосной станции с сочетанием в работе двух методов регулирования.

Несмотря на кажущуюся простоту создания автоматизированной системы управления для стабилизации напора в системе, многим организациям приходится самостоятельно решать множество возникающих проблем при разработке систем управления автоматическими станциями из-за отсутствия методик и рекомендаций в технической литературе.

В качестве объекта автоматизации мы возьмем модульную насосную станцию третьего подъема. Эта станция используется для водоснабжения строительного комплекса и

CC BY

обеспечивает бесперебойную подачу воды в систему пожаротушения в случае пожара. Для этой станции была выбрана схема управления с тремя преобразователями частоты.

Два преобразователя частоты будут работать, а один будет находиться в резерве. Для управления преобразователями частоты предусмотрен программируемый логический контроллер (ПЛК). Этот ПЛК реализует основные функции регулирования и диспетчеризации. Сенсорная панель оператора была предусмотрена для представления технологических параметров и управления. Следует предусмотреть возможность ручного запуска системы в случае отказа контроллера.

Разработанная система управления каскадной насосной станцией имеет трехуровневую иерархическую структуру:

1. Нижний уровень включает исполнительные механизмы (насосы), оборудование контрольно-измерительных приборов и автоматики (датчики давления, реле давления, турбинные расходомеры);

2. Средний уровень реализован на базе шкафа управления SCC-3/3-7,5x3-3-H-GSM;

3. Высокий уровень реализован на ПК оператора с представлением мнемонической схемы технического процесса. Все технологические параметры системы достигают этого уровня.

С нижнего уровня сигналы подаются в шкаф управления, и управляющее воздействие передается насосам. Насосы получают управляющее напряжение с заданной характеристикой U/f по силовым экранированным кабелям. Насосы выбираются с тепловым контактом, регулирующим температуру обмоток для их защиты от перегрева. Реле давления необходимо для реализации функции защиты от «сухого хода». Он установлен на подающем трубопроводе и выпускном коллекторе для определения работы насоса и контроля выходного давления. Датчики давления следует выбирать с выходным сигналом 4–20 мА. Расходомеры установлены на выпускном коллекторе для коммерческого измерения подаваемой воды.

Средний уровень системы обеспечивается полным шкафом управления. В его структуре должны быть предусмотрены преобразователи частоты, высокоскоростной контроллер (время цикла не более 200 мс) и панели HMI. Верхний уровень должен быть организован на промышленном ПК, который находится в комнате оператора. Оператор может контролировать текущий технологический процесс и архив данных, полученных из системы, с помощью этого ПК.

Разработана структурная схема автоматизации насосной станции (рис. 1), где 1 – шкаф управления CC-3/3-7, 5x3-3- H-GSM, 2 – сеть GSM, 3 – ПК оператора, 4 – преобразователь частоты Microdrive FC 51, 5 – программируемый логический контроллер ILC 131 ETH, 6 – сенсорная панель оператора MT8070iE, 7 – насос P1, 8 – насос P2, 9 – насос P3, 10 – реле давления, 11 – датчики давления 4-20 мА, 12 – турбинные расходомеры 4-20 мА.

CC BY

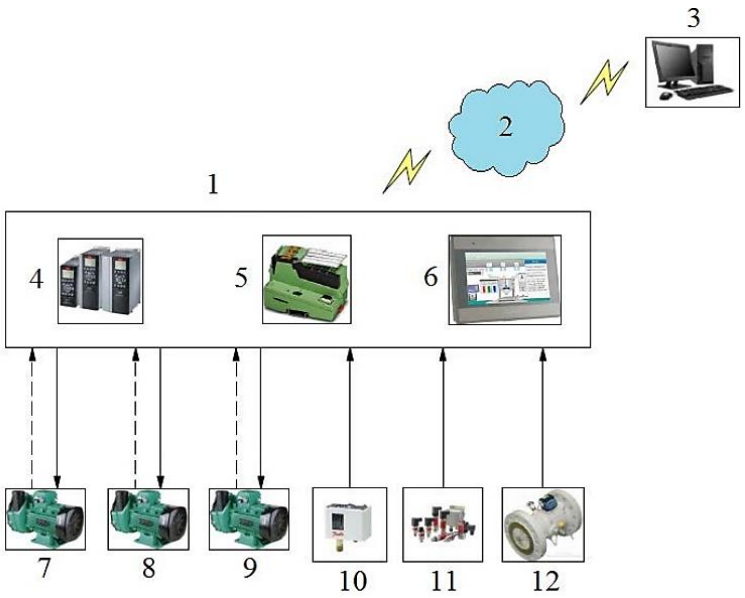


Рис. 1. Структурная схема автоматизации насосной станции

Необходимо регулировать выходное давление на станции третьего подъема. Она не должна превышать и опускаться ниже установленных значений из-за неравномерного течения воды во времени. Поэтому при математическом моделировании необходимо учитывать контур управления скоростью двигателя в зависимости от выходного давления. Разработанная система автоматического регулирования давления представлена на структурной схеме (рис. 2).

Сигнал о назначении давления подается от фильтра. Затем разность этого сигнала и сигнала от датчика давления, преобразующего входное давление в текущий сигнал, передается регулятору. Сигнал регулирования напряжения подается от регулятора на вход преобразователя частоты, частота вращения двигателя насоса является выходным сигналом этого блока. Этот сигнал подается на вход насоса, преобразующего частоту циклического двигателя в давление. Мы получаем давление в трубопроводе после наложения возмущающего воздействия на сигнал насоса. Блоки FC, P и PA представляют собой апериодические связи. В ходе работы были рассчитаны параметры блок-схемы каждого блока с начальными условиями. При имитационном моделировании были использованы начальные условия.

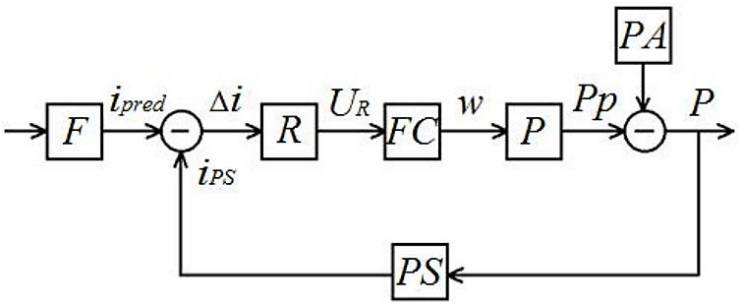


Рис. 2. Структурная схема системы автоматического регулирования давления

1. Блок преобразователя частоты. В общем виде функция передачи для этой ссылки выглядит следующим образом:

$$W_{FC} = K_{FC}/(T_t * p + 1)$$

Мы рассчитаем коэффициент усиления и время отклика. Коэффициент усиления равен:

$$K_{FC} = \frac{\omega}{U_R} = 50$$

T_t = 0,01 ввиду высокой скорости отклика. Таким образом, мы получаем передаточную функцию этой ссылки:

$$W_{FC} = 50/(0.01 * p + 1)$$

2. Блок насоса. В общем виде функция передачи для этой ссылки выглядит следующим образом:

$$W_p = K_p/(T_t * p + 1)$$

Мы рассчитаем коэффициент усиления и время срабатывания. Коэффициент усиления равен:

$$K_p = \frac{P}{\omega} = 0.6$$

T_p = 1 – время разгона насоса. Таким образом, мы получаем передаточную функцию этой ссылки:

$$W_p = 0.6/(p + 1)$$

3. Блок датчика давления является безынерционным и имеет следующий вид:

$$K_{PS} = \frac{i_{PS}}{P} = 0.033$$

4. Блок возмущающего действия. В общем виде функция передачи для этой ссылки выглядит следующим образом:

$$W_{PA} = K_{PA}/(T_{PA} * p + 1)$$

Мы рассчитаем коэффициент усиления и время срабатывания. Коэффициент усиления равен:

$$K_{PA} = \frac{P_{PA}}{Q_{PA}} = 1.11$$

5. TPA = 1 – время открытия клапана. Таким образом, мы получаем передаточную функцию этой ссылки:

$$W_{PA} = 1.11/(p + 1)$$

Моделирование системы управления проводилось в специализированном программном комплексе визуального моделирования Matlab/Simulink. Сначала моделирование проводилось без возмущающего воздействия (рис. 3), затем с внешним возмущающим воздействием (рис. 2) какое отверстие запорного клапана.

CC BY

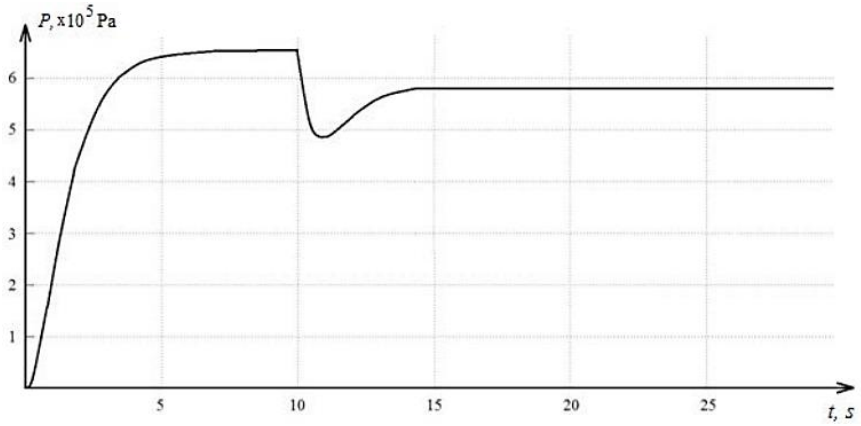


Рис. 3. Результат моделирования

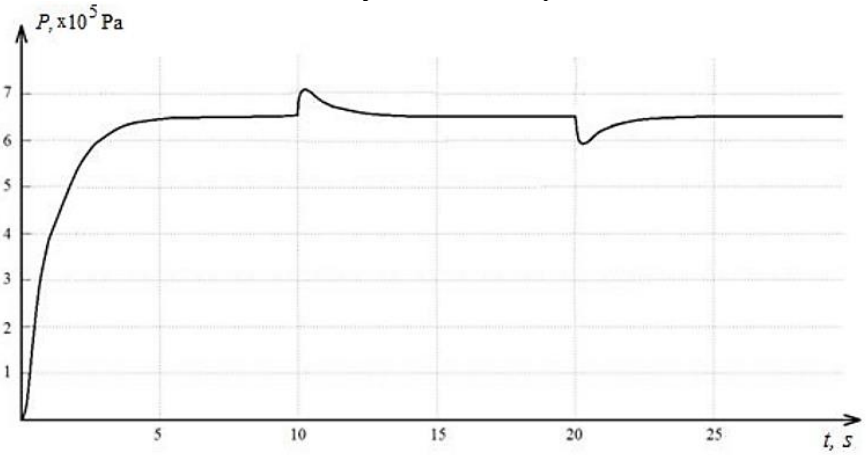


Рис. 4. Результат моделирования

В момент открытия клапана (рис. 3) происходит резкое падение давления, а затем система переходит в новое устойчивое состояние с меньшим значением давления, которое не соответствует требованиям. Поэтому для решения этой проблемы необходимо добавить регулятор давления, который компенсирует действие возмущений в системе. ПИ-регулятор оказался наиболее подходящим.

Мы примем желаемую передаточную функцию разомкнутого контура в качестве критерия эффективности управления при расчете PI-регулятора. Целесообразно применять настройки контура управления на технический оптимум для рассматриваемой системы регулирования. В результате синтеза получена передаточная функция ПИ-регулятора.

$$W_R(p) = \frac{\frac{1}{2 * T_t * p * (T_t + 1)}}{\frac{K_{FC}}{T_t * p + 1} * \frac{K_p}{T_P * p + 1} * K_{PS}} = \frac{T_P * p + 1}{2 * T_t * K_{FC} * K_p * K_{PS} * p} = \frac{p + 1}{4 * p}$$

Позже было проведено моделирование системы с ПИ-регулятором и возмущающими воздействиями.

Об этом можно судить по полученным результатам (рис. 4) система быстро преодолевает возмущение и возвращается в исходное устойчивое состояние с заданными показателями качества, что подтверждает необходимость добавления в систему ПИ-регулятора.

Существующие системы управления насосными станциями не являются универсальными, что приводит к тому, что при создании новой каскадной станции

необходимо адаптироваться к конкретной конфигурации силовой схемы и количеству насосов.

В результате выполнения данной работы была разработана схема автоматизации каскадной насосной станции. Разработана математическая модель трехкаскадной системы. Предлагаемая математическая модель позволяет исследовать системы с различным количеством каскадов до их проектирования и оценки их эффективности. Рассчитанные коэффициенты PI-регулятора следует использовать при создании управляющей программы.

Внедрение результатов работ на насосных станциях водоснабжения поможет улучшить выходные характеристики системы для потребителя, повысить надежность водопроводной сети и снизить износ насосного оборудования и потребление электроэнергии.

Результаты исследований рекомендуются для использования на насосных станциях первого, второго и третьего порядка и прилегающих территориях.

© Сулейменова С.Т., Баяхметова К.Е., 2022

УДК 004.891
https://doi.org/10.36906/AP-2022/19

Гладких Т.Д.
ORCID: 0000-0003-0568-4109, канд. техн. наук
Филиал Тюменского индустриального университета в г. Нижневартовске
г. Нижневартовск, Россия

К ВОПРОСУ АНАЛИЗА НАДЁЖНОСТИ СИСТЕМ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

Аннотация. В работе рассматривается реализация математической модели надёжности технической системы, имеющей временную избыточность. Использована теория Марковских процессов для описания возможных состояний объекта исследования. Представлены этапы расчёта системы дифференциальных уравнений Колмагорова для представленной модели. Отмечены сложности, возникающие при анализе математических моделей с большой размерностью системы дифференциальных уравнений.

Ключевые слова: модель; надёжность; Марковский процесс.

Gladkikh T.D.
ORCID: 0000-0003-0568-4109, Ph.D.
Branch of Industrial University of Tyumen in Nizhnevartovsk
Nizhnevartovsk, Russia

ON THE ISSUE OF RELIABILITY ANALYSIS BY COMPUTER MATHEMATICS SYSTEMS

Abstract. The paper considers the implementation of a mathematical model of the technical system reliability with temporary redundancy. The theory of Markov processes is used to describe the equipment possible states. The calculation stages of the Kolmogorov differential equations system for the presented model are presented. The difficulties arising in the analysis of mathematical models with a large dimension of the differential equations system are noted.

Keywords: model; reliability; Markov process.

Для оценки надёжности технических систем существует множество классических способов на основе статистических и детерминированных подходов, и специфических методик на основе теории нечётких множеств и искусственных нейронных сетей.

При анализе характеристик надёжности в зависимости от поставленной задачи можно применять один из следующих прикладных программных пакетов: Matlab, Matcad, R, Statistica и др.

CC BY

Одним из универсальных методов анализа надёжности можно назвать подходы на основе теории марковских процессов. Например, марковские цепи могут использоваться при оценке качества программного обеспечения, так в работе [2] предложена методика статистического тестирования ПО на основе марковских цепей.

При расчёте показателей безотказности сложных систем теория марковских процессов так же является одним из удобных математических аппаратов [1; 3]. При моделировании таким методом можно учесть множество возможных состояний системы, при этом не учитывая траекторию попадания системы в данное состояния. При использовании метода на основе марковских процессов исследователям необходимо решать систему дифференциальных уравнений Колмагорова большой размерности, что требует компетентности в данной области. И здесь возникает необходимость выбора ПО, которое в большей степени подходит для решения поставленной задачи.

Таким образом, актуальными являются работы, рассматривающие практическую реализацию решения дифференциальных уравнений, составленных на основе графа переходов (цепи Маркова) с применением современных программных средств.

Для определения вероятности безотказной работы системы решается система дифференциальных уравнений Колмагорова, что при современном развитии программного обеспечения (ПО) не является затруднительным.

Однако в некоторых случаях требуется получить решение дифференциальных уравнений в общем виде. Например, для оперирования полученным выражением в функции какого-либо параметра. В этом случае удобно использовать систему компьютерной математики Maple.

При рассмотрении однородного Марковского процесса необходимо выделить следующие этапы:

– определение возможных состояний системы (то есть конкретные условия объекта: его работоспособность, отказ, переключение и др.);

– определение связей между различными состояниями, которые необходимо отобразить на графе переходов;

– составление матрицы переходов или системы дифференциальных уравнений на основе графа;

– решение дифференциальных относительно вероятностей состояний;

– расчёт надёжности в контексте задачи: поиск вероятности отказа системы, вероятности безотказной работы системы, вероятность нахождения в состояниях с определенными условиями.

Одним из важных вопросов является повышение надёжности опасных производств, в таких объектах используют различного вида избыточности (резервирование). Например, для систем управления (информационных систем, электронной техники) таких объектов исследования имеются следующие особенности:

– невозможность восстановления, то есть интенсивность восстановления равна 0;

CC BY

– в определенный момент времени может возникнуть только одна неисправность, то есть предельная вероятность возникновения двух отказов в канале связи пренебрежимо мала;

– известна интенсивность покрытия неисправностей, в системе могут быть два типа дефектов – обнаруживаемые и необнаруживаемые;

– в начале эксплуатации все каналы связи работают в нормальном режиме.

Ниже рассмотрен анализ простейшей модели надёжности с временной избыточностью для восстанавливаемой системы. Анализ невосстанавливаемой системы можно рассматривать как частный случай восстанавливаемой системы, где в модели надёжности интенсивности восстановлений $\mu_i = 0$.

Рассмотрим этапы анализа показателей надёжности для системы, описываемой графом состояний (рис.1) в контексте реализации на ПО Maple. В представленном графе система описывается тремя состояниями: p0 – исправное состояние, p1 – работоспособное состояние (наличие временной избыточности), p3 – состояние отказа.

Предложенный граф при всей своей простоте может отражать такие процессы, как перерыв электроснабжения и наличие временной избыточности в виде источника бесперебойного питания, нарушение технологического процесса в транспортировке углеводородного сырья и наличие временной избыточности в виде свободного объёма в резервуарном парке и т. д.

Система дифференциальных уравнений для графа рисунке 1 имеет вид:

$$\begin{cases} \frac{dp_0}{dt} = -p_0(t)\cdot(\lambda+\lambda) + p_1(t)\cdot(\mu) + p_3(t)\cdot(\mu_2); \\ \frac{dp_1}{dt} = p_0(t)\cdot(\lambda) - p_1(t)\cdot(\mu+\lambda_V); \\ \frac{dp_3}{dt} = p_0(t)\cdot(\lambda) + p_1(t)\cdot(\lambda_V) - p_3(t)\cdot(\mu_2). \end{cases}$$

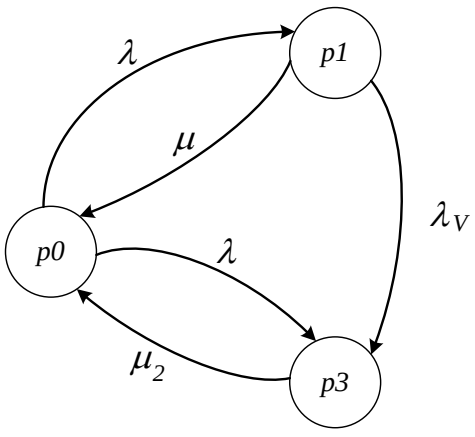


Рис. 1. Граф состояний и переходов для анализа надёжности системы

При решении поставленной задачи использованы следующие исходные данные (пример ввода в ПО Maple):

$$> l := .201;\ m := \frac{1}{.11};\ m2 := \frac{1}{0.0174};\ mv := 4;\ lv := \frac{60\cdot 24}{T};$$

Результатом выполнения строки являются значения параметров, в том числе выраженные через другие величины: значение временной избыточности обозначено переменной T.

Для формирования системы дифференциальных уравнений с нормировочным уравнением можно применить запись вида (здесь и далее скрин-шоты из Maple):

$$> sis := diff(p0(t), t) = -p0(t)\cdot(l + l) + p1(t)\cdot m + p3(t)\cdot m2,\ diff(p1(t), t) = p0(t)\cdot l - p1(t)\cdot(m + lv), p0(t) + p1(t) + p3(t) = 1;$$

Результат выполнения строки следующий:

$$sis := \frac{d}{dt}p0(t) = -2\,p0(t)\,l + p1(t)\,m + p3(t)\,m2,\ \frac{d}{dt}p1(t) = p0(t)\,l - p1(t)\,(m + lv), p0(t) + p1(t) + p3(t) = 1$$

Для задания нулевых начальных условий вводим новую переменную:

$$> ny := p0(0) = 1, p1(0) = 0, p3(0) = 0;$$

Командная строка для решения системы дифференциальных уравнений “sis”, с нулевыми начальными условиями “ny”, относительно вероятностей состояний “p0(t), p1(t), p3(t)” имеет вид:

$$> dsolve(\{sis, ny\}, \{p0(t), p1(t), p3(t)\});$$

Фрагмент выполнения (решения системы дифференциальных уравнений) строки следующий:

$$p0(t) = -\frac{1}{201000000}\,\frac{1}{T}\left(-1000000000\left(-\frac{201}{4000}\left(\left(-66964173461\,T - 1440000000000 + 3\sqrt{260091153435807018649\,T^2 - 156103536892800000000000\,T + 230400000000000000000000}\right)\right.\right.\right.$$
$$\left.\left.\left.e^{\frac{1}{2000000000}\,\frac{\left(-66964173461\,T - 1440000000000 + 3\sqrt{260091153435807018649\,T^2 - 156103536892800000000000\,T + 230400000000000000000000}\right)t}{T}} \cdots\right.\right.\right.$$

По представленному фрагменту видно, что ответ представлен в функции параметра T, таким образом, возникает возможность оценки влияния данного параметра на результат оценки вероятностей состояний.

Для рассмотренного графа функция безотказности – вероятность безотказной работы – есть сумма вероятностей нахождения системы в исправном и работоспособном состоянии:

$$R(t) = p0(t) + p1(t).$$

Для графической оценки влияния параметра временной избыточности T на результирующую функцию надёжности R(t) используется команда plot с заданием интервала. На рисунке 2 представлен результат оценки надёжности для принятых исходных данных.

CC BY

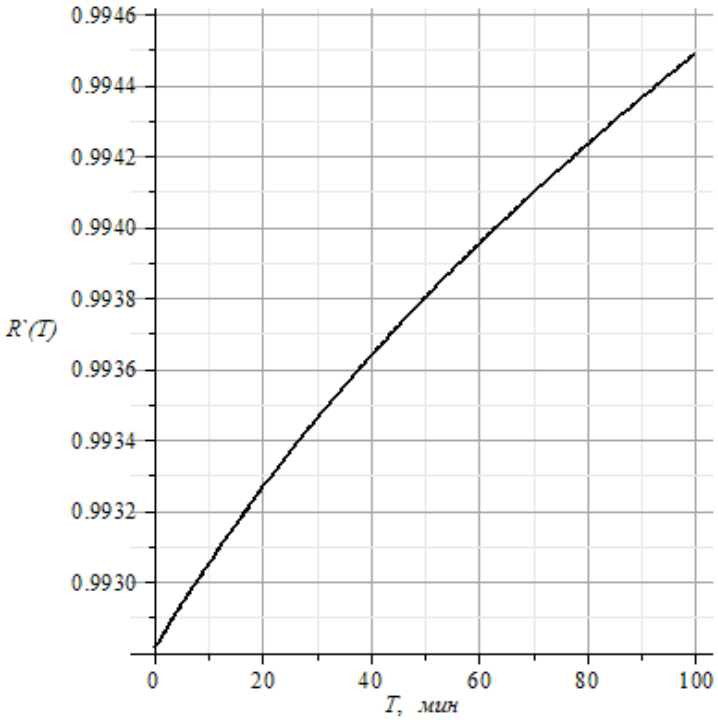


Рис. 2. Пример графической интерпретации результатов анализа надёжности

Тем не менее, применение подобной методики для расчёта системы четырёх и более уравнений Колмагорова в программе Maple является проблематичным. В подобных случаях необходимо использовать на первом этапе решение системы уравнений в преобразовании Лапласа, на втором этапе выполнить обратное преобразование служебной командой “Invlaplace”. При сложностях на втором этапе, следует применить формулу разложения для перехода от изображений к оригиналам. Кроме того, в случаях больших систем дифференциальных уравнений приходится запрашивать только приближенные значения для результатов с применением команды “evalf”, что скажется на точность расчёта.

Таким образом, анализ надёжности технических систем является сложной математической задачей, которую необходимо решать с учётом современного развития программного обеспечения, учитывая специфику объекта исследования. Не смотря на то что системы компьютерной математики имеют прикладной характер, они являются двигателями цифровизации в вопросах повышения надёжности технологических объектов.

Литература

1. Daňhel M., Štěpánek F., Kubátová H. Dependability Prediction Involving Temporal Redundancy and the Effect of Transient Faults // 2017. Euromicro Conference on Digital System Design (DSD). 2017. Pp. 360-363. https://doi.org/10.1109/DSD.2017.77

2. Sun L., Gao S., Wang L. An Automatic Test Sequence Generation Method Based on Markov Chain Model // 2021. World Conference on Computing and Communication Technologies (WCCCT). 2021. Pp. 91-96. https://doi.org/10.1109/WCCCT52091.2021.00024

CC BY

3. Zhou Y., Lin C., Liu Y., Xu H. Analytical Study on the Reliability of Redundancy Architecture for Flight Control Computer Based on Homogeneous Markov Process // IEEE Access. 2018. Vol. 6. Pp. 18290-18298. https://doi.org/10.1109/ACCESS.2018.2812819

© Гладких Т.Д., 2022

УДК 519.85
https://doi.org/10.36906/AP-2022/20

Доржиева Э.С.
Романенков А.М.
ORCID: 0000-0002-7671-3053, канд. техн. наук
Российский государственный университет им. А.Н. Косыгина
(Технологии. Дизайн. Искусство)
г. Москва, Россия

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЧЕЛОВЕКА ПРИ ПРЫЖКЕ

Аннотация. В данной работе была получена математическая модель человека при прыжке. Выбран комплекс программных, который использовался при разработке приложения. Были определены функциональные требования к разрабатываемому веб-приложению. На основе формул, найденных в ходе численного метода моделирования разработано веб-приложение, которое предоставляет функционал ввода параметров человека и выполняет построение динамического графика перегрузок. Анализ графика позволяет экспертам дать рекомендации по уменьшению нагрузки на суставы, а, следовательно, предотвратить перегрузки суставов и травмы.

Ключевые слова: прыжок; модель; человек; моделирование; веб-приложение.

Dorzhieva E.S.
Romanenkov A.M.
ORCID: 0000-0002-7671-3053, Ph.D.
Russian State University A.N. Kosygin
(Technologies. Design. Art)
Moscow, Russia

MATHEMATICAL MODELING OF HUMAN MOVEMENT DURING A JUMP

Abstract. In this paper, a mathematical model of a person jumping was obtained. The software package that was used in the development of the application was selected. The functional requirements for the web application being developed were defined. Based on the formulas found during the numerical modeling method, a web application has been developed that provides the functionality for entering human parameters and builds a dynamic overload graph. The analysis of the schedule allows experts to give recommendations on reducing the load on the joints, and, consequently, to prevent joint overload and injury.

Keywords: jump; model; human; simulation; web-application.

CC BY

Прыжок – короткий, непродолжительный взлёт в воздух, путь в воздухе после отталкивания от какой-нибудь точки опоры. Это эффективное упражнение для развития выносливости и взрывной силы. Это также стандартный тест для измерения выходной мощности спортсмена. Это также стандартный тест для измерения выходной мощности спортсмена.

В таких видах спорта, как прыжки в высоту, прыжки в длину, баскетбол и волейбол, прыжок является основным и необходимым навыком. Во многих других видах спорта с помощью прыжка измеряют физические способности спортсмена. Прыжки на обеих ногах играют важную роль в спорте и связаны с большим количеством травм, например, разрыв сухожилий. Естественно, это заставляет задуматься о параметрах, которые характеризуют прыжок и могут объяснить процесс получения травмы. Данная работа направлена на моделирование движения и изучение механизма возникновения травм человека при прыжке. Данное исследование может сократить риск получения травм, а также дать рекомендации для более эффективного выполнения элемента.

Актуальность работы обусловлена наличием большого количества видов спорта, использующих прыжковые элементы. Прыжок – основная фаза многих физических нагрузок. Анализ сложных движений человека может дать ценную информацию для реабилитации движений после полученных травм, качественной спортивной тренировки, проектирования и управления человекоподобными роботами, а также взаимодействия человека и искусственного интеллекта.

Цель работы заключается в разработка веб-приложения с элементами интеллектуального анализа и обработки данных. Предметом анализа являются масс-инерционные характеристики частей тела человека. В графическом компоненте приложения представлен функционал построения графиков зависимости изменения углов, график зависимости центра масс, график зависимости скорости и график перегрузок.

В соответствии с поставленной целью в работе необходимо решить следующие задачи:

– рассмотреть технику исполнения элемента прыжок

– изучить биомеханику двигательного аппарата человека.

– рассмотреть процесс получения травмы.

– изучить технологии, которые применяются при разработке веб-приложения

– рассмотреть программное обеспечение, которое понадобится при разработке приложения.

Данные математического моделирования элемента «прыжок» помогает выявить ошибки, которые могут возникнуть в результате неправильного управления телом при выполнении этого элемента. Корректировка выявленных ошибок позволит улучшить визуальное восприятие элемента и уменьшить вероятность травмирования.

В результате разработано открытое веб-приложение, которое может быть использовано как средство поддержки принятия решений.

В ходе работы использовались следующие методы исследования: анализ, синтез, дедукция, методы графического представления материалов.

CC BY

В данной работе человеческое тело рассматривается в виде четырёхзвенной модели. Каждое звено представляет собой твёрдый стержень и является частью тела человека (рис.).

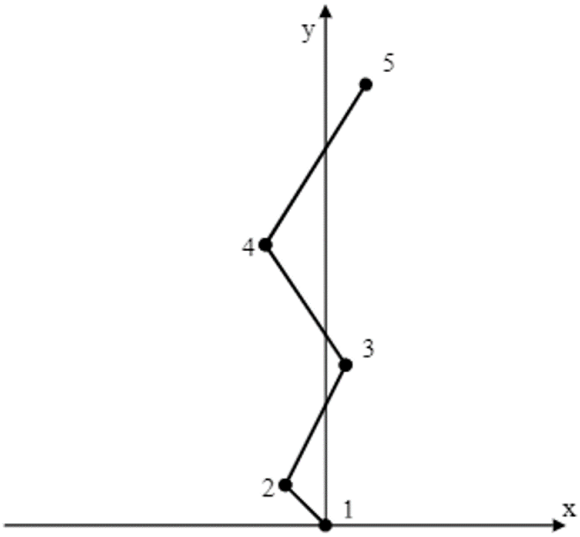


Рис. Четырёхзвенная модель человека, где 1-2: стопа, 2-3: голень, 3-4: бедро, 4-5: туловище

Опираясь на результаты, предложенные в работе «Математическое моделирование движений в художественной гимнастике» найдём координаты центров масс у отдельно взятых стержней.

Центры масс для 1-2

$$X:\ x_1 = -r_1 \sin\varphi_1 \quad (1)$$

$$Y:\ y_1 = r_1 \cos\varphi_1 \quad (2)$$

Центры масс для 2-3:

$$X: x_2 = r_2 \sin\varphi_2 - l_1 \sin\varphi_1 \quad (3)$$

$$Y: y_2 = r_2 \cos\varphi_2 + l_1 \cos\varphi_1 \quad (4)$$

Центры масс для 3-4:

$$X: x_3 = l_2 \sin\varphi_2 - l_1 \sin\varphi_1 - r_3 \sin\varphi_3 \quad (5)$$

$$Y: y_3 = r_3 \cos\varphi_3 + l_2 \cos\varphi_2 + l_1 \cos\varphi_1 \quad (6)$$

Центры масс для 4-5:

$$X: x_4 = l_2 \sin\varphi_2 - l_1 \sin\varphi_1 - l_3 \sin\varphi_3 + r_4 \sin\varphi_4 \quad (7)$$

$$Y: y_4 = r_4 \cos\varphi_4 + l_3 \cos\varphi_3 + l_2 \cos\varphi_2 + l_1 \cos\varphi_1 \quad (8)$$

Общий центр масс:

$$X: x_c = \frac{\mathrm{m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4}}{\mathrm{M}} \quad (9)$$

$$Y: y_c = \frac{\mathrm{m_1y_1 + m_2y_2 + m_3y_3 + m_4y_4}}{\mathrm{M}} \quad (10)$$

Используя координаты центров масс всех четырех стержней, мы можем написать IIзакон Ньютона.

II закон Ньютона для 1-2:

$$X: \mathrm{m_1}\frac{\mathrm{d^2x_1(t)}}{\mathrm{dt^2}} = \mathrm{F_{x1} + R_x} \quad (11)$$

CC BY

$$Y: \mathrm{m}_1 \frac{\mathrm{d}^2 \mathrm{y}_1(\mathrm{t})}{\mathrm{dt}^2} = \mathrm{F}_{\mathrm{y1}} + \mathrm{R}_{\mathrm{y}} + P_1 \quad (12)$$

II закон Ньютона для 2-3:

$$X: \mathrm{m}_2 \frac{\mathrm{d}^2 \mathrm{x}_2(\mathrm{t})}{\mathrm{dt}^2} = \mathrm{F}_{\mathrm{x1}} + \mathrm{F}_{\mathrm{x2}} \quad (13)$$

$$\mathrm{Y}: \mathrm{m}_2 \frac{\mathrm{d}^2 \mathrm{y}_2(\mathrm{t})}{\mathrm{dt}^2} = \mathrm{F}_{\mathrm{y1}} + \mathrm{F}_{\mathrm{y2}} + P_2 \quad (14)$$

II закон Ньютона для 3-4:

$$\mathrm{X}: \mathrm{m}_3 \frac{\mathrm{d}^2 \mathrm{x}_3(\mathrm{t})}{\mathrm{dt}^2} = \mathrm{F}_{\mathrm{x2}} + \mathrm{F}_{\mathrm{x3}} \quad (15)$$

II закон Ньютона для 4-5:

$$\mathrm{m}_4 \frac{\mathrm{d}^2 \mathrm{x}_4(\mathrm{t})}{\mathrm{dt}^2} = \mathrm{F}_{\mathrm{x3}} \quad (16)$$

С помощью IIзакона Ньютона, расписанного для каждого из четырех стержней отдельно выше, мы можем выразить силы, а также в дальнейшем вычислить их, которые действуют на стержни. Также, мы сможем написать модули этих сил.

Выразим силы F_{x3} и F_{y3}:

$$X: \mathrm{F}_{\mathrm{x3}} = \mathrm{m}_4 \frac{\mathrm{d}^2 \mathrm{x}_4(\mathrm{t})}{\mathrm{dt}^2} \quad (17)$$

$$\mathrm{Y}: \mathrm{F}_{\mathrm{y3}} = \mathrm{m}_4 \left(\frac{\mathrm{d}^2 \mathrm{y}_4(\mathrm{t})}{\mathrm{dt}^2} + \mathrm{g} \right) \quad (18)$$

Модуль действующих сил:

$$F_3 = \sqrt{F_{x3}^2 + F_{y3}^2} \quad (19)$$

Выразим силы F_{x2} и F_{y2}:

$$X: \mathrm{F}_{\mathrm{x2}} = \mathrm{m}_3 \frac{\mathrm{d}^2 \mathrm{x}_3(\mathrm{t})}{\mathrm{dt}^2} + \mathrm{F}_{\mathrm{x3}} \quad (20)$$

$$Y: \mathrm{F}_{\mathrm{y2}} = \mathrm{m}_3 \left(\frac{\mathrm{d}^2 \mathrm{y}_3(\mathrm{t})}{\mathrm{dt}^2} + \mathrm{g} \right) + \mathrm{F}_{\mathrm{y3}} \quad (21)$$

Модуль действующих сил:

$$F_2 = \sqrt{F_{x2}^2 + F_{y2}^2} \quad (22)$$

Выразим силы F_{x1} и F_{y1}:

$$\mathrm{X}: \mathrm{F}_{\mathrm{x1}} = \mathrm{m}_2 \frac{\mathrm{d}^2 \mathrm{x}_2(\mathrm{t})}{\mathrm{dt}^2} + \mathrm{F}_{\mathrm{x2}} \quad (23)$$

$$\mathrm{Y}: \mathrm{F}_{\mathrm{y1}} = \mathrm{m}_2 \left(\frac{\mathrm{d}^2 \mathrm{y}_2(\mathrm{t})}{\mathrm{dt}^2} + \mathrm{g} \right) + \mathrm{F}_{\mathrm{y2}} \quad (24)$$

Модуль действующих сил:

$$F_1 = \sqrt{F_{x1}^2 + F_{y1}^2} \quad (25)$$

Выразим силы R_{x} и R_{y}:

CC BY

$$X: \mathrm{R_x} = \mathrm{m_1}\frac{\mathrm{d^2 S_x(t)}}{\mathrm{dt^2}} + \mathrm{F_{x1}} \quad (26)$$

$$Y: \mathrm{R_y} = \mathrm{m_1}\frac{\mathrm{d^2 S_x(t)}}{\mathrm{dt^2}} + m_1\mathrm{g} + \mathrm{F_{y1}} \quad (27)$$

Модуль действующих сил:

$$R = \sqrt{R_x^2 + R_y^2} \quad (28)$$

С помощью координат центра масс всего тела мы можем найти скорость этого тела.

Скорость по оси x:

$$\mathrm{V_x} = \frac{\mathrm{d\,x_c(t)}}{\mathrm{dt}} \quad (29)$$

Скорость по оси y:

$$\mathrm{V_y} = \frac{\mathrm{d\,y_c(t)}}{\mathrm{dt}} \quad (30)$$

Общая скорость тела:

$$V_c = \sqrt{V_x^2 + V_y^2} \quad (31)$$

Момент инерции:

Для 1, 2 (относительно точки 1):

$$M_{F_{y1}} + M_{F_{x1}} + M_{R_x} + M_{R_y} + M_{P_1} = I\varphi'' \quad (32)$$

$$M_{F_{x1}} = 0 \quad (33)$$

$$M_{F_{y1}} = 0 \quad (34)$$

$$M_{P_1} = \frac{l_{12}}{2}\cos\alpha \quad (35)$$

$$M_{F_{y2}} = l_{12}\cos\alpha \quad (36)$$

$$M_{F_{x2}} = l_{12}\cos\alpha \quad (37)$$

$$\frac{l_{12}}{2}\cos\alpha + l_{12}\cos\alpha + l_{12}\cos\alpha = \frac{m_{12}l_{12}^2}{12}\frac{d\omega(\mathrm{t})}{dt} \quad (38)$$

$$5\cos\alpha = \frac{m_{12}l_{12}^2}{6}\frac{d\omega(\mathrm{t})}{dt} \quad (39)$$

$$\frac{d\omega(\mathrm{t})}{dt} = \frac{5\cos\alpha}{6\,m_{12}\,l_{12}} \quad (40)$$

Для 2, 3 (относительно точки 3):

$$M_{P_{\mathrm{II}}} + M_{F_{y^2}} + M_{F_{x^2}} + M_{F_{y^3}} + M_{F_{x^3}} = I\varphi'' \quad (41)$$

$$\mathrm{M_{P_{II}}} = \frac{\mathrm{l_{23}}}{2}\cos\beta \quad (42)$$

$$M_{F_{y2}} = l_{23}\cos\beta \quad (43)$$

$$M_{F_{x2}} = l_{23}\cos\beta \quad (44)$$

$$M_{F_{x3}} = 0 \quad (45)$$

$$M_{F_{y3}} = 0 \quad (46)$$

$$\frac{l_{23}}{2}\cos\beta + l_{23}\cos\beta + l_{23}\cos\beta = \frac{m_{23}l_{23}^2}{12}\frac{d\omega(\mathrm{t})}{dt} \quad (47)$$

CC BY

$$5\cos\beta = \frac{m_{23} l_{23}^2}{6} \frac{d\omega(t)}{dt} \quad (48)$$

$$\frac{d\omega(t)}{dt} = \frac{5\cos\beta}{6\, m_{23}\, l_{23}} \quad (49)$$

Разработано веб-приложение, которое предоставляет функционал ввода параметров спортсмена $(l_1, r_1, m_1, l_2, r_2, m_2, l_3, r_3, m_3, l_4, r_4, m_4, M, g, t)$ и выполняет построение динамических графиков, а именно: график зависимости изменения углов туловищем и конечностями тела, график зависимости центра масс, график перегрузок. Анализ графиков позволяет экспертам дать рекомендации по уменьшению нагрузки на суставы, а следовательно, предотвратить перегрузки суставов и травмы.

© Доржиева Э.С., Романенков А.М., 2022

УДК 622.279
https://doi.org/10.36906/AP-2022/21

Казымов Б.З.
канд. техн. наук
Институт нефти и газа НАН Азербайджана
Баку, Азербайджан
Зейналов Р.М.
Ph.D.
Институт систем управления НАН Азербайджана
Баку, Азербайджан

ОПРЕДЕЛЕНИЕ ДИНАМИКИ ПОТРЕБНОГО КОЛИЧЕСТВА СКВАЖИН ПРИ РАЗРАБОТКЕ ГЛУБОКОЗАЛЕГАЮЩИХ НЕФТЯНЫХ ЗАЛЕЖЕЙ В РЕЖИМЕ РАСТВОРЕННОГО ГАЗА

Аннотация. В статье была исследована задача определения динамики потребного количества скважин при разработке нефтяных залежей в режиме растворенного газа, в случаях нелинейно-упругой и ползучей деформации горных пород. По установленной соответствующей схеме были осуществлены численные расчёты, и, на примере гипотетической залежи, определена динамика потребного количества скважин.

Ключевые слова: нефтяная залежь; режим растворенного газа; деформация; ползучесть; проницаемость; пористость; потребное количество скважин.

Kazimov B.Z.
Ph.D.
Institute of Oil and Gas of the Azerbaijan National Academy of Sciences
Baku, Azerbaijan
Zeynalov R.M.
Ph.D.
Institute of Control Systems of the Azerbaijan National Academy of Sciences
Baku, Azerbaijan

DETERMINATION OF THE DYNAMICS OF THE REQUIRED NUMBER OF WELLS DURING THE DEVELOPMENT OF DEEP-LYING OIL DEPOSITS IN THE DISSOLVED GAS MODE

Abstract. The article investigated the problem of determining the dynamics of the required number of wells during the development of oil deposits in the mode of dissolved gas, in cases of nonlinear elastic and creeping deformation of rocks. According to the established corresponding scheme, numerical calculations were carried out, and, using the example of a hypothetical reservoir, the dynamics of the required number of wells was determined.

CC BY

Keywords: oil deposit; dissolved gas regime; deformation; creep; permeability; porosity; required number of wells.

Соблюдение регулируемых режимов разработки нефтяных и газовых залежей является необходимым условием оптимального освоения их запасов. При различных режимах разработки это условие удовлетворяется за счёт определение потребного количества скважин, обеспечивающее заданную динамику добычи [2; 3; 5]. Анализ степени изученности этого вопроса показывает, что задача определения динамики потребного количества скважин при разработке, в частности, нефтяных залежей в режиме истощения, в случае ползучести горных пород с учётом влиянии этого вида деформации на пористость и проницаемость пласта [1; 4; 5]. Исследование этой задачи позволило-бы при разработке глубокозалегающих нефтяных месторождений по годам разработки, с учётом изменение коллекторских свойств пластов, определить потребного количества скважин, обеспечивающую заданную прогнозируемую динамику добычи нефти из месторождения. Статья посвящена исследованию данного вопроса.

Пусть задана прогнозируемая динамика добычи нефти из залежи ($q(t)$). Требуется определять динамику потребного количества скважин $n(t)$. Зная прогнозируемый среднегодовой дебит одной (центральной) скважины $q_1(t)$, эта величина легко определяется таким образом [2; 3]:

$$n(t) = \frac{q(t)}{q_1(t)}. \tag{1}$$

Динамика добычи нефти из залежи ($q(t)$) в теоретических расчётах, проводимых на этапе проектирование разработки месторождения, может быть осуществляться согласно теоретической формуле

$$q = \frac{q_{\max} t}{t_1} e^{-\left[\frac{t}{t_1}-1\right]}$$

где $q_{\max}$ – максимальное значение добычи из месторождения; t_1 – время достижения максимальное значение добычи q.

В случае разработки нефтяной залежи в режиме растворенного газа дебит одной скважины, на основе принятой сетки скважин, может быть определен согласно формуле [5; 6]:

$$q_1(t) = \frac{\pi h}{\left(\ln\frac{r_k}{r_c} - \frac{1}{2}\right)}\left(\frac{k(p_c)F_н(\sigma_c)}{\mu(p_c)a(P_c)} + \frac{k(p)F_н(\sigma)}{\mu(p)a(p)}\right)(p - p_c) \tag{2}$$

где p и σ – соответственно пластовое давление и нефтенасыщенность пласта; p_c и σ_c – соответственно забойные значения пластового давления и нефтенасыщенности пласта; k – проницаемость пласта; $F_н$ – относительная фазовая проницаемость пласта для нефти; $a(P)$ – объёмный коэффициент нефти; h – толщина залежи; μ – вязкость нефти в пластовых условиях; r_k – эффективный радиус контура питания скважины; r_c – радиус скважины.

CC BY

Эффективный радиус контура питания скважины определяется в зависимости от формы принимаемой сетки скважин и расстояния между скважин (R).

Влияние ползучей деформации горных пород на изменение пористости (m) и проницаемости (k) пласта учитываем согласно выражениям [4]:

$$\overline{m} = 1 + \beta_n (p - p_0) + m_1 \int_0^t e^{-\gamma_m (t-\tau)} (p - p_0) d\tau$$

$$\overline{k} = 1 + \alpha_k (p - p_0) + k_1 \int_0^t e^{-\gamma_k (t-\tau)} (p - p_0) d\tau \qquad (3)$$

где $\overline{m} = \frac{m}{m_0}$; $\overline{k} = \frac{k}{k_0}$; p_0, m_0, k_0 – соответственно начальные значения пластового давления, пористости и проницаемости пласта; β_n – коэффициент сжимаемости горных пород; α_k – коэффициент, характеризующий темп изменение проницаемости в зависимости пластового давления; m_1, γ_m, k_1, γ_k – параметры ползучести.

Средневзвешенные значения пластового давления, нефтенасыщенности и проницаемости пласта, необходимые для определение дебита одной скважины по формуле (2), с учетом влияние ползучести горных пород на значения пористости и проницаемости пласта, определяются на основе следующих уравнений материального баланса нефти и газа в залежи при соответствующих начальных условиях [6]:

$$q(t) = -\Omega_з \frac{d}{dt}\left(\frac{\overline{m}\sigma}{\overline{a}(P)}\right) \qquad (4)$$

$$q(t) \cdot \Gamma \cdot p_{am} = -\Omega_з \frac{d}{dt}\left(a(P_0)\overline{m}P(1-\sigma) + \frac{\overline{m}\sigma}{\overline{a}(P)} \frac{S(P) p_{am}}{\beta}\right)\beta \qquad (5)$$

где $\overline{a}(P) = \frac{a(P)}{a(P_0)}$; Γ – газовый фактор; $S(P)$ - коэффициент растворимости газа в нефти; $\Omega_з$ – запас нефти в залежи; β – температурная поправка для газа; p_{am} – атмосферное давление.

Забойные значения нефтенасыщенности пласта (σ_c) определяется из условия равенства газового фактора в контуре и забое скважины ($\Gamma_c = \Gamma_k$).

Через величину депрессии пласта ΔP забойное давление p_c может определяться, согласно соотношению:

$$p_c = p - \Delta P. \qquad (6)$$

На основе расчётной схемы (1)–(6) при следующих исходных данных проведены расчёты:

$P_0 = 40$ МПа; $\Omega_з = 35 \cdot 10^6$ м^3; $m_0 = 0{,}2$; $q_{\max} = 0{,}5 \cdot 10^6$ м^3/год; $t_1 = 5$ лет; $h = 8$ м; $\beta = 0{,}8$;

$a(P_0) = 1{,}25$; $\alpha_k = 5 \cdot 10^{-3}$ МПа$^{-1}$; $\beta_n = 2{,}5 \cdot 10^{-3}$ МПа$^{-1}$, $\Delta\overline{P} = \Delta P / P_0 = 0{,}1$; $k_0 = 0{,}05$ дарси;

$R = 200$ м; $R = 400$ м; $m_1 = 3{,}4 \cdot 10^{-8}$ (МПа·с)$^{-1}$; $k_1 = 10^{-5}$ (МПа·с)$^{-1}$;

CC BY

$$\gamma_m = 4{,}3\cdot 10^{-6}\ \text{с}^{-1};\ \gamma_k = 10^{-4}\ \text{с}^{-1};\ \mu_н = 0{,}5\cdot \text{e}^{-0{,}05(P-P_0)},\ \text{спз};\ F_н(\sigma) = 1{,}06\sigma^3 - 0{,}06;$$

$$F_г(\sigma) = 1{,}16(1-\sigma)^2;\ \mu_г = 0{,}0054\left(\frac{P}{P_0}\right)^2 + 0{,}0114\frac{P}{P_0} + 0{,}0105,\ \text{спз};$$

$$S(P) = \begin{cases} 3{,}9\cdot P + 15{,}5, & P \ge 5\ \text{МПа} \\ 7\cdot P, & P \le 5\ \text{МПа} \end{cases};\ a(P) = \begin{cases} 0{,}0058\cdot P + 1{,}021, & P \ge 5\ \text{МПа} \\ 0{,}01\cdot P + 1, & P \le 5\ \text{МПа} \end{cases}$$

где $F_г$ – относительная фазовая проницаемость пласта для газа.

Результаты расчётов, проведены на рисунке. На рисунке приведены динамики потребного количества скважин для случаев нелинейно-упругой (пунктирные линии) и ползучей деформации (сплошные линии) горных пород для различных значений расстояния между скважин (R).

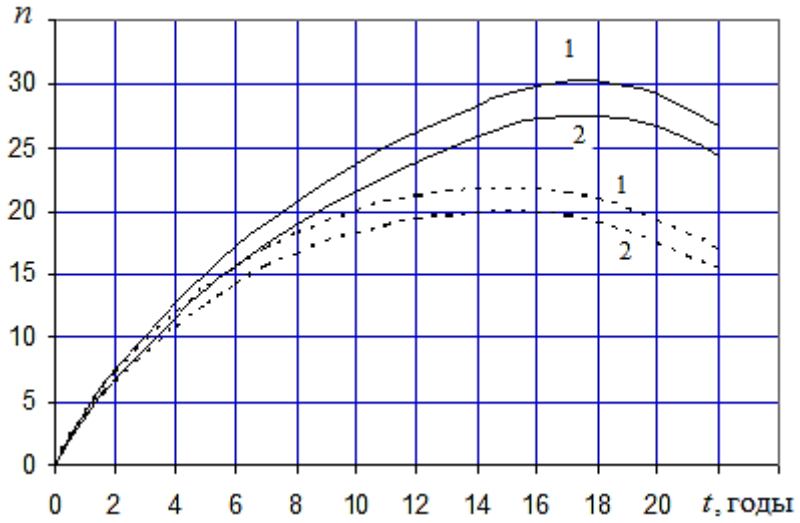


Рис. Изменение во времени потребного количества скважин
1 – $R = 400$ м; 2 – $R = 200$ м

Анализ, представленных кривых, показывает, что:

1) Значения потребного количества скважин в обоих случаях расстояния между скважинами, случаю нелинейно-упругой деформации горных пород пласта, в случае ползучей деформации больше (максимально до 8 скважин);

2) Различные значения расстояния между скважин по течению определенного периода разработки (в данном расчётном случае после 5 лет) на значения потребного количества скважин начинают оказывать более существенное влияние, и этот период разработки определяется временем достижения максимальной добычи нефти из месторождений.

Результаты расчётов свидетельствуют о необходимости учёта ползучей деформации горных пород при определении потребного количества скважин при разработке глубокозалегающих нефтяных месторождений в режиме истощения.

Литература

1. Аметов И.М., Басниев К.С. Фильтрация жидкости и газа в ползучих средах // Известия АН СССР. Механика жидкости и газа. 1981. № 4. С. 150-153.

2. Казымов Б.З., Насирова К.К., Бунятов А.А.Определение динамики потребного количества скважин при разработке нефтяных залежей в режиме растворенного газа // Учёный XXI века. 2021. № 6(77). С. 3-8. https://doi.org/10.15350/24103586.2021.6.1

3. Кулиев А.М., Гаджиев М.А., Эфендиев Р.М. Влияние релаксации горных пород на потребное количество скважин для разработки газовых месторождений // Научные труды НИИ «Геотехнологические проблемы нефти и газа и химия». Баку, 2007. Т. VIII. С. 282-291.

4. Кулиев А.М., Казымов Б.З. Деформация горных пород и её влияние на их фильтрационно-ёмкостные свойства и на процессы фильтрации и разработки месторождений нефти и газа. Баку: Элм, 2009. 88 с.

5. Кулиев А.М., Эфендиев Р.М, Казымов Б.З. Влияние ползучести горных пород на нефтеотдачу залежей, разрабатываемых в естественном режиме // SOCAR Proceedings. 2012. № 3. С. 32-37. https://doi.org/10.5510/OGP20120300120

6. Розенберг М.Д., Кундин С.А., Курбанов А.К., Суворов Н.И., Шовкринский Г.Ю. Фильтрация газированной жидкости и других многокомпонентных смесей в нефтяных пластах. М.: Недра, 1969. 453 с.

© Казымов Б.З., Зейналов Р.М., 2022

УДК 519.85
https://doi.org/10.36906/AP-2022/22

Кузнецов В.Н.
канд. физ.-мат. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия

МАТЕМАТИЧЕСКИЙ АНАЛИЗ ФУНКЦИЙ БУЛЕВОЙ АЛГЕБРЫ: ТАБЛИЧНО-АНАЛИТИЧЕСКОЕ ИСЧИСЛЕНИЕ, СТРОГИЙ ПОРЯДОК КОНЪЮНКТИВНЫХ НОРМАЛЬНЫХ ФОРМ И ПРОИЗВОДНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ ФУНКЦИЙ

Аннотация. В статье рассматриваются вопросы математических основ функций алгебры логики, развитие аналитических методов представления и исследования этих функций.

Ключевые слова: функция; булева алгебра; исчисление; исследование функций.

Kuznetsov V.N.
Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

MATHEMATICAL ANALYSIS OF BOOLEAN ALGEBRA FUNCTIONS: TABULAR-ANALYTICAL CALCULUS, STRICT ORDER OF CONJUNCTIVE NORMAL FORMS AND DERIVED METHODS OF FUNCTION RESEARCH

Abstract. The article deals with the mathematical foundations of the functions of the algebra of logic, the development of analytical methods of representation and research of these functions.

Keywords: function; Boolean algebra; calculus; research of functions.

Со времён Джорджа Буля, заложившего математические основы функций алгебры логики (ФАЛ) [1], стали развиваться аналитические методы представления и исследования этих функций, позволяющие оперировать с ними так же удобно и привычно, как с арифметическими формулами. При этом, по сути являющиеся определяющими, табличные методы не без оснований считались громоздкими (удвоение размеров таблиц при увеличении числа булевых переменных на единицу) и использовались лишь в простейших случаях и/или для иллюстративных целей. В данной работе, вопреки исторически сложившимся предубеждениям, разработано неудлиняющее таблично-аналитическое представления ФАЛ – их исчисление, совмещающее достоинства как табличного, так и аналитического методов. В этих рамках естественно определяется и доказывается фундаментальное свойство конъюнктивных нормальных форм (КНФ) – их строгая упорядоченность (без привлечения

CC BY

каких-либо дополнительных функций). Также рассматривается набор актуализированных методов исследования ФАЛ, что иллюстрируется на примерах задач выполнимости КНФ.

A.1 *zu*↑-представление функций алгебры логики

Для систематического построения нового представления рассмотрим предварительно простой пример обычного табличного представления КНФ, зависящей от трёх булевых переменных и содержащей четыре дизъюнкта.

Таблица 1

Табличное представление КНФ $f(x_1, x_2, x_3) = d_1 \wedge d_2 \wedge d_3 \wedge d_4$

	№	x_1	x_2	x_3	$d_1 \wedge d_2 \wedge d_3 \wedge d_4 = f$				
$d_1 = x_1 \vee x_2 \vee x_3$ $d_2 = x_1 \vee \overline{x_3}$ $d_3 = x_1 \vee \overline{x_2} \vee x_3$ $d_4 = \overline{x_1} \vee \overline{x_2}$	0	0	0	0	0	1	1	1	0
	1	1	0	0	1	1	1	1	1
	2	0	1	0	1	1	0	1	0
	3	1	1	0	1	1	1	0	0
	4	0	0	1	1	0	1	1	0
	5	1	0	1	1	1	1	1	1
	6	0	1	1	1	0	1	1	0
	7	1	1	1	1	1	1	0	0

При рассмотрении таблицы прежде всего особо следует подчеркнуть, что область определения заданных функций – множество трёхзначных двоичных наборов (x_1, x_2, x_3) от (0,0,0) до (1,1,1), упорядоченных по возрастанию их двоичных значений: №№ = 0,1, … ,7.

Рассмотрим теперь простейшие однобуквенные дизъюнкты $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ – проекционные базисные переменные. Их табличные представления – столбцы, выделенные из их общей области определения, – очевидно, являются периодическими последовательностями двоичных цифр 0 и 1. Например, область упорядоченных значений переменной $\boldsymbol{x}_3$ содержит сначала четыре цифры 0, т. е. 2^2=4 раза, после которых следует четыре цифры 1, т.е также 2^2=4 раза. Их итоговая периодическая последовательность может быть описана символическим выражением вида $(0\#2^2)(1\#2^2)$, означающим конкатенацию двух строк равной длины $(0\#2^2)$ и $(1\#2^2)$. Разделительный знак # означает повторение предшествующего выражения указанное число раз. Аналогично, область упорядоченных значений проекционной переменной $\boldsymbol{x}_2$ может быть описана символическим выражением $((0\#2^1)(1\#2^1))\#2^1$.

При упорядоченной организации областей определений булевых функций отрезки нулей и единиц всегда будут иметь длину, кратную 2^k (достаточно экстраполировать область определения в рассматриваемой таблице до n=4). Поэтому для упрощения описания этих областей, как и в любых других описаниях, содержащих степени 2^k, основание 2 (с предшествующим разделительным символом #), будем опускать, всегда предполагая их по умолчанию. В результате получаем описания вида $0^2 1^2$ и $(0^1 1^1)^1$. Наконец, чтобы избежать путаницы в обозначениях булевых цифр 0 и 1, с одной стороны, и показателей 0 и 1 в степенях 2^0 и 2^1, с другой, целесообразно заменить булевы цифры 0 и 1 буквенными обозначениями, например:

$$0 \rightarrow \text{z (zero)}; \qquad (1.1)$$

CC BY

$$1 \to \mathrm{u}\ (\mathrm{unit}). \qquad (1.2)$$

В итоге получаем ***zu*↑**-представление таблиц базисных переменных: $\boldsymbol{x}_1=\mathrm{zu}$, $\boldsymbol{x}_3=\mathrm{z}^2\mathrm{u}^2$. Отсутствие степени после букв z и u в первом предыдущем равенстве формально обозначает её нулевое значение: $\mathrm{zu}=\mathrm{z}^0\mathrm{u}^0=(\mathrm{z}\#2^0)(\mathrm{u}\#2^0)=(\mathrm{z}\#1)(\mathrm{u}\#1)=\mathrm{zu}$.

Обычно младшие переменные (или формулы) рассматриваются в пространстве значений старших переменных (или формул). В этих случаях пространства значений младших переменных (формул) будем расширять (продолжать) до размерности пространства старших переменных (формул) путём периодических удвоений пространства младших. Например, продолжение пространств младших переменных $\boldsymbol{x}_1$ и $\boldsymbol{x}_2$ до размерности пространства старшей переменной $\boldsymbol{x}_3$ имеет вид:

$$(\boldsymbol{x}_1)^2=(\mathrm{zu})^2,\ (\boldsymbol{x}_2)^1=(\mathrm{z}^1\mathrm{u}^1)^1,\ \ \boldsymbol{x}_3=\mathrm{z}^2\mathrm{u}^2.$$

Число повторений пространства младшей переменной (формулы) определяется разностью между размерностями пространств старшей и младшей переменных (формул). В контексте предыдущего примера получаем: для $\boldsymbol{x}_1$: 2=3-1; для $\boldsymbol{x}_2$: 1=3-2; для $\boldsymbol{x}_3$: 0=3-3. (Напомним, полученные размерности 2, 1, 0 следует понимать как степени двойки – 2^2, 2^1, 2^0, соответственно.) Примером периодического продолжения табличной формулы является столбец дизъюнкта $\boldsymbol{d}_4$ в таблице 1.

Для того, чтобы подчеркнуть, что произвольная строка a длины $|a|$ периодически продолжена до пространства размерности $s>|a|$, используем обозначение $a^{[s]}$. Оно также обозначает число 2^s символов в расширенной строке a – т.е. её длину $|a^{[s]}|=2^s$. Примеры: $|\boldsymbol{x}_1^{[3]}|=2^3=8$; $\boldsymbol{d}_4^{[3]}=(\ \bar{x}_1 \vee \bar{x}_2)^{[3]}=(\ \bar{x}_1 \vee \bar{x}_2)^1$.

В контексте определённой выше символики, получаем в итоге следующие общего характера основополагающие ***zu*↑**-представления для n базисных переменных:

$$\boldsymbol{x}_k^{[n]}=(\mathrm{z}^{\,k-1}\mathrm{u}^{\,k-1})^{\,n-k},\quad k=1,\ldots,n, \qquad (2.1)$$

$$\bar{\boldsymbol{x}}_k^{[n]}=(\mathrm{u}^{\,k-1}\mathrm{z}^{\,k-1})^{\,n-k},\quad k=1,\ldots,n, \qquad (2.2)$$

$$\boldsymbol{0}=\mathrm{z}^{\,n}\ -\ \text{тождественно нулевая функция}, \qquad (2.3)$$

$$\boldsymbol{1}=\mathrm{u}^{\,n}\ -\ \text{тождественно единичная функция}. \qquad (2.4)$$

Например, в частном случае k=n=3, получаем $\bar{\boldsymbol{x}}_3^{[3]}=(\mathrm{u}^{3-1}\mathrm{z}^{\,3-1})^{3-3}=(\mathrm{u}^2\mathrm{z}^{\,2})^0=\mathrm{u}^2\mathrm{z}^{\,2}$.

Следует ещё раз подчеркнуть на будущее, что конкатенация $\mathrm{z}^k\mathrm{u}^k$ имеет дину: $|\mathrm{z}^k\mathrm{u}^k|=|\mathrm{z}^k|+|\mathrm{u}^k|=2^k+2^k=2\times2^k=2^{k+1}$, конкатенация $\mathrm{z}^{k+2}\mathrm{u}^{k+2}$ имеет длину 2^{k+3}, …

Дальнейшее развитие ***zu*↑** – представления булевых функций сводится к определению бинарных операций дизъюнкции и конъюнкции над ними. Суть этих определений сводится в конце концов к выполнению этих операций над отдельными символами z и/или u, позиционно одинаково расположенными в столбцах одинаковой высоты в таблицах двух различных булевых функций. Обобщение этих операций на параллельные строки (столбцы) одинаковой длины 2^s даётся нижеследующими равенствами, постулирующими логические свойства этих операций:

$$a^{[s]} \vee \mathrm{z}^s = a^{[s]}, \qquad a^{[s]} \vee \mathrm{u}^s = \mathrm{u}^s, \qquad (3.1)$$

$$a^{[s]} \wedge \mathrm{z}^s = \mathrm{z}^s, \qquad a^{[s]} \wedge \mathrm{u}^s = a^{[s]}. \qquad (3.2)$$

CC BY

где $a^{[s]}$, как отмечалось выше, – произвольная строка, содержащая 2^s букв. Согласно свойству коммутативности операций дизъюнкции и конъюнкции их результат не зависит от порядка следования строк-операндов.

Для выделения параллельных строк (столбцов) одинаковой длины бывает необходимым исходно данные строки выровнять. А именно, более короткую строку $a^{[s]}$ удлинить путём её периодического повторения до размера более длинной строки $a^{[t]}$:

$$a^{[s]} \to (a^{[s]})^{t-s} = a^{[t]}, \quad t>s;$$

а более длинную строку b^t путём её половинного членения представить в виде конкатенации составляющих её частей:

$$b^t = b^{t-1}\, b^{t-1} = (b^{t-2}\, b^{t-2})(b^{t-1})=\dots$$

Например, в случае s=2, t=5, получаем $a^{[2]} \to (a^{[2]})^{5-2} = a^{[5]}$, $b^5 = b^{5-1}b^{5-1} = b^4b^4 = (b^{4-1}b^{4-1})$

$$(b^{4-1}\, b^{4-1}) = (b^3b^3)(b^3b^3) =\dots$$

При построении табличных формул важно контролировать равенство длин подстрок, участвующих в тех или иных операциях. Рассмотрим нижеследующий иллюстрирующий пример.

$$(\boldsymbol{x}_3\vee\boldsymbol{x}_5)^{[7]}=((z^2u^2)^2\vee(z^4u^4))^2=((z^2u^2)^1(z^2u^2)^1\vee(z^4u^4))^2=(((z^2u^2)^1\vee z^4)((z^2u^2)^1\vee u^4))^2=(\mathbf{(z^2u^2)^1u^4})^2.$$

Здесь сначала таблица переменной $\boldsymbol{x}_3$ расширяется до размерности переменной $\boldsymbol{x}_5$ и её формула периодически повторяется $2^{5-3}=2^2$ раз, что отображается её возведением в степень 2^2: $(z^2u^2)^2$. Размерность формулы (z^2u^2), согласно (2.1), равна 3. Размерность формулы $(z^2u^2)^2$ равна 3+2=5, согласно правилу перемножения степеней: $2^3\times2^2=2^{3+2}=2^5$. Эту же степень 5 имеет и формула (z^4u^4), согласно (2.1). В следующем по порядку равенстве формула $(z^2u^2)^2$ размерности 5 декомпозируется на конкатенацию двух подформул $(z^2u^2)^1$, каждая размерности 4. Эти две подформулы почленно участвуют в операции дизъюнкции с двумя подформулами z^4 и u^4, соответственно (каждая также размерности 4). Окончательная формула является конкатенацией двух подформул $(z^2u^2)^1$ и u^4 одинаковой размерности 4 – результаты двух дизъюнкций . В итоге получаем формулу размерности 5, которая и расширяется до размерности 7 её возведением в степень 7-5=2.

Построим теперь $\boldsymbol{zu}\uparrow$ – представления дизъюнктов $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, $\boldsymbol{d}_3$, $\boldsymbol{d}_4$ из таблицы 1:

$\boldsymbol{d}_1= x_1\vee x_2\vee x_3=((zu)^1\vee(z^1u^1))^1\vee(z^2u^2)=((zu)u^1))^1\vee(z^2u^2)=((zu)u^1)u^2.$

$\boldsymbol{d}_2= x_1\vee \bar{x}_3=(zu)^2\vee u^2z^2= u^2(zu)^1.$

$\boldsymbol{d}_3= x_1\vee \bar{x}_2\vee x_3=((zu)^1\vee(u^1z^1))^1\vee(z^2u^2)= (u^1(zu))^1\vee(z^2u^2)= (u^1(zu))u^2.$

$\boldsymbol{d}_4=(\bar{x}_1\vee \bar{x}_2)^{[3]}=((uz)^1\vee(u^1z^1))^{3-2}=(u^1(uz))^1.$

Полученные $\boldsymbol{zu}\uparrow$ – представления дизъюнктов (т. е., по сути дела, их табличные представления) позволяют определить отношение строгого порядка на их множестве. В основу этого определения положим следующие общего характера построения и очевидные утверждения о них.

Разделим всё исходное множество дизъюнктов на два непересекающихся подмножества – в одном, назовём его *P* (pozitive variables), содержатся только "положительные" дизъюнкты – без отрицательных базисных переменных, а в другом, назовём его *N* (negative variables),

CC BY

содержатся дизъюнкты, каждый имеющий хотя бы одну «отрицательную» базисную переменную.

В *P*-секции определим младшим тот дизъюнкт, в начале таблицы которого фигурируют наибольшее число подряд идущих нулей (z). Если в двух разных дизъюнктах в началах их таблиц содержатся одинаковое число нулей подряд, то дальнейшее определение младшего дизъюнкта сводится к сравнению оставшихся частей их таблиц, руководствуясь этим же заданным критерием.

В *N*-секции младшим дизъюнктом определим тот, который в начале своей таблицы содержит наименьшее число единиц (u) подряд. Если в двух разных дизъюнктах в началах их таблиц содержатся одинаковое число единиц подряд, то дальнейшее определение младшего дизъюнкта сводится к сравнению оставшихся частей их таблиц, руководствуясь этим же заданным критерием.

Полагая, по определению, z<u, получаем, что любой дизъюнкт из *P*-секции является младшим по отношению к любому дизъюнкту из *N*-секции.

Для рассмотренного выше примерного множества дизъюнктов (табл. 1) получаем следующий их порядок по возрастанию: $\boldsymbol{d}_1<\boldsymbol{d}_3<\boldsymbol{d}_4<\boldsymbol{d}_2$. В начале таблицы дизъюнкта $\boldsymbol{d}_1$ фигурирует нуль (z), для $\boldsymbol{d}_3$ – 2 единицы подряд (u^1), для $\boldsymbol{d}_4$ – 3 единицы подряд ((u^1u), для $\boldsymbol{d}_2$ – 4 единицы подряд (u^2). Тогда КНФ $\boldsymbol{f}(x_1, x_2, x_3)=\boldsymbol{d}_1\wedge\boldsymbol{d}_3\wedge\boldsymbol{d}_4\wedge\boldsymbol{d}_2$ с перечисленными по возрастанию её дизъюнктов, становится строго упорядоченной.

А.2 ↑*n.p* – представления функций алгебры логики.

Полученные выше частные выводы о строгом упорядочении дизъюнктов можно получить обобщённым способом.

Для этого сначала из строго упорядоченного ***zu***↑ – представления каждого дизъюнкта непосредственно реконструируем его традиционную нотационную структуру. Эта реконструкция требует следующего анализа имеющегося ***zu***↑ – представления дизъюнкта и выполнения следующих действий:

1) отыскивается буква z или u наивысшей степени;

2) содержащая найденную букву строка обособляется скобками, расчленяется пополам, а полученные половинки соединяются знаком дизъюнкции ∨;

3) половинная строка, содержащая одну единственную найденную букву в высшей степени, удваивает свою длину путём её конкатенации с альтернативной буквой в той же степени, при этом конкатенация с альтернативной буквой осуществляется с той стороны, где расположена оставшаяся строка-половинка;

4) оставшаяся строка-половинка удваивается путём заключения её в скобки и возведения этих скобок в степень 1;

5) альтернативные буквы в высших степенях декодируются согласно определениям (2.1) и (2.2) в «классическую» нотацию типа x_k или $\bar{x}_k$;

6) оставшаяся строка-операнд операции дизъюнкции подлежит реконструкции, начиная с п.1.

Примеры:

$\boldsymbol{d}_1 = x_1 \vee x_2 \vee x_3 = ((zu)u^1)u^2 = ((zu)u^1)^1 \vee (z^2u^2) = ((zu)^1 \vee (z^1u^1))^1 \vee x_3 = ((x_1)^1 \vee x_2)^1 \vee x_3 = (x_1)^2 \vee (x_2)^1 \vee x_3$
$= (x_1)^{[3]} \vee (x_2)^{[3]} \vee x_3 = x_1 \vee x_2 \vee x_3;$

$\boldsymbol{d}_2 = x_1 \vee \bar{x}_3 = u^2(zu)^1 = u^2z^2 \vee (zu)^{[3]} = \bar{x}_3 \vee x_1;$

$\boldsymbol{d}_3 = x_1 \vee \bar{x}_2 \vee x_3 = (u^1(zu))u^2 = (u^1(zu))^1 \vee z^2u^2 = (u^1z^1 \vee (zu)^1)^1 \vee z^2u^2 = (\bar{x}_2 \vee x_1)^{[3]} \vee x_3 = \bar{x}_2 \vee x_1 \vee x_3;$

$\boldsymbol{d}_4 = (\bar{x}_1 \vee \bar{x}_2) = u^1z^1 \vee (uz)^1 = \bar{x}_2 \vee (\bar{x}_1)^{[2]} = \bar{x}_2 \vee \bar{x}_1.$

Важно заметить, что реконструированная структура дизъюнкта отличается от исходно заданной порядком перечисления её отрицательных базисных переменных $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$. Именно: инвертированные базисные переменные в нём предшествуют всем остальным. При этом их нумерующие индексы следуют в порядке убывания (см. реконструированную структуру $\boldsymbol{d}_4$). Неотрицательные переменные, следующие за отрицательными, перечисляются в порядке возрастания их нумерующих индексов.

Выявленные особенности реконструированной структуры дизъюнктов позволяют сразу задавать в них требуемую структуру строго табличного порядка переменных. Более того, эти особенности позволяют значительно упростить запись дизъюнктов. Для этого достаточно отделить инвертированные переменные от остальных, скажем, точкой, затем опустить в них знаки инвертирования, а также опустить символический знак $\boldsymbol{x}$, одинаковый для каждой переменной. В итоге в описании дизъюнкта остаются лишь индекс-номера его переменных и разделяющая их точка; слева от точки расположены индекс-номера негативных переменных (если таковых нет, то их будет представлять цифра 0); справа от точки расположены индекс-номера положительных переменных (если таковых нет, то их будет представлять цифра 0). Пример $\uparrow\boldsymbol{n.p}$ – представления:

$$\boldsymbol{f}(x_1, x_2, x_3) = \boldsymbol{d}_1 \wedge \boldsymbol{d}_3 \wedge \boldsymbol{d}_4 \wedge \boldsymbol{d}_2 = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_2 \vee x_1 \vee x_3;) \wedge (\bar{x}_2 \vee \bar{x}_1) \wedge (\bar{x}_3 \vee x_1)$$
$$= (0.123) \wedge (2.13) \wedge (21.0) \wedge (3.1) = 0.123,\ 2.13,\ 21.0,\ 3.1$$

Для сокращения записи КНФ здесь и в дальнейшем будем разделять дизъюнкты между собой запятыми (означающими операцию конъюнкции, как, например, в языке логического программирования Пролог), ***опуская при этом круглые скобки***.

Все сделанные выше построения и утверждения относительно упорядочения заданной КНФ завершаются следующим общим **принципом ранжирования дизъюнктов**. В таблицах двух произвольно данных дизъюнктов $\boldsymbol{d}$ и $\boldsymbol{g}$ попарно сравниваются позиционно одинаково расположенные функционально результирующие двоичные цифровые наборы. Если цифры совпадают, рассматривается следующая по порядку пара наборов. Если цифры – разные, тот дизъюнкт, в результирующем столбце которого фигурирует «непарный» нуль, объявляется «младшим».

Формализуем этот принцип. Пусть произвольно задана пара дизъюнктов $\boldsymbol{d}=n.p$ и $\boldsymbol{g}=m.q$, где n, m и p, q – упорядоченные строки номеров отрицательных и положительных переменных в дизъюнктах $\boldsymbol{d}$ и $\boldsymbol{g}$, соответственно; n_k, m_k и p_k, q_k – k-тые по порядку слева направо (начиная с k=1) номера переменных в строках n, m и p, q.

Утверждение 1. Случай $n = m$. Отношение $\boldsymbol{d}<\boldsymbol{g}$ выполнятся, если и только если выполняется какое-либо одно из условий:

CC BY

(1): $p_k > q_k$; (2): цифра p_k отсутствует. (4)

Если сравниваемые номера переменных совпадают ($p_k = q_k$), то сравнение двух заданных дизъюнктов продолжается со следующей пары номеров их переменных (т. е. с использованием операции k:=k+1 пошагового перебора переменных) в строках p и q.

Доказательство первого условия в (4).

Буквы p_k и q_k обозначают некоторые переменные x_w и x_v с номерами w и v, соответственно. Согласно первому заданному условию (4) $p_k=x_w>q_k=x_v$, т.е. $x_w>x_v$. Отсюда следует, что w>v.

В случае k=1 имеем $p=x_w=z^{w-1}u^{w-1}$, $q=x_v=z^{v-1}u^{v-1}$ – т. е. двоичная строка p в начале содержит большее число нулей (z) подряд, чем строка q. Согласно принципу ранжирования дизъюнктов получаем $\boldsymbol{d} < \boldsymbol{g}$.

Случай k>1. Пусть a – имеющая некоторую размерность s подстрока p, предшествующая переменной p_k, а b – подстрока q, предшествующая переменной q_k. Очевидно, что $b=a$ (т. к. условию $p_k>q_k$ предшествовало равенство $p_{k-1}=q_{k-1}$). Периодически продолжим подстроку $a^{[s]}$ до размерности w переменной $p_k=x_w$ путём её повторения 2^{w-s} раз, после чего к результату применим операцию дизъюнкции с таблично представленной переменной $p_k=z^{w-1}u^{w-1}$:

$$(a^{[s]})^{w-s}\vee z^{w-1}u^{w-1}=((a^{[s]})^{w-s-1}\vee z^{w-1})((a^{[s]})^{w-s-1}\vee u^{w-1})=(a^{[s]})^{w-s-1}u^{w-1}.$$

Аналогичные операции проведём для строк $b^{[s]}=a^{[s]}$ и $q_k=x_v=z^{v-1}u^{v-1}$:

$$(a^{[s]})^{v-s}\vee z^{v-1}u^{v-1}=((a^{[s]})^{v-s-1}\vee z^{v-1})((a^{[s]})^{v-s-1}\vee u^{v-1})=(a^{[s]})^{v-s-1}u^{v-1}.$$

Т. о. мы имеем следующую ситуацию. В таблично представленной строке q в точке, где заканчивается повторяющаяся 2^{v-s-1} раз подстрока $a^{[s]}$, начинается единичная строка u^{v-1}. В таблично представленной строке p в той же самой позиционной точке продолжается повторение подстроки $a^{[s]}$ 2^{w-v} раз (w>v):

$$(a^{[s]})^{w-s-1}u^{w-1}=(a^{[s]})^{\mathbf{v-s-1+w-v}}u^{w-1}=(a^{[s]})^{v-s-1}(a^{[s]})^{\mathbf{w-v}}u^{w-1}.$$

Остаётся заметить, что подстрока a начинается с символа z (как дизъюнкция положительных переменных). Т. о. в одинаковых позиционных точках строка p продолжается символом z, а строка q – символом u. Из неравенства z<u следует неравенство $d < g$. Ниже заключительная фаза доказательства представлена графически.

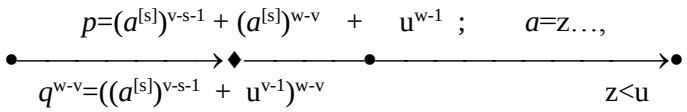


Рис. 1. Различие двоичных значений строк p и q, начиная с табличной позиции ♦.
Знак + означает операцию конкатенации строк

Теперь продолжим **доказательство** для второго заданного в (4) условия – p_k отсутствует. Строку $p=a^{[s]}$ периодически продолжим до размерности пространства переменной $q_k=x_v$:

$$a^{[v]}=(a^{[s]})^{v-s}=(a^{[s]})^{v-s-1}(a^{[s]})^{v-s-1}.$$

Тогда окончание таблицы для строки q представится следующей формулой:

$$q=(a^{[s]})^{v-s}\vee z^{v-1}u^{v-1}=((a^{[s]})^{v-s-1}\vee z^{v-1})((a^{[s]})^{v-s-1}\vee u^{v-1})=(a^{[s]})^{v-s-1}u^{v-1}.$$

CC BY

Сравним строки $a^{[v]}$ и q. В точке половинного деления строки $a^{[v]}$ сочленяются две одинаковые подстроки $(a^{[s]})^{v-s-1}$. Это означает, что правая половина строки $a^{[v]}$ начинается как минимум одной цифрой нуль (напомним, a=z…). Левая половина строки q совпадает с левой половиной строки $a^{[v]}$, а правая – единичная строка u^{v-1}. Сравнивая правые половины строк $a^{[v]}$ и q, имеем z…<u… Отсюда следует, что $d < g$.

Случай отсутствия переменной q_k, неотмеченный в условиях (4), может быть представлен в виде соотношения «$q_k > p_k$, либо цифра q_k не существует», которое, согласно только что доказанному выше, определяет истинность отношения $\boldsymbol{g} < \boldsymbol{d}$ (т.е. $\boldsymbol{d} < \boldsymbol{g}$ – ложно).

Иных реализаций соотношения $p \neq q$ не существует.

Утверждение 2. Случай $\boldsymbol{n} \neq \boldsymbol{m}$. Определим r=max(***n***) и t=max(***m***) – максимальные номера переменных в строках ***n*** и ***m***, соответственно. Строка ***n*** начинается с переменной $\bar{\boldsymbol{x}}_r=u^{r-1}z^{r-1}$, а строка ***m*** – с переменной $\bar{\boldsymbol{x}}_t=u^{t-1}z^{t-1}$. Если r<t, то таблица строки ***n*** в своём начале будет содержать меньшее число единиц, чем таблица строки ***m*** ($u^{r-1}<u^{t-1}$) и, следовательно, $\boldsymbol{d} < \boldsymbol{g}$. Если r=t, то следующая пара максимальных номеров будет отыскиваться и сравниваться в остатках строк ***n*** и ***m***. В предельном случае равенства максимумов завершится исчерпанием одной из строк ($\boldsymbol{n}\neq\boldsymbol{m}$), которая и определит младший дизъюнкт. Если в самом начале какой-либо дизъюнкт окажется положительным, то он, вследствие *N.P* – разделения переменных, является младшим.

Пример: согласно правилу (4) имеем 0.35<0.357. Воспроизведём этот же результат прямым $\boldsymbol{zu}\uparrow$ – преобразованием заданных дизъюнктов, используя уже полученный ранее результат для $(\boldsymbol{x}_3\vee\boldsymbol{x}_5)^{[7]}$:

$$0.35^{[7]}=(\boldsymbol{x}_3\vee\boldsymbol{x}_5)^{[7]}=(\mathbf{(z^2u^2)^1u^4})^2=(\mathbf{(z^2u^2)^1u^4})^1(\mathbf{(z^2u^2)^1u^4})^1;$$

$$0.357=((z^2u^2)^1u^4)^2\vee z^6u^6=((z^2u^2)^1u^4)^1(z^2u^2)^1u^4)^1\vee z^6u^6=(\mathbf{(z^2u^2)^1u^4})^1u^6.$$

Таблица $0.35^{[7]}$ содержит $4=2^2$ раза повторяющуюся (выделенную) строку. После её второго экземпляра следует третий, начинающийся четырёхкратным повторением нуля ($\mathbf{z^2}$). Таблица 0.357 содержит ту же самую дважды ($2=2^1$) повторяющуюся (выделенную) строку. За ней следует единица, повторяющаяся 2^6 раз ($\mathbf{u^6}$). Т.о. после двукратного повторения выделенной строки в обоих дизъюнктах в первом дизъюнкте следуют 4 нуля ($\mathbf{z^2}$), а во втором – 2^6 единиц ($\mathbf{u^6}$) В силу того, что $\mathbf{z^2\ldots<u^6}$, получаем 0.35<0.357.

Ещё один пример правила (4): 0.358<0.357. Действительно,

$$0.358=((z^2u^2)^1u^4)^3\vee z^7u^7=((z^2u^2)^1u^4)^2(z^2u^2)^1u^4)^2\vee z^7u^7=(\mathbf{(z^2u^2)^1u^4})^2u^7=(\mathbf{(z^2u^2)^1u^4})^1(\mathbf{(z^2u^2)^1u^4})^1u^7.$$

В таблице 0.357 после начального сегмента $(\mathbf{(z^2u^2)^1u^4})^1$ следуют единицы (u^6), а в таблице 0.358 после такого же самого начального сегмента $(\mathbf{(z^2u^2)^1u^4})^1$ следует ещё один такой же сегмент, начинающийся с нулей (z^2). В силу того, что $z^2\ldots<u^6$, получаем 0.358<0.357.

Ещё один пример: 41.358<41.357, также согласно (4).

Если теперь встроить правила **утверждений 1** и **2** ранжирования дизъюнктов в любой известный метод сортировки последовательностей (скажем, «пузырька», «быстрой сортировки»), то полученный метод будет сортировать произвольно данную КНФ в $\uparrow\boldsymbol{n.p}$ – представлении в порядке возрастания её дизъюнктов. Итоговая вычислительная сложность

CC BY

преобразования КНФ в строго упорядоченную сводится к тому, что сначала производятся очевидные перестановки переменных в пределах отдельных дизъюнктов, а затем – перестановки самих дизъюнктов.

Таким образом завершается построение обобщённого ***zu*↑↑*n.p*** – представления булевых функций – их таблично-аналитическое исчисление.

А3. Методы исследования КНФ, актуализируемые таблично-аналитическим исчислением ФАЛ

В данной работе простейшие методы исследования КНФ рассматриваются в иллюстрационных целях в контексте известной проблемы выполнимости КНФ.

Строгая упорядоченность КНФ позволяет свести задачу её выполнимости к задаче её оптимизации (в частности, максимизации). На основе заданной КНФ порождаются логически эквивалентные ей новые строго упорядоченные формулировки КНФ. Такими преобразованиями могут быть, например, инвертирование булевых переменных и/или их переименования, которые не изменяют фундаментальных логических свойство любой КНФ (в частности – её выполнимость). Эти формулировки как двоичные числа упорядочиваются в возрастающем порядке. Наибольшая числовая формулировка, если она численно превосходит исходную КНФ, выбирается в качестве объекта дальнейшего исследования. В противном случае следует искать другие логически эквивалентные данной КНФ её преобразования. Подобная последовательная операция максимизации исходной КНФ повторяется до тех пор, пока текущая оптимизируемая версия КНФ не окажется содержащей только негативные дизъюнкты (отсутствуют дизъюнкты, начинающихся с нуля (z) перед разделительной точкой). Результирующая версия КНФ является выполнимой – её выполняет нулевой набор всех переменных. Действительно, каждый её дизъюнкт в двоичном представлении имеет в своём составе хотя бы одну единицу – негативную переменную, которая, после отрицания её нулевого значения, превращается в единицу и обращает данный дизъюнкт в единицу. Таким образом все дизъюнкты негативной КНФ становятся единичными и единичной (истинной) становится и сама КНФ $\tilde{f}(\mathbf{0})=1$. Важнейшим сопутствующим результатом подобной максимизации исходной КНФ является список инвертированных переменных, преобразовавших её к негативной КНФ $\tilde{f}(\boldsymbol{x})$. Этот список и является тем набором, который выполняет исходную КНФ, т.е. корень уравнения $f(\boldsymbol{x})=1$.

Ниже рассматриваются актуализируемые методы исследования КНФ.

М1. Покоординатное инвертирование КНФ

Описанная схема оптимизации исходной КНФ может быть реализована, например, путём последовательного применения операции её покоординатного инвертирования. В ***↑n.p*** – представлении данной КНФ инвертирование одной из её переменных (координаты) сводится к преобразованию всех дизъюнктов, содержащих заданную переменную, следующим образом. Номер заданной переменной, расположенный слева (или справа) от разделительной точки в дизъюнкте, переносится в противоположный сектор – вправо (или, соответственно, влево) от точки. При этом перенесённый номер упорядоченно встраивается среди номеров переменных данного сектора согласно предопределённого ему локальному порядку. В секторе

CC BY

положительных переменных – в обычном привычном числовом возрастающем порядке, в секторе отрицательных переменных – в убывающем порядке (в направлении к точке).

Простейший пример процедуры максимизации КНФ представляет собой инвертирование переменной x_1 в функции $\boldsymbol{f}(x_1, x_2, x_3)$ = 0.123, 2.13, 21.0, 3.1 из таблицы 1:

$$\boldsymbol{f}(\bar{x}_1, x_2, x_3) = \mathbf{1.23}, 2.1, 21.3, 31.0.$$

Инвертирование в полученной функции переменной x3 приводит ко второму корню:

$$\boldsymbol{f}(\bar{x}_1, x_2, \bar{x}_3) = 31.2, 2.1, 321.0, 1.3 = \mathbf{1.3}, 2.1, 31.2, 321.0.$$

Ниже рассматривается максимизация (уже упорядоченной) КНФ $\boldsymbol{f_0}(x_1, x_2, \ldots, x_9)$, определённой в базисе 9 булевых переменных и содержащей 15 дизъюнктов:

$\boldsymbol{f_0}$=0.123, 1.67, 2.57, 3.45, 4.27, 5.39, 53.0, 6.28, 7.38, 73.5, 8.35, 841.0, 9.17, 96.4, 982.0.

Для порождения логически эквивалентных данной КНФ функций применим к ней поочерёдное инвертирование каждой из её переменной по отдельности. Полученные результаты представлены в нижеследующей обобщённой таблице 2. В ней в виде столбцов содержатся как исходная функция $\boldsymbol{f_0}$, так и все девять упорядоченных функций, полученных из $\boldsymbol{f_0}$ раздельным инвертированием её переменных; при этом всё множество функций-столбцов, в свою очередь, упорядочено по возрастанию своих двоичных значений. В титульной строке таблицы указаны номера инвертированных переменных в $\boldsymbol{f_0}$. Функции-столбцы упорядочены от минимальных ($\bar{7} < \bar{5} < \ldots$) до максимальной ($\ldots < \boldsymbol{f_0}$).

Представлены только первые пять строк из каждого функционального столбца.

Таблица 2

Покоординатное инвертирование КНФ f_0.

$\bar{7}$	$\bar{5}$	$\bar{8}$	$\bar{3}$	$\bar{6}$	$\bar{2}$	$\bar{4}$	$\bar{9}$	$\bar{1}$	f_0
0.378	0.359	0.358	0.345	0.268	0.257	0.247	0.179	0.167	0.123
0.123	0.123	0.123	1.67	0.123	1.67	0.123	0.123	1.23	1.67
3.57	1.67	1.67	2.57	2.57	2.13	1.67	1.67	2.57	2.57
3.45	3.5	2.57	3.12	3.45	3.45	2.57	2.57	3.45	3.45
5.39	4.27	3.45	4.27	4.27	42.7	43.5	3.45	4.27	4.27
					. . .				

Как видно, любая из преобразованных функций не даёт предпочтения в направлении дальнейшей максимизации исходной функции $\boldsymbol{f_0}$. Поэтому к $\boldsymbol{f_0}$ следует применить какие-нибудь другие преобразования, отличные от покоординатного инвертирования.

М2. Комбинационное координатное инвертирование КНФ

В качестве нового преобразования реализуем простейший случай комбинационного координатного инвертирования КНФ – инвертирование всех её переменных разом. Т. е. построим строго упорядоченную функцию $\boldsymbol{f_0}(\bar{\boldsymbol{x}})$. Вот она:

$\boldsymbol{f_0}(\bar{\boldsymbol{x}})$ = 0.35, 0.289, 0.148, 321.0, 4.69, 5.37, 53.8, 54.3, 71.9, 72.4 , 75.2, 76.1, 82.6, 83.7, 93.5

Кстати, заметим, полностью инвертированная функция может быть получена зеркальным отражением положительных и отрицательных переменных относительно разделяющей их точки в каждом из дизъюнктов исходной функции, а затем последующего упорядочения этих дизъюнктов.) Далее применим к полученной функции fo($\bar{x}$) процедуру покоординатного инвертирования, уже апробированную ранее в преобразованиях исходной

функции fo (табл. 2). Результаты – первые пять строк из каждого функционального столбца – представлены в нижеследующей таблице 3.

Таблица 3

Покоординатное инвертирование КНФ $f_0(\bar{x})$

$\bar{4}$	$\bar{7}$	$f_0(\bar{x})$	$\bar{6}$	$\bar{1}$	$\bar{2}$	$\bar{9}$	$\bar{8}$	$\bar{5}$	$\bar{3}$
0.469	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.357	0.289
0.35	0.289	0.289	0.289	0.289	0.148	0.148	2.68	0.289	0.148
0.289	0.148	0.148	0.148	1.48	2.89	3.59	3.78	0.148	21.3
321.0	1.79	321.0	321.0	32.1	31.2	321.0	321.0	3.58	3.5
4.18	2.47	4.69	5.37	4.69	4.69	5.37	4.69	321.0	4.69

. . .

Здесь по своим двоичным (табличным) значениям исходную функцию $f_0(\bar{x})$ превышают семь функций-столбцов: $f_0(\bar{x}) < \bar{6} < \bar{1} < \bar{2} < \bar{9} < \bar{8} < \bar{5} < \bar{3}$. Любая из этих семи функций может быть принята в качестве следующего шага дальнейшей максимизации. Заимствуя стратегию "жадных" алгоритмов, выберем следующим шагом максимизации функцию с инвертированной переменной 3. Обозначим эту функцию ***i3***:

i3= 0.289, 0.148, 21.3, 3.5, 4.69, 5.38, 53.7, 543, 71.9, 72.4, 75.2, 76.1, 8.37, 82.6, 9.35

Применяя к этой и последующим результирующим функциям покоординатное инвертирование с выбором очередной функции, руководствуясь "жадным" алгоритмом, получаем следующую последовательность возрастающих оптимальных функций:

i32= 0.148, 1.23, 2.89, 3.5, 4.69, 5.38, 53.7, 543, 7.24, 71.9, 752, 76.1, 8.37, 8.26, 9.35

i326= 0.148, 1.23, 2.89, 3.5, 5.38, 53.7, 543, 64.9,7.24, 7.16, 71.9, 752, 8.37, 86.2, 9.35

i3264=**1.23**, 2.89, 3.5, 4.18, 5.38, 53.7, 53.4, 6.49, 7.16, 71.9, 74.2, 752, 8.37, 86.2, 9.35

Последняя функция оказывается полностью негативной и, таким образом, специфицирует набор предшествующих инвертирующих переменных (3,2,6,4) как набор, выполняющий исходную КНФ $f_0(\bar{x})$. Если теперь в функции $f_0(\bar{x})$ и в порождённом ею наборе переменных (3,2,6,4) одновременно произвести инверсию всех 9-ти переменных, то функция $f_0(\bar{x})$ вновь преобразуется в $f_0(x)$, а набор (3,2,6,4) после повторного инвертирования преобразуется в набор (1,5,7,8,9) остальных переменных. Этот последний набор и выполняет первоначальную функцию $f_0(x)$.

М3. Максимизация с отступом

Функция $f_0(x)$ имеет и ещё один выполняющий её набор (1,2,4,5,6,9), не выявляемый стратегией «жадных» алгоритмов. Усложнённая стратегия допускает ближайшие отступления (снижения) от достигнутого максимума целевой функции, а затем вновь её дальнейшую максимизацию. Стратегия «**максимизация с отступом**» («чтобы подняться на новую вершину надо сначала спуститься с той, на которой находишься») может быть реализована уже на первом шаге покоординатного инвертирования функции $f_0(x)$ (табл. 2). Ниже приведён протокол этой реализации.

$f_0(x)$ = 0.123, 1.67, 2.57, 3.45, 4.27, 5.39, 53, 6.28, 7.38, 73.5, 8.35, 841, 9.17, 96.4, 982

↓ – ***1*** = 0.167, 1.23, 2.57, 3.45, 4.27, 5.39, 53, 6.28, 7.38, 73.5, 8.35, 84.1, 91.7, 96.4, 982

↓ – ***9*** = 0.167, 1.79, 1.23, 2.57, 3.45, 4.27, 53, 6.49, 6.28, 7.38, 73.5, 8.35, 82.9, 84.1, 95.3

CC BY

↑ + ***5*** = 0.167, 1.79, 1.23, 3.5, 4.27, 52.7, 53.4, 6.49, 6.28, 7.38, 753, 82.9, 84.1, 85.3, 9.35

↑ + ***2*** = 0.167, 1.79, 21.3, 3.5, 42.7, 5.27, 53.4, 6.49, 62.8, 7.38, 753, 8.29, 84.1, 85.3, 9.35

↓ – ***4*** = 0.167, 1.79, 2.47, 21.3, 3.5, 5.27, 543, 62.8, 64.9, 7.38, 753, 8.29, 8.14, 85.3, 9.35

↑ + ***6*** = **1.79**, 2.68, 2.47, 21.3, 3.5, 4.69, 5.27, 543, 6.17, 7.38, 753, 8.29, 8.14, 85.3, 9.35

М4. Организация монотонной максимизации КНФ с предварительным переименованием переменных

Рассмотрим теперь пример искусственно организованной монотонной максимизации КНФ. С этой целью достаточно в первом дизъюнкте 0.123 исследуемой функции $f_0(\boldsymbol{x})$ переименовать переменные таким образом, чтобы путём последующего инвертирования переменных КНФ можно было бы уменьшить (как двоичное число). Например, может быть использовано такое переименование переменных:

$$\mathbf{p}_1 = 1\leftrightarrow 9,\ 2\leftrightarrow 8,\ 3\leftrightarrow 7.$$

Тогда результирующая функция приобретает (в операторной записи) вид:

$f_0(\mathbf{p}_1\cdot\boldsymbol{x})$ = 0.789, 1.39, 2.57, 3.27, 4.38, 5.17, 6.28, 61.4, 7.45, 73.5, 75, 8.35, 821, 9.36, 942.0

Далее, следуя стратегии "жадных" алгоритмов, получаем следующий функциональный протокол поочерёдного инвертирования переменных

$f_0(\bar{\boldsymbol{8}})$=0.358, 1.39, 2.57, 21.8, 3.27, 5.17, 61.4, 7.45, 73.5, 75, 8.79, 84.3, 86.2, 9.36, 942.0

$f_0(\bar{\boldsymbol{5}})$=0.157, 1.39, 21.8, 3.27, 5.38, 52.7, 61.4, 7.5, 75.4, 753, 8.79, 84.3, 86.2, 9.36, 942.0

$f_0(\bar{\boldsymbol{1}})$=0.139, 1.57, 2.18, 3.27, 5.38, 52.7, 6.14, 7.5, 75.4, 753, 8.79, 84.3, 86.2, 9.36, 942.0

$f_0(\bar{\boldsymbol{4}})$=0.139, 1.57, 2.18, 3.27, 5.38, 52.7, 64.1, 7.5, 753, 754, 8.79, 8.34, 86.2, 9.36, 92.4

На этом шаге дальнейшая монотонная максимизация становится невозможной. Вследствие этого вновь прибегнем к следующему переименованию переменных:

$$\mathbf{p}_2 = 1\leftrightarrow 5,\ 3\leftrightarrow 6.$$

Полученная при этом переименовании функция приобретает вид

$f_0(\mathbf{p}_2\cdot\mathbf{p}_1\cdot\boldsymbol{x})$ = 0.569, 1.68, 2.58, 21.7, 43.5, 5.17, 6.27, 7.1, 741, 761, 8.79, 8.46, 83.2, 9.36, 92.4

Опять следуя стратегии «жадных» алгоритмов, получаем завершающий функциональный протокол инвертирования переменных:

$f_0(\bar{\boldsymbol{9}})$=0.369, 1.68, 2.58, 2.49, 21.7, 43.5, 5.17, 6.27, 7.1, 741, 761, 8.46, 83.2, 9.56, 98.7

$f_0(\bar{\boldsymbol{3}})$=**1.68**, 2.58, 2.49, 21.7, 3.69, 4.35, 5.17, 6.27, 7.1, 741, 761, 8.46, 8.23, 9.56, 98.7

Для того, чтобы реконструировать выполняющий набор переменных, соответствующий первоначально заданной функции fo(x), достаточно произвести обратное переименование переменных в полученной итоговой функции. С этой целью интерпретируем переименования как операцию перестановок чисел 1..9. Как известно, любая составная перестановка может быть преобразована в множество циклических перестановок, не имеющих общих элементов. В нашем случае составная перестановка $\mathbf{p}_3 = \mathbf{p}_2\cdot\mathbf{p}_1$ разлагается на следующие циклы:

$$\mathbf{p}_3 = [1\leftrightarrow 5,\ 3\leftrightarrow 6]\cdot[1\leftrightarrow 9,\ 2\leftrightarrow 8,\ 3\leftrightarrow 7] = [9\rightarrow 5\rightarrow 1]\cdot[2\leftrightarrow 8]\cdot[7\rightarrow 6\rightarrow 3].$$

CC BY

Применяя к полученному множеству номеров инвертированных переменных обратную перестановку $\mathbf{p_3}^{-1}$, получаем (в перестановке, обратной циклической, направление стрелок меняется на противоположное):

$\mathbf{p_3}^{-1} \cdot (8,5,1,4,9,3) = [1 \to 5 \to 9] \cdot [2 \leftrightarrow 8] \cdot [3 \to 6 \to 7] \cdot (8,5,1,4,9,3) = (2,9,5,4,1,6) = (1,2,4,5,6,9)$

– тот же выполняющий набор, который был получен методом «**максимизации с отступом**».

М5. Поиск выполняющего КНФ набора переменных в динамическом троичном дереве дизъюнктов

Рассмотрим теперь подробнее математические основания эвристической стратегии «**максимизация с отступом**». Это поиск выполняющего данную КНФ набора переменных в динамическом троичном дереве, узлами которого являются дизъюнкты КНФ ФАЛ-3. В простейших случаях такой поиск может свестись к цепному изменению знаков переменных в соответствующих узлах дерева. Именно, изменение знака какой-либо переменной в положительном дизъюнкте обращает его в отрицательный и, одновременно, может обратить какой-то отрицательный дизъюнкт в положительный. Проведение в последнем некоторой другой знакоперемены может вновь породить пару отрицательного и положительного дизъюнктов. Так выстраиваемая цепочка может завершиться тем, что в результирующей КНФ все дизъюнкты оказываются отрицательными.

В качестве примера вновь рассмотрим КНФ

$\boldsymbol{f_0(x)}$ = 0.123, 1.67, 2.57, 3.45, 4.27, 5.39, 53.0, 6.28, 7.38, 73.5, 8.35, 841, 9.17, 96.4, 982.0

Знакоперемена $\overline{\boldsymbol{1}}$ в дизъюнкте 0.123 порождает два дизъюнкта 1.23 и 0.167. Знакоперемена $\overline{\mathbf{6}}$ в 0.167 в свою очередь порождает дизъюнкты 1.67 и 0.268. Знакоперемена $\overline{\mathbf{2}}$ в 0.268 порождает дизъюнкты 6.28 и 0.257. Знакоперемена $\overline{\mathbf{5}}$ в 0.257 порождает дизъюнкты 5.27 и 0.359. Если, далее, в дизъюнкте 0.359 произвести знакоперемену $\overline{\mathbf{3}}$, то первоначальный отрицательный дизъюнкт 53.0 в результате предыдущих знакоперемен преобразуется в положительный 0.53 без дальнейшей возможности его преобразования в отрицательный. Поэтому для построения дальнейшей цепочки взаимосвязанных дизъюнктов и переменных следует в 0.359 произвести знакоперемену $\overline{\mathbf{9}}$, которая порождает дизъюнкты 9.35 и 0.469. Знакоперемена $\overline{\mathbf{4}}$ в последнем порождает отрицательный дизъюнкт 4.69 без возникновения какого-либо сопутствующего положительного дизъюнкта. Таким образом цепочка знакоперемен $\overline{\boldsymbol{1}} \to \overline{\mathbf{6}} \to \overline{\mathbf{2}} \to \overline{\mathbf{5}} \to \overline{\mathbf{9}} \to \overline{\mathbf{4}}$ приводит все дизъюнкты $\boldsymbol{f_0(x)}$ к отрицательным:

$$0.123 \to \overline{\boldsymbol{1}} \to (\mathbf{1.23}, 0.167) \to \overline{\mathbf{6}} \to (\mathbf{1.67}, 0.268) \to \overline{\mathbf{2}} \to (\mathbf{6.28}, 0.257) \to \overline{\mathbf{5}} \to (\mathbf{5.27}, 0.359) \to \overline{\mathbf{9}} \to (\mathbf{9.35}, 0.469) \to \overline{\mathbf{4}} \to \mathbf{4.69}.$$

(Следует заметить, что все произведённые знакоперемены действуют и на все предшествующие уже преобразованные дизъюнкты. Поэтому дизъюнкт **1.23** преобразуется в **21.3**.)

В рамках рассмотренного примера попробуем теперь этим же методом найти какую-нибудь другую знакопеременную цепочку, преобразующую КНФ $\boldsymbol{f_0(x)}$ в дизъюнктивно-отрицательную (т.е., по сути, другой корень уравнения $\boldsymbol{f_0(x)}$=1). Для этого, оказывается, достаточно на втором шаге построения цепочки в дизъюнкте 0.167 изменить знак переменной

CC BY

7 (а не 6). В конечном результате получается цепочка знакоперемен $\bar{1}\rightarrow \bar{7}\rightarrow \bar{8}\rightarrow \bar{5}\rightarrow \bar{9}$, приводящая все дизъюнкты $f_0(x)$ к отрицательным:

$\bar{1}$: 0.16**7**, 1.23, 2.57, 3.45, 4.27, 5.39, 53.0, 6.28, **7**.38, 73.5, 8.35, 84.1, 91.7, 96.4, 982

$\bar{7}$: 0.37**8**, 1.23, 3.57, 3.45, 5.39, 53, 6.28, 7.16, 72.5, 74.2, **8**.35, 84.1, 96.4, 971, 982

$\bar{8}$: 0.35**8**, 1.23, 3.57, 3.45, 4.18, **5**.39, 53.0, 7.16, 72.5, 74.2, 8.37, 86.2, 92.8, 96.4, 971

$\bar{5}$: 0.35**9**, 1.23, 3.5, 4.18, 5.38, 53.7, 53.4, 7.16, 74.2, 752, 8.37, 86.2, **9**2.8, 96.4, 971

$\bar{9}$: **1.23**, 2.89, 3.5, 4.18, 5.38, 53.7, 53.4, 6.49, 7.16, 71.9, 74.2, 752, 8.37, 86.2, 9.35

Следует отметить очевидное: добавление к функции $f_0(x)$ ещё одного дизъюнкта 951.0 обращает её в противоречивую: $f_0(x) \wedge 951.0 \equiv 0$. Отсюда следует, что выявление противоречивой КНФ требует полного перебора и проверки всех возможных направлений поиска в ней, что и ведёт к комбинаторному взрыву объёма вычислений.

На этом завершим рассмотрение методов исследования КНФ $f_0(x)$. Все они являются, по сути, эвристическими, поскольку на каждом шаге развития вычислительного процесса имеют точки ветвления и выбора других направлений вычислений, требующих своего обоснования. Не затронутая здесь процедура построения самой таблицы КНФ непосредственно на основе $zu\uparrow\uparrow n.p$ – представления с первых же шагов наталкивается на все комбинаторные проблемы задачи выполнимости,

Действие разработанных методов носит иллюстративный характер. Однако ничто не препятствует разработке новых методов, в частности – на основе комбинации уже рассмотренных.

Как известно, любая ФАЛ, не являющаяся тождественно истинной, может быть преобразована в КНФ. Поэтому все полученные в данной работе результаты являются универсальными.

Ограничение в заданной КНФ $f_0(x)$ не более, чем трёхлитерными дизъюнктами не является существенным. Как известно [2], КНФ с дизъюнктами любой арности $r>3$ может быть полиномиально преобразована в КНФ ФАЛ-3 с дизъюнктами арности r не более 3. (при этом КНФ ФАЛ-3 входит в класс NP задач, а КНФ ФАЛ-2 остаётся в классе P – по сложности полиномиально разрешимых задач).

Ограничение в заданной КНФ $f_0(x)$ девятью переменными также не является существенным. Оно позволяет избежать технических усложнений, связанных с изображением превышающих значения 9 номеров переменных и операций с ними, и при этом позволяет оставаться в привычном множестве десятичных номеров 0÷9. Иначе требуется списковое представление переменных, которое не проясняет суть рассматриваемых вопросов.

Современное состояние теории ФАЛ обусловлено традиционным определением дизъюнкта как «множества положительных и/или отрицательных булевых переменных», и соответственным определением КНФ, которое укоренилось в учебной литературе, монографиях [2] и математических энциклопедиях [3]. В данной работе даётся математически обоснованное определение дизъюнкта как «упорядоченного множества положительных и/или …». Источником этого определения является обычно рассматриваемое упорядоченное

CC BY

множество элементарных *n*-ок (x_1, x_1, ... x_n) области определения ФАЛ. Основываясь на этой упорядоченности, с одной стороны, и игнорирование её при определении дизъюнктов, содержащих эти *n*-ки (и привлекать затем различные математические конструкции, чтобы получить упорядоченность дизъюнктов) – с другой стороны, представляется, по меньшей мере, непоследовательным.

В формализме развитого таблично-аналитического исчисления простая запись любого дизъюнкта выявляет точное местоположение всех его нулей и единиц по всему 2^n – табличному диапазону. Это приводит к строгому упорядочению всего множества дизъюнктов и, автоматически, – к упорядочению самой КНФ, а, затем, – также и к упорядочению множества эквивалентно преобразуемых друг в друга КНФ. Т. е. упорядочение КНФ становится элементарной операций. Эта простота и актуализирует рассмотренные методы исследования КНФ. Они достаточно очевидны и, по сути, являются эвристическими. Рассмотренная в работе КНФ с 9 переменными и 15–16 дизъюнктами простейшим образом выявляет всю проблематику задачи выполнения КНФ.

Дальнейшего изучения заслуживают как уже рассмотренные, так и другие методы в их комбинационном взаимодействии.

Литература

1. Boole G. An Investigaition of the lows of thought. London, 1854.

2. Ахо А., Хопкрофт Дж., Ульман Дж.. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 536 с.

3. Виноградов И.М. Математическая энциклопедия. Т. 2. М.: Советская Энциклопедия, 1979. 1104 с.

© Кузнецов В.Н., 2022

УДК 532
https://doi.org/10.36906/AP-2022/23

Маликов З.М., Наврузов Д.П.
Институт механики и сейсмостойкости сооружений им. М.Т. Уразбаева АН РУз
г. Ташкент, Узбекистан

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ОСЕСИММЕТРИЧНОЙ СТРУИ НА ОСНОВЕ НОВЫХ МОДЕЛЕЙ ТУРБУЛЕНТНОСТИ

Аннотация. В работе проводится верификация новой двухжидкостной модели турбулентности на примере осесимметричной несжимаемой струи. Данная модель ранее была апробирована для этой задачи, где была использована система уравнений с постоянным давлением. В настоящей работе рассмотрена полная система уравнений турбулентности новой модели. Для сравнения в работе представлены также численные результаты хорошо известной модели Спаларта – Аллмареса SA. Численные реализации уравнений гидродинамики проведены на основе использования метода контрольного объёма FVM (Finite Volumes Method). Для аппроксимации конвективных членов использована схема Леонардо QUICK (Quadratic Upstream Interpolation for Convective Kinematics), а для вязкостных членов центральная конечно-разностная схема. Коррекция к скоростям проводилась по методике SIMPLEC. Полученные численные результаты сопоставлены с опытными данными из базы данных NASA. Показано, что новая двухжидкостная схема более точно описывает турбулентную струю, чем модель SA.

Ключевые слова: двухжидкостная модель турбулентности; модель SA; конечно-разностная схема против потока; схема QUICK; метод контрольного объёма; SIMPLEC.

Malikov Z.M., Navruzov D.P.
Institute of Mechanics and Seismic Stability of Structures
named after M.T. Urazbayev, AS RUz
Tashkent, Uzbekistan

NUMERICAL STUDY OF AN AXISYMMETRIC JET BASED ON A NEW TWO FLUID TURBULENCE MODEL

Abstract. A new two-fluid model of turbulence is verified by the example of an axisymmetric incompressible jet. This model was previously tested for the problem, where a system of equations with constant pressure was used. In the article, a complete system of turbulence equations of the new model is considered. For comparison, the article also presents the numerical results of the well-known Spalart – Allmares (SA) model. Numerical realizations of the equations of hydrodynamics were performed using the Finite Volumes Method (FVM). The Leonardo QUICK (Quadratic Upstream Interpolation for Convective Kinematics) scheme was used to approximate the convective terms, and

CC BY

the central finite-difference scheme was used for the viscosity terms. Velocity correction was conducted according to the SIMPLEC method. The numerical results obtained are compared with experimental data from the NASA database. It is shown that the new two-fluid scheme more accurately describes the turbulent jet than the SA model.

Keywords: two-fluid turbulence model; SA model; upstream finite-difference scheme; QUICK scheme; control volume method; SIMPLEC.

Во многих случаях при движении жидкости и газа возникают так называемые поверхности тангенциального разрыва. Течение жидкости в окрестности такой поверхности называется струёй. Тангенциальный разрыв могут терпеть такие параметры, как скорость течения, температура и концентрация примеси. При этом распределение статического давления оказывается непрерывным. Как известно, на поверхности тангенциального разрыва в связи с её неустойчивостью возникают вихри, беспорядочно движущиеся вдоль и поперёк потока, вследствие этого происходит молярный поперечный перенос количества движения, тепла и примесей. В результате на границе двух струй формируется область конечной толщины с непрерывным распределением скорости, температуры и концентрации примеси, которая называется струйным турбулентным пограничным слоем. Наиболее изученным видом турбулентной струи является струя, распространяющаяся в покоящейся среде, такая струя называется затопленной. При равномерном поле скорости в начальном сечении затопленной струи границы её пограничного слоя представляют собой расходящиеся поверхности, которые пересекаются у кромки сопла. С внешней стороны пограничный слой струи соприкасается с неподвижной жидкостью, причём под внешней границей понимают поверхность, во всех точках которой составляющая скорости по оси x равна нулю $(U=0)$. С внутренней стороны пограничный слой переходит в ядро постоянной скорости, поэтому на внутренней границе пограничного слоя скорость потока равна скорости истечения $U=U_0$ [1].

Рассматриваемая задача имеет большое значение для авиационной и ракетно-космической техники. Моделирование турбулентной затопленной струи вот уже на протяжении почти 100 лет приковывает внимание многих исследователей. Несмотря на множества этих исследований, рассматриваемую задачу нельзя считать полностью завершённой, т. к. она связана до сих пор не решённой проблемой турбулентности. Несмотря на то, что до сегодняшнего дня разработано более 100 различных моделей турбулентности, универсальную модель турбулентности все ещё не существует. Это означает, что эти модели, если удовлетворительно описывают одних классов турбулентности, других могут описывать неверно даже качественно. Однако недавно опубликована работа одного из соавторов настоящей статьи [2], где представлен новый подход к решению проблемы турбулентности. В указанной работе на основе нового подхода получена замкнутая математическая модель турбулентности, которая апробирована для различных задач. Достоинством новой модели являются её способность описывать анизотропную турбулентность, а также в работе показано, что она не страдает «аномалией круглой струи». Суть данного явления заключается в том, что все модели, направленные на замыкание уравнений Навье-Стокса осреднённые по Рейнольдсу

CC BY

(модели RANS) для осесимметричной струи дают в полтора раза большее расширение струи, чем наблюдается в экспериментах. Поэтому в этих моделях вводятся специальные поправки. Кроме того, сравнения численных результатов модели с известными опытными данными показали её хорошую точность. Новая математическая модель была апробирована для ряда задач турбулентности и результаты были сопоставлены с экспериментальными данными, которые представлены в базе данных NASA (https://clck.ru/dncvY). В работе [2] на основе новой двухжидкостной модели проведено исследование затопленной струи. При этом использована упрощённая, параболизованная система уравнений, т.е. предполагалось давление постоянным. Однако не во всех струйных течениях давление можно считать постоянным. Например, во многих технических устройствах струя может протекать в ограниченных пространствах. Поэтому целью настоящей работы является апробация новой модели для затопленной струи при использовании полной системы уравнений турбулентности. В работе для сопоставления приведены также результаты хорошо известной модели SA [3].

Модель турбулентности SA является однопараметрической и на сегодняшний день является одной из наиболее популярных моделей. Поэтому эта модель достаточно хорошо освящена во многих научных работах и учебниках. Следовательно, нет необходимости останавливаться на самой модели SA. Однако отметим, что методики численной реализации модели SA и двухжидкостной модели аналогичны. Что касается новой двухжидкостной модели, то она подробно изложена в работе [2]. Краткое содержание сути новой двухжидкостной модели также представлено в работе [4]. В этих работах показана возможность представления турбулентного потока как гетерогенную смесь двух жидкостей, совершающие относительное движение. Система уравнений несжимаемой турбулентной струи в новой двухжидкостной модели в цилиндрических координатах имеет следующий вид:

$$
\begin{cases}
\dfrac{\partial U}{\partial t}+U\dfrac{\partial U}{\partial x}+V\dfrac{\partial U}{\partial r}+\dfrac{\partial p}{\rho\partial x}=\nu\left(\dfrac{\partial^2 U}{\partial r^2}+\dfrac{\partial U}{r\partial r}+\dfrac{\partial^2 U}{\partial x^2}\right)-\dfrac{\partial r\vartheta u}{r\partial r}-\dfrac{\partial uu}{\partial x},\\
\dfrac{\partial V}{\partial t}+U\dfrac{\partial V}{\partial x}+V\dfrac{\partial V}{\partial r}+\dfrac{\partial p}{\rho\partial r}=\nu\left(\dfrac{\partial^2 V}{\partial r^2}+\dfrac{\partial V}{r\partial r}+\dfrac{\partial^2 V}{\partial x^2}\right)-\dfrac{\partial r\vartheta\vartheta}{r\partial r}-\dfrac{\partial u\vartheta}{\partial x},\\
\dfrac{\partial u}{\partial t}+U\dfrac{\partial u}{\partial x}+V\dfrac{\partial u}{\partial r}=-\dfrac{\partial U}{\partial x}u-\dfrac{\partial U}{\partial r}\vartheta+C_s(\dfrac{\partial U}{\partial r}-\dfrac{\partial V}{\partial x})\vartheta+\\
\qquad+\dfrac{\partial}{\partial x}\left(2\nu_{xx}\dfrac{\partial u}{\partial x}\right)+\dfrac{\partial}{r\partial r}\left(\nu_{xr}r\left(\dfrac{\partial u}{\partial r}+\dfrac{\partial\vartheta}{\partial x}\right)\right)-K_f u,\\
\dfrac{\partial\vartheta}{\partial t}+U\dfrac{\partial\vartheta}{\partial x}+V\dfrac{\partial\vartheta}{\partial r}=-\dfrac{\partial V}{\partial r}\vartheta-\dfrac{\partial V}{\partial x}u-C_s(\dfrac{\partial U}{\partial r}-\dfrac{\partial V}{\partial x})u+\\
\qquad+\dfrac{\partial}{\partial x}\left(\nu_{xr}\left(\dfrac{\partial\vartheta}{\partial x}+\dfrac{\partial u}{\partial r}\right)\right)+\dfrac{\partial}{r\partial r}\left(2\nu_{rr}r\dfrac{\partial\vartheta}{\partial r}\right)-2\nu_{rr}r\dfrac{\vartheta}{r^2}-K_f\vartheta,\\
\dfrac{\partial U}{\partial x}+\dfrac{\partial rV}{r\partial r}=0,
\end{cases}
\qquad (1)
$$

Здесь

$$\nu_{xx} = \nu_{rr} = 3\nu + 2\frac{S}{def(\vec{V})},\ \nu_{xr} = 3\nu + 2\left|\frac{u\vartheta}{def(\vec{V})}\right|, def(\vec{V}) = \sqrt{\left(\frac{\partial U}{\partial r} + \frac{\partial V}{\partial x}\right)^2 + 2\left(\frac{\partial U}{\partial x}\right)^2 + 2\left(\frac{\partial V}{\partial r}\right)^2},$$

$$S = \frac{u^2 J_x + \vartheta^2 J_r}{J_x + J_r}, \quad J_x = \left|\frac{\partial u}{\partial x}\right|, \quad J_r = \left|\frac{\partial \vartheta}{\partial r}\right|, \quad C_s = 0.2, \quad K_f = C_1\lambda_{\max} + C_2\frac{|d\cdot\vartheta|}{d^2}.$$

В приведённых уравнениях U,V – соответственно аксиальная и радиальная составляющие вектора осреднённой скорости потока, u, ϑ – соответственно аксиальная и радиальная относительные скорости, $\nu_{xx},\nu_{rr},\nu_{xr}$ – эффективные молярные вязкости, K_f - коэффициент трения, d – ближайшее расстояние до твёрдой стенки.

Для определения коэффициента трения составим определитель для характеристического уравнения

$$\begin{vmatrix} -\lambda - \frac{\partial U}{\partial x} & -\frac{\partial U}{\partial r} + C_s\left(\frac{\partial U}{\partial r} - \frac{\partial V}{\partial x}\right) \\ -\frac{\partial V}{\partial x} - C_s\left(\frac{\partial U}{\partial r} - \frac{\partial V}{\partial x}\right) & -\lambda - \frac{\partial V}{\partial r} \end{vmatrix} = 0. \qquad (2)$$

Из уравнения (2) находим наибольший вещественный корень

$$\lambda_{\max} = \sqrt{D} \qquad если \quad D > 0$$
$$\lambda_{\max} = 0 \qquad если \quad D < 0 \qquad (3)$$
$$D = \frac{\partial U}{\partial r}\frac{\partial V}{\partial x} - \frac{\partial U}{\partial x}\frac{\partial V}{\partial r} + C_s(1 - C_s)\left[\frac{\partial U}{\partial r} - \frac{\partial V}{\partial x}\right]^2$$

Для численной реализации система уравнений (1) приводилась к безразмерному виду соотнесением всех скоростей к средне-расходной скорости струи на выходе из сопла U_0, а линейные размеры к радиусу сопла R. Исследование проводилось для струи с числом Рейнольдса $\mathrm{Re} = \rho R U_0 / \mu = 5600$. Для получения стационарного решения систем уравнений (1) использовался метод установления. Суть данного метода в том, что решается нестационарная задача и при большом значении безразмерного времени достигается стационарное решение задачи. При интегрировании уравнений использовалась прямоугольная сетка 200X100, безразмерные шаги интегрирования были равны $\Delta t = 0.005$, $\Delta x = 0.2$, $\Delta r = 0.1$. В качестве начального условия рассмотрена идеальная струя без расширения. Стационарное решение достигалось при $t = 100$.

Для численной реализации систем уравнений гидродинамики использован метод контрольного объёма. Для конвективных членов использована схема Леонардо QUICK [5] в явном виде, а вязкостные члены аппроксимировались центральной разностью. Причём вязкостные члены в продольном направлении аппроксимировались в явном виде, а в поперечном направлении в неявном виде. Схема Леонардо QUICK имеет второго порядка точности. Следовательно, расчётная схема имела точность первого порядка по времени и второго по пространству. Для реализации полу-неявной схемы использовался метод прогонки. Связь между давлением и скоростями проводилась процедурой SIMPLEC [6], где была использована итерационный метод верхней релаксации [7].

CC BY

Многочисленные исследования показали, что характеристика струи сильно зависит от потока, который формируется в дозвуковом сопле. Известно, что в дозвуковом сопле происходит подавление турбулентности и на выходе из сопла можно получить достаточно равномерный поток. Поэтому в работе в качестве начальных условий приняты следующие безразмерные параметры:

$$U = 1, \quad V = 0, \; u = 0.055, \; \vartheta = 0.028 r^{1/2}.$$

Выше сказано, что для конвективных членов использована схема Леонардо QUICK. Представим эту схему для одномерного случая

$$\frac{\partial U\Phi}{\partial x} = \frac{U_e\Phi_e - U_w\Phi_w}{\Delta x} \tag{4}$$

$$U_e = \frac{(U_i + U_{i+1})}{2}, \quad U_w = \frac{(U_i + U_{i-1})}{2}.$$

$$\begin{cases} \Phi_e = \frac{3}{16}(\Phi_i + \Phi_{i+1}) + \frac{3}{8}(\Phi_i + \Phi_{i-1}) - \frac{1}{16}(\Phi_{i-2} + \Phi_{i-1}). & U_e > 0, \\ \Phi_e = \frac{3}{16}(\Phi_i + \Phi_{i-1}) + \frac{3}{8}(\Phi_i + \Phi_{i+1}) - \frac{1}{16}(\Phi_{i+1} + \Phi_{i+2}). & U_e < 0, \\ \Phi_w = \frac{3}{16}(\Phi_i + \Phi_{i-1}) + \frac{3}{8}(\Phi_{i-2} + \Phi_{i-1}) - \frac{1}{16}(\Phi_{i-3} + \Phi_{i-2}). & U_w > 0, \\ \Phi_w = \frac{3}{16}(\Phi_{i-2} + \Phi_{i-1}) + \frac{3}{8}(\Phi_i + \Phi_{i-1}) - \frac{1}{16}(\Phi_i + \Phi_{i+1}). & U_w < 0, \end{cases}$$

Приведём численные результаты полученные на основе новой двухжидкостной модели и модели Спаларта-Аллмареса.

На рисунке 1 приведены численные результаты для безразмерной осевой продольной скорости в зависимости от расстояния до сопла.

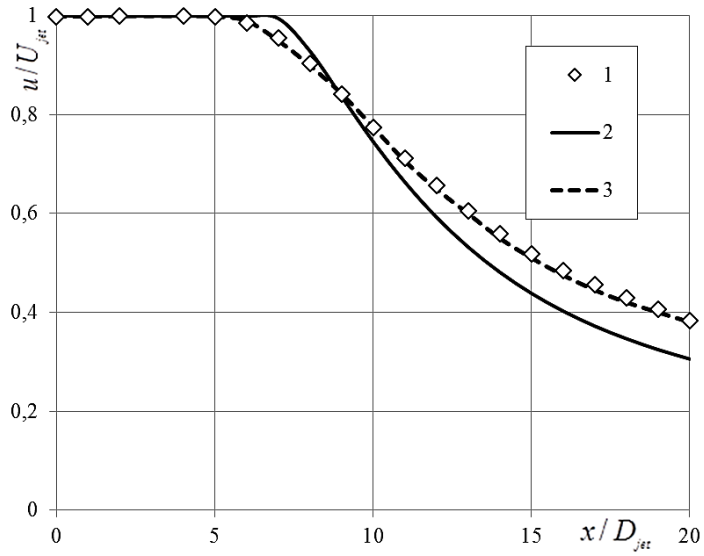


Рис. 1. Безразмерная осевая продольная скорость в зависимости от расстояния до сопла.
1 – эксперимент, 2 – модель SA, 3 – новая двухжидкостная модель

На рисунке 2 представлены профили турбулентного напряжения в различных сечениях струи.

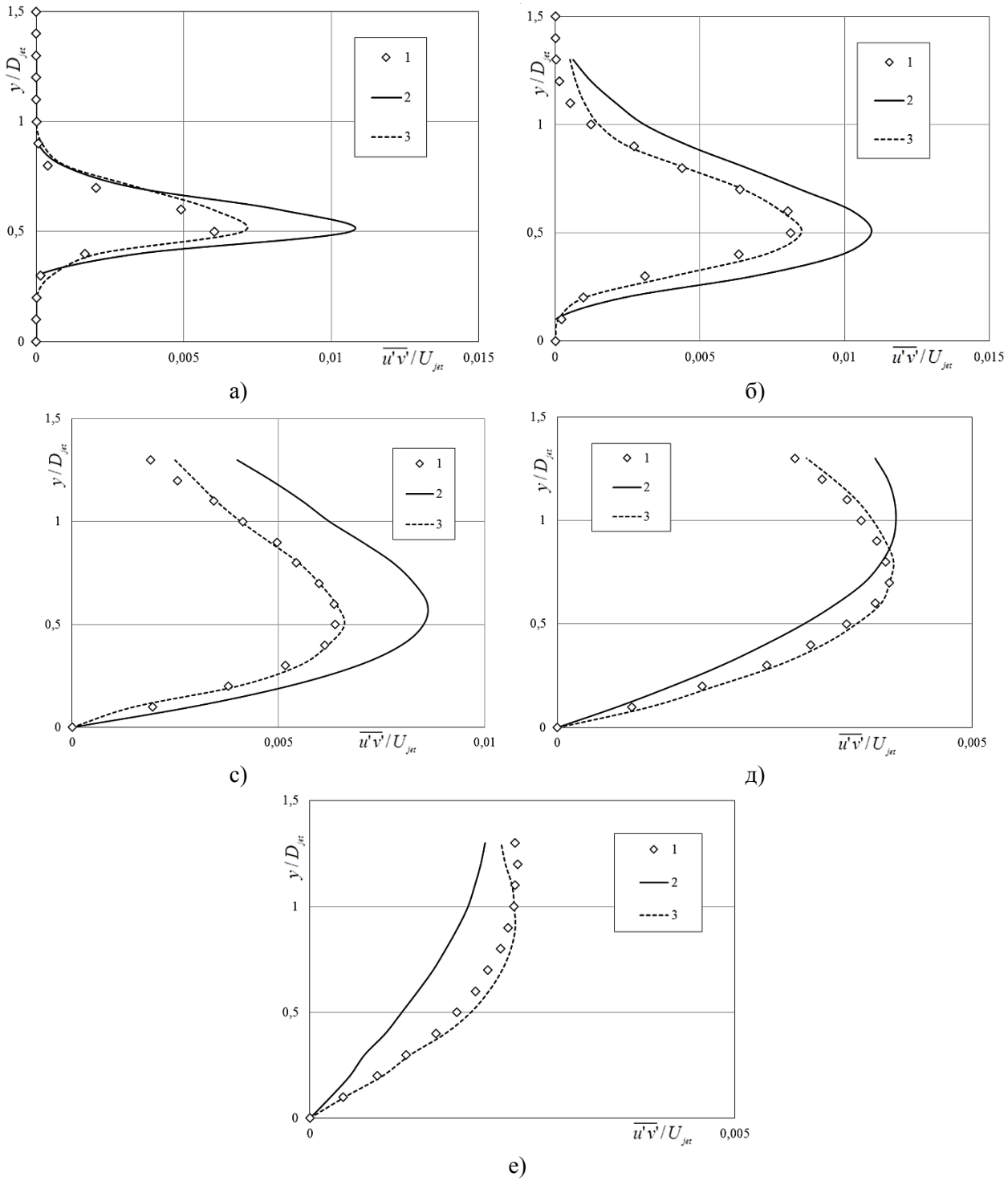


Рис. 2. Профиль турбулентного напряжения.
1 – экспериментальные данные, 2 – модель SA, 3 – двухжидкостная модель.
а) $x/D_{jet}=2$, б) $x/D_{jet}=5$, с) $x/D_{jet}=10$, д) $x/D_{jet}=15$, е) $x/D_{jet}=20$

Необходимо отметить, что в новой двухжидкостной модели турбулентное напряжение $\overline{u'v'} = uv$ и среднеквадратичное значение флуктуирующей продольной скорости $\sqrt{\overline{u'^2}} = u$.

Представленные на рисунках 1-2 показывают, что новая двухжидкостная модель более точно описывает турбулентную осесимметричную струю, чем модель SA. Кроме того, новая двухжидкостная модель более устойчива, чем модель SA. Численные эксперименты показали, что модель SA теряла устойчивость при шаге времени $\Delta t > 0.001$ в то время как

двухжидкостная модель была устойчива даже при $\Delta t > 0.01$. Поэтому новую двухжидкостную модель можно считать практичной для расчётов инженерных задач и имеющей высокую точность.

Литература

1. Абрамович Г.Н. Теория турбулентных струй. М: Наука. 1984. 716 с.

2. Malikov Z.M. Mathematical Model of Turbulence Based on the Dynamics of Two Fluids // Applied Mathematic Modeling. 2020. Vol. 82. Pp. 409-436. https://doi.org/10.1016/j.apm.2020.01.047

3. Spalart P.R., Allmaras S.R. A one-equation turbulence model for aerodynamic flows // 30th aerospace sciences meeting and exhibit. 1992. Pp. 439-478. https://doi.org/10.2514/6.1992-439

4. Malikov Z.M., Madaliev M.E. Numerical Simulation of Two-Phase Flow in a Centrifugal Separator // Fluid Dynamics. 2020. Vol. 55. No. 8. Pp. 1012-1028. https://doi.org/10.1134/S0015462820080066

5. Leonard B.P. A stable and accurate convective modeling procedure based on quadratic upstream interpolation // Computer methods in applied mechanics and engineering. 1979. Vol. 19. Issue 1. Pp 59-98.

6. Patankar S.V. Numerical heat transfer and fluid flow. Hemisphere Publishing Corporation, 1980.

7. Андерсон Д., Таннехилл Дж., Плетчер Р. Вычислительная гидромеханика и теплообмен. М.: Мир, 1990. Т. 1. 384 с.

© Маликов З.М., Навруzов Д.П., 2022

УДК 519.6
https://doi.org/10.36906/AP-2022/24

Наврузов Д.П.
Институт механики и сейсмостойкости сооружений им. М.Т. Уразбаева АН РУз
г. Ташкент, Узбекистан

ВИХРЕВОЕ ТЕЧЕНИЕ ПОСЛЕ ВНЕЗАПНОГО РАСШИРЕНИЯ

Аннотация. С использованием новой турбулентной модели динамики двухжидкостей выполнено исследование структуры течения и процесса распространения дисперсной примеси в турбулентном потоке после внезапного расширения трубы. Изучено влияние частиц на процессы распространения дисперсной фазы, на турбулентность газа и дисперсию твёрдой фазы. Показано, что малоинерционные частицы вовлекаются в циркуляционное движение и могут накапливаться в ней. Крупные частицы проходят в основном через область сдвигового течения и практически не попадают в отрывную зону. Выполнено сопоставление с данными измерений для случая монодисперсных двухфазных отрывных течений.

Ключевые слова: внезапное расширение; динамика двух жидкостей; вихрь; контрольный объем; молярный вязкость.

Navruzov D.P.
Institute of Mechanics and Seismic Stability of Structures
named after M.T. Urazbayev, AS RUz
Tashkent, Uzbekistan

VORTEX FLOW AFTER SUDDEN EXPANSION

Abstract. Using a new turbulent model of the dynamics of two-fluids, the structure of the flow and the process of propagation of a dispersed impurity in a turbulent flow after a sudden expansion of the pipe was studied, the effect of particles on the processes of propagation of the dispersed phase, on gas turbulence and dispersion of the solid phase was studied. It is shown that low-inertia particles are involved in the circulation movement and can accumulate in it. Large particles pass mainly through the shear flow region and practically do not fall into the separation zone. Comparison with the measurement data for the case of monodisperse two-phase separation flows is performed.

Keywords: sudden expansion; dynamics of two liquids; vortex; reference volume; molar viscosity.

Течение после внезапного расширения трубы является одним из часто встречающихся случаев отрывного потока при обтекании острых кромок тел. Процесс отрыва сопровождается значительными изменениями скорости и давления в зоне отрыва. Течение и теплообмен вниз от сечения отсоединения потока зависит от предыстории развития потока и геометрии

CC BY

поверхности в районе сечения отрыва. Внезапное расширение потока в трубах или каналах обычно используется в качестве стабилизатора пламени в камерах сгорания, для интенсификации тепломассы обменных процессов и во многих других технических устройствах. Достаточно часто отрыв- это явление, приводящее к негативным последствиям: увеличение сопротивления газа в трубопроводах, снижение скорости и прочее. Наличие рециркуляционного течения в отрывной зоне оказывает значительное влияние на интенсивность процессов переноса импульса, теплоты и массы и определяет структуру турбулентного течения. Достаточно часто в различных технологических приложениях используются двухфазные течения с отрывом. При этом, помимо указанных выше параметров, на процессы переноса большое влияние оказывает дисперсная фаза. Её воздействие возрастает с ростом диаметра частиц и их концентрации. Цель настоящей работы – проведение численного исследования распространения дисперсной примеси в турбулентном потоке после внезапного расширения трубы при наличии теплообмена двухфазного потока с поверхностью стенки трубы.

Как было отмечено выше, серьёзными тестовыми задачами для моделей турбулентности являются задачи с закрученными потоками. Поэтому в качестве следующей тестовой задачи для верификации новой модели рассмотрим процесс прецессии сильно закрученного потока после внезапного расширения рисунке 1.

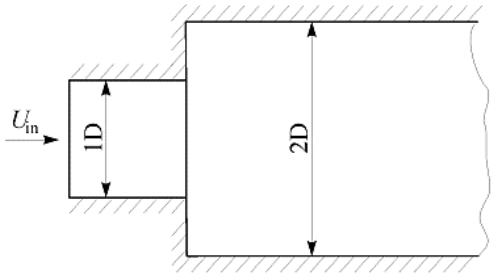


Рис. 1. Труба с внезапным расширением

Теперь рассмотрим математическую модель турбулентности на основе динамики двух жидкостей. Система уравнений для турбулентного потока имеет следующий вид:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial \rho U_i}{\partial x_i} = 0, \\ \dfrac{\partial \rho U_i}{\partial t} + \dfrac{\partial (\rho U_j U_i + \delta_{ij} p)}{\partial x_j} = \dfrac{\partial}{\partial x_j}\left(\Pi_{ij} - \rho \nu_i \nu_j\right), \\ \dfrac{\partial \rho \nu_i}{\partial t} + \rho U_j \dfrac{\partial \nu_i}{\partial x_j} = -\rho \nu_j \dfrac{\partial U_i}{\partial x_j} + \dfrac{\partial \pi'_{ij}}{\partial x_j} + F_{ri} + F_{\perp i}, \\ \Pi_{ij} = \mu\left(\dfrac{\partial U_i}{\partial x_j} + \dfrac{\partial U_j}{\partial x_i}\right), \pi'_{ij} = \rho \nu'_{\nu_i \nu_j}\left(\dfrac{\partial \nu_i}{\partial x_j} + \dfrac{\partial \nu_j}{\partial x_i}\right), \\ \nu'_{\nu_i \nu_j} = 3\nu + \left|\dfrac{\nu_i \nu_j}{def(U)}\right|, F_r = -\rho C_r \nu, F_\perp = 2\rho C_s rot U \times \nu. \end{cases} \quad (1)$$

Необходимо отметить, что π'_{ij} - является тензором. Поэтому в правой части для данного тензора суммирование по повторяющимся индексам не производится.

CC BY

Задачу будем исследовать в цилиндрических координатах. В задаче пренебрежём продольными производными в диффузионных членах. Тогда система уравнений (1) в цилиндрических координатах будет иметь вид:

$$\begin{cases} \dfrac{\partial V_z}{\partial z}+\dfrac{\partial rV_r}{r\partial r}=0, \\ \dfrac{\partial V_z}{\partial t}+V_z\dfrac{\partial V_z}{\partial z}+V_r\dfrac{\partial V_z}{\partial r}+\dfrac{1}{\rho}\dfrac{\partial p}{\partial z}=\dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 V_z}{\partial r^2}+\dfrac{\partial V_z}{r\partial r}\right)-\dfrac{\partial r v_r v_z}{r\partial r}, \\ \dfrac{\partial V_r}{\partial t}+V_z\dfrac{\partial V_r}{\partial z}+V_r\dfrac{\partial V_r}{\partial r}+\dfrac{1}{\rho}\dfrac{\partial p}{\partial r}=\dfrac{V_\varphi^2}{r}+\dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 V_r}{\partial r^2}+\dfrac{\partial V_r}{r\partial r}-\dfrac{V_r}{r^2}\right)-\dfrac{\partial r v_r v_r}{r\partial r}, \\ \dfrac{\partial V_\varphi}{\partial t}+V_z\dfrac{\partial V_\varphi}{\partial z}+V_r\dfrac{\partial V_\varphi}{\partial r}+\dfrac{V_r V_\varphi}{r}=\dfrac{1}{\mathrm{Re}}\left(\dfrac{\partial^2 V_\varphi}{\partial r^2}+\dfrac{\partial V_\varphi}{r\partial r}-\dfrac{V_\varphi}{r^2}\right)-\dfrac{\partial r^2 v_r v_\varphi}{r^2\partial r}, \\ \dfrac{\partial v_z}{\partial t}+V_z\dfrac{\partial v_z}{\partial z}+V_r\dfrac{\partial v_z}{\partial r}=-(1-2C_s)\dfrac{\partial V_z}{\partial r}v_r+\dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\nu'_{zr}\dfrac{\partial v_z}{\partial r}\right)-C_r v_z, \\ \dfrac{\partial v_r}{\partial t}+V_z\dfrac{\partial v_r}{\partial z}+V_r\dfrac{\partial v_r}{\partial r}=-2C_s\dfrac{\partial V_z}{\partial r}v_z-(2C_s\dfrac{\partial rV_\varphi}{r\partial r}-\dfrac{2V_\varphi}{r})v_\varphi+\dfrac{2}{r}\dfrac{\partial}{\partial r}(r\nu'_{rr}\dfrac{\partial v_r}{\partial r})-2\nu'_{rr}\dfrac{v_r}{r^2}-C_r v_z, \\ \dfrac{\partial v_\varphi}{\partial t}+V_z\dfrac{\partial v_\varphi}{\partial z}+V_r\dfrac{\partial rv_\varphi}{r\partial r}=-(1-2C_s)\dfrac{\partial rV_\varphi}{r\partial r}v_r+\dfrac{1}{r^2}\dfrac{\partial}{\partial r}(r^2\nu'_{\varphi r}(\dfrac{\partial v_\varphi}{\partial r}-\dfrac{v_\varphi}{r}))-C_r v_\varphi, \end{cases} \quad (2)$$

Для определения молярных вязкостей воспользуемся соотношениями:

$$\nu'_{zr}=3\nu+2\left|\frac{v_z v_r}{def(U)}\right|$$

$$\nu'_{rr}=3\nu+2\left|\frac{v_r v_r}{def(U)}\right|$$

$$def(U)=\sqrt{(\frac{\partial V_z}{\partial r})^2+(\frac{\partial V_\varphi}{\partial r}-\frac{V_\varphi}{r})^2}$$

Для обеспечения устойчивости при решении уравнений системы (1) в конвективных членах использована разностная схема против потока второго порядка точности по методу контрольного объёма. А для диффузионных членов использовалась центральная разность в неявном виде. Связь полей скорости и давления для несжимаемой жидкости реализовывалась с помощью процедуры SIMPLEC [1]. После формирования квазипериодического режима проводилось усреднение нестационарных полей.

CC BY

$$\begin{cases} \frac{\tilde{U}-U^n}{\Delta t}+U^n\frac{\partial U^n}{\partial \xi}+U^n\eta'\frac{\partial U^n}{\partial \eta}+V^n\times F(z)\frac{\partial U^n}{\partial \eta}+\frac{\partial P}{\partial \xi}+\eta'\frac{\partial P}{\partial \eta}=\frac{F(z)}{r}\frac{\partial}{\partial \eta}\left[r\nu_{eff}F(z)\frac{\partial \tilde{U}}{\partial \eta}\right]-\sum_{i=1}^{N}\frac{\rho_i}{\rho}k_i(U-U_p), \\ \frac{\tilde{V}-V^n}{\Delta t}+U^n\frac{\partial V^n}{\partial \xi}+U^n\eta'\frac{\partial V^n}{\partial \eta}+V^n\times F(z)\frac{\partial V^n}{\partial \eta}-\frac{G^2}{r^3}+F(z)\frac{\partial P}{\partial \eta}=\frac{F(z)}{r}\frac{\partial}{\partial \eta}\left[r\nu_{eff}F(z)\frac{\partial \tilde{V}}{\partial \eta}\right]-\frac{\nu_{eff}}{r^2}\tilde{V}-\sum_{i=1}^{N}\frac{\rho_i}{\rho}k_i(V-V_p), \\ \frac{\partial \tilde{v}}{\partial t}+U\frac{\partial \tilde{v}}{\partial \xi}+U\eta'\frac{\partial \tilde{v}}{\partial \eta}+F(z)V\frac{\partial \tilde{v}}{\partial \eta}=(Pv-Dv)+\frac{F(z)}{r\sigma_v}\frac{\partial}{\partial \eta}\left[r(\nu_{eff})F(z)\frac{\partial \tilde{v}}{\partial \eta}\right]+\frac{C_{b2}}{\sigma_v}\left(F(z)\frac{\partial \tilde{v}}{\partial \eta}\right)^2, \\ \frac{\partial U_p}{\partial t}+U_p\frac{\partial U_p}{\partial \xi}+U_p\eta'\frac{\partial U_p}{\partial \eta}+V_p\times F(z)\frac{\partial U_p}{\partial \eta}=k_i(U-U_p), \\ \frac{\partial V_p}{\partial t}+U_p\frac{\partial V_p}{\partial \xi}+U_p\eta'\frac{\partial V_p}{\partial \eta}+V_p\times F(z)\frac{\partial V_p}{\partial \eta}-\frac{W_p^2}{r}=k_i(V-V_p), \\ \frac{\partial W_p}{\partial t}+U_p\frac{\partial W_p}{\partial \xi}+U_p\eta'\frac{\partial W_p}{\partial \eta}+V_p\times F(z)\frac{\partial W_p}{\partial \eta}+\frac{W_pV_p}{r}=k_i(W-W_p), \end{cases} \quad (3)$$

$$\begin{cases} U^{n+1}=\tilde{U}-\Delta t\left(\frac{\partial \delta p}{\partial \xi}+\eta'\frac{\partial \delta p}{\partial \eta}\right) \\ V^{n+1}=\tilde{V}-\Delta t\times F(z)\frac{\partial \delta p}{\partial \eta} \end{cases}$$

Для численного решения уравнения переноса системы (4) использована конечно-разностная схема против потока, которая имеет точность второго порядка, т. е. $O(\Delta t,\Delta\xi^2,\Delta\eta^2)$

.

$$\begin{cases} \frac{\tilde{U}-U^n}{\Delta t}+U^n\frac{\partial U^n}{\partial \xi}+U^n\eta'\frac{\partial U^n}{\partial \eta}+V^n\times F(z)\frac{\partial U^n}{\partial \eta}+\frac{\partial P}{\partial \xi}+\eta'\frac{\partial P}{\partial \eta}=\frac{F(z)}{r}\frac{\partial}{\partial \eta}\left[r\nu_{eff}F(z)\frac{\partial \tilde{U}}{\partial \eta}\right]-\sum_{i=1}^{N}\frac{\rho_i}{\rho}k_i(U-U_p), \\ \frac{\tilde{V}-V^n}{\Delta t}+U^n\frac{\partial V^n}{\partial \xi}+U^n\eta'\frac{\partial V^n}{\partial \eta}+V^n\times F(z)\frac{\partial V^n}{\partial \eta}-\frac{G^2}{r^3}+F(z)\frac{\partial P}{\partial \eta}=\frac{F(z)}{r}\frac{\partial}{\partial \eta}\left[r\nu_{eff}F(z)\frac{\partial \tilde{V}}{\partial \eta}\right]-\frac{\nu_{eff}}{r^2}\tilde{V}-\sum_{i=1}^{N}\frac{\rho_i}{\rho}k_i(V-V_p), \\ \frac{\partial \tilde{v}}{\partial t}+U\frac{\partial \tilde{v}}{\partial \xi}+U\eta'\frac{\partial \tilde{v}}{\partial \eta}+F(z)V\frac{\partial \tilde{v}}{\partial \eta}=(Pv-Dv)+\frac{F(z)}{r\sigma_v}\frac{\partial}{\partial \eta}\left[r(\nu_{eff})F(z)\frac{\partial \tilde{v}}{\partial \eta}\right]+\frac{C_{b2}}{\sigma_v}\left(F(z)\frac{\partial \tilde{v}}{\partial \eta}\right)^2, \\ \frac{\partial U_p}{\partial t}+U_p\frac{\partial U_p}{\partial \xi}+U_p\eta'\frac{\partial U_p}{\partial \eta}+V_p\times F(z)\frac{\partial U_p}{\partial \eta}=k_i(U-U_p), \\ \frac{\partial V_p}{\partial t}+U_p\frac{\partial V_p}{\partial \xi}+U_p\eta'\frac{\partial V_p}{\partial \eta}+V_p\times F(z)\frac{\partial V_p}{\partial \eta}-\frac{W_p^2}{r}=k_i(V-V_p), \\ \frac{\partial W_p}{\partial t}+U_p\frac{\partial W_p}{\partial \xi}+U_p\eta'\frac{\partial W_p}{\partial \eta}+V_p\times F(z)\frac{\partial W_p}{\partial \eta}+\frac{W_pV_p}{r}=k_i(W-W_p), \end{cases} \quad (4)$$

На входе задавался экспериментальный профиль скорости, измеренный для сечения $z=-5D$, соответствующего расположению входа расчётной области [2].На выходе $z=5D$ задавались «мягкие» условия. Поток имел следующих входных параметров:

$$\mathrm{Re}=\frac{DU_{in}}{\nu}=30000,\quad S=\frac{\int_0^R V_{0\varphi}V_{0z}r^2dr}{R\int_0^R V_z^2rdr}=0.6$$

CC BY

$$v_\varphi = 0.2U_{im},\ v_z = 0.2U_{in},\ v_r = 0.05U_{in},$$

Здесь Re – число Рейнольдса, S – параметр определяющий степень закрутки входящего потока, U_{in} – максимальная осевая скорость на входе. Для расчётов использовалась сетка 100x100.Уравнения в безразмерный вид приводились соотнесением всех скоростей к U_{in}, а пространственные масштабы к диаметру малой трубки. Интегрирование по времени проводилось безразмерным шагом $\Delta t = 0.001$. В качестве начальных условий были заданы

$$V_\varphi = V_{0\varphi}(1 - \frac{0.1z}{D}),\ V_z = V_{0z},\ V_r = 0,\ \text{if z>0 and r<0.5D}$$

$$V_\varphi = V_z = V_r = 0,\ \text{if z>0 and r>0.5D}$$

На рисунке 2–5 сплошными линиями представлены численные результаты. Точками изображены опытные данные полученные в работе [2]. Здесь все пространственные масштабы соотнесены к диаметру входной трубы. На рисунках 2 и 3 представлены профили соответственно аксиальной и тангенциальной скоростей потока в сечении x=0.2. Аналогичные графики представлены на рисунках 4 и 5 для x=0.5.

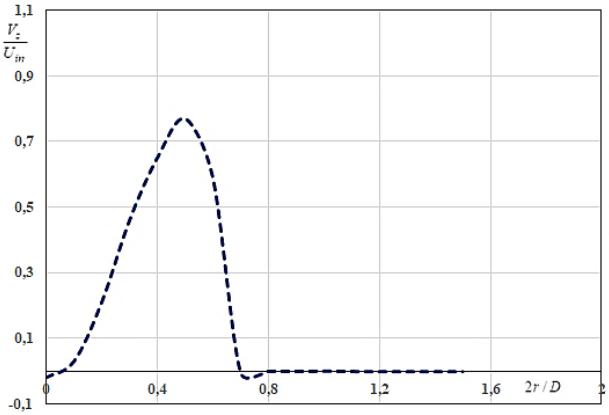


Рис. 2. Профиль аксиальной скорости в сечении x=0.2

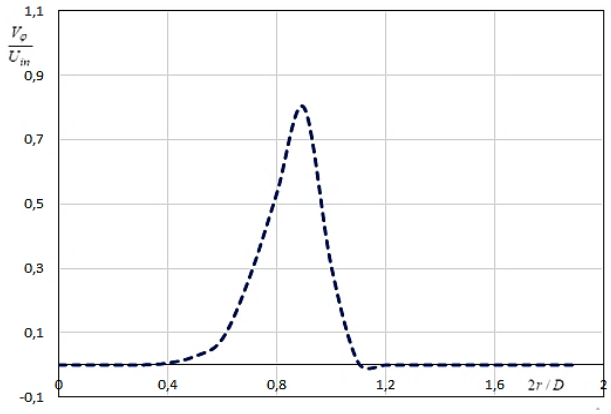


Рис. 3. Профиль тангенциальной скорости в сечении x=0.2

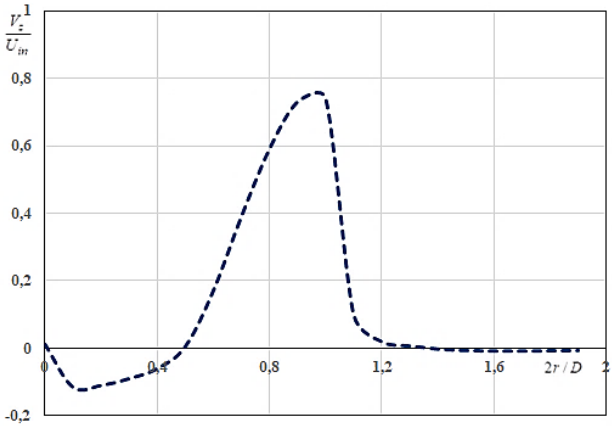


Рис. 4. Профиль аксиальной скорости в сечении x=0.5

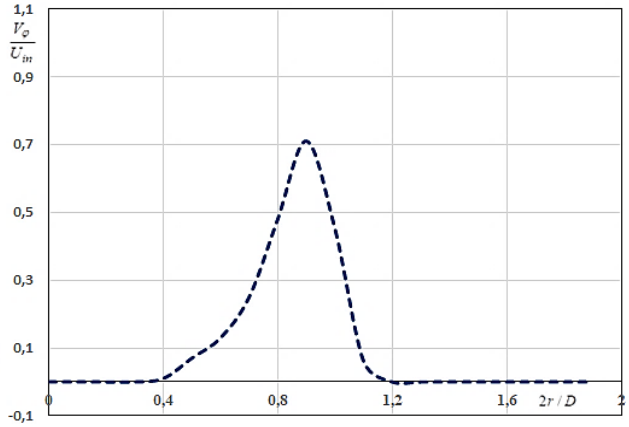


Рис. 5. Профиль тангенциальной скорости в сечении x=0.5

В статье показана возможность моделирования вихревого течения после внезапного расширения с помощью новой модели. Поэтому математическое моделирование проведено на основе динамики двух жидкостей. Калибровка и верификация новой модели проведена на известных задачах, которые хорошо исследованы экспериментально. Несмотря на то, что

природа турбулентности у этих задач разные, новая модель количественно неплохо описывает эти сложные течения.

Литература

1. Patankar S.V. Numerical Heat Transfer and Fluid Flow. Taylor & Francis London. 1980.

2. Dellenback P.A., Metzger D.E., Neitzel G.P. Measurments in turbulent swirling flow through an abrupt expansion // AIAA Journal. 1988. Vol. 26. No. 6. Pp. 669-681. https://doi.org/10.2514/3.9952

© Наврузов Д.П., 2022

УДК 681.5
https://doi.org/10.36906/AP-2022/25

Полкова Е.В.
Кузяков О.Н.
д-р техн. наук
Тюменский индустриальный университет
г. Тюмень, Россия

ВИРТУАЛЬНЫЕ АНАЛИЗАТОРЫ КАК ИНСТРУМЕНТ ДЛЯ ОЦЕНКИ ТЕКУЩЕГО СОСТОЯНИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА ТРУБОПРОВОДА

Аннотация. В статье выявлена актуальная задача нефтегазовой отрасли – повышение качества и эффективности оценивания состояния технологических процессов на предприятии. Описаны достоинства и недостатки существующих методов. Подробно рассмотрены виртуальные анализаторы как математические модели, посредством которых производится косвенный расчёт показателей качества технологического процесса в качестве способа определения качества и эффективности оценки технологических процессов трубопроводов. Изучена модель и принцип действия виртуального анализатора. Подробно расписаны цели и задачи использования виртуальных анализаторов.

Ключевые слова: технологические процессы трубопровода; методы оценки качества и эффективности; виртуальные анализаторы.

Polkova E.V.
Kuzyakov O.N.
Ph.D.
Industrial University of Tyumen
Tyumen, Russia

VIRTUAL ANALYZERS AS A TOOL FOR ASSESSING THE CURRENT STATE OF THE TECHNOLOGICAL PROCESS OF A PIPELINE

Abstract. The article reveals the urgent task of the oil and gas industry – improving the quality and efficiency of assessing the state of technological processes at the enterprise. The advantages and disadvantages of using the methods are described. Virtual analyzers are considered in detail as mathematical models that are produced using indirect calculation of the quality indicators of the technological process as a definition of the quality and efficiency of assessing the technological processes of pipelines. The model and the principle of operation of the virtual analyzer have been studied. The goals and objectives of using virtual analyzers are described in detail.

Keywords: pipeline technological processes; methods for assessing quality and efficiency; virtual analyzers.

CC BY

Наиболее дешёвым способом транспортировки нефти и нефтепродуктов в настоящее время является трубопроводный транспорт. Высокая вероятность аварий связанных с неисправностью в линейной части трубопровода, обусловлена естественным износом нефтепроводов, так как большая часть трубопроводов была построена ещё в конце XX века. Разливы нефти из-за изношенных трубопроводов приносят огромный ущерб экологии, ликвидация таких разливов процесс дорогостоящий и трудозатратный. Во избежание разливов, и с целью минимизации ущерба для экологии применяются меры при мониторинге показателей течения и режимов перекачки.

Повышение качества и эффективности оценивания состояния технологических процессов является первостепенной и наиболее актуальной задачей нефтегазовой отрасли на сегодняшний день. Мониторинг текущих технологических процессов является распространённым методом управления производством. Актуальное состояние технологического процесса отслеживается посредством таких параметров, как температуры, давления, расхода и иных, значения о показателях которых поступают от датчиков, которые находятся на отдельной технологической установке.

В большинстве практических случаев оперативное управление технологическими процессами реализуются на основе использования результатов мониторинга производственной ситуации. Результаты производственного мониторинга лежат в основе оперативного управления технологическими процессами. Мониторинг состоит из этапов сбора, первичной обработки данных состоящей из результатов измерительных средств и комплексов, входящих в состав автоматизированных систем управления технологических процессов, и лабораторных анализов промежуточной и товарной продукции.

В современном мире с целью оценки текущего состояния технологических процессов трубопровода активно применяются три способа: лабораторные анализы, данные поточных анализаторов, а также данные виртуальных анализаторов.

Лабораторные анализы облают высокой точностью, относительно недорогие, но не проводятся постоянно, что не даёт возможность оперативно реагировать на изменения в технологическом процессе. Результаты анализов из заводских лабораторий обычно не могут обещать необходимые уровни полноты и оперативности и зачастую при практическом опыте, результаты лабораторных анализов показывают неудовлетворительную достоверность. Основная причина - несоответствие пропускной способности и технологичности лабораторных средств анализа проб необходимостям производства. Поточные анализаторы хоть и определяют физико-химические свойства продуктов непрерывно в режиме реального времени, имеют высокую точность, самодиагностику и визуализацию показаний, но требуют высококвалифицированное обслуживание, имеют высокую стоимость и необходимость периодичной калибровки.

Виртуальные анализаторы служат дополнением к вышеописанным методам и являются моделью, предназначенной для косвенного измерения качественных показателей процесса, построенную на основе данных лабораторных анализов и архивных производственных

CC BY

данных [1]. Виртуальные анализаторы способны повысить своевременность результатов контроля состояния материальных потоков, но имеют большую стоимость и регулярное высококвалифицированное эксплуатационное обслуживание и не способны обеспечить полноту информационного обеспечения с точки зрения создания автоматизированных контуров оптимального управления технологических процессов. Так, можно сформулировать проблему повышения полноты, оперативности и достоверности информационного обеспечения технологического персонала за счёт создания и внедрения системы виртуального мониторинга технологических процессов [2].

Виртуальные анализаторы представляют наиболее оптимальный способ определения качества технологических процессов трубопроводов, не уступая в точности другим способам оценки. Принцип действия виртуальных анализаторов основывается на монотонном измерении качества посредством запрограммированных математических моделей, которые описывают их взаимосвязь с настоящими значениями измеряемых технологических переменных. На рисунке представлена общая модель виртуального анализатора.

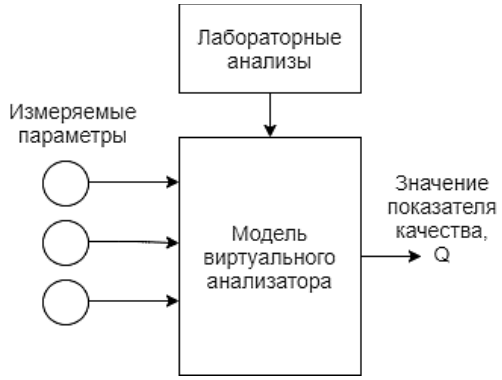


Рис. Общая модель виртуального анализатора

Виртуальные анализаторы представляют возможность оценивать необходимые показатели качества технологических процессов трубопровода по следующим измеряемым параметрам продукта: температура, давление, расход, которые, в свою очередь, непрерывно контролируются современными системами управления [3].

Виртуальные анализаторы представляют из себя математические модели, посредством которых производится косвенный расчёт показателей качества технологического процесса. В качестве данных для расчёта используются архивные результаты производственных и лабораторных анализов. ВА интегрируются посредством специального программного обеспечения, которые, в свою очередь, реализуются в базовой системе управления [4].

Принимая воздействие различных возмущений, технологический процесс периодически отклоняется относительно оптимальных режимных параметров, что, в свою очередь, требует его корректировки. Параллельно с этим, качество работы трубопроводного транспорта определяется не только показателями основных технологических параметров, но и, непосредственно, составом сырья и показателями качества производимых продуктов. Посредством виртуальных анализаторов предоставляется возможность в настоящий момент времени получать и использовать в управлении данных о трудноизмеримых параметрах

CC BY

технологического процесса. Наряду с этим, ВА выполняют функцию обычных измерительных устройств и могут использоваться в системах управления совместно с ними.

Отметим основные цели использования виртуальных анализаторов:

– мониторинг трудноизмеримых параметров;

– повышение оперативности управления технологических процессов;

– использование в системах оптимизации технологических процессов;

– контроль достоверности параметров.

Задачи, которые способны решать виртуальные анализаторы:

– прогноз качества продукта на любом этапе технологического процесса;

– создание оптимальных технологических решений при заданных критериях качества;

– поиск решений и новых технологических режимов по управлению технологическими процессами для критериев эффективности;

– обеспечение устойчивой и безопасной эксплуатации объекта;

– изучение значимости и функций влияния отдельных технологических параметров;

– выявление и учёт взаимозависимостей технологических параметров;

– изучение экологической и промышленной безопасности эксплуатации объекта, своевременное оповещение и обнаружение опасности;

– основа для построения системы гибкого планирования процессов контроля и ремонта контрольно-измерительных средств;

– обеспечение возможности управления технологическим объектом персоналом с недостаточным уровнем квалификации.

Кроме того, дополнительными задачами можно выделить:

– информационное дублирование отдельных измерительных средств для оперативного контроля их состояния;

– использование в качестве вспомогательной контрольно-диагностической системы технологической установки в интересах раннего обнаружения возможных неисправностей [5].

Виртуальный мониторинг способствует получению новых знаний о эволюции, текущем состоянии и динамике технологических процессов с использованием математической обработки оперативных и ретроспективных данных, полученных контрольно-измерительными средствами. Виртуальные анализаторы входят в систему виртуального мониторинга и реализуют возможность изучения эволюции и текущего состояния технологических процессов.

Так, можно сформулировать общую цель виртуальных анализаторов – повышение уровня информационно-аналитического обеспечения персонала и создание условий (достаточной информационной базы) для формирования оптимального управления как одного технологического процесса, так и всего технологического цикла.

В связи с тем, что показатели качества технологического процесса во многом характеризуют эффективность работы системы автоматизации и технологического объекта в целом, а также могут входить в целевую функцию оптимизации, виртуальные анализаторы

CC BY

являются основой для построения таких систем оптимизации в реальном времени, как системы усовершенствованного управления технологическими процессами.

Таким образом, основной целью представленной статьи являлось изучение использования виртуальных анализаторов в качестве инструмента для оценки текущего состояния технологического процесса трубопровода. В результате работы были изучены такие аспекты, как: актуальность повышения качества и эффективности оценивания состояния технологических процессов; виртуальные анализаторы в качестве способа определения качества технологических процессов трубопроводов; модель и принцип действия виртуального анализатора; цели и задачи использования виртуальных анализаторов [6].

В заключение необходимо отметить, что автоматизация занимает ключевое место в развитии технологических процессов, посредством которой значительно повышается качество и эффективность в решении множества задач, одной из которых является оценка текущего состояния технологических процессов. Виртуальные анализаторы являются перспективным средством, повышающим эффективность оценивания текущего состояния технологического процесса. Исходя из этого, их изучение и повышение качества и эффективности измерения должно являться ключевой задачей из области автоматизации технологических процессов.

Литература

1. Лежнин Д.В. Опыт создания моделей технологических процессов для предприятий нефтехимической промышленности // ИСУП. 2015. № 3(57). С. 87-90.

2. Кремлев И.А., Громаков Е.И. Использование виртуальных анализаторов для управления технологическими процессами // Молодёжь и современные информационные технологии: сб. трудов XVI Международной научно-практической конференции студентов, аспирантов и молодых учёных (г. Томск, 03-07 декабря 2018 года). Томск, 2019. С. 287-288.

3. Данилов А.Д., Мугатина В.М. Верификация и тестирование сложных программных продуктов на основе нейросетевых моделей // Вестник Воронежского государственного технического университета. 2016. Т. 12. № 6. С. 62-67.

4. Клокотов И.Ю. Автоматизация технологических процессов // Международный журнал прикладных наук и технологий Integral. 2019. № 4-1.

5. Лобанова В.А., Воронина О.А. Электронные и виртуальные средства контроля качества нефтепродуктов // Фундаментальные и прикладные проблемы техники и технологии. 2011. № 5. С. 109-113.

6. Мусаев А.А., Тураносов А.В. От on-line анализаторов к системе тотального мониторинга качества продукции // Нефтепереработка и нефтехимия. Научно-технические достижения и передовой опыт. 2012. № 12. С. 46-55.

© Полкова Е.В., Кузяков О.Н., 2022

УДК 519.852.6
https://doi.org/10.36906/AP-2022/26

Садыгов А.Б.
д-р техн. наук
Зейналов Р.М.
Ph.D.
Институт систем управления НАН Азербайджана
г. Баку, Азербайджан
Мамедли Р.Э.
канд. физ.-мат. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия

ИССЛЕДОВАНИЕ РЕШЕНИЙ ОДНОЙ ГРАНИЧНОЙ ЗАДАЧИ

Аннотация. Данная работа посвящена исследованию решений одной из граничных задач для уравнения Коши-Римана на плоской полосе с единичной шириной. Исходя из фундаментального решения по направлению x_2 уравнения Коши-Римана получены необходимые условия. Эти необходимые условия, примененные к граничной задаче, приводят ее к разностному уравнению. Решение полученного разностного уравнения сводится к определению граничного значения искомой функции. Далее граничная условия определяет следующий граничное значение и наконец решения поставленное граничной задач получается из основного соотношения.

Ключевые слова: Уравнения Коши-Римана; плоская полоса; граничные условия; дельта-функция Дирака; функция Хевисайда.

Sadigov A.B.
Ph.D.
Zeynalov R.M.
Ph.D.
Institute of Control Systems of the Azerbaijan National Academy of Sciences
Baku, Azerbaijan
Mammadli R.E.
Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

INVESTIGATION OF SOLUTIONS TO A SINGLE BOUNDARY VALUE PROBLEM

Abstract. The presented work is devoted to the study of solutions of one boundary value problem for the Cauchy-Riemann equation on a flat strip with unit width. Based on the fundamental

CC BY

solution in the direction of x_2 , the Cauchy-Riemann equation, we obtain the necessary conditions. This necessary condition is applied to the boundary condition to reduce this condition to a difference equation. The solution of the obtained difference equation can be reduced to some definition of the boundary value of the required function. Further, the boundary condition determines the next boundary value, and finally the solution to the formulated boundary value problem is obtained from the basic relation.

Keywords: Cauchy-Riemann equations; flat strip; boundary conditions; Dirac delta function; Heaviside function.

Как известно, фредгольмовости граничной задачи для уравнения Коши-Римана посвящены многочисленные работы [1-6]. Целью данной работы является нахождение аналитического вида решения граничной задачи. Основное место занимает найденное в работе регулярное необходимое условие, которое получается при помощи фундаментального по направлению решения уравнения Коши-Римана.

Это фундаментальное решение отличается от фундаментальных решений, примененных в работах выше. Отметим, что в вышеприведенных работах необходимые условии имеем сингулярности, которые после регуляризации совместно с заданными граничными условиями давали достаточные условия для фредгольмовости поставленных граничных задач. В данной работе полученные необходимые условия не содержат сингулярные интегралы.

Постановка задачи. Рассмотрим следующую граничную задачу:

$$\frac{\partial u(x)}{\partial x_2} + i\frac{\partial u(x)}{\partial x_1} = 0,\ x \in D = \{X = (x_1, x_2) : x_1 \in R, x_2 \in (0,1)\}, \quad (1)$$

$$u(x_1,1) = \alpha(x_1)u(x_1,0) + \varphi(x_1),\ x_1 \in R, \quad (2)$$

где $i = \sqrt{-1}$, $\alpha(x_1)$ и $\varphi(x_1)$ – непрерывные, комплекснозначные функции. $D \in R^2$ – полоса с шириной единица. D – ограниченная, выпуклая по направлению x_2 плоская область, граница $\Gamma = \partial D$ –линия Ляпунова. Если область D спроектировать на оси x_1 параллельно оси x_2 то граница Γ разбивается на две части Γ_1 и Γ_2 , уравнения которых имеют следующий вид: $x_2 = \gamma_1(x_1)$ и $x_2 = \gamma_2(x_1)$ при $x_1 \in [a_1, b_1]$.

Фундаментальное по направлению x_2 решения уравнения (1) имеет вид

$$U(x-\xi) = \theta(x_2 - \xi_2)\delta(x_1 - \xi_1 - i(x_2 - \xi_2)), \quad (3)$$

где $\theta(t)$ – единичная функция Хевисайда, $\delta(z)$ – дельта-функция Дирака с комплексными аргументами.

Основное соотношения: Умножая уравнения (1) на фундаментальное решения (3), интегрируя по области D и применяя формулу интегрирования по частям, имеем [7]:

CC BY

$$0 = \int_D dx_1 \int_0^1 \frac{\partial u(x)}{\partial x_2} U(x-\xi)dx_2 + i\int_0^1 dx_2 \int_D \frac{\partial u(x)}{\partial x_1} U(x-\xi)dx_1 = \int_D u(x)U(x-\xi)\,|_{x_2=0}^1\, dx_1 + i\int_0^1 u(x)U(x-\xi)\,|_{x_1=-\infty}^{\infty}\, dx_2 -$$
$$- \int_D dx_1 \int_0^1 u(x)\frac{\partial u(x-\xi)}{\partial x_2} dx_2 - i\int_0^1 dx_2 \int_D u(x)\frac{\partial u(x-\xi)}{\partial x_1} dx_1 \quad (4)$$

или же учитывая

$$\frac{\partial u(x-\xi)}{\partial x_2} + i\frac{\partial u(x-\xi)}{\partial x_1} = \delta(x-\xi), \quad (5)$$

из (4) получим:

$$\int_R u(x_1,1)\theta(1-\xi_2)\delta(x_1-\xi_1-i(1-\xi_2))dx_1 - \int_R u(x_1,0)\theta(1-\xi_2)\delta(x_1-\xi_1+i\xi_2)dx_1 = \begin{cases} u(\xi),\ \xi \in D, \\ \frac{1}{2}u(\xi),\ \xi_1 \in R, \end{cases} \quad (6)$$

где при получении основного соотношения (6) предполагаем, что имеет место ограничение

$$\lim_{x_1\to\infty} \int_0^1 u(x)\theta(x_2-\xi_2)\delta(x_1-\xi_1-i(x_2-\xi_2))dx_2 = \lim_{x_1\to-\infty} \int_0^1 u(x)\theta(x_2-\xi_2)\delta(x_1-\xi_1-i(x_2-\xi_2))dx_2. \quad (7)$$

Первое выражение основного соотношения (6) даёт общее решения уравнения (1) определенное в области D , а второе соотношение является необходимым условием.

Необходимые условии. Из (6) имеем следующие необходимые условия:

$$\frac{1}{2}u(\xi_1,0) = \int_R u(x_1,1)\delta(x_1-\xi_1-i)dx_1 - \frac{1}{2}u(\xi_1,0),$$

и

$$u(\xi_1,0) = (\xi + i,1). \quad (8)$$

Таким образом получено следующие утверждение.

Теорема 1. Если имеет место ограничение (7), то тогда каждая аналитическая функция, определенная на полосе D , удовлетворяет условиям (8).

Теперь учитывая необходимые условия (8) в граничном условии (2) имеем:

$$u(x_1,1) = \alpha(x_1)u(x_1+i,1) + \varphi(x_1), \quad (9)$$

К полученному уравнению (9) будем применять метод последовательных подстановок. Так как

$$u(x_1+i,1) = \varphi(x_1+i) + \alpha(x_1+i)u(x_1+2i,1),$$

то из (9) получим:

$$u(x_1,1) = \varphi(x_1) + \alpha(x_1)\left[\varphi(x_1+i) + \alpha(x_1+i)u(x_1+2i,1)\right] =$$
$$= \varphi(x_1) + \alpha(x_1)\varphi(x_1+i) + \alpha(x_1)\alpha(x_1+i)u(x_1+2i,1).$$

Если ещё раза произвести эти действия, имеем:

$$u(x_1,1) = \varphi(x_1) + \alpha(x_1)\varphi(x_1+i) + \alpha(x_1)\alpha(x_1+i)\varphi(x_1+2i) +$$
$$+ \alpha(x_1)\alpha(x_1+i)\alpha(x_1+2i)u(x_1+\xi i,1).$$

Таким образом методом математической индукции легко установить следующее соотношение:

CC BY

$$u(x_1,1)=\varphi(x_1)+\sum_{k=1}^{n}\left(\prod_{\varsigma=0}^{k-1}\alpha(x_1+\varsigma i)\right)\varphi(x_1+ki)+\left(\prod_{\varsigma=0}^{n}\alpha(x_1+\varsigma i)\right)u(x_1+(\varsigma+1)i,1) \quad (10)$$

Тогда установлено:

Теорема 2. При условиях теоремы 1, если $|\alpha(z)|\le q\le 1$, $|\varphi(z)|\le K$, то для ограниченного решения задачи (1)-(2) справедливо соотношение:

$$u(x_1,1)=\varphi(x_1)+\sum_{k=1}^{n}\left(\prod_{\varsigma=0}^{k-1}\alpha(x_1+\varsigma i)\right)\varphi(x_1+ki). \quad (11)$$

Решение граничной задачи. Возвращаемся к основному соотношению (6) для решения граничной задачи (1)-(2) и имеем:

$$u(\xi)=\int_R u(x_1,1)\delta(x_1-\xi_1-i+i\xi_2)dx_1=u(\xi_1-i\xi_2+i,1), \quad (12)$$

где $u(\xi_1-i\xi_2+i,1)$ дано в виде (11).

Таким образом получаем следующее:

Теорема 3. При условиях теоремы 2 существует решения задачи (1)-(2) имеющий вид (12). Нет решения (12)

Литература

1. Зейналов Р.М., Алиев Н.А. Задача Зарембы-Стеклова для уравнения Коши-Римана // Вестник Дагестанского государственного университета. 2015. Т. 30. № 6. С.74-79.

2. Алиев Н.А., Зейналов Р.М. Исследование решения задачи Стеклова для уравнения Коши-Римана при граничном условии, содержащем глобальный член // Известия Национальный Академии Наук Азербайджана. Серия физико-технических и математических наук. 2010. Т. 30. № 3. С. 75-80.

3. Зейналов Р.М. Задача Стеклова для уравнения Лапласа с линейными граничными условиями, содержащими интегралы // Вестник ОмГУ. 2016. № 2(80). С. 6-10.

4. Aliyev N.A, Abbasova A.H. The New Approach to Boundary Problems for Equation Cauchy-Riemann // Abstracts of International Conference on Modern problem of Applied Mathematics and Information Technologies (Tashkent, 19-21 September 2009). Uzbekistan, 2009. Pp. 28.

5. Aliyev N., Fatemi M., Jahanshahi M. Analytic solution for the Cauchy-Riemann equation with non-local boundary conditions in the first-quarter // Quarterly Journal of Science Tarbiat Moallem University. 2009. Vol. 9. Issue 1. Pp. 29-40.

6. Садыгов А.Б., Мамедли Р.Э., Зейналов Р.М. Математические модели автоматизированного управления системами противокоррозионной защиты магистральных трубопроводов // Современное программирование: Мат-лы II Международной научно-практической конференции (г. Нижневартовск, 14-16 ноября 2019 года). Нижневартовск, 2019. С. 68-75.

7. Владимиров В.С. Уравнения математической физики. М.: Наука, 1981. 512 с.

© Садыгов А.Б., Зейналов Р.М., Мамедли Р.Э., 2022

УДК 519.688
https://doi.org/10.36906/AP-2022/27

Сергеев А.М.
ORCID: 0000-0002-4788-9869, канд. техн. наук
Санкт-Петербургский государственный университет аэрокосмического приборостроения
г. Санкт-Петербург, Россия

ПОИСК БИНАРНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ФОРМИРОВАНИЯ ЦИКЛИЧЕСКИХ БЛОКОВ СИММЕТРИЧНЫХ МАТРИЦ АДАМАРА ТРЁХБЛОЧНОЙ КОНСТРУКЦИИ БАЛОНИНА-СЕБЕРРИ

Аннотация. Обсуждается проблема вычисления трёхблочных симметричных матриц Адамара конструкции Балонина-Себерри больших порядков. Рассматриваются вопросы генерации и обработки бинарных последовательностей, формирующих циклические блоки для построения таких матриц. Практический результат работы заключается в описании подхода в работе со сверхбольшими последовательностями как основой подбора вариантов комбинаций первых строк циклических блоков, порождающих матрицу Адамара. Описанный подход может быть использован для последующего улучшения алгоритмов поиска.

Ключевые слова: симметричная матрица Адамара; трёхблочная конструкция Балонина-Себерри; бинарные последовательности; циклические блоки.

Sergeev A.M.
ORCID: 0000-0002-4788-9869, Ph.D.
Saint-Petersburg State University of Aerospace Instrumentation
Saint Petersburg, Russia

SEARCH FOR BINARY SEQUENCES FOR FORMING CYCLIC BLOCKS OF SYMMETRIC HADAMARD MATRICES OF THREE-BLOCK BALONIN-SEBERRY CONSTRUCTION

Abstract. The problem of calculating three-block symmetric Hadamard matrices of the Balonin-Seberry design of large orders is discussed. The issues of generation and processing of binary sequences forming cyclic blocks for constructing such matrices are considered. The practical result of the work is to describe an approach to working with super-large sequences as the basis for selecting invariants of combinations of the first rows of cyclic blocks generating the Hadamard matrix. The described approach can be used to further improve search algorithms.

Keywords: symmetric Hadamard matrices; three-block Balonin-Seberry construction; binary sequence; cyclic blocks.

CC BY

Вычисление ортогональных матриц Адамара больших порядков (более 200) с элементами 1 и -1, является вычислительно сложной задачей, решаемой сегодня на суперкомпьютерах во всемирно известных научных центрах. Прогресс в этой области компьютерных вычислений достигается различными путями: разработкой принципиально новых методов, например, основанных на оптимизационных процедурах [6], или совершенствованием известных методов за счёт адаптации их под конкретные структуры вычисляемых матриц [4; 5; 7; 9; 10].

Одними из наиболее сложных структур вычисляемых матриц Адамара являются симметричные структуры, успешно применяемые в обработке информации. Одной из стабильных на порядках симметричных структур является трёхблочная конструкция Балонина-Себерри [8], являющаяся модификацией массива Вильямсона [2]:

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \\ \mathbf{C} & -\mathbf{D} & -\mathbf{A} & \mathbf{B} \\ \mathbf{B} & -\mathbf{A} & \mathbf{D} & -\mathbf{C} \\ \mathbf{D} & \mathbf{C} & -\mathbf{B} & -\mathbf{A} \end{pmatrix}.$$

Массив Вильямсона допускает перевод к симметричному виду Балонина-Себерри перестановкой средних строк, смены знаков, при условии равенства пары блоков **B** = **C** при известных ограничениях [2; 11], сводящихся к виду:

$$\mathbf{A}^2 + 2\mathbf{B}^2 + \mathbf{C}^2 = 4m\mathbf{I},$$

где m – порядок блока.

Блоки не обязательно ортогональны, но имеют циклическую структуру [11]. При такой стабильной конструкции основную вычислительную трудность представляет поиск бинарных последовательностей из 1 и -1, как основы блоков **A**, **B** и **D** порядка $m = n/4$, образующих **H** порядка n.

Для указанных блоков, имеющих циклическую структуру, бинарные последовательности являются первыми строками, смещаемыми вправо. Цикличность блоков позволяет сократить объём памяти при хранении матрицы **H** и вычислительные затраты, сохраняя возможность её быстрого воссоздания.

На рисунке 1 для примера представлен портрет симметричной матрицы Адамара конструкции Балонина-Себерри не большого порядка 28, где поле тёмного цвета соответствует -1, а белого цвета – единице.

Фиксация структуры симметричной матрицы Адамара в виде конструкции Балонина-Себерри позволяет искать варианты комбинаций первых строк, порождающих три циклических блока **A**, **B** и **D** вместо четырёх. Однако, сложность их поиска с ростом порядка n существенно возрастает, делая необходимым совершенствование алгоритмов для принципиальной возможности их программной реализации на персональных компьютерах.

CC BY

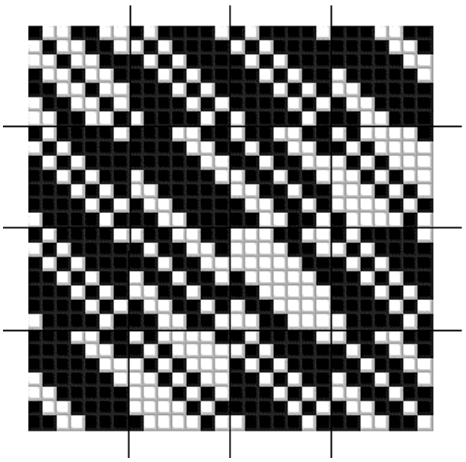


Рис. 1. Симметричная матрица Адамара конструкции Балонина-Себерри с выделенными блоками

Алгоритм условно состоит из трёх подзадач:

– генерация последовательностей;

– фильтрация последовательностей с целью отбора предпочтительных;

– поиск совместимых блоков, формирующих матрицу **H** конструкции Балонина-Себерри искомого порядка, на основе анализа отобранных последовательностей.

Каждая из указанных подзадач имеет несколько вариантов реализации. Сочетание различных подходов позволяет получать алгоритмы с определённой эффективностью.

При раздельной генерации последовательностей для формирования первых строк блоков **A**, **D** и **B** необходимо их хранение в оперативной памяти. При совпадении логических адресов (PAF-адресов) **A** и **D** формируется пара с новым адресом PAF(**A**)+PAF(**D**), размещается в отдельном каталоге [1; 7; 11]. Если для пары **A**, **D** и **B** выполняется PAF(**A**,**D**) = -2PAF(**B**), то синтезируется троица блоков **A**, **B** и **D**.

Такой подход не приводит к дублированию информации, поскольку хранятся только исходные последовательности, а пары формируются через ссылки. Поскольку блоки **A** и **D** не всегда образуют пару, а для одной **A** может подходить несколько различных **D**, поэтому для исключения потери пар сохраняются оба накопленных исходных наборов. При этом PAF-адрес служит фильтром, исключающим не образующие пар **A** и **D**.

Например, поиск матрицы Адамара конструкции Балонина-Себерри порядка 156 дал несколько вариантов решения при найденных только 10000 пар. Около 10000000 пар были отфильтрованы на этапе формирования каталога. 1000000 вариантов **B** были сравнены с найденными парами **A** и **D**.

Дальнейшее совершенствование алгоритма поиска заключается в применении кодирования фрагментов последовательностей целыми числами. Например, число 9 (1001) соответствует сочетанию 1-1-11 в последовательности. Это позволяет существенно сократить объем использованной памяти для хранения при программной реализации и затраты на вычисление PAF, однако требует специальных генераторов, оперирующих наборами битов.

У каждого типа генерации есть определённые преимущества и недостатки [3], определяющие скорость формирования групп из 1 и -1 и их уникальность. Временем поиска

CC BY

можно варьировать, не меняя алгоритм, а лишь заменив генератор на основе прецедента удачного накопления последовательностей, прямо влияющего на разрешимость задачи поиска симметричных матриц Адамара конструкции Балонина-Себерри.

Последовательности как основы блока **B** имеют малую ценность, поэтому их нужно хранить с возможностью замены с минимальными вычислительными затратами. Оптимальным выглядит вариант с полной заменой последовательности, если в ней не нашлось подходящей для образования **B**.

Последовательности для блоков **A** и **D** по отдельности также имеют малую ценность, но только до объединения в пару. При этом, как говорилось выше, к последовательности **A** может найтись несколько **D**, поэтому при реализации алгоритма лучше осуществлять обновление.

Найденные пары **A**, **D** – половина решения. Они наиболее ценны, их требуется сохранить. При этом в случае появления новой пары, можно освободить место, удалив одну из тех пар, что не привели к решению.

В заключении следует отметить, что поиск ортогональных матриц Адамара больших порядков является сложным, а реализующие его программы – длительными по времени выполнения на самых современных компьютерах.

Поиск симметричных матриц Адамара, основанный на использовании математических закономерностей, характерных для конструкции Балонина-Себерри, значительно сокращает временные затраты, поскольку ищутся варианты совместимых трёх блоков, а не четырёх, как в конструкции Вильямсона.

Эффективность алгоритмов зависит от реализации трёх ключевых подзадач – генератора, фильтра и поиска совместимых блоков. Разделение на подзадачи добавляет гибкости при дальнейшей модификации алгоритмов, позволяет проводить более качественный и детальный анализ всех этапов работы и их результатов.

Литература

1. Абузин Л.В., Балонин Ю.Н., Куртяник Д.В., Сергеев А.М. Генерация, фильтрация и поиск экстремума в сверхбольшом каталоге бинарных последовательностей // Обработка, передача и защита информации в компьютерных системах: Первая Всероссийская научная конференция (г. Санкт-Петербург, 14-22 апреля 2020 года). СПб., 2020. С. 121-124.

2. Acevedo S., Dietrich H. New infinite families of Williamson Hadamard matrices // Australian Journal of Combinatorics. 2019. Vol. 73(1). P. 207-219.

3. Balonin Y., Abuzin L., Sergeev A., Nenashev V. The Study of Generators of Orthogonal Pseudo-Random Sequences // Intelligent Decision Technologies. Smart Innovation, Systems and Technologies. 2019. Vol. 143. Singapore: Springer. P.125-133. https://doi.org/10.1007/978-981-13-8303-8_11

4. Балонин Н.А., Балонин Ю.Н., Востриков А.А., Сергеев М.Б. Вычисление матриц Мерсенна-Уолша // Вестник компьютерных и информационных технологий. 2014. № 11(125). С. 51-55.

CC BY

5. Балонин Н.А., Балонин Ю.Н., Сергеев М.Б. О модификации метода Скарпи вычисления матриц Мерсенна для задач преобразования изображений // Информационные технологии. 2014. № 4. С. 48-51.

6. Балонин Н.А., Сергеев М.Б., Суздаль В.С. Динамические генераторы квазиортогональных матриц семейства Адамара // Труды СПИИРАН. 2017. № 5(54). С. 224-243.

7. Balonin N.A., Balonin Y.N., Djokovic D., Karbovskiy D.A., Sergeev M.B. Construction of Symmetric Hadamard Matrices // Information and Control Systems. 2017. № 5. Pp. 2-11. https://doi.org/10.15217/issn1684-8853.2017.5.2

8. Balonin N.A., Seberry J. The Propus Construction for Symmetric Hadamard Matrices. 29 November 2015. https://arxiv.org/pdf/1512.01732.pdf

9. Bulutoglu D., Kotsireas I., Koutschan C., TurnerJ. Legendre pairs of lengths l = 0 (mod 5). https://doi.org/10.48550/arXiv.2111.02105

10. Востриков А.А. Матричные витражи и регулярные матрицы Адамара // Информационно-управляющие системы. 2021. №5. С. 2-9. https://doi.org/10.31799/1684-8853-2021-5-2-9

11. Seberry J., Balonin N.A. Two infinite families of symmetric Hadamard matrices // Australian Journal of Combinatorics 2017. Vol. 69(3). Pp. 349-357.

© Сергеев А.М., 2022

УДК 519.687.7
https://doi.org/10.36906/AP-2022/28

Чертовской В.Д.
ORCID: 0000-0002-87-40-6702
д-р техн. наук
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И Ульянова (Ленина)
г. Санкт-Петербург, Россия

АДАПТИВНЫЙ ПЕРЕХОД НА ВЫПУСК ПРОИЗВОДСТВОМ НОВОЙ ПРОДУКЦИИ

Аннотация. Отмечено, что в производствах в последнее время все шире применяется процедура оперативного перехода на выпуск новой продукции. Отмечено, что для математического описания процедуры может применяться аппарат как статического, так и динамического линейного программирования. Показано использование для названной цели статического линейного программирования.

Ключевые слова: производство; новая продукция; алаптивный переход; линейное программирование.

Chetrtovskoi V.D.
ORCID: 0000-0002-87-40-6702 Ph.D.
Saint Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia

ADAPTIVE TRANSITION TO PRODUCTION OF NEW PRODUCTS

Abstract. It is noted that in manufacturing in recent years, the procedure for the operational transition to the release of new products has been increasingly used. It is noted that the apparatus of both static and dynamic linear programming can be used for the mathematical description of the procedure. Shows the use for the named purpose of static linear programming.

Keywords: manufacturing; new products; adaptnive transition; linear programming.

Настоящая работа является продолжением публикации [1]. В ней отмечено, что для описания в многоуровневой производственной системы процесса управления возможно применение аппарата статического линейного программирования [2]. Одним из способов управления служит оперативный переход на выпуск новой продукции, математическому описанию которого посвящена настоящая работа.

Постановка задачи. Рассматривается трёхуровневая систем руководитель (уровень h=3) – диспетчер (уровень h=2) - начальники цехов (уровень h =1).

CC BY

Описание по уровням имеет следующий вид.

Уровень $h = 1$.

$$\sum_{i=0}^{N-1} \mathbf{D}_1^m \, \mathbf{p}_k(t_i) \le \mathbf{b}^m(0),$$

$$\sum_{i=0}^{N-1} \mathbf{p}_K(t_i) \le \mathbf{P}(T),$$

$$\mathbf{D}_k{}^{\psi}\mathbf{p}_k(t_{i+1}) \le \mathbf{b}_k{}^{\psi}(t_i),$$

$$\mathbf{D}_k{}^m\mathbf{p}_k(t_{i+1}) \le \mathbf{b}_k{}^m(t_i), \qquad (1)$$

$$G_k = \mathbf{F}_k\mathbf{P}_k(T) \rightarrow \max, \qquad (2)$$

$$i = 0, N\ \ 1,\ t_i = iv,\ t_0 = 0,\ T = Nv,$$

где **z**, **p** – вектор-столбцы (планового) незавершённого производства и ежедневного плана, **R** – вектор-столбец спроса; **D** – матрица норм расходов ресурсов; **b** – вектор-столбец наличного количества ресурсов; $\mathbf{b}^m(0)$ – вектор количества материальных ресурсов, которыми располагает уровень $h = 3$; $\Delta\mathbf{b}$ - поступление ресурсов; **P** – вектор-столбец плана уровня $h = 3$; **F** – вектор-строка прибыли от выпуска единицы продукции; **A**, **B**, **C** единичная матрицы соответствующих размерностей; v, T – минимальный интервал времени и время моделирования; $m = 1, M$ – виды материальных ресурсов; $\psi = 1, \Psi$ - виды прочих ресурсов; $i = 1, I$ – моменты времени; $k = 1, k$ – номер подразделения.

Уровень $h = 2$.

Выражения (1), (2) заменяются на выражения (3), (4)

$$\mathbf{D}_k{}^m\mathbf{p}_k(t_{i+1}) \le \mathbf{p}_{k\text{-}1}(t_i) \qquad (3)$$

$$G = \sum_{i=1}^{K} G_k \rightarrow max, \qquad (4)$$

Уровень $h = 3$.

$$\mathbf{P}(T) \ge \mathbf{R}(T),\ \mathbf{P}(t_i) = \mathbf{P}(t_{i\text{-}1}) - \mathbf{p}(t_i),$$

$$\mathbf{Dp}(t_i) \le \mathbf{b}(t_{i\text{-}1}),$$

$$G = \mathbf{FP}(T) > \rightarrow \max,$$

где **z**, **p**, **P** – незавершённое производство, планы текущий и с накоплением; **R** – спрос; **D** – матрица норм расходов; **b** – наличное количество ресурсов; **F** – прибыль от выпуска единицы продукции, T, v – интервалы времени; $T = Nv$.

При традиционном режиме может изменяться спрос на старую, выпускаемую продукцию.

$$R_c(t) = R^c_c(t) + \Delta R_c * 1(t - \theta),$$

где R_c – вектор спроса на старую продукцию; ΔR_c – изменения вектора.

При переходе на новую продукцию имеет место спрос

$$R_н(t) = \Delta R_н * 1(t - \theta),$$

где $\Delta R_н$ – вектор спроса на новую продукцию.

CC BY

Схема перехода на выпуск новой продукции. Все виды продукции (план выпуска) делятся на 4 группы P_1 - P_4. Виды продукции $P_4[\tau]$ и $P_3[\tau]$, $P_3[\tau]$ и $P_2[\tau]$ имеют (попарно) общие ресурсы У них нет общих ресурсов с продукцией $P_{1k}[\tau]$. Изменяется P_4, и, как следствие, P_3 и P_2, а P_1 остаётся без изменения и потому далее не рассматривается.

Выпуск новой продукции может сопровождаться снижением спроса на старую продукцию с величины R_c до величины R'_c.

Рассмотрим *k*-тый элемент уровня *h*=1, поскольку описание для остальных элементов будет идентичным.

Возможны следующие варианты запуска новой и снятия старой продукции.

1. Продукция P_{ck} снимается постепенно за время $T_c = \tau_c = \max \tau_c$, где $t_c = \{\tau_{cj}, j \in J_3 \subset J$;

J, J_3 – множества выпускавшейся ранее, τ_{ck} – вектор постоянных времени, а для новой продукции используется:

1) последовательный непрерывный способ запуска (только после снятия старой продукции) с мгновенным выпуском;

2) последовательный непрерывный способ с выпуском продукции через время T_c.

3) последовательный прерывный способ;

4) параллельный способ запуска (в процессе снятия старой продукции);

5) параллельный способ запуска (в процессе снятия старой продукции);

II. Старая продукция P_{ck} снимается мгновенно в момент времени (t) = (τ – 1):

1) мгновенно производится новая продукция $P_н[\tau] \geq R_c$;

2) начинается выпуск новой продукции с временем $T_п$ выхода на устойчивый уровень.

Нетрудно видеть, что в вариантах I.2, I.3, I.5 приходится учитывать динамику самого процесса планирования.

На выходе объекта управления фиксируется готовая продукция j (j = 1, J), на входе имеют место соответствующие комплекты ресурсов, готовые для. запуска в производство. Полагаем, что комплектование выполнено до начала процесса производства.

В этом режиме спрос $\mathbf{R}(t) = \{\mathbf{R}_c^T(t), \mathbf{R}_н^T(t)\}^T$, где $\mathbf{R}_4(t)$ – спрос на новую продукцию; ^{T} – символ транспонирования. Пусть в момент времени (t) = (τ 1) возникает необходимость в оперативном переходе на выпуск новой продукции $\mathbf{P}_н[\tau] = \{P_{нj}[\tau], j \in 1, J_4\}$. При этом старая продукция $\mathbf{P}_c[\tau]$ из $\mathbf{P}[\tau]$ снимается с производства полностью ($\mathbf{P}'_c[\tau] = 0$) или частично ($\mathbf{P}'_c[\tau] < \mathbf{P}_c[\tau]$).

Отметим, что, не снижая общности, можно считать (t) =(τ 1) = 0. В противном случае можно пересчитать начальные условия.

Имеют место При переходе на выпуск новой продукции возможны два случая:

а) матрица ограничений **D** заполнена полностью;

б) матрица **D** разрежена, что позволяет уменьшить размерность решаемой задачи.

Обратимся к векторам P_1 - P_4 рассмотрим случай *а*).

Ограничения старого плана имеют вид

$$\mathbf{D}_1\mathbf{P}_3[\tau] \leq \mathbf{b}. \qquad (5)$$

CC BY

С появлением новой продукции описание получает вид

$$(\mathbf{D}_1\ \mathbf{D}_2)\begin{pmatrix}\mathbf{P}_3[\tau]\\ \mathbf{P}_4[\tau]\end{pmatrix} \le \mathbf{b}(0), \qquad (6)$$

если новые ресурсы не требуются и

$$\begin{pmatrix}\mathbf{D}_1 & \mathbf{D}_2\\ 0 & \mathbf{D}_3\end{pmatrix}\begin{pmatrix}\mathbf{P}_3[\tau]\\ \mathbf{P}_4[\tau]\end{pmatrix} \le \begin{pmatrix}\mathbf{b}(0)\\ \mathbf{b}_1(0)\end{pmatrix}. \qquad (7)$$

Вычитая из (6) равенство (5) с величиной $\mathbf{b}^*$, получим

$$\mathbf{D}_1(\mathbf{P}'_3[\tau] - \mathbf{P}_3[\tau]) - \mathbf{D}_2\mathbf{P}_4[\tau] \le \mathbf{b}^* - \mathbf{b} \qquad (8)$$

Переход целесообразен, если

$$\mathbf{F}_1(\mathbf{P}'_3[\tau] - \mathbf{P}_3[\tau]) + \mathbf{F}_2\mathbf{P}_4[\tau] \ge 0.$$

В случае (7) к ограничению (8) добавляется

$$\mathbf{D}_3\mathbf{P}_4[\tau] \le \mathbf{b}_1(0),$$

а целевая функция

$$G = \mathbf{F}_1\mathbf{P}'_3[\tau] + \mathbf{F}_2\mathbf{P}_4[\tau] \rightarrow \max.$$

Перейдём случай *б*).

Введём новые обозначения.

Пусть в выражении новая продукция $\mathbf{P}_4[\tau]$ и часть старой продукции $\mathbf{P}_3[\tau]$, части старой продукции $\mathbf{P}_3[\tau]$ и $\mathbf{P}_2[\tau]$ имеют попарно общие ресурсы и не имеют общих ресурсов с $\mathbf{P}_1[\tau]$.

Очевидно, что старый план $\mathbf{P}^c[\tau] = \{\mathbf{P}_1^m[\tau], \mathbf{P}_2^m[\tau], \mathbf{P}_3^m[\tau]\}$, $|\mathbf{P}^c| = |\mathbf{P}_1| + |\mathbf{P}_2| + |\mathbf{P}_3|$, где $|.|$ – размерность вектора, $\mathbf{b}^{cT}(\tau\ \ 1) = \{\mathbf{b}_1^m(\tau\ \ 1), \mathbf{b}_2^m(\tau\ 1), \mathbf{b}_3^m(\tau\ \ 1)\}^T$, $\mathbf{F}^c = \{\mathbf{F}_i, i = 1, 3\}$.

Для выпуска новой продукции выделяются дополнительные ресурсы $\mathbf{b}_4(\tau\ \ 1)$, $\mathbf{b}^1{}_3(\tau\ \ 1)$, $\mathbf{b}^1{}_2(\tau\ \ 1)$. Предполагаем, что наличие ресурсов $\{\mathbf{b}_2(\tau\ \ 1) + \mathbf{b}'_2(\tau\ \ 1)\}$, $\{\mathbf{b}_3(\tau\ \ 1) + \mathbf{b}'_3(\tau\ \ 1)\}$, $\mathbf{b}_4(\tau\ \ 1)$ обеспечивает выполнение нового плана $\mathbf{P}^{нT}[\tau] = \{\mathbf{P}_1{}^T[\tau], \mathbf{P}_2{}^T[\tau], \mathbf{P}_3{}^T[\tau], \mathbf{P}_4{}^T[\tau]^T$ при $\mathbf{P}_4[\tau] \ge \mathbf{R}_4[\tau]$. Иначе следует решить вопрос ресурсного обеспечения.

Тогда новый план связан с ограничениями

$$\begin{pmatrix}\mathbf{D}_1 & 0 & 0 & 0\\ 0 & \mathbf{D}_2 & \mathbf{D}_6 & 0\\ 0 & 0 & \mathbf{D}_9 & \mathbf{D}_{11}\\ 0 & 0 & 0 & \mathbf{D}_{12}\end{pmatrix}\begin{pmatrix}\mathbf{P}_1[\tau]\\ \mathbf{P}_2[\tau]\\ \mathbf{P}_3[\tau]\\ \mathbf{P}_4[\tau]\end{pmatrix} \le \begin{pmatrix}\mathbf{b}_1(\tau-1)\\ \mathbf{b}_2(\tau-1) - \mathbf{b}''_2(\tau-1)\\ \mathbf{b}_3(\tau-1) - \mathbf{b}''_3(\tau-1)\\ \mathbf{b}_4(\tau-1)\end{pmatrix} \qquad (9)$$

где $\mathbf{D}_1$, $\mathbf{D}_2$,$\mathbf{D}_6$, $\mathbf{D}_9$, $\mathbf{D}_{11}$, $\mathbf{D}_{12}$ – подматрицы соответствующей размерности матрицы $\mathbf{D}$ норм расхода ресурсов.

Старый план был рассчитан до появления новой продукции и потому в нем, в предположении, что истрачены все ресурсы, неравенство можно заменить на равенство

$$\begin{pmatrix}\mathbf{D}_1 & 0 & 0\\ 0 & \mathbf{D}_2 & \mathbf{D}_6\\ 0 & 0 & \mathbf{D}_9\end{pmatrix}\begin{pmatrix}\mathbf{P}_2[\tau]\\ \mathbf{P}_2[\tau]\\ \mathbf{P}_3[\tau]\end{pmatrix} = \begin{pmatrix}\mathbf{b}_1(\tau-1)\\ \mathbf{b}_2(\tau-1)\\ \mathbf{b}_3(\tau-1)\end{pmatrix}. \qquad (10)$$

Основная идея формирования модели нового плана – запись в отклонениях (11) нового плана (9) от известного старого плана (10), что позволяет получить результат для нового плана с неизвестными значениями $\mathbf{P}_4[\tau]$ и $\mathbf{P}_3'[\tau]$

$$\mathbf{P}_4[\tau] <= \mathbf{R}_4[\tau],$$

CC BY

$$\begin{pmatrix} \mathbf{D}_6 & 0 \\ \mathbf{D}_9 & \mathbf{D}_{11} \\ 0 & \mathbf{D}_{12} \end{pmatrix} \begin{pmatrix} \mathbf{P}_3''[\tau] - \mathbf{P}_3[\tau] \\ \mathbf{P}_4[\tau] \end{pmatrix} \le \begin{pmatrix} \mathbf{b}_2''(\tau - 1) \\ \mathbf{b}_3''(\tau - 1) \\ \mathbf{b}_4(\tau - 1) \end{pmatrix} \qquad (11)$$

где $\mathbf{D}_6$, $\mathbf{D}_9$, $\mathbf{D}_{11}$, $\mathbf{D}_{12}$ – подматрицы соответствующей размерности матрицы $\mathbf{D}$ норм расхода ресурсов.

Поскольку часто $|\mathbf{P}_4| << |\mathbf{P}|$, $|\mathbf{P}_3| << |\mathbf{P}|$, то возможно существенное снижение размерности (а следовательно, и времени расчёта) задачи (11) по сравнению с исходной задачей.

Если значение $\mathbf{P}'_3[\tau]$ задано и фиксировано, целевая функция получает вид

$$G_4 = \mathbf{F}_4\mathbf{P}_4[\tau] \rightarrow \max.$$

Заметим, что приведённые выражения позволяют определить экономические условия, при которых выгодно переходить на выпуск новой продукции.

Расчёт в адаптивном (нестационарном) режиме можно, таким образом, свести к эквивалентному расчёту в стационарном режиме. Правда, задача динамического линейного программирования даёт возможность точнее учитывать экономические характеристики.

Забегая вперёд, отметим, что для уровня $h = 2$ выражение (11) получает вид

$$\begin{pmatrix} \mathbf{D}_{6k} & 0 \\ \mathbf{D}_{9k} & \mathbf{D}_{11k} \\ 0 & \mathbf{D}_{12k} \end{pmatrix} \begin{pmatrix} \mathbf{P}_{3k}''[\tau] - \mathbf{P}_{3k}[\tau] \\ \mathbf{P}_{4k}[\tau] \end{pmatrix} \le \begin{pmatrix} 0 \\ \mathbf{P}_{3k-1}''[\tau] - \mathbf{P}_{3k-1}[\tau] \\ \mathbf{P}_{4k-1}[\tau] \end{pmatrix}$$

Стационарный процесс планирования начинается после завершения нестационарного переходного процесса планирования.

Описанный приём годится и для исследования стационарного планирования. При количественном изменении спроса $\mathbf{R}_3'$ получим выражения

$$\begin{pmatrix} \mathbf{D}_2 & \mathbf{D}_6 \\ 0 & \mathbf{D}_9 \end{pmatrix} \begin{pmatrix} \mathbf{P}_2''[\tau] - \mathbf{P}_2[\tau] \\ \mathbf{P}_3''[\tau] - \mathbf{P}_3[\tau] \end{pmatrix} \le \begin{pmatrix} \mathbf{b}_2''(\tau - 1) \\ \mathbf{b}_3''(\tau - 1) \end{pmatrix}$$

$$(\mathbf{F}_2 \ \mathbf{F}_3) \begin{pmatrix} \mathbf{P}_2''[\tau] - \mathbf{P}_2[\tau] \\ \mathbf{P}_3''[\tau] - \mathbf{P}_3[\tau] \end{pmatrix} \rightarrow max$$

Аналогичное выражение получается при изменениях ресурсного обеспечения $\Delta\mathbf{b}_3$.

При изменениях цены $\Delta\,\mathbf{F}_3$ получится

$$\begin{pmatrix} \mathbf{D}_2 & \mathbf{D}_6 \\ 0 & \mathbf{D}_9 \end{pmatrix} \begin{pmatrix} \mathbf{P}_2''[\tau] - \mathbf{P}_2[\tau] \\ \mathbf{P}_3''[\tau] - \mathbf{P}_3[\tau] \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

$$(\mathbf{F}_2 \ \mathbf{F}_3 + \Delta\mathbf{F}_3) \begin{pmatrix} \mathbf{P}_2''[\tau] - \mathbf{P}_2[\tau] \\ \mathbf{P}_3''[\tau] - \mathbf{P}_3[\tau] \end{pmatrix} \rightarrow max.$$

Заключение. Представлено математическое описание процедуры оптимального перехода производства на выпуск новой продукции. Замечено, что данный аппарат может быть использован и для описания процесса управления в традиционной (стационарной) процедуре. Описание послужило основой для программирования.

Литература

1. Чертовской В.Д. Процедура программирования модели многоуровневой адаптивной системы управления производством // Современное программирование: Мат-лы III

Международной научно-практической конференции (г. Нижневартовск, 26-27 ноября 2020 года). Нижневартовск, 2021. С. 140-142. https://doi.org/10.36906/AP-2020/24

2. Чертовской В.Д. Моделирование процессов адаптивного автоматизированного управления производством. СПб: Лань. 2019. 200 с.

© Чертовской В.Д., 2022

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ПРИКЛАДНЫХ НАУКАХ (ФИЗИКА, ХИМИЯ, БИОЛОГИЯ И ДР.)

УДК 536.46
https://doi.org/10.36906/AP-2022/29

Кабилов М.М.
ORCID: 0000-0002-3157-6876, канд. физ.-мат. наук
Халимов И.И.
ORCID: 0000-0001-7713-2579, канд. физ.-мат. наук
Холов О.А.
ORCID: 0000-0001-6602-5196
Российско-Таджикский (Славянский) университет
г. Душанбе, Таджикистан
Баротов А.С.
ORCID: 0000-0003-2661-9576
Дангаринский государственный университет
г. Дангара, Таджикистан

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ СРАВНЕНИЯ ОСНОВНЫХ ПАРАМЕТРОВ ФИЛЬТРАЦИОННОГО ГОРЕНИЯ ПРОПАНО-ВОЗДУШНОЙ СМЕСИ

Аннотация. В статье приводятся результаты сравнения скорости волны и равновесной температуры фильтрационного горения пропано-воздушной смеси, рассчитанные по соотношениям и с помощью созданного программного комплекса. Соотношения, по которым определяются скорость волны, соответствуют некоторым состояниям волны горения и получены в результате учёта или несущественности каких-то процессов. Созданный программный комплекс является очень удобным средством для проведения серии вычислительных экспериментов, что не реально в лабораторных условиях и даёт возможность анализировать исследуемые параметры. Программный комплекс позволяет нарисовать кривые зависимости скорости волны, равновесной температуры и их относительные погрешности от скорости вдува смеси в пористый блок при вариации состава смеси. Анализ кривых перечисленных зависимостей показывает, что основные параметры волны являются возрастающими функциями скорости вдува смеси.

Ключевые слова: температура; концентрация; горение; пропан; скорость волны; пористая среда; диффузия; программа; погрешность; сравнения.

Kabilov M.M.
ORCID: 0000-0002-3157-6876, Ph.D.

CC BY

Khalimov I.I.
ORCID: 0000-0001-7713-2579, Ph.D.
Kholov O.A.
ORCID: 0000-0001-6602-5196
Russian-Tajik (Slavonic) University
Dushanbe, Tajikistan
Barotov A.S.
ORCID: 0000-0003-2661-9576
Dangara State University
Dangara, Tajikistan

SOFTWARE COMPLEX FOR COMPARING BASIC PARAMETERS OF FILTERING COMBUSTION OF PROPANE-AIR MIXTURE

Abstract. The article presents the results of comparing the wave velocity and the equilibrium temperature of filtration combustion of a propane-air mixture, calculated from the ratios and using the created software package. The ratios used to determine the wave speed correspond to certain states of the combustion wave and are obtained as a result of taking into account or insignificance of some processes. The created software package is a very convenient tool for carrying out a series of computational experiments, which is not realistic in laboratory conditions and makes it possible to analyze the parameters under study. The software package allows you to draw curves of the dependence of the wave speed, equilibrium temperature and their relative errors on the rate of injection of the mixture into the porous block when varying the composition of the mixture. An analysis of the curves of the above dependences shows that the main parameters of the wave are increasing functions of the mixture blowing speed.

Keywords: temperature; concentration; combustion; propane; wave speed; porous medium; diffusion; program; error; comparisons.

Разработка программных комплексов в научной области является перспективным направлением, поскольку сбор, хранение, обработка, использование, наглядное представление и анализ информации посредством современной информационной технологии выполняется с высокой точностью и быстро. Разработка программного обеспечения для научных исследований, имеет свою специфику, и оно создаётся силами небольших исследовательских групп и ими же используется [1, с. 59; 2, с. 52; 3, с. 164].

Настоящая работа посвящена представлению разработанного комплекса программ для сравнения основных параметров фильтрационного горения пропановоздушной смеси в инертной пористой среде рассчитанные по различным соотношениям. Соотношения, по которым определяются скорость волны, соответствуют некоторым состояниям волны горения и получены в результате учёта или несущественности каких-то процессов. Созданный программный комплекс является очень удобным средством для проведения серии

CC BY

вычислительных экспериментов, что не реально в лабораторных условиях по многим причинам и даёт возможность анализировать исследуемые параметры. Программный комплекс позволяет нарисовать кривые зависимости скорости волны, равновесной температуры и их относительные погрешности от скорости вдува смеси в пористый блок при вариации состава смеси.

Воспользовавшись программным комплексом, выясняем отклонения основных параметров волны, рассчитанные по нижеприведённым соотношениям, то есть насколько влияют те или процессы на скорость распространения и температуры фронта горения. Следует заметить, что все рассматриваемые соотношения для скорости волны получены из однотемпературной модели фильтрационного горения газов. Эта модель соответствует случаю интенсивного межфазного теплообмена, то есть тепло выделяющееся в результате химической реакции в газе мгновенно передаётся твёрдой фазе. Соотношение (1) получено при дополнительном условии несущественности диффузии недостающего компонента [4, с. 445]:

$$(u-\upsilon_{10})^2=\frac{k_0\lambda T_0 RT_e}{\rho_{10}Q\eta_0 E}\exp(-E/RT_e) \qquad (1)$$

$$T_e=T_0+\frac{Q\eta_0}{c_p(1-\rho_2c_2u/\rho_{10}c_p(\upsilon_{10}-u))}$$

В [5, с. 39] соотношение (2) для скорости волны горения получено при симметричности профилей температуры пористой среды и концентрации реагирующего компонента газовой смеси, где обобщённое число Льюиса равно единице ($u_\varphi Le=\rho_{1*}c_pD_*/\lambda=1,\ \lambda=\alpha_1\lambda_1+\alpha_2\lambda_2$), то есть при равнозначности процессов диффузии и теплопроводности

$$(\upsilon_{10}+u)^2=k_0\exp\left(-\frac{1}{\beta}\right)\frac{\gamma^2\lambda_1\Lambda}{\rho_{10}^0c_pu_\varphi}\cdot\frac{T_0}{T_e}\cdot\frac{1}{1+\gamma} \qquad (2)$$

$$\gamma=\frac{RT_e^2}{E(T_e-T_0)},\quad T_e=T_0+\frac{Q\eta_0}{c_p(1+\rho_2c_2u/\rho_{10}c_p(\upsilon_{10}+u))}$$

В [6, с. 138] соотношение (3) скорости волны получено при относительно малом значение обобщённого числа Льюиса в сравнении с единицей

$$(u+\upsilon_{10})^2=\frac{2\left(J\frac{\lambda}{c_p}\right)_*}{\rho_{10}{}^2u_\varphi}. \qquad (3)$$

где

$$u_\varphi=1+\frac{\rho_2c_2u}{\rho_{10}(u+\upsilon_{10})c_p},\qquad J_*=\rho_{1*}n_*k_0\exp(-E/RT_*),$$

$$\rho_{1*}=\rho_{10}T_0/T_*,\ n_*=1,\quad T_*=T_0+0{,}59\cdot(T_e-T_0),\ T_e=T_0+\frac{Q\eta_0}{c_p(1+\rho_2c_2u/\rho_{10}c_p(\upsilon_{10}+u))}.$$

T_0,T_*,T_e – начальная, характерная и равновесная температуры; R – универсальная газовая постоянная; E – энергия активации; k_0 – предэкспонент; η_0 – массовая концентрация пропана

CC BY

в исходном состоянии; Q – тепловой эффект реакции; υ_{10} – скорость вдува смеси в пористый блок; u – скорость распространения волны горения; ρ_{10}, c_p – приведённая плотность и теплоёмкость смеси газов соответственно; ρ_2, c_2 – те же величины для пористой среды; λ_1, λ_2 – коэффициенты теплопроводности газа и пористой среды; α_1, α_2 – объёмные содержания газа и пористой среды.

Для нахождения значения скорости волны в зависимости от значения исходных данных используется программа подбор параметра. Подобранные значения скорости удовлетворяют соответствующие соотношения с точностью до 10^{-6} порядка. Для расчёта были использованы следующие значения физико-химических величин пористой среды и смеси газов: $\rho_1^0 = 1.163\ кг/м^3$, $\rho_2^0 = 3900\ кг/м^3$, $c_p = 1006\ м^2/(c^2 K)$, $c_2 = 800\ м^2/(c^2 K)$, $T_0 = 300\ K$, $\lambda_2 = 1 кг/(см^3 K)$, $\lambda_1 = 0.25 кг/(м^3 \cdot c \cdot K)$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $E = 116393.04\ кгм^2/(м^2 \cdot c \cdot моль)$, $Q = 48122727\ м^2/(c^2)$, $k_0 = 1.874 \cdot 10^{12} c^{-1}$, $d = 1 мм$, $R = 8.314\ кгм^2/(c^2 моль K)$, $\eta_0 = 0.0399$ (стехиометрия).

Главное окно программы содержит значения управляющих и определяющих параметров, а также расчётные характеристики волны горения и различные кнопки (**Расчёт**, **Сравнение**, **График**, **Данные**) (рис. 1)

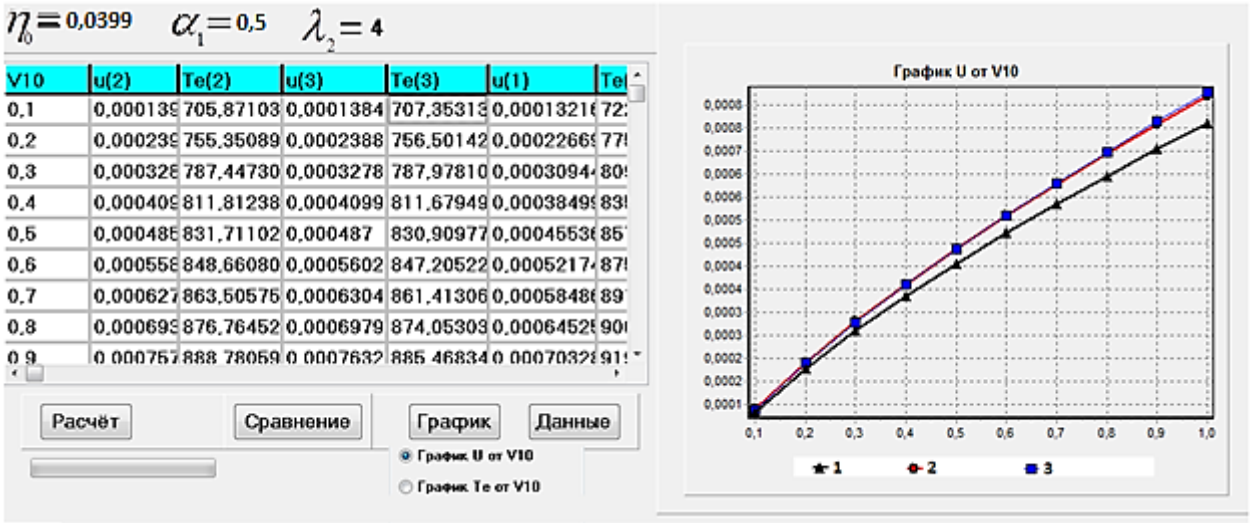


Рис. 1. Главное окно программы

На рис.1 в столбике V10 приведены значения скорости вдува в единицах, м/с. На других столбиках приведены наборы значений скорости волны горения u (м/с) и равновесной температуры T_e (K) соответствующие к значениям скорости вдува смеси и номерам соотношения в статье. С помощью элемента «**График**» выбирается зависимость: скорость волны от скорости вдува или равновесной температуры от скорости вдува, например, в правой части рисунке 1 приведены графики зависимости скорости волны от скорости вдува соответствующие трём соотношениям.

CC BY

В окне по необходимости можно изменить значения исходных параметров с помощью кнопки **«Данные»**, а также значения параметров на верхней части окна. При нажатии кнопки «**Расчёт»** происходит операции подбора значений скорости волны по трём соотношениям, приведённым в статье, далее по определённым формулам вычисляются равновесная температура и относительные погрешности соответственно. При нажатии кнопки «**Сравнение»** открывается ниспадающее меню (рис. 2), где приведены относительные погрешности равновесных температур и скорости волны соответствующие следующим названиям: **Delta(1-2), Delta(1-3), Delta(2-3)**.

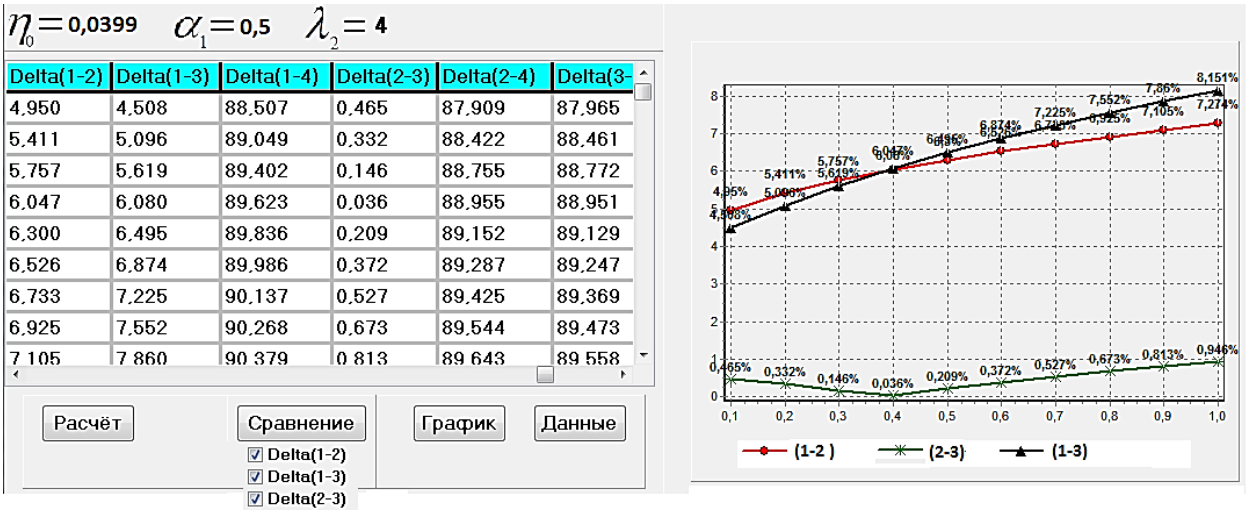


Рис. 2. Ниспадающее меню кнопки **«Сравнение»**

1. Кривые зависимости скорости волны, равновесной температуры и относительные погрешности скорости и температуры от скорости вдува предварительно перемешанной пропановоздушной смеси в пористый блок, определённые по различным соотношениям показывают, возрастание скорости волны и равновесной температуры в пределах изменения скорости вдува и незначительное влияние диффузии реагирующего компонента на скорость волны.

2. Относительно малые значения до одного процента относительной погрешности скорости волны, рассчитанные по соотношениям полученным при приближённо-аналитическом и аналитическом решении модельной задачи фильтрационного горения газа, говорят о несущественности влияния диффузии реагирующего компонента на скорость волны.

3. Созданный программный комплекс простой, удобен в эксплуатации и расширяемый для включения дополнительных соотношений скорости волны. С её помомщью можно нарисовать кривые зависимости скорости волны, равновесной температуры и относительные погрешности скорости и температуры определённые по разным соотношениям от скорости вдува.

CC BY

Литература

1. Массель Л.В., Болдырев Е.А. Моделирование и разработка современных программных комплексов для исследований энергетики // Вычислительные технологии. 2002. Т. 7. № 2. С. 59-70.

2. Юнуси М.К., Юнуси М.М., Юнуси Ф.М. Компьютерный инструмент на VB для прогнозирования численности трудовых ресурсов // Вестник Таджикского национального университета. Серия естественных наук. 2017. № 1-4. С. 52-57.

3. Халимов И.И. Пакет программ расчёта характеристик волны фильтрационного горения газовоздушной смеси // Математическое и компьютерное моделирование физических процессов: сб. статей республиканской научно-практической конференции (г. Душанбе, 25-октября 2019). Душанбе, 2019. С. 164-168.

4. Кабилов М.М., Садриддинов П.Б. Скорость распространения фронта горения смеси газов в инертной пористой среде // Доклады Академии наук Республики Таджикистан. 2009. Т. 52. № 6. С. 443-448.

5. Кабилов М.М., Гулбоев Б.Дж. Фильтрационное горение газов при симметричности профилей температуры пористой среды и концентрации компонентов газовой смеси // Доклады Академии наук Республики Таджикистан. 2013. Т. 56. № 1. С. 35-43.

6. Кабилов М.М., Холов О.А. Приближённо-аналитическое решение модельной задачи фильтрационного горения газов // Доклады Академии наук Республики Таджикистан. 2018. Т. 61. № 2. С. 134-139.

© Кабилов М.М., Халимов И.И., Холов О.А., Баротов А.С., 2022

УДК 004.4, 534.1, 625.46
https://doi.org/10.36906/AP-2022/30

Истратова Е.Е.
ORCID: 0000-0001-9759-2249, канд. техн. наук
Кожевников А.Н.
ORCID: 0000-0002-8869-7978
Штрайх А.Е., Николаева Е.Д.
Новосибирский государственный технический университет
г. Новосибирск, Россия

ОПРЕДЕЛЕНИЕ ДИНАМИЧЕСКИХ ХАРАКТЕРИСТИК ОПОР КОНТАКТНОЙ СЕТИ ГОРОДСКОГО ЭЛЕКТРИФИЦИРОВАННОГО ТРАНСПОРТА ПУТЕМ МОДЕЛИРОВАНИЯ В СРЕДЕ SOLID WORKS

Аннотация. В статье представлены результаты компьютерного моделирования опор контактной сети городского электрифицированного транспорта в среде Solid Works. Цель исследования заключалась в построении модели и изучении с её помощью динамических характеристик опор контактной сети трамвайных путей города Новосибирска. Расчётная оценка производилась как с учётом жёсткого закрепления опоры, так и при моделировании дополнительной массы бетона в подземной части конструкции. Полученные результаты позволили сделать вывод о том, что снижение общей размерности задачи по числу узловых перемещений даёт возможность выполнять расчёты за меньшее время, чем при детальном моделировании условий закрепления опоры и бетонного основания в грунте.

Ключевые слова: Solid Works; динамические характеристики; опоры контактной сети; метод конечного элемента; городской электрифицированный транспорт.

Istratova E.E.
ORCID: 0000-0001-9759-2249, PhD.
Kozhevnikov A.N.
ORCID: 0000-0002-8869-7978
Shtrayh A.E., Nikolaeva E.D.
Novosibirsk State Technical University
Novosibirsk, Russia

PUBLIC ELECTRIFIED TRANSPORT OVERHEAD LINE PYLON EIGENVALUES IDENTIFICATION SPECIALITIES IN SOLID WORKS PROGRAM

Abstract. In this paper, results of computational discrete modelling of public electrified transport overhead line pylon are provided. Main idea of the paper is creation of the pylon model and investigation of the pylon dynamic characteristics based on Novosibirsk tram lines. Numerical

CC BY

research has been performed with ideal fixation of the pylon taken into consideration as well as introduction in the model additional concrete mass at underground part of the pylon. Obtained results should be considered as basis to stipulate total model nodal degrees of freedom decreasing without adversely raised affection on computational time or tolerance with respect to detailed modelling of concrete fundament.

Keywords: Solid Works; eigenvalues; overhead line pylon; finite element method; public electrified transport.

Данная работа выполнена в рамках гранта, по итогам конкурса мэрии города Новосибирска на предоставление грантов в форме субсидий в сфере научной и инновационной деятельности.

Городской электрифицированный транспорт города Новосибирска представлен достаточно разнообразно: 14 троллейбусных маршрутов протяжённостью 133,4 км и 11 трамвайных маршрутов протяжённостью 117 км. Зачастую происходит дублирование маршрутов городского транспорта, что приводит к совместному использованию опор контактной сети для одновременного поддержания проводов электропитания двух видов транспорта. Отказ одной опоры контактной сети может послужить причиной остановки движения сразу всего электрифицированного транспорта на данном участке или даже на всем маршруте. Следовательно, необходимо, чтобы, эксплуатируемые в подобных условиях, опоры обладали более высокими показателями эксплуатационной надёжности.

В настоящее время основным критерием для ремонта и замены опор контактной сети является существенный крен конструкции, особенно поперёк направления передачи электрической энергии, или явные признаки разрушения опоры: выкрашивание бетона, растрескивание, отсутствие отдельных элементов. Для организации эффективной системы предупреждения возникновения аварийных ситуаций необходимо разработать систему периодического мониторинга технического состояния опор контактной сети [1, с. 39].

Методику мониторинга целесообразно организовать на основании ГОСТ 31937-2011 «Здания и сооружения. Правила обследования и мониторинга технического состояния». В стандарте предлагается отслеживать изменение частот собственных колебаний сооружений в процессе их эксплуатации. Снижение частот собственных колебаний свидетельствует о том, что произошло снижение жёсткости конструкции – в ней накопились повреждения, негативно влияющие на её несущую способность. В зависимости от величины показателя снижения частот делают выводы о том, насколько сооружение безопасно для последующей эксплуатации и принимают решение о необходимости планово-предупредительного ремонта или даже замены отдельной опоры.

Реализация описанного подхода требует наличия измерительного комплекса для проведения экспресс-оценки динамических характеристик конструкций и значений частот собственных колебаний для опор в исправном состоянии. Если для жилых и промышленных зданий эталонные значения динамических характеристик представлены в ГОСТ 31937-2011,

CC BY

то для других классов сооружений подобные данные в литературе не представлены. Поэтому первым шагом является проведение расчётов с целью определения значений частот низших собственных колебаний конструкций в неповреждённом состоянии – непосредственно после сборки и монтажа.

Конструкция типовой опоры контактной сети городского электрифицированного транспорта представляет собой две трубы, объединённые стыковочным кольцом. На вершине цилиндрической части располагается кронштейн для ламп уличного освещения (рис. 1). Геометрические размеры и материал данной части опоры не удалось найти, поэтому в представленном далее расчёте она не учитывается. Дополнительные элементы, установленные на основной стойке опоры, такие как: светофоры, знаки дорожного движения, оттяжки и грузо-компенсаторы, также исключены из расчётной модели.

Для построения расчётной модели определения частот собственных колебаний стержня переменного сечения необходимо использовать один из конечно-элементных комплексов. В рамках данной работы предлагается построить геометрическую и конечно-элементную модели в среде Solid Works в силу наглядности получаемых результатов и достаточно простого редактирования геометрических моделей. Недостатком комплекса выступает сложность настройки параметров разбиения геометрической модели на конечные элементы.

По проектной документации на обследованном участке улицы Блюхера установлены опоры тип X и тип XV. В дальнейшем будем представлять результаты для опор типа XV. Они отличаются большим диаметром относительного опор типа X и, как следствие, повышенными значениями частот собственных колебаний.

Построение геометрической модели отдельной конструкции будем проводить с использованием команды «вращение».

Рис. 1. Примеры обследованных опор контактной сети города Новосибирска

CC BY

Подобный подход является одним из наиболее эффективных при моделировании объектов с осевой симметрией, поскольку использование операции «выдавливание» потребует трёхкратного применения для всех перепадов диаметров и форм модели опоры [2, с. 74]. В этом случае необходимо отрисовать продольное сечение опоры и затем повернуть его на необходимый угол.

Следующим этапом определения частот собственных колебаний выступает построение конечно-элементной модели и ее закрепление. При разбиении геометрической модели на конечные элементы в среде Solid Works не предусмотрен выбор типа конечного элемента, в то время, как размер и отношение длин граней элемента могут быть кастомизированы. Каждый конечный элемент представлен тетраэдром, поскольку эта форма элемента позволяет описать широкий класс перепадов форм сечений и наличие отверстий.

Отдельной задачей выступает закрепление конструкции. Опора помещается в грунт на глубину 2,5 метра, следовательно, необходимо ограничить перемещения конструкции на указанной отметке от ее нижнего торца. Для реализации такого подхода модель должна иметь в выбранной области некоторую поверхность, к которой будет применено выбранное граничное условие (рис. 2). На внутренней поверхности модели на уровне грунта была смоделирована проточка на доли миллиметра, боковая поверхность которой будет зафиксированной.

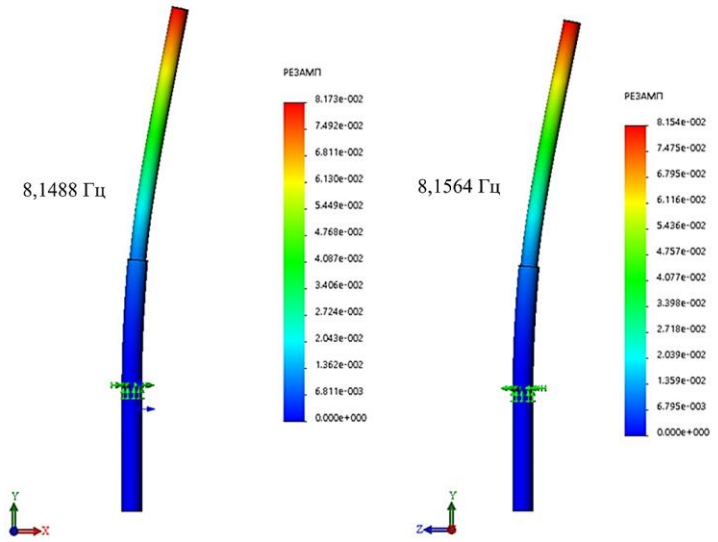


Рис. 2. Низшие формы собственных колебаний опоры при жёстком закреплении

Дополнительно закрепим нижний торец опоры от вертикальных перемещений, поскольку на глубине 2,5 метра располагается вечная мерзлота, которая может быть рассмотрена как неподвижная поверхность. Полученные в данном расчётном случае формы собственных колебаний исключают перемещения части опоры между зафиксированными поверхностями, что противоречит наличию податливого грунта вокруг опоры (рис. 2).

Различие между полученными значениями частот собственных колебаний не превышает 0,093%, что позволяет рассматривать рассчитанные частоты как одно и то же собственное значение с ортогональными формами колебаний. Рассматриваемый результат является закономерным, поскольку исходная конструкция обладает осевой симметрией.

CC BY

Следующим подходом к определению частот собственных колебаний стало моделирование более корректного закрепления опоры в грунте. В реальности каждая опора помещается в бетонное кольцо: конструкция вертикально устанавливается в котлован диаметром 800 мм и глубиной 2,5 метра и заливается бетоном. Соответственно более корректным будет фиксация продольных перемещений нижнего основания опоры и бетонного основания, и исключение горизонтальных смещений опоры на уровне грунта [3, с. 82]. На первом этапе предполагаем идеальное сцепление материала опоры и бетона. Полученные формы низших балочных форм собственных колебаний представлены на рисунке 3.

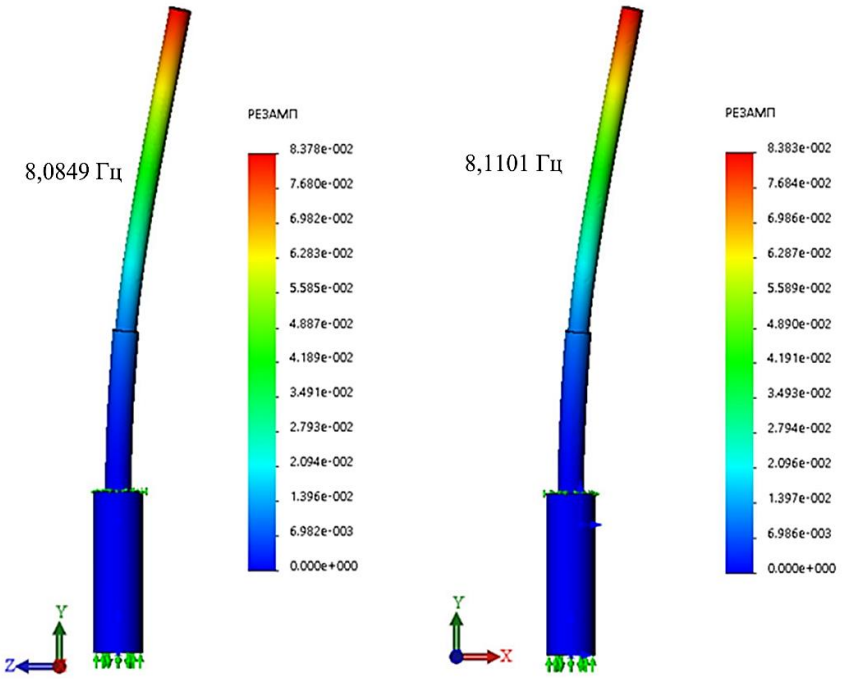


Рис. 3. Низшие формы собственных колебаний опоры с учётом бетонного основания

Различие между полученными значениями частот собственных колебаний не превышает 0,312%, что позволяет рассматривать рассчитанные частоты как одно и то же собственное значение с ортогональными формами колебаний. Рассматриваемый результат является закономерным, поскольку геометрическая модель опоры обладает осевой симметрией.

Добавление в геометрическую модель бетонного основания привело к снижению значений частот собственных колебаний опоры. Таким образом, можно сделать вывод о том, что бетонное основание вносит больший вклад в массовые характеристики, нежели в упругие свойства всей конструкции. Несмотря на это, снижение значений частот собственных колебаний не превышает 0,47%, что не следует рассматривать как существенное.

Таким образом, полученный результат позволяет исследовать поведение всей опоры при учёте жёсткого закрепления и без учёта бетонного основания. Положительным эффектом выступает снижение общей размерности задачи по числу узловых перемещений, что позволяет выполнять расчёты за меньшее время, чем в случае детального моделирования условий закрепления опоры и бетонного основания в грунте.

CC BY

Литература

1. Ли В.Н., Демина Л.С., Власенко С.А. Организация контроля состояния опорного хозяйства системы тягового электроснабжения // Электроэнергетические комплексы и системы: история, опыт, перспектива: Сб. научных трудов Всероссийской научно-технической конференции с международным участием (г. Хабаровск, 19-20 ноября 2020 года). Хабаровск, 2020. С. 36-41.

2. Ли В.Н., Сапов А.С. Использование современных программных комплексов для расчёта и моделирования устройств контактной сети // Электрификация и развитие инфраструктуры энергосбережения тяги поездов на железнодорожном транспорте: Тез. докладов Шестого международного симпоз. «Eltrans-2011». СПб., 2011. С. 74.

3. Нуждин Л.В., Сердакова М.В. Численные исследования колебаний свайных фундаментов в программном комплексе SOLID WORKS // Вестник Сибирского государственного университета путей сообщения. 2011. № 23. С. 78-84.

© Истратова Е.Е., Кожевников А.Н., Штрайх А.Е., Николаева Е.Д., 2022

УДК 004.942
https://doi.org/10.36906/AP-2022/31

Лаврёнов А.Н.
ORCID: 0000-0001-7384-3621, канд. физ.-мат. наук
Хитрушко В.В.
ORCID: 0000-0003-1557-2397
Белорусский государственный педагогический университет им. Максима Танка
г. Минск, Беларусь

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ПРИКЛАДНЫХ НАУКАХ

Аннотация. В статье рассматривается проблема использования визуальной наглядности на уроках. Делаются выводы: 1) изучение основ компьютерного моделирования является одной из важных задач образовательного процесса; 2) существуют множество реальных процессов, которые мы не можем получить из-за масштабности размеров или опасности исследований. Вводится термин «многослойная компьютерная модель», с помощью которой сделана попытка рассмотреть движение тела, брошенного под углом к горизонту. Детально рассмотрен первый математический слой модели с помощью программного кода в среде программирования Pascal.

Ключевые слова: модель; компьютерная модель; физическое явление; задача; движение.

Lavrenov A.N.
ORCID: 0000-0001-7384-3621, Ph.D.
Khitrushko V.V.
ORCID: 0000-0003-1557-2397
Belarusian State Pedagogical University named after Maxim Tank
Minsk, Belarus

COMPUTER SIMULATION IN APPLIED SCIENCES

Abstract. The article discusses the problem of using visual clarity in the classroom. The following conclusions are drawn: 1) the study of the foundations of computer modeling is one of the important tasks of the educational process; 2) there are many real processes that we cannot get due to the scale of the size or the danger of research. The term "multilayer computer model" is introduced, with the help of which an attempt is made to consider the motion of a body thrown at an angle to the horizon. The first mathematical layer of the model is considered in detail using the program code in the Pascal programming environment.

Keywords: model; computer model; physical phenomenon; task; movement.

CC BY

В современном мире проблема использования визуальной наглядности на уроках остаётся актуальной в силу её эффективности. Идеальным средством, которое позволяет нам качественно повысить эту наглядность, является компьютер. Например, на уроках физики очень удобно использовать компьютерные модели. Но, как известно, физика – наука экспериментальная. Поэтому, наглядная демонстрация физических явлений или процессов на уроках является основой обучения физики в школе. Она способствуют созданию физического мышления у обучающихся, делает более понятными объяснения учителя при изложении нового материала, пробуждает познавательный интерес у обучающихся к предмету. Стоит специально отметить, что в условиях школьной лаборатории не всегда возможно продемонстрировать физические процессы или провести эксперименты. Здесь причины бывают разные:

• отсутствие необходимого оборудования;

• опасность проведения эксперимента в данных условиях;

• невозможность проведения демонстрации в реальных условиях.

Конечно, можно объяснить физическое явление по картинкам или «на пальцах», но будет ли от этого результат? Насколько правильно представят то или иное явление ученики? Чтобы понять физическое явление, а потом уметь его объяснить, необходимо его визуально представить. Визуальное представление физического явления или процесса позволяет ученику лучше его запомнить и изучить. Именно поэтому на каждом уроке физики должна присутствовать демонстрация, визуальное представление физических явлений. Если это невозможно провести в данных условиях, то значит нужно использовать компьютерную модель.

Компьютерные модели – это модели, созданные с помощью компьютерных программ для имитирования физических явлений, экспериментов. Очень удобно использовать компьютерные модели в демонстрационном варианте при объяснении нового материала. Например, гораздо проще и нагляднее показать, как будет двигаться снаряд, выпущенный из пушки вертикально вверх, используя модель «Движение тела, брошенного под углом к горизонту», чем объяснять это с помощью доски и мела. Такие модели можно демонстрировать на интерактивной доске, а также на ученических компьютерах, причём, каждый ученик сможет сам управлять интерактивной моделью. При индивидуальной работе учащиеся с большим интересом работают с компьютерными моделями.

Однако, среди них выделяется особый класс так называемых метапредметных моделей, в которых затрагиваются тематики многих дисциплин. Чтобы подчеркнуть, что в реализации такой рассматриваемой метапредметной модели необходимо решать последовательно задачи, каждая из которых относится к своему предмету, то для этих целей, с нашей точки зрения, лучше использовать термин «многослойная модель». Здесь каждый слой может содержать свою тематику и, соответственно, нацелен достигать свои определенные цели. Если теперь это все находит своё претворение в жизнь на компьютере, то тогда мы имеем многослойную компьютерную модель. Для пояснения сказанного рассмотрим подробнее ранее упомянутую тему «Движение тела, брошенного под углом к горизонту» из курса физики 9 класса.

CC BY

Как осуществить её изучение в рамках многослойной компьютерной модели? Уточним вначале свой выбор слоёв – математический, физический, технически-конструктивный, армейский и статический. Названия даём немного условные, но поясняющие основную тематическую направленность слоя. Вначале идёт математическое описание формализованного природного явления, которое подчиняется определённым физическим законам. Для визуализации и наглядности собирается конструкция, которая технически воспроизводит выбранный нами в природе процесс на основе такого армейского инструмента войны как пушка. С помощью неё производятся многочисленные манипуляции, которые позволяют получить кучу данных по различным параметрам. На каждом слое компьютер будет отрабатывать свою роль. Таким образом, в некотором роде получим обсуждаемое компьютерное моделирования в прикладных науках как его реализацию на разных слоях. В нашем варианте такими науками получаются – математика, физика, конструирование, армейское дело и статистика. Так как проведение детального анализа каждого слоя требует соответствующего объёмного наполнения, то в данной работе будет выполнен только первый этап – анализ математического слоя, а остальные – в последующих публикациях.

Текстовая задача – вербальная форма определённого явления. Для того, чтобы найти решение подобной задачи, необходимо перевести её на язык математических операций, т.е. создать её математическую модель. Математическая модель – приближенное представление какого-либо класса явлений, предметов и объектов реального мира на языке математики. Чёткое представление перемещения тела, брошенного под углом к горизонту допустимо только лишь при определённом безупречном случае. Так как реальное перемещение тел в земной атмосфере совершается согласно линии движении, значительно отличающейся от параболической из-за противодействия атмосферы, то в таком случае будем полагать, что влиянием воздуха на движение можно пренебречь. Главными параметрами объекта моделирования считаются:

• начальная скорость тела v_0, *м/с*;
• угол бросания α, °;
• время полёта тела *t, с*;
• координаты тела в любой момент времени (*x, y*), *м*;
• дальность полёта тела *L, м*;
• высота подъёма тела *h, м*;
• ускорение свободного падения *g, м/с*^{2}.

Из них исходными данными являются:

• начальная скорость тела v_0;
• угол бросания α;

Таким образом, из условия задачи можно сформулировать допущения:

• мяч – материальная точка (он достаточно мал по сравнению с Землёй);
• ускорение свободного падения $g = 9{,}8$ *м/с*^{2};
• движение по оси OY равноускоренное;
• сопротивление воздуха не учитываем;

CC BY

• движение по оси OX равномерное.

Сформулируем задачу более чётко и конкретно. Например, следующим образом:

Попадёт ли мяч, брошенный со скоростью 18 м/сек под углом 30°, в площадку протяжённостью 1 м, находящуюся на расстоянии 25 метров?

В этом случае начальная скорость тела v_0 =18 *м/сек*, α = 30°, l = 1 *м*, L = 25 *м*. Решение математической задачи возможно путём построения компьютерной модели пошаговых вычислений. Для этого можно использовать метод построения таблиц значений функции в электронных таблицах. Для наглядности в Microsoft Excel можно построить траекторию полёта мяча как диаграмму графика функции (рис. 1).

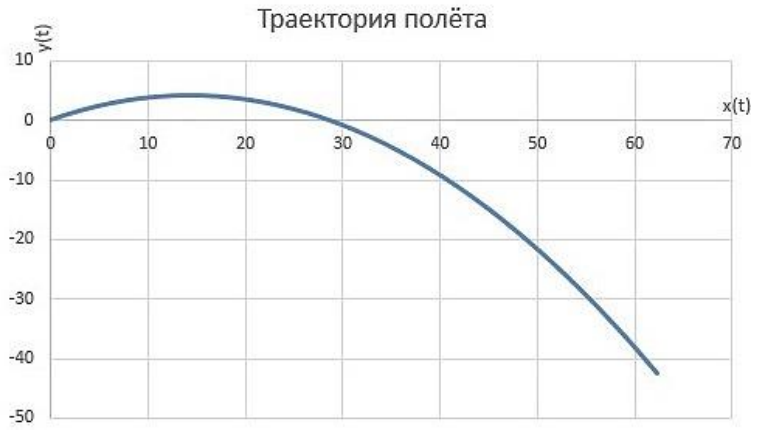


Рис. 1. Диаграмма траектории полёта мяча

Далее формальную модель задачи можно преобразовать в компьютерную с использованием системы программирования Pascal ABC.NET. Рассмотрим данную компьютерную модель подробнее. Необходимо написать программу, в которой будет представлен рисунок площадки, траектория движения мячика и, задавая все необходимые начальные параметры, сделать текстовый вывод «*Попадание*», «*Недолет*», «*Перелет*». Для создания графической и анимационной программы подключим модуль GraphABC, который представляет собой простую графическую библиотеку. Рисование будет осуществляется в специальном графическом окне. После запуска программы «*Движение тела, брошенного под углом к горизонту.pas*» вниманию пользователя предоставляется удобное и понятное в использовании окно программы. В нём появляются поля «*Введите начальную скорость*», «*Введите угол броска*», «*Введите длину площадки*», «*Введите расстояние до площадки*», предназначенные для задания основных параметров программы. Необходимо указать соответствующие значения и программа автоматически выполнит необходимые вычисления, выводя график на экран. В появившемся окне в полях «*Введите начальную скорость*», «*Введите угол броска*», «*Введите длину площадки*» и «*Введите расстояние до площадки*» введём последовательно числа – 18, 30, 1 и 25. На экране появится графический рисунок и текстовое сообщение о том, что мячик перелетел площадку (рис. 2), что является верным с учётом введённых значений.

CC BY

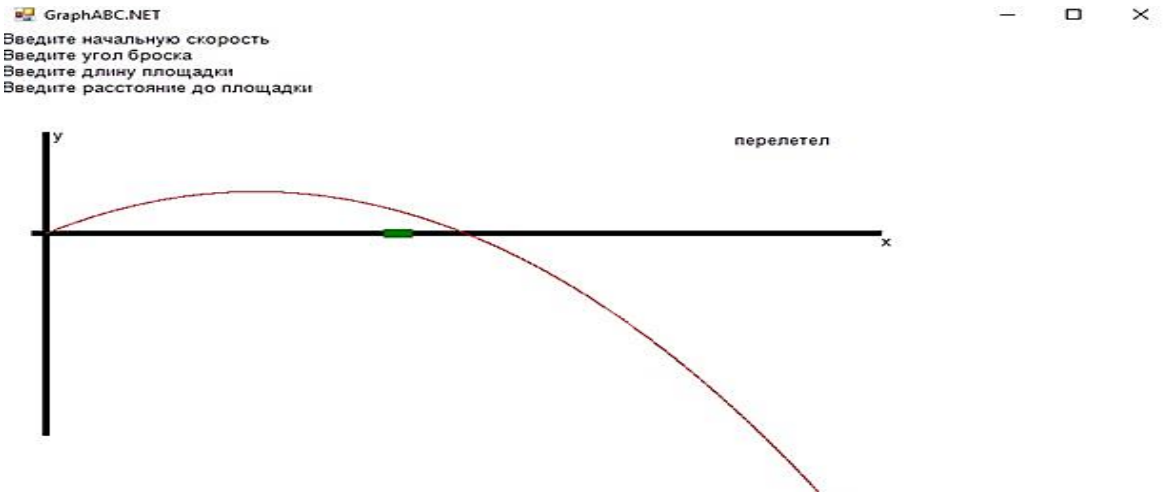


Рис. 2. Скриншот работы программы при перелёте

Если ввести те же значения начальной скорости, длины площадки и расстояния площадки, а угол бросания мячика уменьшить, например на 15°, то в данном случае мячик не долетит до площадки. При введённых значениях начальной скорости 18 *м/с*, угла бросания – 25°, длины площадки – 3 *м*, расстояния до площадки – 25 *м*, мячик попадёт в площадку (рис. 3).

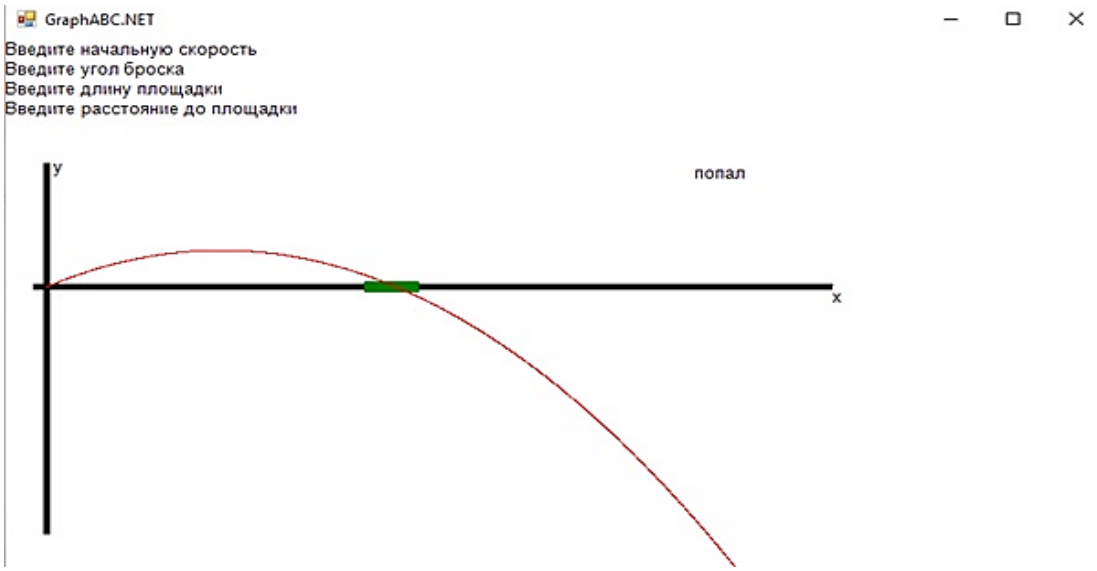


Рис. 3. Скриншот работы программы при попадании

Практической важностью этой компьютерной модели является автоматизирование расчётов физических величин движения тела, брошенного под углом к горизонту и, равно как результат, экономия времени преподавателей и учащихся при решении задач на уроках физики. Кроме того, компьютерное моделирование этой задачи в настоящее время является актуальным не только в учебном предмете физика, но и, например, для военных: при запуске снарядов из пушки, баллистических ракет, мин и прочего. Для решения подобных задач в армейском деле используется электронно-вычислительная техника – компьютеры, с помощью которых моделируется полёт снарядов из пушки, т.к. в условиях высокоскоростной стрельбы нет возможности отдельно наводить орудие для каждого выстрела, даже если бы

вычислительная система и производила бы расчёты индивидуальной траектории снаряда [1]. Таким образом, компьютерное моделирование в прикладных науках реализуется не только в физико-математических областях, но и в армейском деле, в статистике.

Литература

1. Мельников П.Н., Сазонов А.А. Моделирование траектории полёта артиллерийского снаряда // Известия ТулГУ. Технические науки. 2018. № 11. С. 52-58.

© *Лаврёнов А.Н., Хитрушко В.В., 2022*

УДК 004.94
https://doi.org/10.36906/AP-2022/32

Kharitonov L.S.
ORCID: 0000-0001-5237-9452
Evart T.E.
ORCID: 0000-0002-4426-5823, Ph.D.
Arzamas Polytechnic Institute of Nizhny Novgorod State Technical University named after R.E. Alekseev
Arzamas, Russia

CARDIAC TISSUE MAPPING BY MOTION CAPTURE

Abstract. This paper describes an algorithm to map cardiac tissue using motion capture tools. This problem is especially relevant for tissue engineering. The algorithm and its program code are developed for the motion capture of cardiac muscle tissue contraction. The program code is tested on a practical example: the contractions of cardiomyocytes at the microscopic level are visualized by a vector plot over the animation. The contraction and relaxation zones are also mapped: in the animation of the cardiac muscle tissue, they are highlighted in blue and red, respectively. The calculations are performed in *Wolfram Mathematica*.

Keywords: cardiac tissue mapping; motion capture; vector field; video stream; information technology in microscopy; tissue engineering.

Харитонов Л.С.
ORCID: 0000-0001-5237-9452
Эварт Т.Е.
ORCID: 0000-0002-4426-5823, канд. физ.-мат. наук
Арзамасский политехнический институт (филиал)
Нижегородского государственного технического университета им. Р.Е. Алексеева
г. Арзамас, Россия

КАРТИРОВАНИЕ СЕРДЕЧНОЙ ТКАНИ С ПОМОЩЬЮ ВИДЕОАНАЛИЗА

Аннотация. В работе описывается алгоритм картирования сердечной ткани с использованием средств видеоанализа. Данная задача является особенно актуальной для тканевой инженерии. В ходе работы был разработан алгоритм программа для реализации видеоанализа сокращения ткани сердечной мышцы. Программный код был тестирован на примере решения конкретной задачи – в виде векторной диаграммы поверх анимации были подробно визуализированы сокращения кардиомиоцитов под микроскопом. Также были картированы области сжатий и расширений: на анимации сокращения ткани сердечной

CC BY

мышцы они были выделены синим и красным цветами соответственно. Вычисления были выполнены в системе компьютерной алгебры *Wolfram Mathematica*.

Ключевые слова: картирование сердечной ткани; видеоанализ; векторное поле; видеопоток; информационные технологии в микроскопии; тканевая инженерия.

1. Introduction

Motion capture is recording, processing, and analyzing video data containing, as a rule, some motion. It serves to solve many science and technology problems, being most widespread in medicine [1; 2; 3] and biomedical engineering [4; 5]. For the first time, motion capture was applied in biomechanical research. Nowadays, it is used, e.g., in rehabilitation centers to obtain quantitative data for assessing treatment's effectiveness. Motion capture systems are used in general and pediatric neurology, traumatology, and orthopedics to study various types of motor disorders [2], determine the nature of pathology, assess the results of treatment of cerebral palsy, Parkinson's disease [3], stroke, etc. Other application domains of motion capture are cinema, animation, and computer games.

For biomedical engineering (particularly tissue engineering, see V.A. Tsvelaya's dissertation [6]), a topical problem is to map the excitations and contractions of living cardiac tissue cells and create detailed models.

Therefore, for studying the nature of cardiac muscle tissue contractions, it is necessary to develop and test motion capture-based algorithms and programs for mapping living cardiac tissue and creating detailed models. This paper aims to use motion capture for cardiac tissue mapping.

To achieve the goal, we identify the following tasks:

• develop an algorithm and write a program implementing the motion capture of cardiac muscle tissue contraction;

• test the program code on a particular example, i.e., visualize the contractions of cardiomyocytes and map the cardiac tissue on the animation.

2. Using Motion Capture to Map Cardiac Muscle Contractions

2.1. Algorithm description

To solve this problem, we will use *Wolfram Mathematica 12.2*. Due to its rich functionality for procedural, structural, and functional programming, data analysis, numerical and symbolic calculations, visualization, graphics, and data processing, this computer algebra system is suitable for various applied problems of science and technology.

The first (preparatory) stage is the frame-by-frame import of a video file from the PC's hard drive into *Mathematica* in the form of animation using the Import function, indicating "ImageList" in the specification parameters. For reducing the amount of calculations, the Take function is used to extract a video clip from it, consisting of a relatively small number of frames.

In the second stage, the frames are compared to find the difference between them and detect motion. Since there are no markers on the video, the markerless motion capture technology will be used [7]. It is necessary to analyze the mutual arrangement of the inhomogeneous parts of the object's images in successive frames. Therefore, we will apply the *ImageDisplacements* function. This

CC BY

function estimates horizontal and vertical offsets between successive images: compares the successive images $image_i$ and $image_{i+1}$ and returns the horizontal and vertical offset matrix $\{\delta_x, \delta_y\}$ for each pixel of $image_i$. All input images for this function must have the same size.

After obtaining the offset arrays for all frames and pixels, we order the data stream using a Gaussian filter (the *GaussianFilter* function). This filter is commonly used in image processing for anti-aliasing, noise reduction, and image derivation calculations. However, in the case of resampling, it makes the image strongly blurred. Gaussian filtering is widespread in computer graphics and computer vision as one of the stages of edge detection in computer vision algorithms. This filter is based on a convolution involving a Gaussian matrix as its base kernel.

In the third stage, it is necessary to visualize the vector field of cardiac tissue motion. For the convenience of observations, it is displayed by a vector plot over the animation. This can be done using the *ListVectorPlot* function (building vector plots of frames) and the *ListAnimate* function (displaying frames as animation). This function generates a vector plot from an array of vector field values. *ListVectorPlot* displays a vector field using arrows. By default, the arrow indicates the vector's direction, and the color indicates its magnitude. The function has many additional parameters for flexible visualization.

It is important to obtain information about the heart rate. Therefore, in the fourth stage, we write to memory a data array containing the stream values in each animation frame and find frames with a peak stream using the *FindPeaks* function. The pulse (heart rate) can be visualized by a 2D line plot using the *ListLinePlot* function. This function builds a line plot (linear chart) by points. *ListLinePlot* goes through a given list of points sequentially left to right. This function has many additional parameters for flexible visualization.

In the last (fifth) stage, it is necessary to map the contraction and relaxation zones. To do this, we calculate the divergence of the video stream field by a separate function. For this purpose, we use the *GaussianFilter* function for the calculation of derived images:

```
DivVideoStream[σ_?Positive] :=
    flow ↦ GaussianFilter[flow[[All, All, 1]], σ {3, 1}, {0, 1}, Padding → "Fixed"] –
           GaussianFilter[flow[[All, All, 2]], σ {3, 1}, {1, 0}, Padding → "Fixed"];
```

In the animation, the expansion areas are highlighted in red and the compression areas in blue: the colors are applied over the animation and combined in the CIE LAB colorspace (CIE 1976 L*a*b*) using the *ColorCombine* function. This function unites single-channel images $image_1$, $image_2$, … into a multi-channel image, combining the sequence of channels in $image_i$. All images $image_i$ must be the same size. When combining a color image with a grayscale image, *ColorCombine* creates an image in the same colorspace with an alpha channel.

In the Lab colorspace, the lightness value is separated from the chromatic component of the color (hue, saturation). Lightness is given by the coordinate *L* varying from 0 to 100 (darkest to lightest). The chromatic component is given by two Cartesian coordinates *a* and *b*: the former shows

CC BY

the color's position in the range from green to red; the latter, from blue to yellow. This feature provides mapping and highlighting of the contraction and relaxation zones in different colors.

The Lab space uniquely identifies the color, unlike RGB or CMYK colorspaces, representing a set of hardware data for reproducing color on paper or monitor screens. (Color may depend on the type of printing machine, ink brand, humidity in the workshop, and monitor brand and settings.) Therefore, Lab has found wide application in image processing software as an intermediate colorspace to convert data between other colorspaces (e.g., from an RGB scanner to a CMYK printing device). Due to special properties, editing in this colorspace is a powerful color correction tool.

With the color definition in Lab, we can separately affect the image's brightness, contrast, and color. In many cases, this feature accelerates image processing, e.g., in prepress. Lab allows selectively affecting different colors in the image and enhancing the color contrast. Among other tools of this colorspace, note noise reduction in digital photographs [8].

Finally, using the ListAnimate function, we animate the resulting array of images, highlighting the contraction and relaxation zones.

The proposed algorithm is shown by a flow chart in Fig. 1.

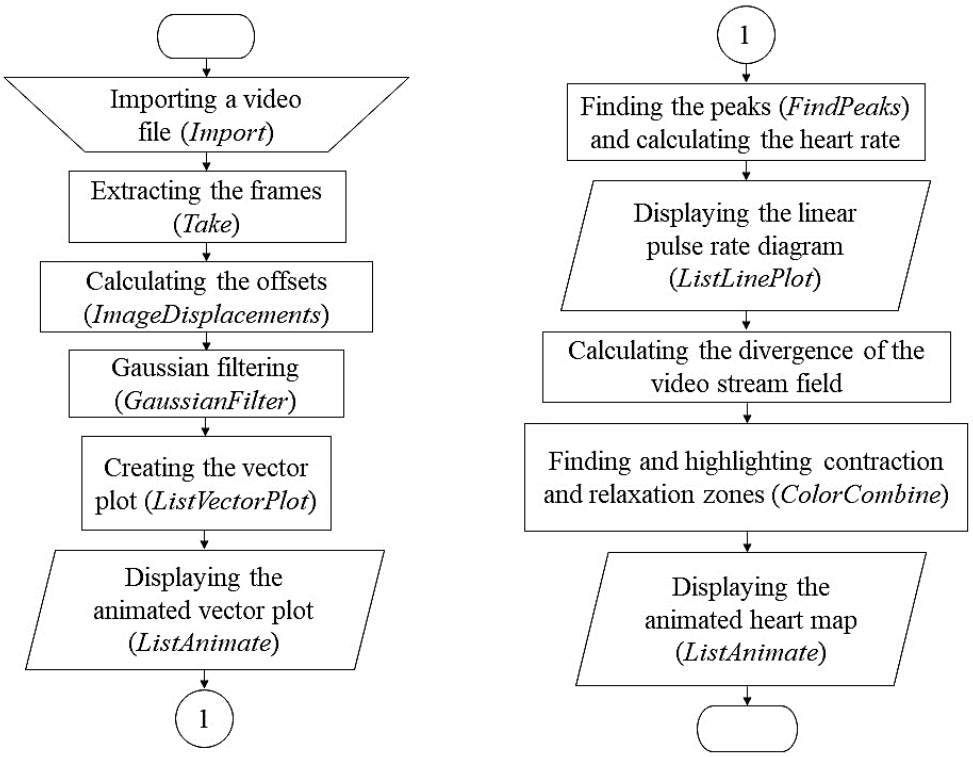


Fig. 1. Motion capture algorithm for cardiac tissue: flow chart

2.2. Algorithm testing on a practical example

We tested the algorithm on a practical example, analyzing a video file with the contractions of cardiomyocytes at the microscopic level. We measured the nature of movements and contractions and performed cardiac tissue mapping. The original video (https://clck.ru/eRmX6) was provided by Cell Applications, Inc. (San Diego, USA), a company studying stem cells.

CC BY

After the frame-by-frame import of this file from the PC's hard drive into *Mathematica*, for convenient and fast calculations, we extracted a 60-frame video clip from it using the Take function (frames from 11 to 70). Sixty frames are quite enough since at least a few beats are accurately captured on them.

Next, we determined the local vertical and horizontal offsets between neighbor frames for all frames sequentially using the *ImageDisplacements* function. This function compares two successive frames, returning the offset matrix $\{\delta_x, \delta_y\}$ for each pixel of the first frame.

Then we applied the *GaussianFilter* function to the frames (for anti-aliasing and noise reduction) and visualized the vector field of cardiac tissue motion, displaying it as a vector plot using the *ListVectorPlot* function; see Fig. 2.

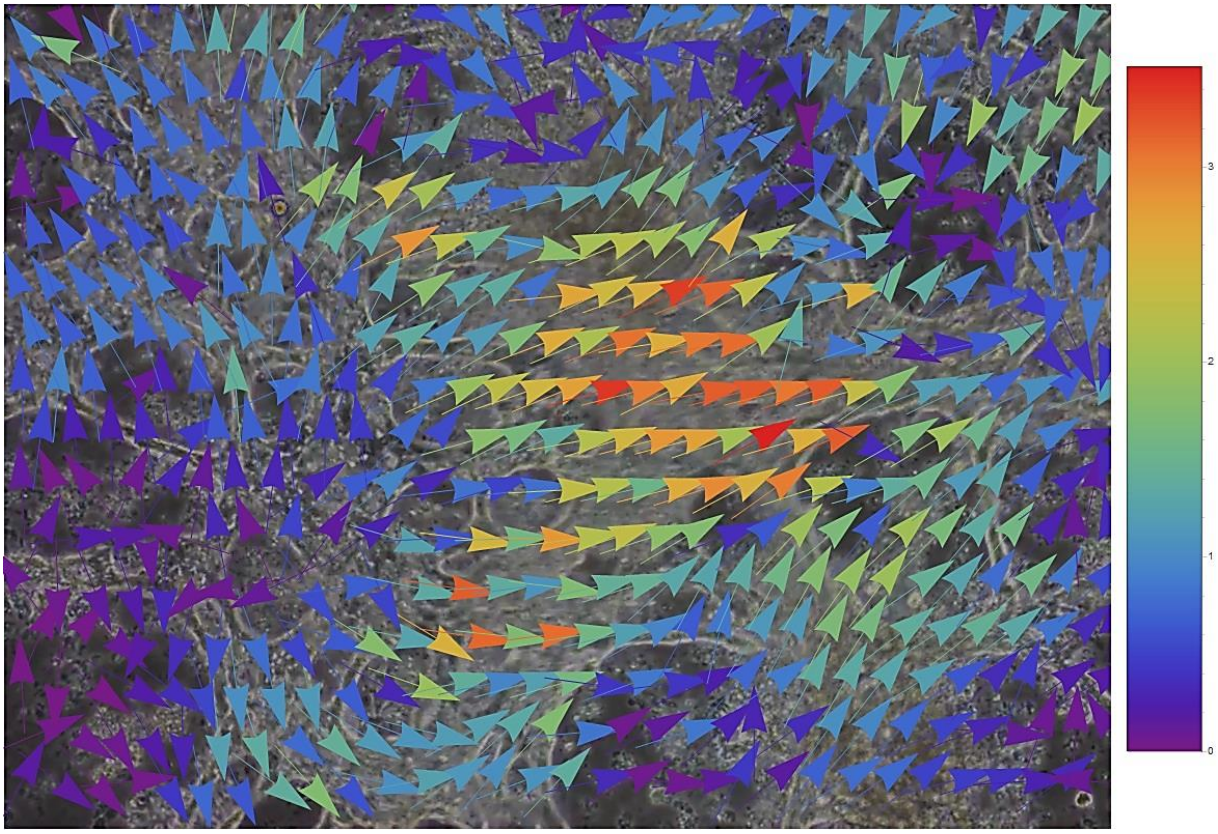


Fig. 2. A frame displaying the vector field of cardiac muscle motion.
Colors indicate the size of the video stream; arrows, the direction of motion

To determine the heart rate, we wrote into memory a data array containing the stream values in each animation frame and found the peak stream frames using the *FindPeaks* function. We visualized the heart rate as a 2D line plot using the *ListLinePlot* function.

Despite that the video data was analyzed, the model is quite detailed. It yields a graph similar to an electrocardiogram, which is taken by a special device.

Clearly, the graph resembles a cardiogram with all waves (Fig. 3). However, the waves are arranged in the opposite direction. (Perhaps, the video was recorded from the end.) The red circle in Fig. 4 indicates a phenomenon called the splitting of the R wave. Also, we calculated the heart rate by averaging. The result is about 80 beats per minute.

CC BY

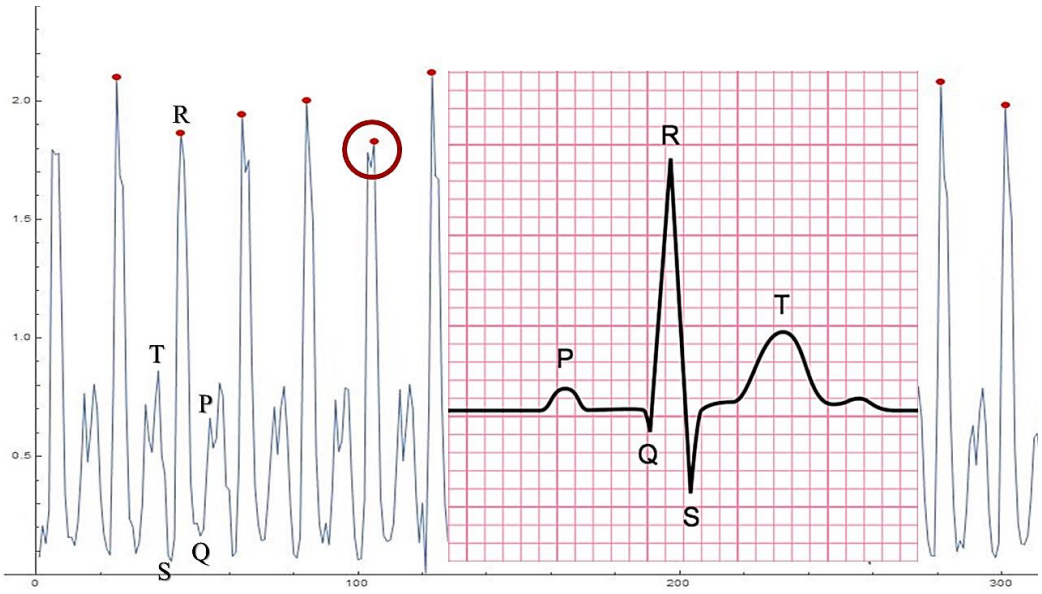


Fig. 3. Motion capture graph vs. ECG

To map the contraction and relaxation zones, we calculated the divergence of the video stream field using the function described above.

Then, we highlighted the relaxation zones in red and the contraction zones in blue, applying the colors over the animation and combining them in the LAB colorspace using the *ColorCombine* function (Fig. 4). It is common to unite several single-channel images into a multi-channel one. To better color the contraction and relaxation zones, the animation frames were converted to the black-and-white colorspace using the *ColorConvert* function.

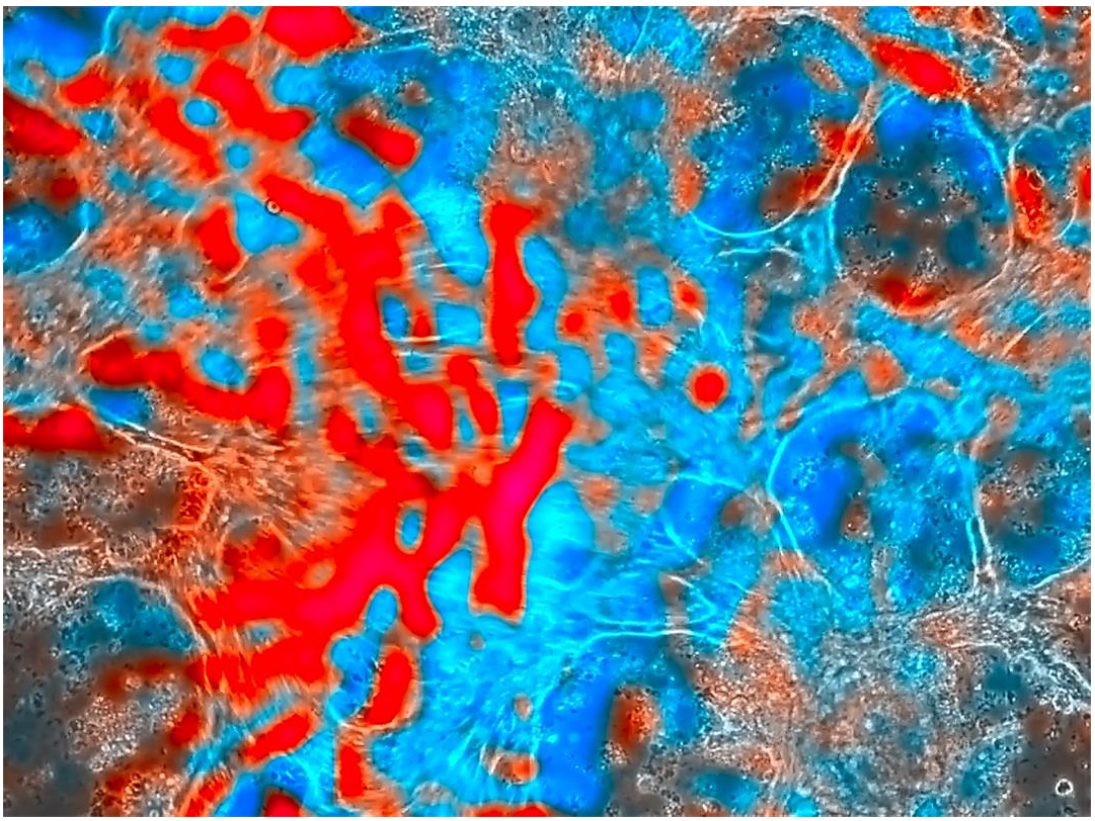


Fig. 4. Cardiac tissue mapping. Frame with highlighted contraction (blue) and relaxation (red) zones

CC BY

3. Conclusions

This paper has presented a motion capture approach to map cardiac tissue motion on the animation at the microscopic level. Note that the line plot of cardiac muscle contractions resembles the ECG.

In particular:

• A motion capture algorithm and its program code have been developed for cardiac muscle tissue contraction.

• The program code has been tested on a practical example: the contractions of cardiomyocytes at the microscopic level have been visualized by a vector plot over the animation, and their contraction and relaxation zones have been highlighted.

The scientific novelty consists in developing and applying a new motion capture algorithm for cardiac tissue mapping and measurement.

The practical value consists in using this algorithm to study cardiac tissue contractions at the microscopic level by biotechnologists, bioengineers, and other researchers.

Литература

1. Borzikov V.V., Rukina N.N., Vorobyova O.V., Kuznetsov A.N., Belova A.N. Human motion video analysis in clinical practice (review) // Sovremennye Tehnologii v Medicine. 2015. Vol. 7. Issue 4. P. 201-210. https://doi.org/10.17691/stm2015.7.4.26

2. Малков А.Б., Кондратьев С.Н., Аброськина М.В. Симметричность и гендерные различи момента аддукции коленного сусутава // Клиническая практика. 2020. Т. 11. № 3. С. 43-49. https://doi.org/10.17816/clinpract17640

3. Прокопенко С.В., Можейко Е.Ю., Аброськина М.В., Ондар В.С., Исмаилова С.Б., Субочева С.А., Хомченкова А.А., Гуревич В.А., Зубрицкая Е.М., Малков А.Б., Кондратьев С.Н. Персонализированная реобилитационная оценка локомоторных функций при болезни Паркинсона с использованием трехмерного видеоанализа движений // Российский неврологический журнал. 2021. Т. 26. № 1. С. 23-33. https://doi.org/10.30629/2658-7947-2021-26-1-23-33

4. Christoph J., Luther S. Marker-free tracking for motion artifact compensation and deformation measurements in optical mapping videos of contracting hearts // Frontiers in Physiology, 2018. 9:1483. https://doi.org/10.3389/fphys.2018.01483

5. Sala L., van Meer B.J. et al. MUSCLEMOTION: A Versatile Open Software Tool to Quantify Cardiomyocyte and Cardiac Muscle Contraction In Vitro and In Vivo // Circulation Research. 2017. Vol. 122, No. 3. P. 5-16. https://doi.org/10.1161/CIRCRESAHA.117.312067

6. Цвелая В.А. Тканевая инженерия сердца как средство для исследования фундаментальных процессов возникновния реентри: Дисс. … канд. биол. наук. Москва, 2019. 164 с.

7. Persson T., Lanshammar H., Medved V. A marker-free method to estimate joint centre of rotation by video image processing // Computer Methods and Programs in Biomedicine. 1995. Vol. 46. Issue 3. P. 217-224. https://doi.org/10.1016/0169-2607(95)01621-Y

CC BY

8. Margulis D. Photoshop LAB Color: The Canyon Conundrum and Other Adventures in the Most Powerful Colorspace, 2nd ed. Peachpit, 2015.

© Kharitonov L.S., Evart T.E., 2022

УПРАВЛЕНИЕ РАЗРАБОТКОЙ ПРОГРАММНЫХ КОМПЛЕКСОВ И ВЫЧИСЛИТЕЛЬНЫМИ ПРОЦЕССАМИ. ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004
https://doi.org/10.36906/AP-2022/33

Антонов И.М.
Мелюков С.А.
Баскаков С.А.
Чемоданов В.Б.
канд. техн. наук
Московский авиационный институт
г. Москва, Россия

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МОБИЛЬНОГО ПИЛОТАЖНОГО СТЕНДА И ОСОБЕННОСТИ ЕГО РАЗРАБОТКИ

Аннотация. В настоящей статье рассмотрено программное обеспечение и особенности его разработки для мобильного пилотажного стенда. Описаны принятые решения и удачные подходы.

Ключевые слова: Git; Qt; Python; C++; QML.

Antonov I.M.
Milyukov S.A.
Baskakov S.A.
Suitcases V.B.
Ph.D.
Moscow Aviation Institute
Moscow, Russia

SOFTWARE FOR MOBILE FLIGHT SIMULATOR AND FEATURES OF ITS DEVELOPMENT

Abstract. This article discusses the software and features of its development for a mobile flight simulator. The solutions adopted and successful approaches are described.

Keywords: Git; Qt; Python; C++; QML.

CC BY

При выполнении работы была поставлена задача создать мобильный пилотажный стенд для внедрения в учебный процесс, отработки различных алгоритмов в области систем управления летальным аппаратом (ЛА), а также оценки психофизического состояния пилота. Пилотажный стенд близок по своему функционированию к авиасимулятору, но допускает проведение моделирование динамических свойств и движения ЛА различных типов.

В рамках поставленной задачи был изготовлен мобильный стенд на основе реальных органов управления ранее выпускавшегося реактивного ЛА, чтобы процесс был максимально приближен к реальному [1].

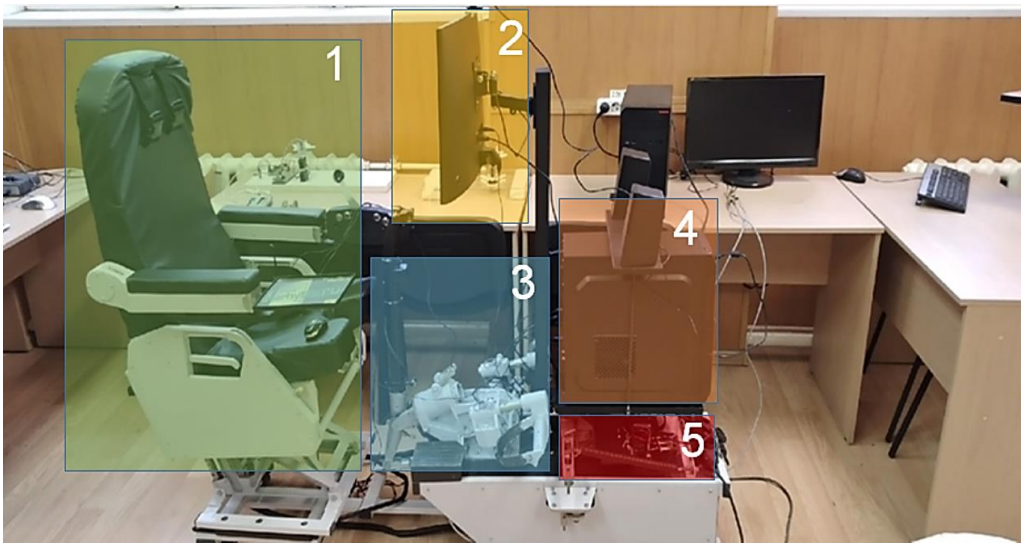


Рис. 1. Мобильный пилотажный стенд

Как видно из рисунка 1, стенд обладает модульной структурой. В него входят основные модули, такие как:

1. Кресло пилота с ручкой управления двигателем (РУД);
2. Вогнутый монитор для отображения информации;
3. Ручка управления самолётом (РУС) и педали;
4. Персональный компьютер (ПК);
5. Устройство, для передачи сигналов с РУД, РУС и педалей в ПК.

Модульность обеспечивает гибкость, наращиваемость и простоту ремонта. А также позволяет разрабатывать программное обеспечение для каждого модуля независимо друг от друга. Для удобства разработки использовался Git – распределённая система управления версиями. Под разработку каждого модуля были создан свой отдельный репозиторий.

Одной из первоочередных задач была обработка управляющих сигналов с помощью АЦП и передача их на ПК. Для этого разработка велась на устройстве на основе микроконтроллера ATmega2560, Arduino Mega 2560. Использование уже готовой платформы вместо просто микроконтроллера значительно сократило время для выполнения этой задачи. На выходе из устройства информация о управляющих сигналах передаётся через USB интерфейс на виртуальный COM порт ПК.

CC BY

На ПК управляющие сигналы принимаются и обрабатываются в каком-либо математическом пакете прикладных программ, для дальнейшего использования для математического моделирования или в отдельной самописной программе с математической моделью ЛА. Помимо этого возможно использование эмулятора джойстика для прямой передачи сигналов с органов управления в авиасимулятор.

Параллельно с обработкой сигналов велась разработка математической модели ЛА и системы автоматического управления для неё. Для этого использовалась российская среда динамического моделирования технических систем Simintech. Она позволяет очень просто и наглядно позволяет решать многие вычислительные задачи. Особенно этому способствует представление программы в виде блоков. Отдельно велась разработка математической модели и разработка интерфейса для связи с авиасимулятором.

В качестве основного авиасимулятора используется FlightGear в силу простоты работы с ним. Он позволяет отображать динамику полёта летательного аппарата в режиме реального времени с помощью передачи UDP пакетов в него из любой другой программы. А также использовать собственные 3D-модели (рис. 2) вместо встроенных.

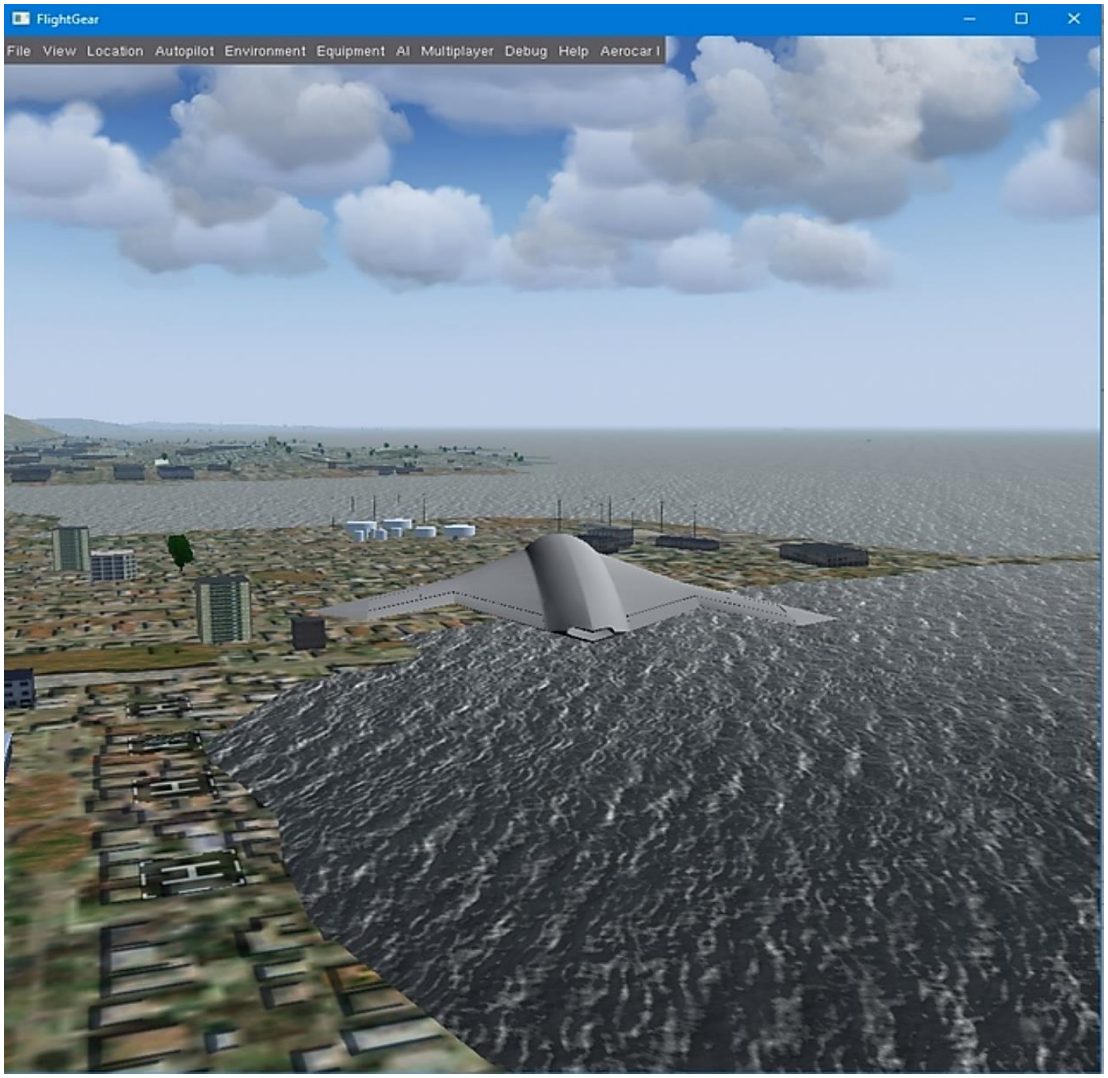


Рис. 2. Своя 3D-модель летающего крыла в FlightGear

CC BY

Одним из дополнительных модулей стенда является программно-аппаратный комплекс оценки психофизического состояния пилота. Аппаратную часть составляют мышка, джойстик, камера и трекер. Программную же часть две программы: отслеживающая трекер и программа для прохождения тестов. Подробнее о них написано в [1].

Для быстрого прототипирования программы для прохождения различных тестов использовался язык программирования Python версии 3.6 и библиотеки PyGame и matplotlib. После получения требуемых точностей измерения, для повышения быстродействия программы и её кроссплатформенности, она была переписана на ЯП C++ с использованием фреймворка Qt.

Одним из способов работы со стендом является использование его для проведения различных занятий по техническим дисциплинам. Для этого разрабатываются программы разного рода. Одной из них является программа (рис. 3) для демонстрации и объяснения работы газотурбинного двигателя (ГТД).

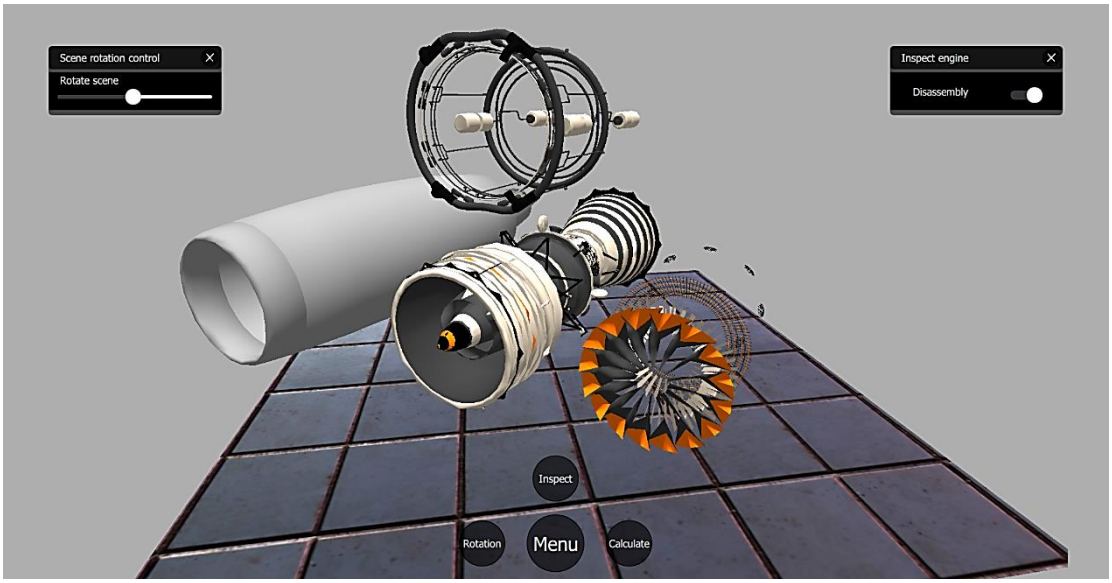


Рис. 3. Программа для демонстрации работы ГТД

Она написана на C++/Qt с, преимущественно, использованием QML для создания интерфейса программы. Математическая модель написана на C++ для быстроты её расчёта. Так как она является упрощённой, то прототипирование, как в случае с программно-аппаратным комплексом оценки психофизического состояния пилота, не потребовалось.

Таким образом применение вышеописанных подходов существенно сократило время разработки программного обеспечения для мобильного пилотажного стенда.

Литература

1. Антонов И.М., Мелюков С.А., Иванов Б.А. Чемоданов В.Б. Мобильный пилотажный стенд // Сб. тезисов докладов XLVII Международной молодёжной научной конференции «Гагаринские чтения-2021» (г. Москва, 20-23 апреля 2021 года). М., 2021. С. 482-483.

© Антонов И.М., Мелюков С.А., Баскаков С.А., Чемоданов В.Б., 2022

УДК 629.066
https://doi.org/10.36906/AP-2022/34

Брагинский М.Я.
ORCID: 0000-0003-3679-4579, канд. техн. наук
Тараканов Д.В.
ORCID: 0000-0003-1851-1039, канд. техн. наук
Сургутский государственный университет
г. Сургут, Россия

РАСПОЗНАВАНИЕ ТРАНСПОРТНЫХ СРЕДСТВ НА ИЗОБРАЖЕНИИ ИСКУССТВЕННЫМИ НЕЙРОННЫМИ СЕТЯМИ

Аннотация. В представленной работе описана процедура создания и обучения искусственных нейронных сетей для распознавания типа транспортных средств на изображении. В качестве среды разработки использованы PyCharm и Google Colab. В работе используется двухступенчатая процедура распознавания информационных признаков на изображении. На первом этапе осуществляется идентификация самого транспортного средства. Затем на втором этапе происходит поиск государственного регистрационного знака на цифровой фотографии и распознавание символов на нем. Для повышения помехоустойчивости системы используются процедуры фильтрации изображений. Результатом работы является система распознавания и классификации на изображении транспортных средств.

Ключевые слова: распознавание; классификация; нейронная сеть.

Braginskii M.Y.
ORCID: 0000-0003-3679-4579, Ph.D.
Tarakanov D.V.
ORCID: 0000-0003-1851-1039, Ph.D.
Surgut State University
Surgut, Russia

RECOGNITION OF VEHICLES IN THE IMAGE BY ARTIFICIAL NEURAL NETWORKS

Abstract. The presented work describes the procedure for creating and training artificial neural networks for recognizing the type of vehicles in the image. PyCharm and Google Colab were used as a development environment. The work uses a two-stage procedure for recognizing information signs in the image. At the first stage, the identification of the vehicle is carried out. Then, at the second stage, a search for the state registration plate in a digital photograph and recognition of characters on

CC BY

it are performed. To increase the noise immunity of the system, image filtering procedures are used. The result of the work is a system of recognition and classification of vehicles in the image.

Keywords: recognition; classification; neural network.

Работа по управлению транспортными потоками и создание необходимой транспортной инфраструктурой требует всестороннего анализа дорожной обстановки. Такой анализ должен учитывать время суток, соотношение типов транспортных средств в общем потоке, время прохождения транспортных средств отдельных участков трассы. Несмотря на появление специальных инструментальных сред моделирования, позволяющих строить транспортные модели (PTV Visum и Aimsun), самым трудоёмким этапом является сбор информации о транспортных средствах, который нередко осуществляется вручную (видеосъёмка и расшифровка видеозаписей). В работе [1] предлагается метод распознавания транспортных средств путём обработки потокового видео с камеры беспилотного летательного аппарата. Также распознавание транспортных средств используется в интеллектуальных системах помощи водителю и системах управления беспилотными транспортными средствами [2; 3], где в первую очередь решается задача вычисления расстояния до транспортных средств и построение траекторий их движения в окружающем 3D-пространстве.

Представленная работа направлена на создание экономичной, помехоустойчивой «софтверной» системы распознавания и классификации транспортных средств на основе искусственной нейронной сети.

При появлении транспортного средства в зоне видимости камеры видеонаблюдения происходит фиксация изображения. После предварительной обработки данное изображение поступает в обученную искусственную нейронную сеть, которая решает задачи детектирования и классификации транспортных средств.

В качестве классификатора в представленной работе используется свёрточная нейронная сеть ConvNet [2], архитектура которой представлена на рисунке 1.

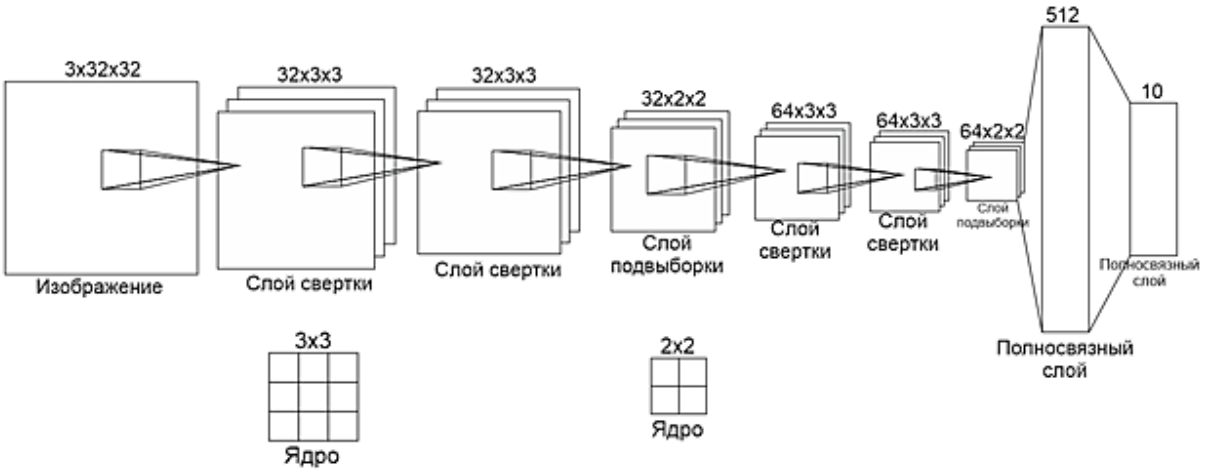


Рис. 1. Модель нейронной сети ConvNet

Как видно из рисунка 1, для решения поставленной задачи была выбрана модель с 7 слоями. Входным воздействием для нейронной сети является изображение размером 32x32 пикселя в формате RGB. Слои свёртки имеют одинаковое ядро размером 3x3 пикселя, но могут отличаться разными классификаторами признаков (в одном случае это 32, а в другом 64). Слои, расположенные до полносвязного слоя, являются средствами предобработки изображения, а дальнейшая классификация изображения выполняется полносвязной сетью прямого распространения. Слои, располагающиеся до полносвязного слоя, используются для выделения информационных признаков, которые затем подаются на слоя, выполняющего процедуру классификации транспортного средства.

На основе вышеуказанной модели был разработан код для построения и обучения нейронной сети на языке программирования Python:

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same', input_shape=(32, 32, 3), activation='relu'))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes, activation='softmax'))
```

Model Sequential представляет собой линейный стек слоёв. Conv2D (сверточный 2D-слой) имеет ядро 3x3, а также классификаторы признаков, в каждом сверточном слое используется функция активации ReLU. Dense – полносвязный слой. В нем, также как в сверточном слое, используется функция активации. Кроме того, данный слой отвечает за классификацию изображения. Dropout – слой, обнуляющий в модели НС часть весов, используется для предотвращения переобучения. Flatten – слой, преобразующий 2D-данные в 1D-данные, используется для конвертации входных данных в данные меньшей размерности. MaxPooling2D – слой подвыборки, снижающий размерность поступивших на него данных. Также, как и сверточный слой, он имеет ядро размером 2x2.

Для обучения нейронной сети нужна обучающая выборка, и следующий код выполняет загрузку и первоначальную настройку набора данных:

```
(X_train, y_train), (X_test,y_test) = datasets.cifar10.load_data()
X_train.shape
X_test.shape
classes = ["bus", "car", "truck"]
X_train = X_train.astype('float32')
```

CC BY

```
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255
batch_size = 64
nb_classes = 10
nb_epoch = 20
Y_train = np_utils.to_categorical(y_train, nb_classes)
Y_test = np_utils.to_categorical(y_test, nb_classes)
```

Первоначальная загрузка выполняется с помощью функции datasets.cifar10.load_data(), с сервера загружается архив в формате «.tar.gz». (X_train, y_train) является обучающей выборкой, а (X_test, y_test) соответственно тестовой. Обучение осуществлялось с учителем, то есть для каждого файла в датасете заранее определен его класс. После загрузки необходимо проверить количество изображений, и в каком формате они были получены. С помощью X_train.shape и X_test.shape проверялось количество изображений, их размеры и количество каналов у изображений.

Для корректной работы искусственной нейронной сети необходимо осуществить процедуру нормализации входных данных, которая заключается в приведении значений яркости пикселов изображений к диапазону [0, 1].

Следующим этапом осуществляется установка параметров оптимизатора и запуск обучения нейронной сети:

```
sgd = SGD(learning_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy',
              optimizer=sgd,
              metrics=['accuracy'])
model.fit(X_train, Y_train,
              batch_size=batch,
              epochs=nb_epoch,
              validation_split=0.1,
              shuffle=True)
```

Оптимизаторы являются важными компонентами архитектуры нейронных сетей. Они играют важную роль в процессе тренировки нейронных сетей, помогая им делать всё более точные прогнозы. SGD – это итерационный метод оптимизации дифференцируемой целевой функции, стохастическая аппроксимация оптимизации градиентного спуска [4]. После установки параметров оптимизатора нужно запустить процесс обучения нейронной сети. Для этого надо указать следующие значения:

X_train – набор фотографий для обучения;

Y_train – метки фотографий;

batch_size – количество фотографий, которые будут поступать в одной итерации;

epochs – количество эпох обучения;

CC BY

validation_split – параметр, который отсеивает часть общей выборки для создания элемента случайности в обучении.

Общая блок-схема обучения нейронной сети отображена на рисунке 2.

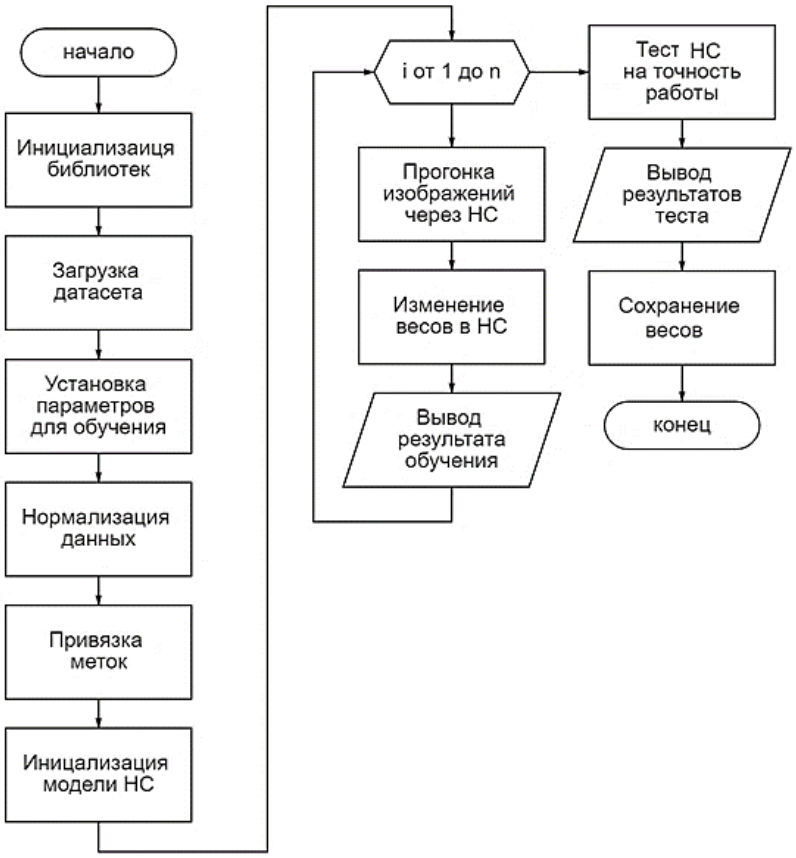


Рис. 2. Блок-схема алгоритма обучения нейронной сети

На рисунке 3 изображён график обучаемости свёрточной нейронной сети для решения задачи классификации транспортных средств при выбранных оптимальных параметрах настойки сети.

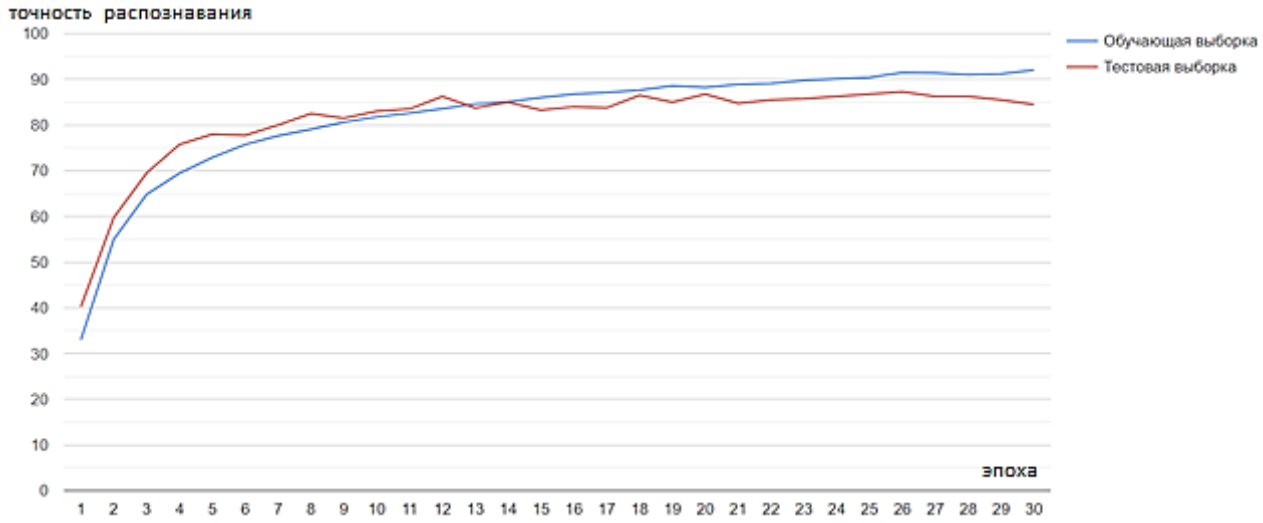


Рис. 3. График обучаемости свёрточной нейронной сети

CC BY

Из графика на рисунка 3 видно, что для обучения нейронной сети необходимо 20–25 эпох обучения модели. Дальнейшее обучение практически не приводило к повышению точности работы нейронной сети.

Во время тестирования кода программы не было выявлено ошибок при распознавании номера и классификации изображения, весь функционал работал правильно и стабильно.

На рисунке 4 представлено исходное изображение транспортного средства. На первом этапе над данным цифровым изображением осуществляется предварительная фильтрация и далее цифровое изображение поступает на вход первой свёрточной нейронной сети. Данная обученная сеть решает задачу классификации объекта на цифровом изображении.

Рис. 4. Исходное изображение

На рисунке 5 представлен как объект классификации, так и результат работы свёрточной нейронной сети (в верхнем левом углу цифрового изображения указан класс объекта).

Рис. 5. Классифицированный объект на изображении

После осуществления идентификации транспортного средства предполагается выполнение процедуры детектирования рамки номерного знака и распознавание

CC BY

государственного регистрационного номера (https://clck.ru/dVibz). Для этой задачи в работе необходимо использовать отдельную свёрточную нейронную сеть.

Анализ работы представленной программы, основанной на использовании свёрточных нейронных сетей и типовых методов фильтрации, показал необходимую помехоустойчивость и требуемое качество распознавания и классификации транспортных средств.

Литература

1. Полищуков Р.М. Распознавание транспортных средств в потоковом видео с машинным обучением // Актуальные исследования. 2020. № 8 (11). С. 20-24.

2. Юдин Д.А., Горшкова Н.Г., Кныш А.С., Фролов С.В. Распознавание транспортных средств и регистрация их траектории движения на последовательности изображений // Вестник БГТУ имени В.Г. Шухова. 2016. № 6. С. 139-148.

3. Rezaei M., Terauchi M., Klette R., Zealand N., Zealand N. Robust Vehicle Detection and Distance Estimation Under Challenging Lighting Conditions // IEEE Transactions on Intelligent Transportation Systems. 2015. Vol. 16. No. 5. Pp. 2723-2743.

4. Taddy M. Stochastic Gradient Descent // Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions. New York: McGraw-Hill, 2019. Pp. 303-307.

© Брагинский М.Я., Тараканов Д.В., 2022

УДК 004.9
https://doi.org/10.36906/AP-2022/35

Есенбаева С.К.
Оренбургский государственный университет
г. Оренбург, Россия

КОРПОРАТИВНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА УЧЁТА ГОРЮЧЕ-СМАЗОЧНЫХ МАТЕРИАЛОВ

Аннотация. Предложен проект корпоративной информационной системы учёта горюче-смазочных материалов. Построена диаграмма потоков данных.

Ключевые слова: корпоративная информационная система; горюче-смазочные материалы; материально-техническое обеспечение.

Esenbaeva S.K.
Orenburg State University
Orenburg, Russia

CORPORATE INFORMATION SYSTEM FOR FUEL ACCOUNTING

Abstract. A project of corporate information system for fuel accounting is proposed. The article contains the presentation of data flow diagram.

Keywords: corporate information system; fuel, MRP system.

Деятельность группы материально-технического обеспечения предприятия в космической отрасли связана с обеспечением материальными ресурсами, горюче-смазочными материалами (ГСМ) и специальными жидкостями (СЖ). Зачастую работа связана с планированием и анализом расходов, для чего требуются пакеты прикладных программ для расчётов.

Однако вести учёт, производить расчёты и оформлять требуемые отчёты гораздо легче и удобнее в специализированной автоматизированной системе, разработанной под конкретные нужды. Внедрение крупных MRP-систем (Material Requirements Planning) на малых предприятиях не целесообразно, ввиду их стоимости.

В данной работе рассматривается проект корпоративной информационной системы (КИС) учёта ГСМ.

Входными данными системы являются:

– масса топлива;

– информация о соответствии сорта, марки и характеристики поступающего топлива данным, указанным в удостоверениях о качестве и паспортах, техническим условиям

CC BY

поставки, предусмотренным в договорах (температуры вспышки, содержание серы, зольность, вязкость);

– объем и плотность топлива;

– результаты контроля качества топлива согласно ГОСТ-2517-85 «Нефть и нефтепродукты. Методы отбора проб» (https://clck.ru/dneN9);

– информация о заказах;

– нормативные документы.

Выходные данные:

– отчёт о заказах топлива;

– еженедельный отчёт по учёту топлива;

– ежемесячный отчёт по учёту ГСМ и СЖ;

– отчёты о списании ГСМ и СЖ;

– план годовой потребности в ГСМ и СЖ.

Информационный поток процесса управления учётом ГСМ выглядит следующим образом (рис. 1).

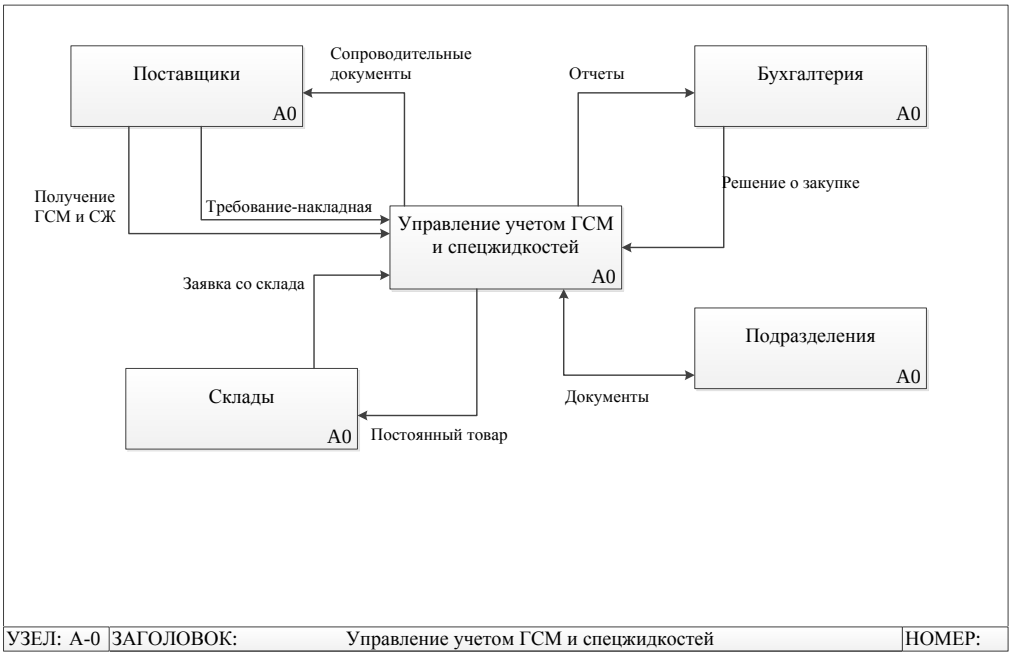


Рис. 1. Контекстная диаграмма DFD

КИС содержит следующие подсистемы: подсистему учёта поступаемого и расходуемого топлива; подсистему отчётов; базу данных, подсистему прогнозирования спроса, подсистему списания ГСМ.

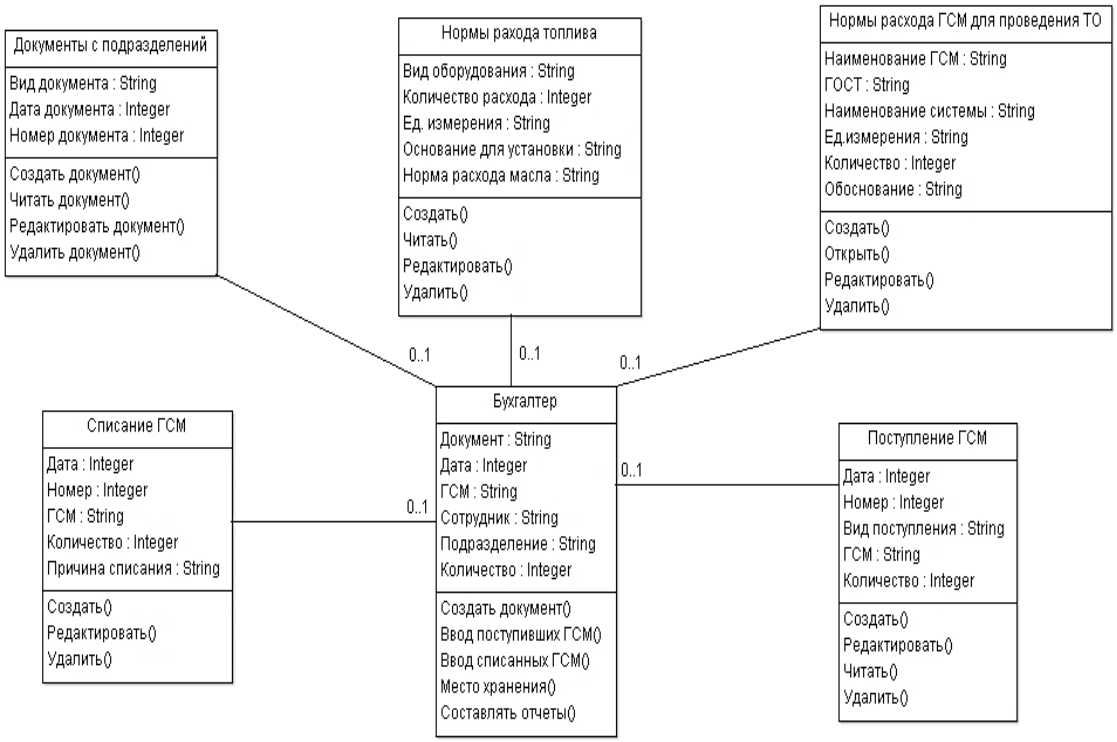


Рис. 2. Диаграмма классов

Основными сущностями являются «Документы с подразделений», «Нормы расхода топлива», «Нормы расхода ГСМ для проведения технического обслуживания», «Списание ГСМ», «Поступление ГСМ», «Бухгалтер» (рис. 2).

КИС разрабатывается для бухгалтера, занимающимся учётом ГСМ. Основываясь на специфике информационной системы, решено использовать локальную базу данных. Клиент-серверную модель базы можно построить при дальнейшем усовершенствовании этой системы [1].

В заключение, стоит отметить, что корпоративная информационная система позволит избежать ошибок, уменьшить материальные и временные затраты, а также повысить скорость работы сотрудников предприятия.

Литература

1. Хомененко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. СПб.: КОРОНА-Век, 2009. 736 с.

© Есенбаева С.К., 2022

УДК 330.44, 658.51
https://doi.org/10.36906/AP-2022/36

Мокряков А.В., Зяблицева Д.К.
Российский государственный университет им. А.Н. Косыгина
(Технологии. Дизайн. Искусство)
г. Москва, Россия

ВОПРОСЫ АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ ОРГАНИЗАЦИИ НА БАЗЕ ПЛАТФОРМЫ 1С

Аннотация. В данной статье описано исследование об автоматизации бизнес-процессов. Авторы описывают насколько можно улучшить эффективность предприятия за счёт внедрения программных модулей на базе платформы 1С. Также обзор используемых объектов для разработки. В работе используется описание бизнес-процессов и применение.

Ключевые слова: бизнес-процесс; автоматизация; оптимизация предприятия; 1С: Предприятие; программное обеспечение.

Mokryakov A.V., Zyablitseva D.K.
Russian State University im. A.N. Kosygin
(Technologies. Design. Art)
Moscow, Russia

QUESTIONS OF AUTOMATION OF BUSINESS PROCESSES OF AN ORGANIZATION BASED ON THE 1C PLATFORM

Abstract. This article describes a study on business process automation. The author describes how much it is possible to improve the efficiency of an enterprise by introducing software modules based on the 1C platform. Also an overview of the objects used for development. The work uses a description of business processes and application.

Keywords: business process; automation; enterprise optimization; 1C: Enterprise; software.

В значительном количестве предприятий российского рынка поднимается вопрос о внедрении автоматизированных систем управления, к которым бизнес выдвигает требованием возможность оперативно адаптироваться в соответствии со спецификой актуальных для конкретной компании бизнес-процессов. Востребованность подобных систем обусловлена достижением предприятием корпоративных целей, таких, как: необходимость производить оперативный обмен информацией между внутренними, а также внешними, источниками; обеспечение непрерывного функционирования предприятия, исключив человеческий фактор. Кроме того, у предприятий есть экономические цели: увеличение доли предприятия на рынке; увеличение рыночной стоимости предприятия; повышение рентабельности предоставляемых

CC BY

услуг и эффективности труда. Учитывая отсутствие полностью универсального решения, построение автоматизированной системы для отдельного предприятия может сопровождаться некоторыми трудностями:

– данный процесс является весьма трудоёмким и значительно использует как финансовые, так и временные ресурсы. Кроме того, сотрудниками компании, которыми внедряемая система будет эксплуатироваться, этот процесс может быть встречен негативно в связи с предстоящими переподготовкой и повышением квалификации;

– внедряемая система должна быть достаточно гибкой и иметь возможность оперативно внести изменения, согласно актуальному регламенту бухгалтерского учёта и юридических процессов.

Автоматизацией бизнес-процесса называется процесс внедрения программного продукта и практик, направленных на конвейерное автоматическое исполнение шаблонных задач. Кроме того, успешному внедрению способствует модульность внедряемой системы, что позволяет использовать исключительно востребованные компоненты. Для повышения эффективности в предприятиях все чаще используют информационные технологии, и все чаще их применяют как решения проблем, которые появляются от сложных бизнес-процессов в предприятии. Помимо проблем, которые могут быть решены информационными технологиями, большое внимание можно уделить преимуществам: в разы увеличивается эффективность качества обслуживания и выполнения бизнес-процессов. Каждый бизнес-процесс состоит из определенных процедур и алгоритмов. При исполнении таковых, необходимо чтобы все соответствовало стандартам, также вся необходимая информация и данные должны быть актуальными для правильного функционирования процессов. Для автоматизации необходима информационная система, и наиболее распространённая – экономическая информационная система. Она предназначена для сбора, поиска, хранения обработки и выдачи информации деятельности экономического объекта по запросам пользователей**.** К экономическим информационным системам относят банковские, бухгалтерские, страховые, финансовые информационные системы (https://clck.ru/eRnKY).

Самый популярный среди экономических информационных системы на текущий момент является комплекс программ «1С» (https://v8.1c.ru/). Разработки на основе данной программы хорошо применяются для автоматизации бизнес-процессов. В данной статье будет разбор на примере разработанной конфигурации, для строительной компании, которая осуществляет услуги и продажу сопутствующих товаров. Чтобы осуществить автоматизацию бизнес-процессов, необходимо было разработать сначала конфигурацию. Для неё использовались такие объекты конфигурации как справочники, документы, регистры, которые включают в себя регистры сведений, накопления, бухгалтерии, расчёта. Для разработки системы и автоматизации был использован встроенный язык программирования 1С, который в значительной степени помог описать все важные алгоритмы, происходящие в бизнес-процессах. Также упрощали разработку механизм запросов и различные конструкторы. Данные инструменты автоматизировали движения документов и изменения в объектах конфигурации. В каждом документе был отражён аналитический учёт для ведения

CC BY

бухгалтерских и налоговых движений. Стоит уделить внимание каждому объекту конфигурации. Для хранения данных в информационной базе, которые бы имели структуру и списочный вид, используются справочники. В данной системе следующие: Сотрудники, Клиенты, Склады, Номенклатура, Субконто и т. д. В каждой системе присутствуют такие справочники, независимо от назначения программного продукта. Для отражения информации о операциях в системе используют документы. Например, Приходная накладная, Оказание услуг, Начисления сотрудникам. Для того чтобы вывести всю информацию о поступлениях, расходах, и любых изменениях в организации используется объект конфигурации Отчёт. Пользователю необходимо контролировать все изменения по товарам и денежным средствам, и в этом помогают отчёты Оказание услуг, Перечень услуг, Начисления сотрудникам, Количество материалов. И для изменений в системе 1С предусмотрены регистры, с помощью которых можно автоматизировать движения бизнес-процессов. Например, в регистре сведений отражаются данные в разрезе измерений. Цены на материалы, виды графиков работы и значения свойств номенклатуры. В регистрах накопления хранятся движения по документам. В регистре Остатки товаров записываются обороты и остатки, для отражения количества товара на предприятии. Регистр расчёта Начисления. Используется в механизме сложных периодических расчётов и служит для хранения записей о тех или иных видах расчёта, которые необходимо выполнить, также для хранения промежуточных данных и самих результатов выполненных расчётов. В данной системе используется для автоматизации расчёта зарплаты сотрудникам. Выполняются вычисления из данных оклада, премии, а также пропусков рабочих часов по каким-либо причинам. Для автоматизации процесса расчёта выплат, в модуль регистра расчёта Начисления можно добавить расчёт дополнительных пособий или вычетов налогов. В системе 1С присутствуют регламентные задания, которые создают разработчики для удобства пользователей. Позволяют создавать события, которые бы происходили в фоновом режиме, параллельно с работой пользователя, выполняя определенные процедуры и алгоритмы по определенному заданному расписанию. Данные задания пользователь может сам отменить или изменить расписание.

С каждым днём информационные системы развиваются и не стоят на месте, поэтому всем предприятиям и бизнесу, необходимо адаптироваться и улучшать эффективность всех инструментов, одним из которых и является автоматизированная информационная система. Большое количество организаций перешло либо на полный отказ, либо на частичное сокращение бумажного документооборота и контроль всех операций человеческими ресурсами. Сейчас это все уже можно упростить и автоматизировать, что значительно удешевит процесс и сократит вероятность каких-либо ошибок в бизнес-процессах. Автоматизированная система позволит оперативно контролировать все операции и изменения, что сократит время, при непредвиденных ситуациях, и поможет сохранить денежные ресурсы. Внедрение типового продукта подойдёт для малых предприятий, который можно легко подстроить под учёт, и упростит будущее обновления системы к изменениям в налоговом или бухгалтерском учёте. Для средних и больших также подойдёт типовой продукт, но с

CC BY

доработками под требования самого предприятия, которые бы упрощали и максимально автоматизировали все важные бизнес-процессы.

© Мокряков А.В., Зяблицева Д.К., 2022

УДК 004.89
https://doi.org/10.36906/AP-2022/37

Имомова Ш.М.
ORCID: 0000-0002-2811-7409
Бухарский государственный университет
г. Бухара, Узбекистан

ЦЕЛИ, ФУНКЦИИ И МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ

Аннотация. Одним из наиболее актуальных вопросов сегодня является разработка цели, структуры и технологий применения обучения специальностям на основе интеллектуальных систем как нового подхода к организации процесса профессионального и индивидуального развития студентов в образовательном процессе. В статье представлена информация о назначении, функциях и методологии интеллектуальных систем управления.

Ключевые слова: интеллектуальная система; искусственный интеллект; методика обучения; учебный процесс; экспертная система; система связи.

Imomova Sh.M.
ORCID: 0000-0002-2811-7409
Bukhara State University
Bukhara, Uzbekistan

OBJECTIVES, FUNCTIONS AND METHODS OF INTELLIGENT CONTROL SYSTEMS

Abstract. One of the most pressing issues today is the development of the goal, structure and technologies for applying training in specialties based on intelligent systems as a new approach to organizing the process of professional and individual development of students in the educational process. The article provides information on the purpose, functions and methodology of intelligent control systems.

Keywords: intelligent system; artificial intelligence; teaching methods; educational process; expert system; communication system.

Преимущества экспертной системы перед другими информационными системами заключаются в следующем: Возможность решения, оптимизации и (или) оценки нового класса сложных задач, которые до недавнего времени считались трудными для решения в ЭВМ или вообще не решались; обеспечение возможности непрограммному пользователю (конечным пользователям) вести диалог на своём языке и применять методы визуализации информации для эффективного использования компьютера; самостоятельное изучение экспертной системы с целью получения более достоверного и квалифицированного заключения или принятия решения, накопление опыта использования знаний, информации, знаний; возможность

CC BY

создания индивидуальных специализированных экспертных систем за счёт отсутствия у пользователя информации или чрезмерной информационной пестроты, усовершенствованных инструментов и использования пользователем этой системы личного опыта специалиста; основой экспертной системы является совокупность знаний (база знаний), сформированная с целью формирования процесса принятия решений.

База знаний-это совокупность моделей, правил, факторов (данных), генерирующих анализ и выводы для решения сложных задач определенных предметных областей.

Когда говорят об искусственном интеллекте, обычно понимают способность компьютерных систем выполнять такие действия, которые, если бы эти действия выполнял человек, можно было бы назвать интеллектуальными.

В большинстве случаев здесь подразумеваются способности, связанные с мышлением человека. Работы в области искусственного интеллекта не ограничиваются экспертными системами. Они также включают в себя создание роботов, систем, которые моделируют нервную систему человека, его способность слышать, видеть, чувствовать, читать и учиться. Решение конкретных вопросов или задач требует специальных знаний. Но не каждая компания может содержать в своём штате специалистов по всем вопросам, связанным с её работой, или даже приглашать их каждый раз, когда возникают такие проблемы. Главная суть использования технологии экспертных систем заключается в получении от эксперта его знаний и занесении их в память компьютера, используя каждый раз, когда возникает такая необходимость. Экспертные системы являются одним из основных приложений искусственного интеллекта и состоят из компьютерных программ, преобразующих опыт экспертов в той или иной области знаний в виде эвристических правил. Эвристики в принятии решений не гарантируют получение оптимального результата с такой же уверенностью, как и простые алгоритмы, используемые для решения проблем в рамках технологий поддержки. Но часто даёт решения, достаточно оптимальные для их практического использования. Все это даёт возможность использовать экспертные системы в качестве систем-консультантов.

Интеллектуальная система обучения является практическим результатом применения методов и инструментов искусственного интеллекта в области автоматизированного обучения и представляет собой новое поколение образовательных систем. Для достижения высокого уровня сформированности знаний и умений студентов в учебном процессе преподаватель специальностей использует три основных типа знаний: знание изучаемой специальности, знание методики обучения и знания обучаемого. В традиционных автоматизированных системах обучения многие части этих знаний строго включены в отдельные разделы предмета в соответствии с выбранной методикой обучения. Знания, необходимые в интеллектуальной системе образования, выделяются и демонстрируются с использованием различных методов и технологий искусственного интеллекта. При представлении учебного материала в интеллектуальной системе обучения, используя эти знания и исходя из психофизиологических и интеллектуальных возможностей учащегося, выявлять и регламентировать наиболее эффективные методы обучения, приёмы и ставки, содержание науки, объем и уровень возможна сложность задач.

CC BY

Интеллектуальная система обучения специальным предметам состоит из базы знаний, системы управления, системы обучения и системы коммуникации. База знаний – основа интеллектуальной системы преподавания специальных предметов. База знаний по каждой специальности, в свою очередь, состоит из нескольких взаимосвязанных баз данных, каждая из которых составляет определенную часть изучаемого предмета: «Термины и основные понятия», «Теория», «Практико-лабораторные работы», «Самостоятельные работает».

Знания о методах обучения собраны в базе данных «Задания», в которой дидактические материалы по всем разделам предмета систематизированы и представлены в упорядоченном виде, в соответствии с методами обучения. Информация об учащемся собирается в базе данных результатов обучения. Для каждого студента создаётся индивидуальная карта, в которой содержится информация об уровне и качестве знаний, полученных студентом по предмету, особенностях его психофизиологического развития.

Система управления базой знаний предназначена для ввода и накопления новых знаний. Первый этап – изучение системы, все знания по специальности новые, и сбор этих знаний осуществляется в процессе диалога между системой и экспертом. На следующих этапах поиск новых знаний осуществляется автоматически: вся полученная информация сравнивается с существующей, новая информация сортируется, анализируется на основе имеющихся знаний, и их классификация представляется в одном или нескольких вариантах. Например, при обнаружении нового метода обработки предмета предоставляется информация о том, как это сделать, о необходимых инструментах и оборудовании, используемых материалах и областях применения.

Система обучения предназначена для организации и поддержки целенаправленного учебного процесса и состоит из трёх модулей: управление обучением, постановка задач и анализ результатов. На каждом этапе обучения по результатам анализа индивидуальной карты студента определяется цель обучения, выбираются или разрабатываются учебные задания. Если студент работает самостоятельно, его действия отслеживаются, его ошибки и трудности фиксируются, когда необходимо вмешиваться в процесс обучения, оказывать помощь, вносить изменения в индивидуальную карту студента. Когда требуется помощь, подбираются оптимальные формы её изложения (указание на ошибку, предоставление студенту информационных материалов для исправления собственных ошибок или заполнения того, чего он не знает, и т. д.). Поскольку все решения принимаются на основе индивидуальных характеристик студента, интеллектуальная система гарантирует, что процесс обучения оптимален для студента.

Коммуникационная система служит для обеспечения взаимного общения при работе с получателем знаний в индивидуальном и сетевом режиме, в процессе дистанционного обучения. Данная система направлена на выявление личности обучающегося, обеспечение его доступа к базе знаний, информации и учебно-информационным материалам системы обучения. Общение ученика с системой должно осуществляться в удобных для понимания формах и на простом языке.

CC BY

Общий вид процесса реализации интеллектуальной системы обучения предметам специальности можно представить в виде следующей схемы: после подтверждения личности пользователя определяется его право доступа к системе, модуль системы управления обучением анализирует состояние индивидуальной карточки обучающегося, определяет цель, методику и задачи дальнейшего обучения. Модуль формулирования учебных заданий разрабатывает обучающую задачу на основе данных выбранной методики и базы знаний. Студент выполняет поставленную задачу (изучение теоретического материала, выполнение практической работы, контрольной работы, написание ответов на вопросы и т. д.), при этом обеспечивается связь с системой, то есть работа студента контролируется и, при необходимости, оказывается помощь. Модуль анализа результатов выявляет допущенные ошибки, возникающие трудности и вносит изменения в индивидуальную карту учащегося. На этом цикл обучения завершается, и система переходит в состояние готовности к следующему процессу. Пользователь сам может определить задачу (вывести на экран нужную часть теоретического материала или задания, обратиться к имеющимся словарям и ресурсам, просмотреть индивидуальную карточку и т. д.).

Интеллектуальная система обучения предметам специальности, учитывающая индивидуальные особенности обучающегося, позволяет осуществлять в замкнутой системе, с автоматизированным управлением, в самостоятельной, индивидуальной, коллективной и дистанционной формах образовательный процесс, ориентированный на достижение личностных, конкретных целей.

Экспертные системы сегодня объединяют в себе несколько тысяч различных программных комплексов, которые можно классифицировать по различным признакам.

Классификация по решаемой задаче. Интерпретация данных. Это одна из традиционных задач для экспертных систем. Под интерпретацией понимается определение смысла информации, результаты которой должны быть согласованы и точны. Обычно подразумевается многовариантный анализ данных. Интерпретация данных. Это одна из традиционных задач для экспертных систем. Под интерпретацией понимается определение смысла информации, результаты которой должны быть согласованы и точны. Обычно подразумевается многовариантный анализ данных.

Диагностика (диагностика). Диагностика – процесс поиска неисправности в какой-либо системе. Коррупция – отклонение от нормы. Такая интерпретация позволяет рассматривать как отказ техники и технических средств, так и болезни живых организмов и возможные природные аномалии с единой теоретической точки зрения. Необходимость понимания функциональной структуры (анатомии) диагностируемой системы является важной особенностью.

Мониторинг. Основная задача мониторинга – сообщить, что при фактическом использовании времени данные постоянно интерпретируются и что та или иная мера превышает допустимый предел. Основная проблема – обратная функция «пропуска» и «лжи» о тревожной ситуации. Сложность этих проблем заключается в необходимости учитывать контекст, в котором утихают симптомы (симптомы) тревожной ситуации.

CC BY

Дизайн. Дизайн – обучение специалистов проектированию «объектов» с заранее заданными характеристиками. Под специализацией понимается весь набор необходимых документов – чертежи, аннотации и т. д. Основная проблема здесь – получить чёткое структурное описание знаний об «объекте» и проблеме «следа». Чтобы в большей степени организовать эффективное проектирование и редизайн, необходимо сформулировать не только сами проектные решения, но и причины их принятия. Таким образом, два основных процесса, которые выполняются в структуре дизайна, тесно связаны: процесс заключения и процесс интерпретации.

Прогноз. Разрабатываемые системы учитывают возможные последствия данной ситуации. В проектируемой системе обычно используется параметрическая динамическая модель, в которой значение параметров адаптируется к данной ситуации. Результаты, полученные с помощью этой модели, составляют основу для прогнозов с вероятностными оценками.

Планирование. Планирование – процесс поиска планов действий для объектов, способных выполнять определенные задачи. Он использует модель поведения реальных объектов для логического вывода результатов запланированной деятельности.

Обучение. Система образования диагностирует ошибки в изучении любого предмета с помощью компьютера и подсказывает правильные решения. Он суммирует знания гипотетического «ученика» и его или её конкретные ошибки, а затем может диагностировать слабые места в знаниях учащегося на работе и найти соответствующие инструменты для их устранения. Они также планируют документ для общения со студентом на основе его успехов с целью передачи знаний. В целом, все системы, основанные на знаниях, можно разделить на системы, решающие задачу анализа, и системы, решающие задачу синтеза.

Основные различия между задачами анализа и задачами синтеза заключаются в том, что если большинство решений в задаче анализа можно подсчитать и систематизировать, то большинство решений в задачах синтеза строятся на основе решений потенциальных компонентов или задач.

© Имомова Ш.М., 2022

УДК 004.053
https://doi.org/10.36906/AP-2022/38

Казаченко Т.А., Мелехин М.И.,
Литовка А.Д., Батыркаев А.Г.
Аитов А.И.
ORCID: 0000-0003-4941-9104, канд. физ.-мат. наук
Пермский государственный аграрно-технологический университет
им. академика Д.Н. Прянишникова
г. Пермь, Россия

АРХИТЕКТУРНЫЙ ТЕХНИЧЕСКИЙ ДОЛГ В AGILE-ПРОЕКТАХ

Аннотация. Вопрос образования архитектурного технического долга и управления им особенно актуален в промышленных разработках программных продуктов с использованием популярных гибких технологий. Архитектурный технический долг наиболее тяжелый в обслуживании, серьезно влияет на качество создаваемого кода и требует значительных затрат времени и усилий на изменение. Знание причин образования такого технического долга даст возможность сформулировать принципы, соблюдение которых позволит им управлять.

Ключевые слова: гибкие технологии; архитектурное решение; архитектурный технический долг.

Kazachenko T.A., Melekhin M.I.,
Litovka A.D., Batyrkaev A.G.
Aitov A.I.
ORCID: 0000-0003-4941-9104, PhD
Perm State Agro-Technological University named after Academician D.N. Pryanishnikov
Perm, Russia

ARCHITECTURAL TECHNICAL DEBT IN AGILE PROJECTS

Abstract. The issue of the formation of architectural technical debt and its management is especially relevant in the industrial development of software products using popular flexible technologies. Architectural technical debt is the hardest to maintain, seriously affects the quality of the generated code and requires a significant amount of time and effort to change. Knowing the reasons for the formation of such a technical debt will make it possible to formulate principles that will allow them to manage.

Keywords: agile technologies; architectural design decisions; architectural technical debt.

Согласно отчёту ScrumTrek (https://clck.ru/dVnaf) в России все чаще используют гибкие технологии в разработках программных продуктов. Как отмечает эта консалтинговая

CC BY

компания, главным привлекательным эффектом от внедрения Agile является высокая скорость поставки продукта бизнес-заказчику. Для гибкого подхода характерно инкрементальное проектирование, то есть архитектура наращивается в рамках каждого спринта итеративным и экспериментальным способом. Такой подход позволяет постоянно уточнять архитектуру разработки с изменением знаний команды разработчиков о ней. Поэтому в agile-проектах в отличие от waterfall-проектов не существует предварительной архитектуры разрабатываемого ПО. Об этом же говорят практикующие разработчики: «Нельзя прийти и создать сразу идеальный проект. Каждый день мы будем что-то узнавать о предметной области, и эти новые знания нужно сразу внедрять. Вначале это будут костыли, из которых копится технический долг, – и это нормально. Ведь мы не можем, получая новую информацию, каждый раз переписывать продукт заново. Бизнес нас не поймёт» Алексей Некрасов, лидер направления Python в МТС, программный директор направления Python в Skillbox (https://clck.ru/dVnXy). Таким образом, некоторые архитектурные решения (АР), принятые во время разработки программного обеспечения, влекут за собой технический долг, который может быть как преднамеренным, так и случайным.

Особенностью решений архитектурного проектирования является то, что они оказывают наибольшее влияние на качество программной системы, и после реализации их трудно изменить, это хорошо осознается и практиками, и исследователями. Именно поэтому сначала в рамках спринта архитекторы определяют архитектурное решение, не требующего технического обслуживания, соответствующее функциональным и нефункциональным требованиям заказчика. Такое поддерживаемое архитектурное решение может служить целью последующего рефакторинга альтернативных решений и работы с техническим долгом.

Если заранее известно, что новые требования к разрабатываемому ПО в последующих итерациях должны основываться на принятом в настоящее время архитектурном решении, если есть намерение повторно использовать разработанные программные компоненты в других продуктах, то целесообразно найти способ реализовать найденное поддерживаемое архитектурное решение, не влекущее технических долгов. В этом случае разработка стабильной и поддерживаемой, а не хрупкой и сложной в обслуживании архитектуры продукта может облегчить выполнение этих требований.

Однако, в силу ряда условий, присущих гибким технологиям разработки, о которых скажем ниже, как правило, не удаётся реализовать поддерживаемое решение, не влекущее технический долг. Поэтому параллельно с таким решением формируются альтернативные, в которых технический долг уже преднамеренно заложен. Архитектурные решения, влекущие за собой преднамеренное возникновение долгов, представляют собой особый тип АР с заранее известными рисками и отрицательными последствиями для ремонтопригодности и возможности развития программной системы. Работая над этими решениями, инженеры-программисты стремятся достичь в первую очередь двух основных целей:

• *выполнение видимых требований заказчика*, т. е. решение, влекущее за собой возникновение долгов, должно соответствовать всем функциональным требованиям заказчика и атрибутам качества во время выполнения;

CC BY

• *локализации возникшей технической задолженности*, т. е. архитектурное решение, вызывающее техническую задолженность, должно быть заключено в ограниченное количество компонентов. Это упростит его реинжиниринг до поддерживаемого решения в будущем выпуске.

Для выработки таких альтернативных решений делаются одноразовые прототипы MVP (Minimum Viable Product) пользовательских историй, позволяющие при минимальных затратах синхронизировать разработку с исследованиями реакции на продукт целевой аудитории. Окончательный выбор альтернативного АР делается на основе подтверждённых гипотез пользовательских историй, включённых в работу, по принципу «решить в последний ответственный момент», когда техническая неопределённость разработки существенно снижена. Наиболее часто встречающимися ограничениями при разработке архитектурных решений, влекущих технический долг, являются несколько факторов:

• *строгие временные ограничения*. Например, фиксированная дата доставки продукта на рынок до выпуска аналогичной разработки конкурентами;

• *доступность ресурсов*. Например, если одна из команд занята разработкой других функций и не может участвовать в разработке поддерживаемого архитектурного решения, то предпочтительным будет решение, влекущее за собой возникновение долгов;

• *лояльность к продуктам компании*. Инженеры-программисты могут быть вынуждены выбирать в качестве инструментов разработки продукты своей компании, даже если выбранные продукты не могут быть оптимальным решением проблемы. Это заставляет инженеров-программистов адаптировать и настраивать эти продукты, чтобы преодолеть их ограничения, что в свою очередь вызывает создание обходных путей и генерирует технический долг.

Согласование предварительно выбранного альтернативного архитектурного решения с соответствующими заинтересованными сторонами является одной из сложных задач по обслуживанию технического долга, которую приходится решать лицам, принимающим решения в проекте. Решение, связанное с возникновением долгов, может повлиять на скорость разработки в следующих итерациях из-за выплаты процентов по техническому долгу и стать причиной ошибок в будущих выпусках, которые повлияют на способность к масштабированию и внешнее качество продукта. Менеджеры без соответствующего технического опыта с трудом согласуют запланированные работы команды по обслуживанию технического долга, поскольку не видят долгосрочные затраты, связанные с ним. Поэтому инженеры-программисты, представляя план по перепроектированию решения, связанного с возникновением долгов, в поддерживаемое решение, определенное в начале процесса проектирования, обязаны сделать как можно более ясными для менеджеров риски принятия архитектурного решения, приносящего технический долг. В идеале люди, контролирующие бюджет проекта, должны хорошо разбираться в технических вопросах, тогда они будут знать, на какой риск они пойдут. Особенно понятными для бизнеса будут риски, которые могут привести к его остановке, – падающие сайты, системы поддержки клиентов, т. е. риски, связанные с разработкой front-end компонентов, которые делают бизнес «видимым» для

CC BY

клиентов, но в то же время подвержены наибольшим динамическим изменениям, поскольку содержат больше всего неопределённости. Именно в этих продуктах технический долг накапливается быстрее всего.

Наиболее трудно определить технический долг, полученный в результате архитектурных решений, случайно приводящих к такому долгу. В некоторых исследованиях указываются наиболее частые причины, влекущие появление такого долга [1]:

• *неглубокий архитектурный анализ из-за гибких и экспериментальных подходов.* Хотя гибкий подход эффективен для разработки и продажи продуктов, но анализ рынка, проектирование и разработка продукта ведутся параллельно и в короткий период времени. Поэтому увеличивается риск пропуска архитектурно-значимых требований и, как следствие, введение архитектурных решений, связанных с долгами, в следующих итерациях. В качестве неожиданных архитектурно-значимых требований могут быть не только функциональные требования (например, добавление новой функции к продукту), но и атрибуты качества, которые не имели достаточного приоритета при первоначальном проектировании архитектуры программного обеспечения. Например, неожиданное требование повторного использования компонента, изначально разработанного для использования одной командой, а затем необходимость использования того же компонент другими командами, которые также участвуют в разработке продукта, но уже в другом месте. Это технический долг, связанный с необходимостью поделиться компонентом с кем-то ещё;

• *отсутствие архитектурных навыков*. Для agile-проектов характерны небольшие продуктовые группы, состоящее из 7–20 человек, в которых разделение между архитектурой и программированием очень размыто. Каждый программист – это своего рода архитектор, независимо от того, хочет он этого или нет. Кроме того, официальный архитектор – это, как правило, разработчик-практик, у которого проектные навыки могут быть существенно ниже, чем навыки разработчика. Поэтому недостаточный уровень архитектурных навыков может быть причиной создания неоптимальных АР и, как следствие, возникновение непреднамеренного технического долга;

• *отсутствие подходящих альтернативных технологий*. Из-за отсутствия подходящих альтернатив при выборе технологического инструментария (например, библиотеки или фреймворка) инженеры-программисты иногда вынуждены выбирать решение с потенциальными недостатками, которые могут поставить под угрозу ремонтопригодность системы при будущих изменениях и нести технический долг;

• *отсутствие поддержки со стороны поставщиков технологий*. Поставщики технологий могут прекратить развитие и поддержку устаревших технологических решений. Использование таких устаревших инструментов при разработке ПО может создавать непреднамеренный технологический долг, который трудно преодолеть. Для предотвращения подобного архитектурного технического долга необходимо отслеживать приоритеты вендоров в отношении поставляемых инструментов;

• *отсутствие документации с обоснованием проектного решения.* В agile-проектах нередки изменения в составе группы разработки. При этом фиче-команды сосредоточены на

CC BY

создании нового функционала и мало времени уделяют документированию обоснования проектных решений. Поэтому новые члены команды не имеют возможности понять причины существующих АР в полной мере. Впоследствии это приводит к внесению изменений, которые непреднамеренно отклоняются от намеченной цели архитектуры и затрудняют реализацию, например, масштабирования разработки, поскольку архитектурная целостность является ключевым свойством системы для её будущих изменений.

Из вышесказанного следует, для создания в agile-проектах программного продукта, имеющего потенциал развития, а также для эффективного управления архитектурным техническим долгом целесообразно соблюдать ряд принципов:

• придерживаться инкрементального принципа создания архитектуры разрабатываемого ПО, выращивая её от спринта к спринту, стремясь сделать её не всеохватывающей, а лёгкой для рефакторинга, широко используя механизмы обеспечения гибкости: интерфейсы, полиморфизм, шаблоны проектирования, структуры внедрения зависимостей, указатели на функции и многое другое (https://clck.ru/dVnZM);

• подбирать в качестве архитектора программиста-эксперта, имеющего высокую квалификацию проектирования, способного выбрать сбалансированное архитектурное решение на каждом этапе создания кода;

• выполнять регулярное ревью кода для ранней идентификации неявного технического долга, планировать рефакторинг и выплату технического долга, разбивая его на малые задачи, чётко обозначенные целями и сроками. Это позволит не сильно замедлить скорость поставки продукта, сохранить мотивацию команды на исправление написанного кода, а также снижать объем технического долга или хотя бы держать его на фиксированном уровне;

• при согласовании с бизнесом затрат на рефакторинг и работ по техническому долгу говорить на его языке: объяснять финансовые риски и приводить примеры из жизни. Поскольку бизнес имеет потенциал прибыли от развития продукта гораздо больший, чем затраты на обслуживание технического долга, то для бизнеса борьба с техническим долгом – это не убытки, а отложенная ценность, способствующая повышению выгоды;

• в каждом спринте после реализации функционала регулярно проводить воркшопы по SAD (System Architecture Documentation) для создания адаптивной документации по системной архитектуре, что позволит участникам команды лучше понять разработанную архитектуру и находить решения для её улучшения.

В agile –проекте, как, впрочем, и в других проектах, основными принципом, позволяющими, удерживать технический долг на низком уровне, является работа на опережение, не позволяющая проблеме разрастаться.

Литература

1. Soliman M., Avgeriou P., Li Y. Architectural design decisions that incur technical debt – An industrial case study // Information and Software Technology. 2021. Vol. 139, 106669. https://doi.org/10.1016/j.infsof.2021.106669

© Казаченко Т.А., Мелехин М.И., Литовка А.Д., Батыркаев А.Г., Аитов А.И., 2022

УДК 004.9
https://doi.org/10.36906/AP-2022/39

Карифуллаева И.С.
Оренбургский государственный университет
г. Оренбург, Россия

РАЗРАБОТКА CRM-СИСТЕМЫ IT-ПРЕДПРИЯТИЯ МАЛОГО БИЗНЕСА

Аннотация. Предложен проект CRM-системы. Описана диаграмма классов и кооперативные диаграммы взаимодействия объектов.

Ключевые слова: CRM-система; IT-предприятие; малый бизнес.

Karifullaeva I.S.
Orenburg State University
Orenburg, Russia

DESIGNING CRM-SYSTEM OF IT-COMPANY FOR SMALL BUSINESS

Abstract. An article focused on designing of CRM-system for IT-company. A class diagram and collaboration diagrams are described.

Keywords: CRM-system; IT-company; small business.

Основная задача CRM-системы – повысить эффективность бизнес-процессов по поиску и удержанию потребителей, как в продажах, так и в сервисном обслуживании. Управлять взаимоотношениями – значит привлекать новых покупателей и параллельно превращать нейтральных клиентов в лояльных [1; 2].

CRM-система состоит из двух частей: базы данных со всей информацией о клиентах, а также приложений, выполняющих следующие функции: сбор информации о клиентах; обработка этой информации; а также её анализ и (или) вывод в удобном для пользователя виде.

В данной работе рассматривается проектирование CRM-системы для малого IT-предприятия, осуществляющего оптовую продажу компонентов компьютеров, аксессуаров и проводов.

Целью проектирования системы является:

– обеспечение сбора, хранения и обработки исходной информации, необходимой для подготовки отчётности;

– улучшение качества взаимодействия с существующими и будущими потенциальными клиентами;

– сокращение времени сбора и первичной обработки исходной информации;

– сокращение времени, затрачиваемое на информационно-аналитическую деятельность.

В системе предлагается выделить следующие функциональные подсистемы:

CC BY

– подсистема сбора, обработки и загрузки данных, которая предназначена для реализации процессов сбора данных из систем источников, приведения указанных данных к виду, необходимому для наполнения подсистемы хранения данных;

– подсистема формирования и визуализации отчётности;

– подсистема взаимодействия с клиентами;

– подсистема учёта продаж.

На рисунке 1 представлена диаграмма классов [3]. Основными сущностями являются «Сотрудники», «Клиенты», «Продажи», «Спрос», «Каталог товаров».

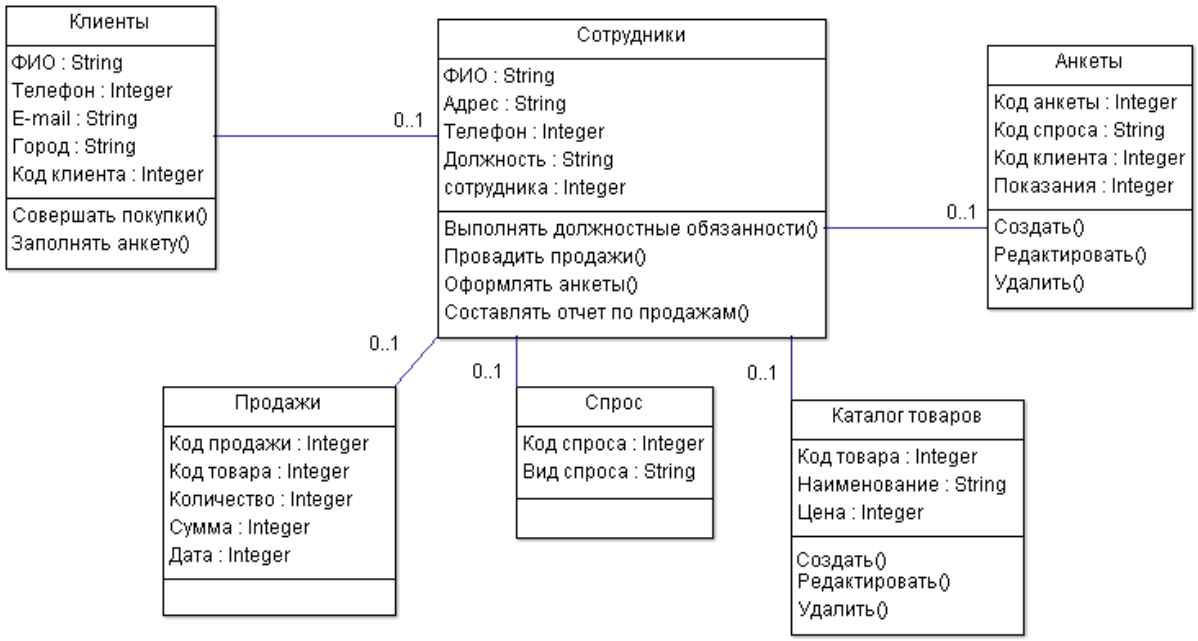


Рис. 1. Диаграмма классов

На кооперативной диаграмме в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта и его класс (рис. 2).

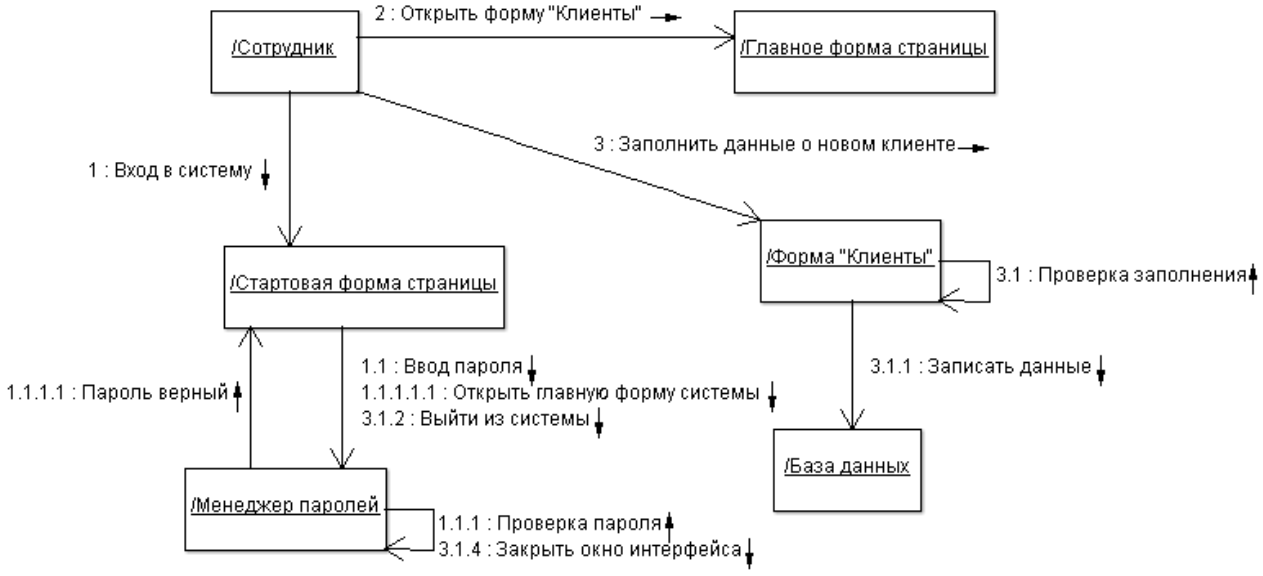


Рис. 2. Кооперативная диаграмма «Внесение информации о клиентах»

Как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. Над соединительными линиями располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации [4].

На рисунке 3 представлена диаграмма процесса проведения анализа продаж.

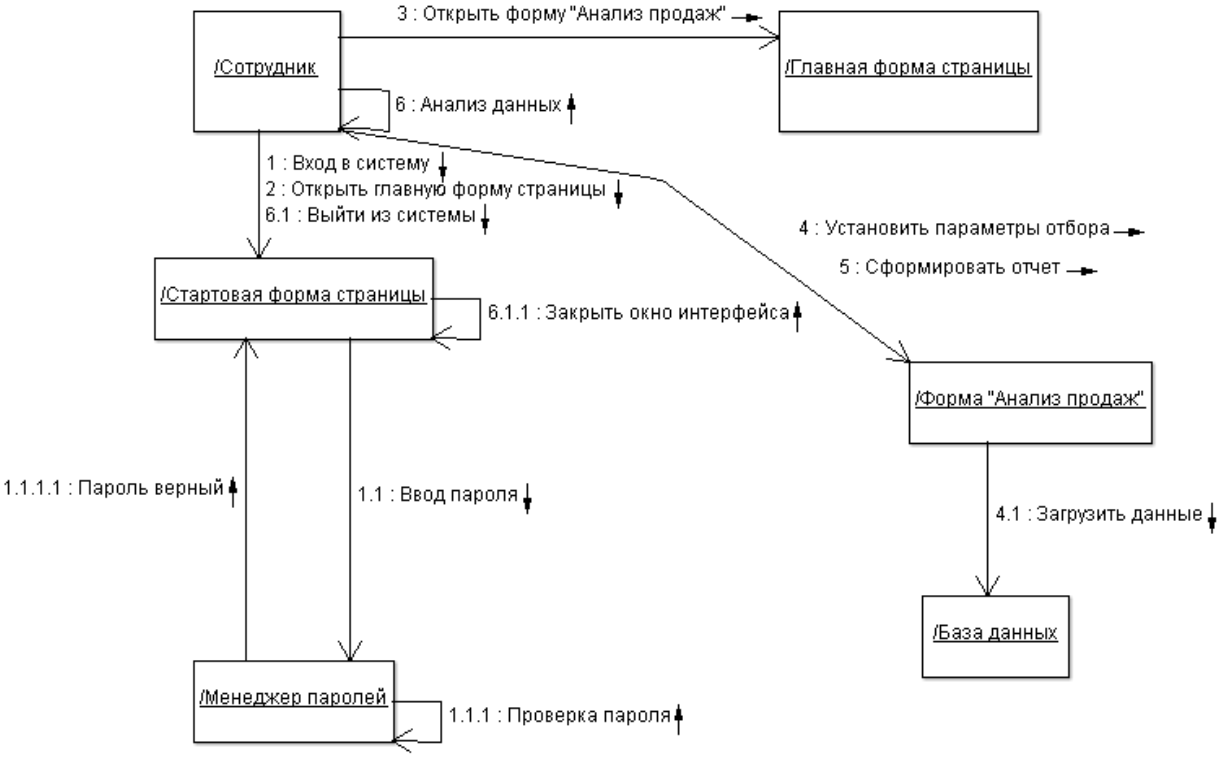


Рис. 3. Кооперативная диаграмма «Проведение анализа продаж»

Таким образом, в заключение, среди выгод от внедрения CRM-систем можно выделить:

– расширение возможностей компании по привлечению новых клиентов и удержанию уже приобретённых;

– повышение ценности клиента для компании;

– повышение уровня услуг без привлечения дополнительных расходов.

Инвестиции в CRM-системы можно рассматривать не столько как инвестиции в оборудование и программное обеспечение, сколько как инвестиции в долговременные отношения с клиентами.

Литература

1. Успенский В.И. Интернет-маркетинг. СПб.: СПГУЭиФ, 2003. 350 с.

2. Дейян А., Троадек А. Стимулирование сбыта и реклама на месте продажи. М.: Прогресс, 2014. 152 с.

3. Фаулер М. UML. Основы. СПб: Символ Плюс, 2004. 192 с.

4. Корчагин Р.Н., Поленова Т.М., Сафонова Т.Е. Информационные системы в экономике. Методические рекомендации к практичеким занятиям по курсу. М.: РАГС, 2009. 101 с.

© Карифуллаева И.С., 2022

УДК 004.032.26
https://doi.org/10.36906/AP-2022/40

Катермина Т.С.
ORCID: 0000-0003-4637-8861, канд. техн. наук
Туманов В.И., Зинченко С.С.
Нижневартовский государственный университет
г. Нижневартовск, Россия

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОПРЕДЕЛЕНИЯ COVID-19

Аннотация. Данная статья посвящена анализу нейронных сетей, их архитектур и моделей, наборов данных, использующихся при решении задачи по определению коронавирусной инфекции в организме человека без применения исследований методами полимеразной цепной реакции и иммуноферментного анализа, а также лабораторных исследований на антитела IgGи IgM.

Ключевые слова: нейронная сеть; SARS-CoV-2.

Katermina T.S.
ORCID: 0000-0003-4637-8861, Ph.D.
Tumanov V.I., Zinchenko S.S.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

USAGE OF NEURAL NETWORKS FOR DIAGNOSING COVID-19

Abstract. This article contains information about neural networks, its architectures and models, datasets that helps to tackle the task of diagnosing coronavirus disease in the body of the person without the usage of tests via polymerase chain reaction and enzyme-linked immunosorbent assay methods, and also without tests for antibodies IgG and IgM.

Keywords: neural network; SARS-CoV-2.

Согласно данным Всемирной Организации Здравоохранения, с 31 декабря 2019 года по 14 ноября 2021 года включительно, в мире было зарегистрировано 253163330 (двести пятьдесят три миллиона сто шестьдесят три тысячи триста тринадцать) подтверждённых случаев заболевания COVID-19 и 5098174 (пять миллионов девяносто восемь тысяч сто семьдесят четыре) смерти, причиной которых являлась коронавирусная инфекция (https://clck.ru/dneku). В России, с начала отчёта статистики в марте 2020 года и 14 ноября этого года, было выявлено 9109094 (девять миллионов сто девять тысяч девяносто четыре) случаев заболевания и 256567 (двести пятьдесят шесть тысяч пятьсот шестьдесят семь) смертей (https://clck.ru/dneoq), что составляет 3,6% и 5,03% от мировой статистики. Россия находится

CC BY

на пятом месте по количеству смертей в мире, находясь после США, Бразилии, Индии и Мексики (https://clck.ru/dneku).

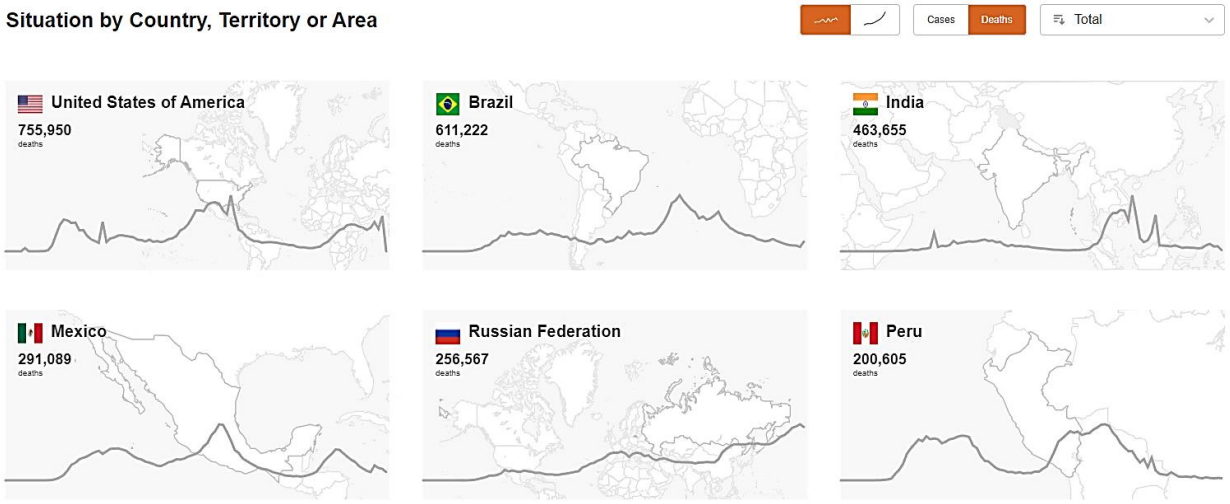


Рис. 1. Страны с наибольшим числом смертей от COVID-19

Наилучшим способом увеличения эффективности лечения и снижения негативных последствий заболевания является его точная и быстрая диагностика. При большом количестве пациентов повышается риск появления ошибки при диагностике. Для дополнительной проверки можно привлекать помощь со стороны в виде нейронной сети, анализирующей данные рентгенографии и/или компьютерной томографии, которые являются одним из важнейших аспектов диагностики коронавируса.

Для обучения нейронной сети необходимо правильно подобрать набор данных (датасет), который должен максимально удовлетворять параметрам задачи и другим требованиям к данным для обучения.

После середины 2020 года в свободном доступе появилось множество датасетов, которые включают в себя результаты компьютерной томографии (КТ) пациентов, имеющих подтверждённый COVID-19. Среди таких датасетов можно отметить COVIDxCT-A. Он основан на 194922 снимках КТ у 3745 пациентов. Данные были собраны у пациентов нескольких стран: Китай, Иран, Италия, Турция, Украина, Бельгия, Австралия, Афганистан, Шотландия, Ливан, Англия, Алжир, Перу, Азербайджан и Россия; медианный возраст пациентов (при расчёте учитывались лишь те случаи, где возраст был известен) –51 год, общий диапазон возрастов 0–93 года.

Следующим этапом работы с нейронными сетями является выбор ее архитектуры и модели.

CC BY

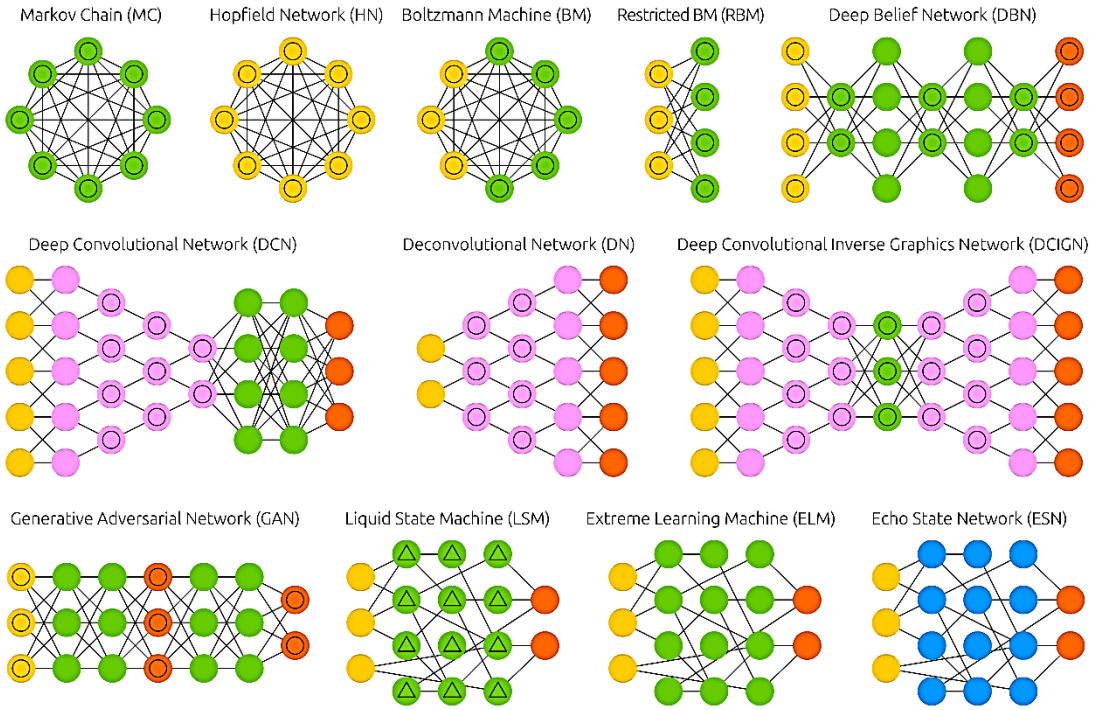


Рис. 2. Различные архитектуры нейронных сетей

Имеется большое количество различных архитектур нейронных сетей и подходят они для разных задач. Для распознавания изображений оптимальным вариантом является свёрточная нейронная сеть [3], дополнительно можно добавить полносвязные слои для принятия решений. Приведём несколько примеров таких нейронных сетей и результаты опубликованных тестирований.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Рис. 3. Модель сверточной нейронной сети VGG-19.

CC BY

На рисунке 3 в колонке E представлена популярная модель нейронной сетиVGG-19. Она имеет несколько наборов сверточных слоёв (conv), слои выборки (maxpool) и полносвязные слои (FC) [1].

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Рис. 4. МодельResNet-50

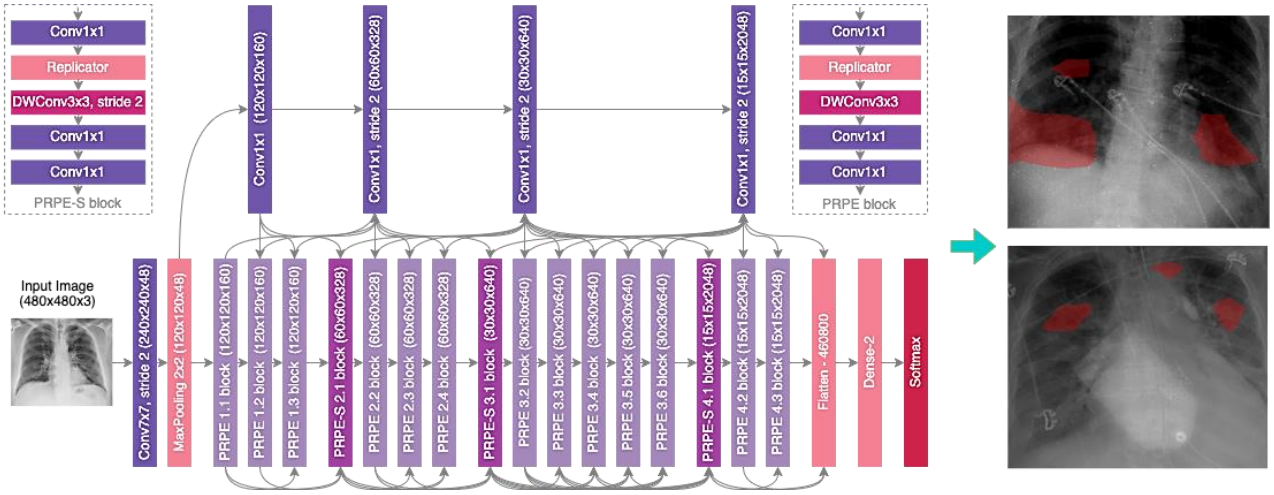


Рис. 5 Модель COVID-Net

На рисунках 4–5 представлены ещё две модели – ResNet-50 и COVID-Net. Основными отличиями, от указанной ранее, является количество слоёв, размер ядер свёртки и вид слоя выборки. Протестировав данные на этих моделях, можно получить достаточно высокую точность работы нейронных сетей (рис. 6) [2].

Architecture	Params (M)	MACs (G)	Acc. (%)
VGG-19	20.37	89.63	83.0
ResNet-50	24.97	17.75	90.6
COVID-Net	**11.75**	**7.50**	**93.3**

Рис. 6. Производительность протестированных моделей на тестовом наборе данных COVIDx.

Можно отметить, что COVID-Net обеспечивает хорошую точность, достигая 93,3%, тем самым подчёркивая эффективность использования совместного проектирования человека и машины для создания архитектур глубоких нейронных сетей с учётом задач, данных и требований. Это особенно важно для таких случаев, как выявление заболеваний, когда новые

случаи и новые данные собираются непрерывно, и необходима способность быстро генерировать новые архитектуры глубоких нейронных сетей.

Литература

1. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // arXiv:1409.1556 [cs]. 2015. https://doi.org/10.48550/arXiv.1409.1556

2. Wang L., Lin Z. Q., Wong A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images // Scientific Reports. 2020. Vol. 10. Issue 1. C. 19549. https://doi.org/10.1038/s41598-020-76550-z

3. Разинкин В.Б., Катермина Т.С. Распознавание лица по фотографии // International Journal of Advanced Studies. 2018. № 1-2. С. 171-180.

© Катермина Т.С., Туманов В.И., Зинченко С.С., 2022

УДК 004.032.26
https://doi.org/10.36906/AP-2022/41

Катермина Т.С.
ORCID: 0000-0003-4637-8861, канд. техн. наук
Лазоренко Е.В.
ORCID 0000-0001-5881-4304
Нижневартовский государственный университет
г. Нижневартовск, Россия

СИСТЕМА АВТОМАТИЧЕСКОГО КОНТРОЛЯ АВТОМОЙКИ ПРИ ПОМОЩИ НЕЙРОННОЙ СЕТИ

Аннотация. Статья посвящена проблеме построения автоматического контроля корректности работы системы автомойки, подсчёта выезжающих машин из бокса автомойки и сравнивания с контрольным числом. Рассматриваются основные библиотеки и модули для нейронных сетей. В статье также анализируются архитектуры нейронных сетей с учетом пригодности к решению поставленной задачи.

Ключевые слова: нейронная сеть; распознавание образов; анализ видеопотока; архитектура нейронной сети.

Katermina T.S.
ORCID: 0000-0003-4637-8861, Ph.D.
Lazorenko Ye.V.
ORCID 0000-0001-5881-4304
Nizhnevartovsk State University
Nizhnevartovsk, Russia

SYSTEM FOR AUTOMATICALLY CONTROLLING A CAR WASH WITH THE AID OF A NEURAL NETWORK

Abstract. The article is devoted to the problem of constructing automatic control of the correctness of the operation of a car wash system, counting of departing cars from a car wash box and comparison with a control number. Main libraries and modules for neural networks are considered. The article also analyzes neural network architectures according to their suitability for the task.

Keywords: neural network; pattern recognition; video stream analysis; neural network architecture.

Популярность нейронных сетей [1] в настоящий момент настолько велика, что их повсеместное использование приводит к появлению все новых фреймворков и библиотек, упрощающих разработку интеллектуальных информационных систем. Частично уже

роботизируемый процесс работы автомойки также может быть подвергнут модернизации системами с использованием элементов искусственного интеллекта.

Любое программное обеспечение может быть подвержено взломам и обходам. Как и любое ПО, система роботизированной автомойки (робомойка) не может быть защищена на 100%. Систему можно обойти как на программном, так и на физическом уровне. Для того чтобы система робомойки была более защищённой, разрабатывается нейронная сеть, которая позволит подсчитывать выезжающие машины и сравнивать результат с числом, подсчитанным робомойкой.

Модули и Фреймворки для нейронных сетей

Для моделирования и быстрого прототипирования нейронных сетей используются следующие Фреймворки и модули.

TensorFlow – комплексная платформа для машинного обучения с открытым исходным кодом. Как и большинство Фреймворков и модулей глубокого обучения, TensorFlow имеет API на Python поверх механизма C и C ++, что ускоряет его работу.

Платформа представляет собой понятный высокоуровневый API-интерфейс, например, Keras с быстрым выполнением, что обеспечивает немедленную итерацию модели и простую отладку.

Для запуска модели машинного обучения на смартфоне или IoT-устройстве, есть среда глубокого обучения с открытым исходным кодом TensorFlow Lite (https://clck.ru/YuBSx).

Если необходимо создать и обучить модель на JavaScript, а после развернуть её в браузере или на Node.js, то можно воспользоваться библиотекой TensorFlow.js.

Таблица 1

Плюсы и минусы TensorFlow

Плюсы	Минусы
Возможность использования в производственных процессах	Отсутствие четких шаблонов проектирования
Оптимизирует использование вычислительных ресурсов.	Необходимо контролировать используемую видеопамять.
Активное и обширное сообщество разработчиов	Недружелюбный фреймворк.
	Придерживается своих стандартов
	Недостаточно разработанная документация.
	Сложен в использовании и освоении.

Проекты, которые используют Фреймворк TensorFlow:

– DeepSpeech – система распознавания речи.

– Mask R-CNN – модель, которая генерирует ограничительные рамки и маски сегментации для каждого объекта на изображении.

– BERT – предобученная нейронная сеть, используемая для решения задач обработки естественного языка.

PyTorch – библиотека с открытым исходным кодом, на Python, обеспечивающая тензорные вычисления с GPU-ускорением, подобно NumPy. Данный фреймворк больше подходит для быстрого прототипирования в исследованиях, а также для любителей и небольших проектов.

CC BY

За счёт глубокой интеграции Фреймворка с кодом C++ разработчики могут программировать на C и C++ с помощью API-расширения на основе FFI для Python.

PyTorch отличается от других Фреймворков машинного обучения тем, что здесь не используются статические расчётные графы – определяемые заранее, как в TensorFlow, Caffe2 или MXNet. Расчетные графы в PyTorch динамические и определяются во время работы. Таким образом, при каждом вызове слоев в модели PyTorch динамически определяется новый расчётный граф (https://clck.ru/YuBSx).

Таблица 2

Плюсы и минусы PyTorch

Плюсы	Минусы
Имеет множество модульных элементов, которые легко комбинировать.	Необходимо самостоятельно писать тренировочный код.
Легко создавать собственные типы слоев и работать на GPU	Недостаточный уровень разработки документации.
Имеет широкий выбор предварительно обученных моделей	

Проекты, которые используют Фреймворк PyTorch:

– PyText – библиотека для обработки устной и письменной речи.

– vid2vid – генеративная нейросеть для подмены деталей и свойств видеопотока.

– pix2pix – алгоритм, который превращает пользовательские наброски в фотографии.

Keras – библиотека для Python с открытым исходным кодом, которая позволяет легко создавать нейронные сети. Библиотека совместима с TensorFlow, Microsoft Cognitive Toolkit, Theano и MXNet. Tensorflow и Theano являются наиболее часто используемыми численными платформами на Python для разработки алгоритмов глубокого обучения, но они довольно сложны в использовании (https://clck.ru/YuBSx).

Таблица 3

Плюсы и минусы Keras

Плюсы	Минусы
Удобен в использовании.	Не подходит для больших проектов.
Лёгок в освоении.	
Быстроразвивающийся фреймворк.	
Хорошая документация.	
Встроен в TF.	

Проекты, которые используют Фреймворк Keras:

– Mask R-CNN – модель, которая генерирует ограничительные рамки и маски сегментации для каждого объекта на изображении

– face_classification – алгоритм для распознавания лиц в режиме реального времени и классификации эмоций и пола.

– YOLOv3 – нейронная сеть для обнаружения объектов в режиме реального времени.

Darknet – фреймворк с открытым исходным кодом, написанный на языке C с использованием программно-аппаратной архитектуры параллельных вычислений CUDA.

CC BY

Фреймворк написан на C и не имеет другого API. По этой причине, будет сложно перейти на другой язык программирования. Фреймворк не рекомендуется использовать для сложных проектов (https://clck.ru/YuBSx).

Таблица 4

Плюсы и минусы Darknet

Плюсы	Минусы
Простой	Кроме задач с обнаружением больше нигде не используется
Быстрый	Не рекомендуется для больших проектов
Удобный	Плохая документация

Проекты, которые используют Фреймворк Darknet:

– YOLOv3 – нейронная сеть для обнаружения объектов в режиме реального времени.

– Tiny-YOLO 3 – компактная нейронная сеть для обнаружения объектов.

XGBoost – это Фреймворк с открытым исходным кодом, который предлагает систему градиентного бустинга для C++, Java, Python, R, Julia. Он разработан для обеспечения высокой эффективности, гибкости и портативности.

XGBoost фокусируется на скорости вычислений и производительности модели и подходит для решения задач регрессии, классификации и упорядочивания. Если данные можно представить в виде таблицы, то точность и производительность будут существенно выше, чем у DeepLearning-решений.

Фреймворк совместим с операционными системами Windows, Linux и OS X, а также поддерживает кластеры AWS, Azure и Yarn, хорошо работает с Flink, Spark (https://clck.ru/YuBSx).

Таблица 5

Плюсы и минусы XGBoost

Плюсы	Минусы
Очень быстрый и удобный инструмент для тренировки моделей типа «дерево решений»	Узко специализированный
Отлично подходит для проверки гипотез	

Caffe – Фреймворк, реализованный на C++, имеет хороший интерфейс Python. Он поддерживает CNN и сети прямого распространения, а также подходит для тренировки моделей, обработки картинок и усовершенствования существующих нейронных сетей.

Таблица 6

Плюсы и минусы Caffe

Плюсы	Минусы
Предлагает предварительно обученные модели для создания демонстрационных приложений	Плохая документация
Быстрый, масштабируемый	Сложность компиляции
Отлично работает с другими фреймворками, такими как PyTorch, и в конечном итоге будет интегрирован в PyTorch 1.0	Ограниченная поддержка сообществом
Обеспечивает оптимизированную работу с сервером	

Deeplearning4j – библиотека программ на языке Java и Scala, используемая как Фреймворк для глубокого обучения. Включает реализацию ограниченной машины Больцмана,

CC BY

глубокой сети доверия, глубокого авто кодировщика, стекового авто кодировщика с фильтрацией шума, рекурсивной тензорной нейронной сети, word2vec, doc2vec и GloVe. Она также поддерживает различные типы нейронных сетей (таких как CNN, RNN, RNTN, или LTSM).

Этот Фреймворк имеет большой потенциал в области обработки естественного языка, распознавания изображений, поиске уязвимостей и анализе текста (https://clck.ru/YuBSx).

Таблица 7

Плюсы и минусы Deeplearning4j

Плюсы	Минусы
Надёжный, гибкий и эффективный;	Большие затраты по времени при разработке проекта
Может обрабатывать большие объёмы данных без потери скорости;	
Работает с Apache Hadoop и Spark, на распределённых CPU или GPU;	
Имеет хорошую документацию;	
Имеет community и enterprise версию.	

Таблица 8

Таблица характеристик (https://clck.ru/Ej72L)

Наименование	Открытое ПО	Платформа	Язык	Интерфейс	Поддержка CUDA	RNN	CNN
TensorFlow	Да	Linux, Mac OS X, Windows	C++, Python	Python, C/C++, Java, Go	Да	Да	Да
Torch	Да	Linux, Mac OS X, Windows, Android, iOS	C, Lua	Lua, LuaJIT, C, библиотека для C++/OpenCL	Да	Да	Да
Keras	Да	Linux, Mac OS X, Windows	Python	Python	Да	Да	Да
Caffe	Да	Linux, Mac OS X, Windows	C++	Python, MATLAB	Да	Да	Да
Deeplearning4j	Да	Linux, Mac OS X, Windows, Android	Java	Java, Scala, Clojure, Python (Keras)	Да	Да	Да
Darknet	Да		C	C	Да	Да	Да
XGBoost	Да	Windows, Linux и OS X, AWS, Azure и Yarn, хорошо работает с Flink, Spark	C++, Java, Python, R, Julia				

Выбор архитектуры нейронной сети для поставленной задачи.

Главные задачи системы подсчёта машин – задача детектирования, локализации и классификации объектов, которые попадают в область видимости [2]. Проблема состоит в определении события, когда машина выезжает из бокса автомойки, определение марки машины и ведения подсчёта выезжающих машин из бокса, а также сравнивание исходного количества машин, подсчитанных с помощью нейронной сети, с числом, которое определило программное обеспечение робомойки. Видеопоток – простой набор изображений, следующих друг за другом. Для распознавания изображения используется архитектура нейронных сетей YOLO, R-CNN, Fast R-CNN, Faster R-CNN и Mask R-CNN.

CC BY

Архитектура R-CNN (Region-Based Convolutional Neural Network) Находит объекты на изображении и разбивает их на регионы с помощью метода *selective search*. Метод selective search сегментирует изображение на регионы, которые затем в ходе итерирования с помощью жадного алгоритма объединяются в более крупные регионы. Далее извлекает признаки полученных регионов с помощью нейронных сетей. Классифицирует обработанные признаки методом опорных векторов (SVM, Support Vector Machine), после чего происходит уточнение границы регионов с помощью линейной регрессии (https://clck.ru/dnerj).

Архитектура Fast R-CNN является следующим шагом в развитии ветки архитектуры R-CNN. Изображение подаётся на вход нейронной сети и обрабатывается selective search. В итоге, имеем карту признаков и регионы предположительных объектов. Координаты регионов предположительных объектов преобразуются в координаты на карте признаков. Полученная карта признаков с регионами передаётся слою RoI (Region of Interest) polling layer. Здесь на каждый регион накладывается сетка. Затем применяется MaxPolling для уменьшения размерности. Полученные признаки подаются на вход полно связного слоя (Fully-conectedlayer), который передаётся двум другим полно связным слоям. Первый с функцией активацией softmax определяет вероятность принадлежности классу, второй – границы области предположительного объекта (https://clck.ru/dnerj).

Следующим шагом стало улучшение архитектуры Fast R-CNN до версии Faster R-CNN. Архитектура Faster R-CNN реализована следующим образом: Изображение подаётся на вход нейронной сети. Карты признаков обрабатываются слоем RPN (Region Proporsal Networks). Здесь скользящее ядро проходится по карте признаков. Центр ядра связан с центром якорей. Якорь – такая область, имеющая разные соотношения сторон и разные размеры. На основе метрики IoF (intersection-over-union), пересечения якорей и реальных размеченных прямоугольников, выносится результат о текущем регионе – есть объект или нет. Далее используется алгоритм FastCNN: карты признаков с полученными объектами передаются слою RoI с последующей обработкой полно связных слоёв и классификацией, а также с определением смещения регионов потенциальных объектов (https://clck.ru/dnerj).

Архитектура Mask R-CNN основана на Faster R-CNN и улучшает её путём добавления ещё одной ветки, которая предсказывает положение маски, покрывающей найденный объект определённым цветом. Таким образом решается задача сегментации, которая позволяет визуализировать результат.

YOLO или You Only Look Once в настоящее время самая популярная архитектура CNN, которая используется для распознавания множественных объектов на изображении. Главной особенностью архитектуры является то, что большинство систем применяют CNN несколько раз к разным областям изображения, тогда как в YOLO CNN применяется один раз ко всему изображению сразу. Сеть делит изображение на клетки и предсказывает bounding boxes и вероятности того, что там есть искомый объект для каждого участка (https://clck.ru/dnexA).

Так как машины, которые выезжают из бокса могут и заезжать обратно задним ходом, для данной задачи отлично подойдёт архитектура свёрточной нейронной сети YOLO. Эта

CC BY

архитектура не только быстродейственна, но и позволяет учитывать контекст из видеопотока. Это позволяет ей запомнить предыдущий результат движения автомобиля.

Предполагаемая система действует следующим образом:

1. машина заезжает в бокс, после срабатывания датчиков, происходит процесс мойки;

3. во время заезда и выезда срабатывают датчики, что позволяет подсчитать количество машин. Также подсчёт ведётся при оплате;

3. после окончания мойки, машина выезжает. Камера с системой нейронной сети, определяет, что машина начала выезжать из бокса, и общее количество обслуженных машин увеличивается;

4. в конце дня сравнивается количество обслуженных машин, посчитанное системы видеонаблюдения и физической системой робомойки. Т. к. сеть может классифицировать марку машины, то она может вести подсчёт въезжающих и выезжающих машин. Тем самым увеличивая надёжность защиты от взлома физической системы робомойки.

Литература

1. Шолле Ф. Глубокое обучение на python. СПб.: Питер, 2021. 400 с.

2. Разинкин В.Б., Катермина Т.С.. Распознавание лица по фотографии // International Journal of Advanced Studies. 2018. № 1-2. С. 171-180.

© Катермина Т.С., Лазоренко Е.В., 2022

УДК 519.852.6
https://doi.org/10.36906/AP-2022/42

Мамедли Р.Э.
канд. физ.-мат. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия

УПРАВЛЕНИЯ С МЕТОДАМИ БЛОКИРОВКИ В ТРАНЗАКЦИЯХ

Аннотация. В работе рассматриваются различные методы управления блокировками в транзакциях. Обоснована необходимость настройки производительности SQL и настройки производительности СУБД. Приведены примеры управления с методами блокировки в транзакциях для принятия важных решений по повышению эффективности обработки запросов.

Ключевые слова: транзакция; настройка производительности базы данных; методы блокировки; степень детализации блокировки; двоичная блокировка; разделяемая блокировка; исключительная блокировка; двухфазная блокировка; тупики.

Mammadli R.E.
Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

CONTROLS WITH BLOCKING METHODS IN TRANSACTIONS

Abstract. The paper discusses various methods for managing locks in transactions. The necessity of SQL performance tuning and DBMS performance tuning is substantiated. Examples of management with blocking methods in transactions are given for making important decisions to improve the efficiency of query processing.

Keywords: transaction; database performance tuning; locking methods; lock granularity; binary lock; shared lock; exclusive lock; two-phase lock; deadlocks.

Настройка производительности базы данных является критически важной темой, однако она обычно получает минимальное освещение в учебной программе по базам данных. Большинство баз данных, используемых в университете, имеют только несколько записей на таблицу. В результате основное внимание часто уделяется тому, чтобы запросы SQL выполняли намеченную задачу без учёта эффективности процесса запроса. Фактически, даже самая эффективная среда запросов не даёт видимых улучшений производительности по сравнению с наименее эффективной средой запросов, когда запрашивается только 20 или 30 строк (записей) таблицы. Отсутствие внимания к эффективности запросов может привести к

CC BY

недопустимо медленным результатам в реальном мире, когда запросы выполняются для десятков миллионов записей.

Методы блокировки являются одним из наиболее распространённых методов, используемых в управлении параллелизмом, поскольку они облегчают изоляцию элементов данных, используемых при одновременном выполнении транзакций. Блокировка гарантирует исключительное использование элемента данных для текущей транзакции. Другими словами, транзакция T2 не имеет доступа к элементу данных, который в данный момент используется транзакцией T1. Транзакция получает блокировку до доступа к данным; блокировка снимается, когда транзакция завершается, так что другая транзакция может заблокировать элемент данных для его исключительного использования. Эта серия блокирующих действий предполагает, что параллельные транзакции могут пытаться манипулировать одним и тем же элементом данных в одно и то же время. Использование блокировок, основанных на предположении о вероятности конфликта между транзакциями, обычно называют пессимистической блокировкой.

Согласованность данных не может быть гарантирована во время транзакции; база данных может находиться во временном несовместимом состоянии при выполнении нескольких обновлений. Следовательно, блокировки необходимы для предотвращения чтения несогласованными данными другой транзакции.

Современные СУБД автоматически инициируют и применяют процедуры блокировки. Вся информация о блокировке обрабатывается менеджером блокировок, который отвечает за назначение и контроль блокировок, используемых транзакциями.

Степень детализации блокировки показывает уровень использования блокировки. Блокировка может осуществляться на следующих уровнях: база данных, таблица, страница, строка или даже поле (атрибут).

При блокировке уровня базы данных вся база данных блокируется, что предотвращает использование любых таблиц в базе данных транзакцией T2 во время выполнения транзакции T1. Этот уровень блокировки хорош для пакетных процессов, но не подходит для многопользовательских СУБД.

При блокировке на уровне таблицы вся таблица блокируется, предотвращая доступ к любой строке транзакции T2, пока транзакция T1 использует таблицу. Если транзакция требует доступа к нескольким таблицам, каждая таблица может быть заблокирована. Однако две транзакции могут обращаться к одной и той же базе данных, если они обращаются к разным таблицам. Блокировки на уровне таблицы, хотя и менее строгие, чем блокировки на уровне базы данных, вызывают пробки, когда многие транзакции ожидают доступа к одной и той же таблице. Такое условие особенно утомительно, если блокировка вызывает задержку, когда разные транзакции требуют доступа к разным частям одной и той же таблицы, то есть когда транзакции не будут мешать друг другу. Следовательно, блокировки на уровне таблицы не подходят для многопользовательских СУБД.

При блокировке на уровне страницы СУБД блокирует всю страницу диска. Страница диска, или страница, является эквивалентом дискового блока, который может быть описан как

CC BY

непосредственно адресуемый раздел диска. Страница имеет фиксированный размер, например, 4КБ, 8КБ или 16КБ. Например, если необходимо записать только 73 байта на страницу 4КБ, всю страницу необходимо прочитать с диска, обновить в памяти и записать обратно на диск. Таблица может занимать несколько страниц, а страница может содержать несколько строк одной или нескольких таблиц. В настоящее время блокировки на уровне страниц являются наиболее часто используемым методом блокировки для многопользовательских СУБД.

Блокировка на уровне строк гораздо менее ограничительная. СУБД позволяет одновременным транзакциям получать доступ к разным строкам одной и той же таблицы, даже если строки расположены на одной странице. Хотя подход блокировки на уровне строк улучшает доступность данных, его управление требует больших накладных расходов, поскольку существует блокировка для каждой строки в таблице базы данных, участвующей в конфликтующей транзакции. Современные СУБД автоматически повышают блокировку с уровня строки до уровня страницы, когда сеанс приложения запрашивает несколько блокировок на одной странице.

Блокировка на уровне поля позволяет одновременным транзакциям получать доступ к одной и той же строке, если они требуют использования разных полей (атрибутов) в этой строке. Хотя блокировка на уровне поля однозначно обеспечивает наиболее гибкий многопользовательский доступ к данным, она редко реализуется в СУБД, поскольку требует чрезвычайно высокого уровня издержек компьютера и потому что блокировка на уровне строк гораздо более полезна на практике.

Независимо от уровня блокировки, СУБД может использовать разные типы или режимы блокировки: двоичная или разделяемая/исключительная.

Двоичная блокировка имеет только два состояния: заблокировано (1) или разблокировано (0). Если объект, такой как база данных, таблица, страница или строка, заблокирован транзакцией, никакая другая транзакция не может использовать этот объект. Если объект разблокирован, любая транзакция может заблокировать объект для его использования. Каждая операция с базой данных требует, чтобы уязвимый объект был заблокирован. Как правило, транзакция должна разблокировать объект после его завершения. Поэтому каждая транзакция требует операции блокировки и разблокировки для каждого элемента данных, к которому осуществляется доступ. Такие операции автоматически управляются и планируются СУБД; пользователь не блокирует и не разблокирует элементы данных.

Исключительная блокировка существует, когда доступ зарезервирован специально для транзакции, которая заблокировала объект. Исключительная блокировка должна использоваться, когда существует вероятность конфликта. Общая блокировка существует, когда параллельным транзакциям предоставляется доступ на чтение на основе общей блокировки. Общая блокировка не вызывает конфликта, если все параллельные транзакции доступны только для чтения.

CC BY

Общая блокировка выдаётся, когда транзакция хочет прочитать данные из базы данных, и для этого элемента данных не установлена исключительная блокировка. Исключительная блокировка выдаётся, когда транзакция хочет обновить (записать) элемент данных, и в настоящий момент никакие блокировки для этого элемента данных не удерживаются какой-либо другой транзакцией. Используя концепцию общей/исключительной блокировки, блокировка может иметь три состояния: разблокирована, общая (чтение) и исключительная (запись).

Две транзакции конфликтуют только тогда, когда хотя бы одна транзакция является записью. Поскольку две транзакции чтения могут быть безопасно выполнены одновременно, общие блокировки позволяют нескольким транзакциям чтения одновременно считывать один и тот же элемент данных. Например, если транзакция T1 имеет общую блокировку элемента X данных, и транзакция T2 хочет прочитать элемент X данных, T2 также может получить общую блокировку элемента X данных.

Если транзакция T2 обновляет элемент данных X, T2 требует исключительную блокировку для элемента данных X. Исключительная блокировка предоставляется, если никакие другие блокировки не удерживаются для элемента данных (это условие известно, как правило взаимоисключающего доступа: только одна транзакция за раз может владеть исключительной блокировкой объекта.) Следовательно, если совместно используемая (или исключительная) блокировка для элемента данных X уже удержана транзакцией T1, исключительная блокировка не может быть предоставлена транзакции T2, и T2 должен ждать начать, пока T1 не совершит. Другими словами, общая блокировка всегда будет блокировать исключительную (запись) блокировку; следовательно, снижение параллелизма транзакций.

Хотя использование общих блокировок делает доступ к данным более эффективным, схема общей/эксклюзивной блокировки увеличивает накладные расходы менеджера блокировок по нескольким причинам:

• Тип блокировки должен быть известен до того, как она может быть предоставлена.

• Существуют три операции блокировки: READ_LOCK для проверки типа блокировки, WRITE_LOCK для выдачи блокировки и UNLOCK для снятия блокировки.

• Схема была усовершенствована, чтобы разрешить обновление блокировки с общего на эксклюзивный и понижение блокировки с эксклюзивного на общий.

Хотя блокировки предотвращают серьёзные несоответствия данных, они могут привести к двум основным проблемам:

• Итоговое расписание транзакций может быть не сериализуемым.

• Расписание может создать тупики. Взаимная блокировка возникает, когда две транзакции бесконечно ждут друг друга, чтобы разблокировать данные. Блокировка базы данных, которая аналогична блокировке трафика в большом городе, возникает, когда две или более транзакций ждут друг друга, чтобы разблокировать данные.

К счастью, обеими проблемами можно управлять: сериализуемость достигается с помощью протокола блокировки, известного как двухфазная блокировка, а тупиковыми

CC BY

ситуациями можно управлять с помощью методов обнаружения и предотвращения взаимных блокировок. Эти методы рассматриваются в следующих двух разделах.

Двухфазная блокировка (2ФБ) определяет, как транзакции получают и снимают блокировки. Двухфазная блокировка гарантирует сериализуемость, но не предотвращает взаимоблокировки. Две фазы:

Растущая фаза, на которой транзакция получает все необходимые блокировки без разблокировки каких-либо данных. Как только все блокировки были получены, транзакция находится в своей заблокированной точке.

Фаза сжатия, при которой транзакция снимает все блокировки и не может получить новую блокировку.

Протокол двухфазной блокировки регулируется следующими правилами:

Две транзакции не могут иметь конфликтующие блокировки.

Операция блокировки не может предшествовать операции блокировки в той же транзакции.

На данные не влияют, пока не будут получены все блокировки, то есть до тех пор, пока транзакция не окажется в своей заблокированной точке.

Взаимная блокировка возникает, когда две транзакции бесконечно ждут друг друга, чтобы разблокировать данные. Например, тупик возникает, когда две транзакции, T1 и T2, существуют в следующем режиме:

T1 = доступ к элементам данных X и Y

T2 = доступ к элементам данных Y и X

Если T1 не имеет разблокированного элемента данных Y, T2 не может начаться; если T2 не разблокировал элемент данных X, T1 не может продолжить. Следовательно, и T1 и T2 ждут, пока другой разблокирует требуемый элемент данных. Такой тупик также известен как смертельные объятия.

В СУБД одновременно может быть выполнено гораздо больше транзакций, что увеличивает вероятность возникновения тупиковых ситуаций. Взаимоблокировки возможны только тогда, когда одна из транзакций хочет получить монопольную блокировку элемента данных; между общими блокировками не может быть тупиковых условий.

Три основных метода управления тупиками:

• Предотвращение тупиковой ситуации. Транзакция, запрашивающая новую блокировку, прерывается, когда существует вероятность возникновения тупика. Если транзакция отменяется, все изменения, сделанные этой транзакцией, откатываются и все блокировки, полученные транзакцией, снимаются. Затем транзакция переносится на исполнение. Предотвращение взаимоблокировки работает, потому что оно избегает условий, которые приводят к взаимоблокировке.

• Обнаружение тупика. СУБД периодически проверяет базу данных на наличие взаимоблокировок. Если обнаружена взаимоблокировка, транзакция «жертвы» прерывается (откатывается и перезапускается), а другая транзакция продолжается.

CC BY

• Избегание от тупиковой ситуации. Транзакция должна получить все необходимые блокировки, прежде чем она может быть выполнена. Этот метод позволяет избежать отката конфликтующих транзакций, требуя, чтобы блокировки были получены последовательно. Однако назначение последовательной блокировки, необходимое для предотвращения тупика, увеличивает время отклика на действие.

Выбор используемого метода управления взаимоблокировкой зависит от среды базы данных. Если вероятность взаимоблокировки низкая, рекомендуется обнаружение взаимоблокировки. Если вероятность взаимоблокировки высока, рекомендуется предотвращение взаимоблокировки. Если время отклика не является высоким в списке приоритетов системы, может быть использовано предотвращение тупиковых ситуаций. Все существующие СУБД поддерживают обнаружение взаимоблокировок в транзакционных базах данных, в то время как некоторые СУБД используют сочетание методов предотвращения и предотвращения для других типов данных, таких как хранилища данных или данные XML.

© Мамедли Р.Э., 2022

УДК 004.057.4
https://doi.org/10.36906/AP-2022/43

Никонова Е.З.
ORCID: 0000-0003-4416-2629, канд. пед. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия

СОВРЕМЕННЫЕ КОНЦЕПЦИИ РАБОТЫ С ДАННЫМИ

Аннотация. В статье рассматриваются подходы к управлению данными в их историческом развитии, причины перехода от работы с данными с помощью СУБД к современным технологиям. Также сравниваются возможности технологий «больших данных», бизнес-аналитики и Data Science.

Ключевые слова: данные; большие данные; бизнес-аналитика; Data Science.

Nikonova E.Z.
ORCID:0000-0003-4416-2629, Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

MODERN CONCEPTS OF WORKING WITH DATA

Abstract. The article discusses approaches to data management in their historical development, the reasons for the transition from working with DBMS to modern technologies. The possibilities of "big data" technologies, business analytics and Data Science are also compared.

Keywords: data; big data; Business Intelligence; Data Science.

Технологии работы с данными не теряют своей актуальности вот уже на протяжении многих десятилетий. И сегодня смысл ставшей когда-то знаменитой фразы Ротшильда «Кто владеет информацией, тот владеет миром» можно передать словами Майка Лукидиса из его доклада «Что такое наука о данных?»: «Будущее принадлежит компаниям и людям, способным превратить данные в продукты» [1].

Согласно стандарту ISO/IEC 2382-1 под данными понимают «…поддающееся многократной интерпретации представление информации в формализованном виде, пригодном для передачи, связи, или обработки» [5].

Объёмы данных постоянно растут, так, в 2020 году количество произведённых данных в мире составило более 64 зеттабайт, что, по мнению IDC (International Data Corporation) – международная исследовательская и консалтинговая компания), было обусловлено глобальной пандемией, заставившей людей перейти на дистанционные формы работы и учёбы [2].

CC BY

Исторически первой технологией работы с данными были СУБД, не уделявшие серьёзного внимания данным как самостоятельной сущности, воспринимая их только как «мешок битов и байтов».

Но уже в начале XXI века отношение к данным кардинально изменилось, не случайно о них заговорили как о «новой нефти», подразумевая их исключительно важное значение в экономике.

Однако сама концепция больших данных зародилась отнюдь не в аналитических бизнес-исследованиях, а в научных публикациях. Так, например, ещё в XVIII веке английский астроном Томас Симпсон отмечал преимущества использования чисел в астрономических наблюдениях.

В 2008 году возрастающую роль технологий обработки больших объёмов, данных отметил редактор британского научного редактора «Nature» Клиффорд Линч, предложив для этого направления термин «Big Data» по аналогии с существующими на тот момент понятиями Большой нефти, Большой руды, отражающими переход количества в иное качество.

В 2001 году в качестве источников Больших данных специалисты указывали непрерывно поступающие данные с измерительных устройств, сообщения из социальных сетей, метеоданные, данные от сетей сотовой связи, устройств аудио- и видео регистрации. О размерах этих данных можно судить по следующим примерам: датчики авиадвигателя вырабатывают около 10 терабайт каждые 30 минут, а сервис Twitter генерирует 8 терабайт сообщений в сутки при установленных ограничениях их длины в 140 символов [2]. Помимо объёмов, генерируемых данных такая вариативность их источников приводила к постоянной изменчивости их состава и структуры. В результате данные уже невозможно было обрабатывать традиционными способами. Кроме этого, изменились и другие характеристики работы с данными в базах Больших данных (см. табл.).

Таблица

Характеристики хранения данных

Характеристика	Традиционная база данных	База Больших данных
Единицы измерения объёма данных	Гигабайт - терабайт	Петабайт - эксабайт
Структурированность данных	Чёткая структура	Отсутствует
Способ хранения	Централизованный	Децентрализованный
Взаимосвязь данных	Сильная	Слабая

Как отмечали аналитики, используя технологии Больших данных, компании могли преодолеть ограничения, обозначенные как «5 V’s of Big Data» [3]:

– Volume (объем) – предел объёма данных, которые можно одновременно обработать;

– Velocity (скорость) – количество транзакций/объем данных, совершаемых за единицу времени;

– Variety (вариативность) – разные типы и структуры/отсутствие структуры данных;

– Veracity (достоверность) – данные поступают из различных источников, не всегда гарантирующих требуемую для принятия решений точность;

CC BY

– Value (значение) – данные должны представлять вполне определенную ценность для компании;

– Variability (изменчивость) – как данные, так и зависимости между ними зависят от окружения и непрерывно изменяются.

Таким образом, рост количества различных технологий, устройств, интернет-сервисов привёл к тому, что Большие данные стали востребованным инструментом в самых различных сферах деятельности человека, таких как научные исследования деятельность, государственное управление, бизнес. К 2014 году компании научились не только накапливать информацию, но и извлекать из неё существенную выгоду в таких направлениях как поиск новых источников дохода, расширение номенклатуры товаров и услуг, привлечение новых клиентов.

Но не смотря на достигнутые результаты, уже в 2015 году аналитики Gartner исключили Большие данные из своего отчёта о ведущих информационных технологиях, объясняя это тем, что Большие данные перешли из разряда инноваций в категорию стандартных инструментов.

Новое направление развития Большие данные получили в связи возможностью реализации двухуровневой модели обработки данных, согласно которой на первом уровне выполняется традиционная аналитика не в режиме реального времени. На втором уровне данные подвергаются «аналитике на лету», т. е. в режиме реального времени, позволяя тем самым оказывать влияние на события, которые ещё не закончились во времени.

Инструментальными средствами Больших данных стали «платформы больших данных» (BDP, Big Data Platform), которые можно рассматривать как своеобразные аналоги СУБД, но со значительно большими возможностями обработки больших объёмов данных. Как правило, в состав BDP входят ХД (хранилища данных), БД, серверы, средства управления данными, технологии и решения, обеспечивающие хранение и анализ больших данных (Hadoop – одно из таких решений), а также инструменты аналитики, представленные специальным типом ПО – Big Data Analytics Software.

Платформы BDP позволяют решить такие задачи как:

– сбор, интеграция данных из различных источников;

– извлечение, преобразование, загрузка данных;

– поддержка хранилища данных;

– потоковая обработка данных (получение, предобработка данных из различных источников и сведение их в один поток);

– анализ данных с использованием методов машинного обучения;

– управление контентом, организационное управление, менеджмент данных.

В качестве примера одного из успешных вариантов BDP можно привести платформу Converged Data Platform, разработанную компанией MapR, отличающуюся универсальностью, возможностью работы с базами данных в режиме реального времени с поддержкой корпоративной безопасности данных [1].

Как уже отмечалось, важнейшей функцией Больших данных является возможность анализа и прогнозирования, для чего используются различные методы и технологии:

CC BY

1. Краудсорсинг – привлечение большого количества людей для решения нерегулярных задач модификации данных.

2. Инструменты ETL (Extract-Load-Transform) – извлечение данных из различных источников, их трансформация, т.е. приведение к единому формату, и загрузка в хранилище.

3. Машинное обучение и нейронные сети – разработка программ с целью получения рекомендаций для принятия решений на основе анализа данных.

4. Предиктивная аналитика – предсказательный анализ, построение прогнозов.

5. Имитационное моделирование – исследование поведения компьютерной модели, построенной на основе больших данных.

6. Статистический анализ – вычисление различных статистических показателей выборки данных, выявление взаимосвязей с помощью корреляционного анализа, оценка изменения данных методом динамических рядов.

7. Data mining (интеллектуальный анализ данных) – выявление тенденций в больших данных методами классификации, кластеризации, регрессионного анализа, анализа отклонений.

8. Визуализация данных – представление результатов анализа больших данных в удобном для восприятия виде.

В перечне наиболее востребованных технологий работы с данными наряду с термином Big Data очень часто встречается понятие Data Science (DS), формальный перевод которого как «наука о данных» не совсем точно определяет его суть. Более точной формулировкой является «наука работы с данными».

Исторически наука о работе с данными возникла ранее концепции Больших данных, в начале 1960-х годов, когда Джон В. Тьюки в своей статье «Будущее анализа данных» впервые предположил, что «…анализ данных и те разделы статистики, которые его придерживаются, должны… приобрести черты отдельной науки, а не математики» [5].

Автором термина Data Science и концепции цифровых данных, проходящих свой жизненный цикл от появления, преобразования до возможности использования в различных научных областях, является датский учёный Петер Наур, разработчик языка программирования Алгол 60. В своей книге «Краткий обзор компьютерных методов» Наур определил науку о данных как «…науку о работе с данными после их установления, в то время как связь данных с тем, что они представляют, делегируется другим областям [науки] и наукам» [4].

Но потребовалось несколько десятков лет, прежде чем DS получила, наконец, официальное признание специалистов Комитета по науке о данных для науки и технологий (CODATA) Международного совета по науке (ICSU), начавшего с 2002 года выпуск Data Science Journal [3].

В 2005 году Национальный научный совет США в своём отчёте впервые определил требования к специалистам по работе с данными, которые затем постоянно уточнялись и сложились в определение как «Data Scientist». Современная квалификация Data Scientist определяется как «…профессиональная деятельность, связанная с эффективным и

CC BY

максимально достоверным поиском закономерностей в данных, извлечение знаний из данных в обобщённой форме, а также их оформление в виде, пригодном для обработки заинтересованными сторонами (людьми, программными системами, управляющими устройствами) в целях принятия обоснованных решений» [2].

Данное определение в некоторой степени отражает технологическую цепочку Data Science проектов (см. рис.).

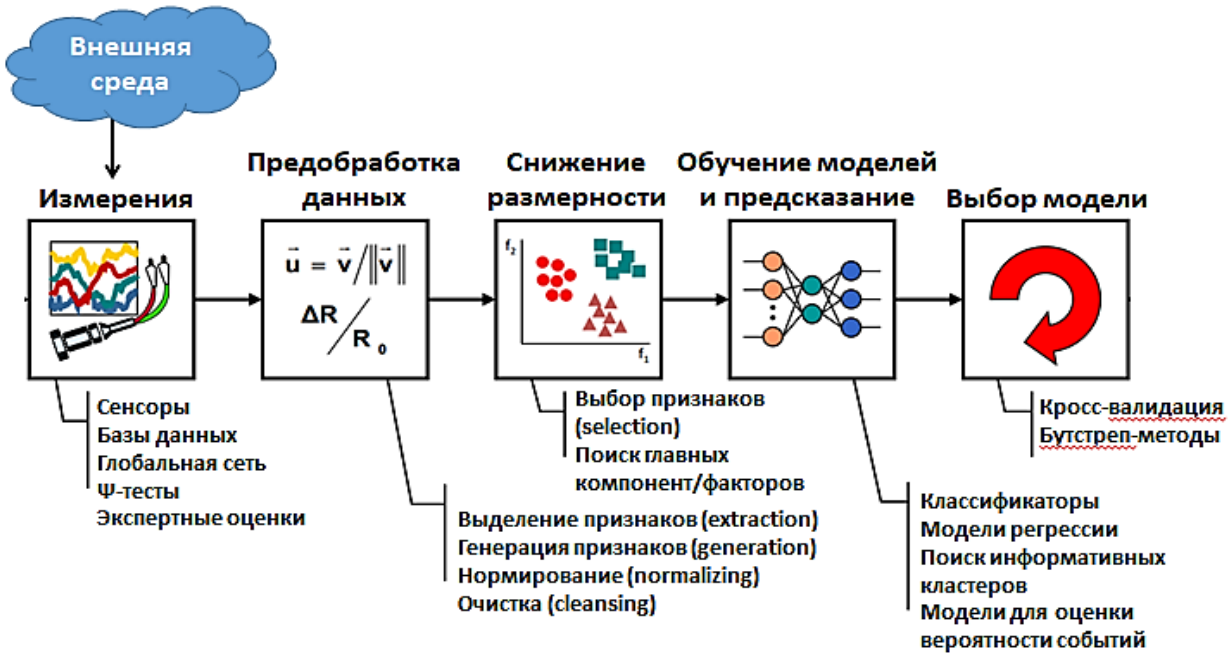


Рис. Технологическая цепочка Data Science проектов (https://clck.ru/eYHUs)

Таким образом, выбор того или иного подхода работы с данными зависит, прежде всего от характера и объёма данных, а также целей обработки данных:

– аналитический анализ небольшой по объёму экономической информации, т. е. бизнес-анализ (Business Intelligence), выполняется традиционными аналитическими инструментами ПО типа Excel, Tableau с использованием методов математической статистики. Корреляционного анализа и т. п.;

– для работы с ограниченными объёмами структурированных данных используются такие решения как БД типа SQL или Oracle;

– разнородные по структуре и типам данные больших объёмов, генерируемые с высокой скоростью, требуют решений для Больших данных (Hadoop, Spark, No SQL);

– извлечение знаний из данных различных предметных областей требует технологий Data Science.

В качестве перспективных направлений развития технологий работы с данными можно отметить как развитие уже известных подходов, таких как технологии искусственного интеллекта, распределённая обработка, Cloud First (облачная инфраструктура), так и появление принципиально новых направлений: Decision intelligence (наука о принятии решений), Маркетплейсы данных (рынки данных), графовые базы данных, экосистемные альянсы, Data Driven (управляемый данными).

CC BY

Литература

1. Карышев М.Ю. Data Science / Big Data: современный вызов статистике и статистикам // Статистика в цифровой экономике: обучение и использование: мат-лы международной научно-практической конференции (г. Санкт-Петербург, 01-02 февраля 2018 года). СПб., 2018. С. 122-123.

2. Назаров А.Д. Эволюция моделей цифрового маркетинга: Data Science // BI-технологии и корпоративные информационные системы в оптимизации бизнес-процессов: Мат-лы VII Международной научно-практической конференции (г. Екатеринбург, 27 ноября 2019 года). Екатеринбург, 2020. С. 57-59.

3. Xu Z., Frankwick G. L., Ramirez E. Effects of big data analytics and traditional marketing analytics on new product success: A knowledge fusion perspective // Journal of Business Research. 2016. Vol. 69. Issue 5. P. 1562-1566. https://doi.org/10.1016/j.jbusres.2015.10.017

4. Novikov S.V. Data science and big data technologies role in the digital economy // TEM Journal. 2020. Vol. 9. Issue 2. Pp. 756-762. https://doi.org/10.18421/TEM92-44

5. IEEE Conference on Data Science: IEEE Conference on Data Science. 2018, St. Petersburg, 15 июня 2018 года. St. Petersburg, 2018. 557 p.

© Никонова Е.З., 2022

УДК 004.891+004.421+004.62
https://doi.org/10.36906/AP-2022/44

Ражков А.Ф.
ORCID: 0000-0001-5200-4167
Объединённый институт проблем информатики
Национальной академии наук Беларуси
г. Минск, Беларусь
Тимощенко Е.В.
ORCID: 0000-0003-1373-5113, канд. физ.-мат. наук
Могилевский государственный университет им. А.А. Кулешова
г. Могилев, Беларусь

ОПТИМИЗАЦИЯ ГИПЕРПАРАМЕТРОВ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ ДАННЫХ

Аннотация. Рассмотрена программная реализация алгоритма оптимизации решетчатого поиска с перекрёстной проверкой для настройки гиперпараметров алгоритмов, применяемых в машинном обучении для решения задач классификации. На примере прикладной задачи с набором медицинских данных проведено сравнение производительности алгоритмов машинного обучения, настроенных с помощью конфигурации гиперпараметров. Найдены наилучшие значения гиперпараметров распространенных моделей машинного обучения.

Ключевые слова: машинное обучение; классификация данных; оптимизация гиперпараметров; решетчатый поиск; перекрёстная проверка.

Razhkov A.F.
ORCID: 0000-0001-5200-4167
The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus
Minsk, Belarus
Timoschenko E.V.
ORCID: 0000-0003-1373-5113, Ph.D.
Mogilev State A. Kuleshov University
Mogilev, Belarus

HYPERPARAMETER OPTIMIZATION FOR CLASSIFICATION MACHINE LEARNING ALGORITHMS

Abstract. A software implementation of a grid search optimization algorithm with cross validation for tuning the hyperparameters of algorithms used in machine learning to solve classification problems is considered. Using the example of an applied problem with a set of medical

CC BY

data, we compared the performance of machine learning algorithms tuned using hyperparameter configuration. Found the best hyperparameter values for common machine learning models.

Keywords: hyperparameter optimization; machine learning; data classification; grid search; cross validation.

Алгоритмы машинного обучения широко используются в различных областях, так как они являются универсальными, демонстрируют высокую производительность в задачах анализа данных и подходят для различных типов задач и наборов данных [9]. Построение эффективной модели машинного обучения является сложным и трудоёмким процессом, который включает в себя нахождение подходящего алгоритма и получение оптимальной архитектуры модели путём настройки её гиперпараметров – параметров, которые настраиваются непосредственно перед обучением модели, а не в процессе машинного обучения. Примером гиперпараметра является шаг градиентного спуска, в то же время примером параметра модели машинного обучения – вес градиентного спуска, который может быть инициализирован и настроен во время обучения. Библиотека scikit-learn (или sklearn) [7] предлагает стандартные методы поиска оптимальных значений гиперпараметров. Наиболее распространенным методом является решетчатый поиск с перекрестной проверкой GridSearchCV, сутью которого является перебор всех возможных комбинаций интересующих параметров для каждого разбиения в перекрёстной проверке [2].

Общий процесс обучения модели включает в себя несколько этапов: разбиение данных, оптимизацию гиперпараметров (в нашем случае запуск решетчатого поиска) и оценку итоговых параметров. Схематично алгоритм обучения проиллюстрирован на рисунке.

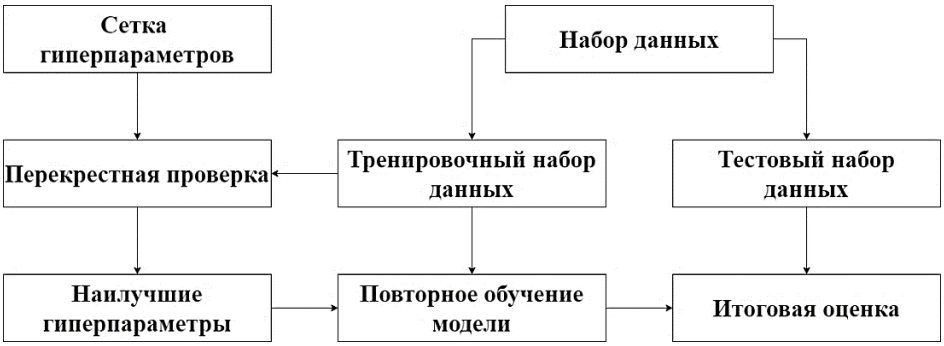


Рис. Общий процесс обучения модели машинного обучения

Рассмотрим программную реализацию обучения модели логистической регрессии с оптимизацией гиперпараметров на языке программирования Python с использованием библиотеки scikit-learn. В связи с тем, что использование всех возможных гиперпараметров модели и добавление большого количества их возможных значений сильно влияет на время обучения модели, рекомендуется использовать только те гиперпараметры, которые могут иметь сильное влияние на точность модели.

Покажем на примере анализа полного списка гиперпараметров и их значений логистической регрессии, почему на практике предпочтительно использовать только

некоторые из них [3]. Для этого приведём листинги кода, при этом комментируя каждый ключевой момент.

Вначале выполняется инициализация списков гиперпараметров. При этом каждый список является коллекцией возможных значений гиперпараметров.

```
penalty = ["l1", "l2", "elasticnet"]
dual = [False, True]
tol = [x / 100.0 for x in range(1, 500, 50)]
C = [x / 100.0 for x in range(1, 100, 20)]
fit_intercept = [True, False]
intercept_scaling = [x / 100.0 for x in range(1, 50, 10)]
class_weight = [None, "balanced"]
random_state = [0]
solver = ["lbfgs", "newton-cg", "liblinear", "sag", "saga"]
max_iter = [5000]
multi_class = ["auto", "ovr", "multinomial"]
verbose = [1]
warm_start = [False, True]
n_jobs = [1]
l1_ratio = [x /10.0 for x in range(1, 10, 1)]
```

Далее проводим инициализацию одного списка, используя функцию product() модуля itertools, которая возвращает декартово произведение ранее представленных инициализированных списков гиперпараметров.

```
all_params = list(product(penalty, tol, dual, C, fit_intercept, intercept_scaling, class_weight,
random_state, solver,max_iter, multi_class, verbose, warm_start, n_jobs, l1_ratio))
```

Затем выполняется инициализация словаря из существующих списков с условными конструкциями, которые позволяют избежать определенных комбинаций взаимоисключающих гиперпараметров.

```
params = [{
        'penalty': [penalty],
         'tol': [tol],
        'dual': [dual],
         'C': [C],
         'fit_intercept': [fit_intercept],
         'intercept_scaling': [intercept_scaling],
        'class_weight': [class_weight],
        'random_state': [random_state],
        'solver': [solver],
        'max_iter': [max_iter],
        'multi_class': [multi_class],
        'verbose': [verbose],
```

```
            'warm_start': [warm_start],
            'n_jobs': [n_jobs],
            'l1_ratio': [l1_ratio]
        }
    for penalty, tol, dual, C, fit_intercept, intercept_scaling, class_weight, random_state, solver,
    max_iter, multi_class, verbose, warm_start, n_jobs, l1_ratio in all_params
    if not (penalty == 'l1' and solver == 'lbfgs')
    and not (penalty == 'l1' and solver == 'newton-cg')
    and not (penalty == 'l1' and solver == 'sag')
    and not (penalty == 'elasticnet' and solver != 'saga')
    and not (dual is True and solver != 'liblinear')
    and not (dual is True and penalty != 'l2')
    and not (l1_ratio != None and 0 <= l1_ratio <= 1 and penalty != 'elasticnet')
    and not (l1_ratio == 0 and penalty != 'l2')
    and not (l1_ratio == 1 and penalty != 'l1')
    and not (solver == 'liblinear' and multi_class == 'multinomial')
    ]
```

На следующем этапе выполняется инициализация логистической регрессии и инициализация класса GridSearchCV.

```
clf = LogisticRegression()
model = GridSearchCV(clf, params, cv=5, error_score="raise")
```

Далее запускается процесс обучения модели на заранее подготовленных данных с фиксацией времени, затраченного на ее обучение.

```
data = get_dataset(dataset_name)
X = data.drop(["Diagnosis"], axis=1)
Y = data['Diagnosis'].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
start_time_fit = time.time()
model.fit(X_train, Y_train)
end_time_fit = time.time()
```

На последнем этапе выполняется инициализация словаря, который включает в себя точность, время обучения, гиперпараметры лучшей модели.

```
dict_best_params = {}
dict_best_params['Точность(%)'] = round(model.best_score_ * 100, 2)
dict_best_params['Время выполнения(с)'] = round((end_time_fit - start_time_fit), 4)
dict_best_params.update(model.best_params_)
```

В качестве прикладной задачи было выбрано прогнозирование наличия сердечно-сосудистого заболевания у пациента с использованием представленного выше алгоритма машинного обучения [14]. Для обучения модели использовался набор данных Heart Disease Data Set из репозитория UCI Machine Learning Repository по ссылке

https://archive.ics.uci.edu/ml/datasets/Heart+Disease. Предложенный набор данных содержит 76 атрибутов, но в большинстве экспериментов используется подмножество, содержащее 14 атрибутов. Описание атрибутов представлено в таблице 1.

Таблица 1

Используемые атрибуты набора данных Heart Disease Data Set

Атрибут	Описание	Тип, возможные значения
Age	Возраст	Число
Sex	Пол	0: женский 1: мужской
Cp	Тип болей в груди	0: бессимптомная 1: атипичная стенокардия 2: неангинальная 3: типичная стенокардия
Trestbps	Артериальное давление в состоянии покоя	Число
Chol	Общий холестерин в сыворотке крови	Число
Fbs	Уровень сахара в крови натощак	0: менее 120 мг / дл 1: более 120 мг / дл
Restecg	Результаты электрокардиографии в покое	0: норма 1: наличие аномалии ST-T 2: наличие вероятной или определенной гипертрофии левого желудочка по критериям Эстеса
Thalach	Максимальная достигнутая частота сердечных сокращений	Число
Exang	Стенокардия, вызванная физической нагрузкой	0: Нет 1: Да
Oldpeak	Депрессия ST-сегмента, вызванная упражнениями по сравнению с отдыхом	Число
Slope	Подъем пикового сегмента ST при физической нагрузке	0: косонисходящая 1: платообразная 2: косовосходящая
Ca	Количество крупных сосудов, окрашенных флуороскопией	0, 1, 2, 3
Thal	Талассемия	0: отсутствует 1: исправленный дефект 2: нормальный 3: обратимый дефект
Target	Наличие заболевания	0: отсутствие заболевания 1: наличие заболевания

Используя алгоритмы машинного обучения, можно разработать базу знаний, которую можно использовать не только для моделирования процессов в различных областях медицины, но и в сфере образования [13]. Для поиска оптимального алгоритма машинного обучения при прогнозировании наличия сердечно-сосудистого заболевания у пациента аналогичным образом были проанализированы представленные ниже алгоритмы машинного обучения для классификации данных:

• Линейный дискриминантный анализ (Linear Discriminant Analysis, LDA).

• Метод K-ближайших соседей (K-nearest neighbors algorithm, KNN) [4].

• Деревья решений. Деревья регрессии и классификации (Classification and Regression Tree, CART) [4; 12].

CC BY

• Наивный байесовский классификатор (Naive Bayes Classifier, NB) [4].

• Метод опорных векторов (C-Support Vector Classification, C-SVC) [5].

• Линейный метод опорных векторов (Linear Support Vector Classification, LSVC).

• Многослойный персептрон (Multilayer Perceptron Classifier, MLP).

• Бэггинг (Bagging Classifier, BG) [6].

• Случайный лес (Random Forest Classifier, RF) [4; 11].

• Классификатор экстремально рандомизированных деревьев (Extra Trees Classifier, ET) [10].

• Адаптивный бустинг (AdaBoost Classifier, AB) [4].

• Градиентный бустинг (Gradient Boosting Classifier, GB) [8].

• Лёгкий градиентный бустинг (Light Gradient Boosting Machine, LGBM) [1].

• Экстремальный градиентный бустинг (Extreme Gradient Boosting, XGB) [4].

Ниже, в таблице 2, представлены результаты, полученные в ходе исследования.

Таблица 2

Точность и значения гиперпараметров алгоритмов

Алгоритм	Значения гиперпараметров	Точность (%)
LR	C = 0.01, class_weight = “balanced”, fit_intercept = False, intercept_scaling = 0.01, l1_ratio = 0.2, max_iter = 5000, multi_class = “auto”, n_jobs = 1, penalty = “elasticnet”, random_state = 0, solver = “saga”, tol = 1.01	84.8
LDA	Solver = “svd”, store_covariance = False, tol = 0.8	83.1
KNN	Algorithm = “auto”, leaf_size = 2, n_jobs = 1, n_neighbors = 9, p = 1, weights = “uniform”	84.4
CART	Criterion = “gini”, max_depth = 16, min_samples_leaf = 6, min_samples_split = 2, min_weight_fraction_leaf = 0, splitter = “random”	84.4
NB	Var_smoothing = 0.396	83.5
C-SVC	C = 0.61, degree = 1, gamma = “scale”, kernel = “poly”, random_state = 0, tol = 0.201, verbose = False	83.1
LSVC	Solver = “svd”, store_covariance = False, tol = 0.8	83.1
MLP	Activation = “relu”, alpha = 0.4, hidden_layer_sizes = 130, random_state = 0, tol = 0.001, verbose = False, warm_start = False	82.3
BG	Bootstrap = True, bootstrap_features = True, n_estimators = 50, n_jobs = 1, random_state = 0, verbose = 0, wart_start = False	84.7
RF	Bootstrap = True, criterion = “gini”, max_depth = 2, max_features = “auto”, n_estimators = 110, random_state = 0	83.9
ET	Criterion = “entropy”, max_depth = 10, min_samples_leaf = 2, min_samples_split = 2, min_weight_fraction_leaf = 0, n_estimators =30, n_jobs = 6, random_state = 0	84.8
AB	Algorithm = “SAMME”, learning_rate = 0.36, n_estimators = 405, random_state = 0	82.3
GB	Criterion = “friedman_mse”, learning_rate = 0.1, loss = “exponent”, max_depth = 1, n_estimators = 400, random_state = 0, tol = 0.0001	82.3
LGBM	Boosting_type = “dart”, learning_rate = 0.379, max_depth = -1, n_jobs =6, num_leaves = 2, objective = “binary”, random_state = 0	83.5
XGB	Learning_rate = 0.051, max_depth = 1, n_estimators, n_thread = 6	83.9

В данной статье рассмотрена программная реализация решетчатого поиска с перекрестной проверкой, позволяющего оптимизировать гиперпараметры алгоритмов

CC BY

машинного обучения для классификации данных, на Python с помощью библиотеки scikit-learn. На примере решения прикладной задачи прогнозирования наличия сердечно-сосудистого заболевания у пациента получены эффективные модели машинного обучения с тщательно настроенными гиперпараметрами. В результате анализа точности и значений гиперпараметров алгоритмов машинного обучения высокую точность (84.8%) показали логистическая регрессия и классификатор экстремально рандомизированных деревьев для исследуемого набора данных. Дальнейшие исследования будут направлены на увеличение точности алгоритмов машинного обучения путем использования доступных библиотек и фреймворков, разработанных для задач автоматической оптимизации гиперпараметров.

Литература

1. Alzamzami F., Hoda M., Saddik A.E. Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation // IEEE Access. 2020. Vol. 8. P. 101840–101858. https://doi.org/10.1109/access.2020.2997330

2. Bergstra J., Bardenet R., Bengio Y., Kegl B. Algorithms for hyper-parameter optimization // Proceedings of the 24th International Conference on Neural Information Processing Systems. 2011. P. 2546-2554.

3. Bisaso K.R., Karungi S.A., Kiragga A., Mukonzo J.K., Castelnuovo B. A comparative study of logistic regression based machine learning techniques for prediction of early virological suppression in antiretroviral initiating HIV patients // BMC Med Inform Decis Mak. 2018. Vol. 18. P. 77. https://doi.org/10.1186/s12911-018-0659-x

4. Caruana R., Niculescu-Mizil A. An empirical comparison of supervised learning algorithms // Proceedings of the 23rd international conference on Machine learning (ICML '06). 2006. P. 161-168. https://doi.org/10.1145/1143844.1143865

5. Chapelle O., Vapnik V., Bousquet O., Mukherjee S. Choosing multiple parameters for support vector machines // Machine Learning. 2002. Vol. 46. No. 1. P. 131–159. https://doi.org/10.1023/A:1012450327387

6. Dietterich T.G. Ensemble methods in machine learning // Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 2000. Vol. 1857. Pp. 1-15. https://doi.org/10.1007/3-540-45014-9_1

7. Pedregosa F., Varoquaux G., Gramfort A., Michel V. Scikit-learn: Machine learning in Python // The Journal of Machine Learning Research. 2011. Vol. 12. P. 2825-2830

8. Friedman J.H. Greedy function approximation: A gradient boosting machine // The Annals of Statistics. 2001. Vol. 29. No. 5. Pp. 1189-1232. https://doi.org/10.1214/aos/1013203451

9. Jordan M.I., Mitchell T.M. Machine Learning: Trends, Perspectives and Prospects // Science. 2015. Vol. 349. Issue 6245. Pp. 255-260. https://doi.org/10.1126/science.aaa8415

10. Geurts P., Ernst D., Wehenkel L. Extremely randomized trees // Machine Learning. 2006. Vol. 63. Issue 1. Pp. 3-42. https://doi.org/10.1007/s10994-006-6226-1

CC BY

11. Probst P, Wright MN, Boulesteix A-L. Hyperparameters and tuning strategies for random forest // WIREs data mining and knowledge discovery. 2019. Vol. 9. № 3. P. 1-15. https://doi.org/10.1002/widm.1301

12. Reif M., Shafait F., Dengel A. Prediction of Classifier Training Time Including Parameter Optimization // Advances in Artificial Intelligence. KI 2011. Lecture Notes in Computer Science. 2011. Vol. 7006. Pp. 260-271. https://doi.org/10.1007/978-3-642-24455-1_25

13. Ражков А.Ф., Тимощенко Е.В. Виртуальный лабораторный практикум «Интеллектуальный анализ данных для прогнозирования заболеваний» // Качество подготовки специалистов в техническом университете: проблемы, перспективы, инновационные подходы: V международная научно-методическая конференция (г. Могилев, 19-20 ноября 2020 года). Могилев, 2020. С. 172-175.

14. Тимощенко Е.В., Ражков А.Ф. Методы интеллектуального анализа биомедицинских данных // Итоги научных исследований учёных МГУ им. А.А. Кулешова 2019 г.: Мат-лы научно-методической конференции (г. Могилев, 29 января-10 февраля. 2020 года). Могилев, 2020. С. 106-107.

© Ражков А.Ф., Тимощенко Е.В., 2022

УДК 004.42
https://doi.org/10.36906/AP-2022/45

Туренко С.К.
д-р техн. наук
Рейхерт В.С.
Тюменский индустриальный университет
г. Тюмень, Россия

РАЗРАБОТКА АЛГОРИТМА РАСЧЁТА ЗОН ДРЕНИРОВАНИЯ ДОЖДЕВЫХ КОЛОДЦЕВ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ РЕКОНСТРУКЦИИ СИСТЕМ ГОРОДСКОГО ВОДООТВЕДЕНИЯ

Аннотация. В данной статье рассматривается алгоритм расчёта зон дренирования дождевых колодцев в рамках диссертационного исследования «Разработка интеллектуальной системы поддержки принятия решений при реконструкции городской системы водоотведения». Алгоритм расчёта зон дренирования дождевых колодцев делиться на несколько этапов: ввод исходных данных по дождевым колодцам, расчёт триангуляции между дождевыми колодцами и расчёт диаграмм Вороного для каждого дождевого колодца, что и будет являться зонами дренирования. Зоны дренирования необходимы для расчёта расхода по каждому дождевому колодцу, которые в свою очередь необходимы для расчёта подземной части систем городского водоотведения. В дальнейшем это понадобиться для расчёта оптимальных параметров для реконструкции систем городского водоотведения, что позволит оптимизировать временные затраты в подборе оптимального варианта исправления проблемных участков данной системы.

Ключевые слова: алгоритм; зоны дренирования; дождевой колодец; триангуляция Делоне; система; водоотведение; диаграмма Вороного.

Turenko S.K.
Ph.D.
Reikhert V. S.
Industrial University of Tyumen
Tyumen, Russia

DEVELOPMENT OF AN ALGORITHM FOR CALCULATION OF DRAINAGE ZONES FOR RAINWAYS FOR INTELLECTUAL DECISION SUPPORT SYSTEM FOR RECONSTRUCTION OF URBAN WATER DISCHARGE SYSTEMS

Abstract. This article discusses an algorithm for calculating drainage zones of rain wells within the framework of the dissertation research “Development of intellectual decision support system for the reconstruction of a city drainage system”. The algorithm for calculating drainage zones for rain

CC BY

wells is divided into several stages: inputting initial data on rain wells, calculating triangulation between rain wells, and calculating Voronoi diagrams for each rain well, which will be the drainage zones. Drainage zones are necessary to calculate the flow for each rain well, which, in turn, are necessary for calculating the underground part of urban drainage systems. In the future, this will be needed to calculate the optimal parameters for the reconstruction of urban sewerage systems, which will optimize the time spent in selecting the optimal option for correcting problem areas of this system.

Keywords: algorithm; drainage zones; rain well; Delaunay triangulation; system; water disposal; Voronoi diagram.

В настоящее время городская инфраструктура, за последние 50 лет, претерпела огромных изменений. На примере Тюмени можно увидеть, на сколько территория города расширилась за эти годы. Там, где раньше были пустыри, возникают новые районы с новыми подземными коммуникациями, в том числе и системами безнапорного городского водоотведения. Такие системы служат для отвода воды в полном объёме с территории города. При проектировании таких систем учитывается множество факторов и гидродинамических параметров для того, чтобы система городского водоотведения функционировала в полном объёме. С течением времени, например, в старых районах города, система водоотведения перестаёт справляться с большим объёмом воды. Первопричиной такой проблемы является развитие города (возникновение новых дорого, домов и других инфраструктур). Такое развитие является нагрузкой на поверхность, тем самым происходит деформирование положений труб под землёй, смещение дождевых колодцев и иных моментов, которые нарушают работу существующей системы городского водоотведения. Возникают проблемы с пропускной способностью всей системы в целом и это сопровождается изливами воды на улице, как, например, это происходит в Тюмени (рис. 1) (https://clck.ru/Yw8hN).

Рис. 1. Излив воды на ул. 30 лет Победы, г. Тюмень

Проектные институты решают данную проблему путём расчётов оптимальных параметров системы городского водоотведения. К таким расчётам относятся: уклоны, диаметры, максимальный расход, скорости течения и время протекания. Также,

дополнительно, рассчитываются объёмы резервуаров для того, чтобы излишки воды стекали в них. Но это всё трудозатратно по времени.

Для того, чтобы прийти к таким расчётам, нужно знать, сколько воды будет притекать с поверхности по дождеприёмникам. Такие расчёты производятся, в основном в Excel. Поэтому, возникла одна из задач диссертационного исследования, автоматизировать данный расчёт, начиная с расчёта зон дренирования дождевых колодцев. В качестве языка программирования используется C# для реализации библиотеки классов по расчёту поверхностных и подземных данных. В рамках статьи рассматривается часть из расчётов поверхностной части – зоны дренирования.

Зоны дренирования – это зоны притока жидкости к дождевым колодцам, с поверхности. Исходными данными для их расчётов являются следующие данные: дождевые колодцы (точки – X, Y, Z) и внешний полигон (элементы полигона – множество точек X, Y, Z, соединяющиеся друг за другом), который будет ограничивать область расчёта. Когда все данные собраны, вторым этапом расчёта зон дренирования является расчёт триангуляции Делоне.

Триангуляция – это планарный граф, все внутренние области которого являются треугольниками (рис. 2).

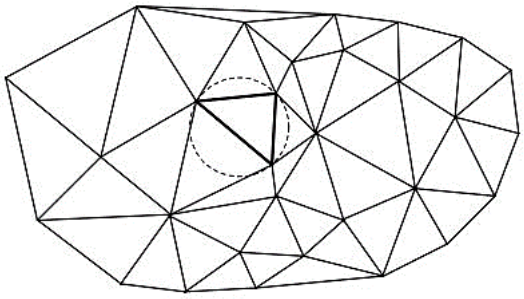


Рис. 2. Схема триангуляции Делоне [1, с. 9]

Алгоритм для построения триангуляции был выбран «Перестраивай и строй» (рис. 3), в реализации который является простым. Суть алгоритма в том, чтобы уменьшить количество пересечение рёбер треугольников, алгоритм считается завершённым только в том случае, когда не останется пересечений в треугольниках. При использовании этого алгоритма не создаются дополнительные точки, с которыми невозможно было бы построить правильные зоны дренирования.

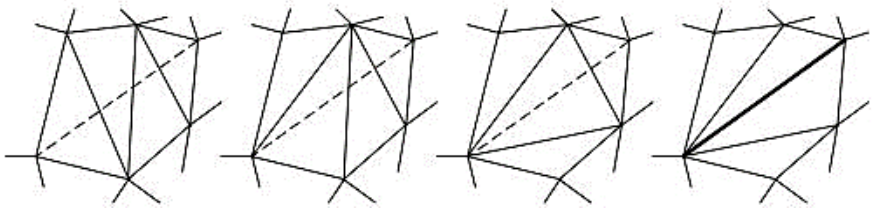


Рис. 3. Алгоритм «Перестраивай и строй» [1, с. 73]

CC BY

При построении триангуляции по дождевым колодцам, формируются треугольники, которые состоят из узлов и рёбер. Узлами треугольников являются дождевые колодцы, а ребра – это соединение между узлами.

После реализации алгоритма построения триангуляции, следующим шагом является реализация алгоритма построения Диаграмм Вороного (https://clck.ru/U7uyF). Диаграмма Вороного (рис. 4) (https://clck.ru/YwB5u) представляет собой разбиение плоскости, где каждая площадь такого разбиения образует узлы, которые близки к исходным точкам плоскости. Исходными точками будут являться узлы триангуляционной плоскости, то есть дождевые колодцы. Дождевые колодцы будут принадлежать этому разбиению.

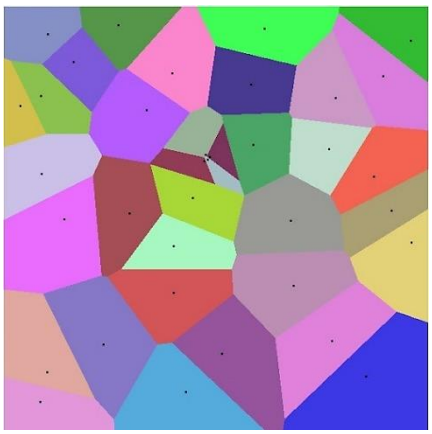


Рис. 4. Диаграмма Вороного

При построении диаграммы Вороного используется распространённый алгоритм – алгоритм Форчуна (https://clck.ru/dWNuD). Алгоритм Форчуна – алгоритм, позволяющий отстраивать диаграмму Вороного путём «заметающей прямой», образуя многоугольники для каждого узла триангуляционной плоскости. Данные многоугольники не должны быть «самопересекающимися». Эти многоугольники и будут являться зонами дренирования каждого дождевого колодца. После реализации данного алгоритма, было проверено на работоспособность в виде плагина в приложение Autodesk Civil 3D, а в качестве исходных данных были взяты реальные данные с Тюменского проектного института ЗАО «Тюменькоммунстрой» (рис. 5).

CC BY

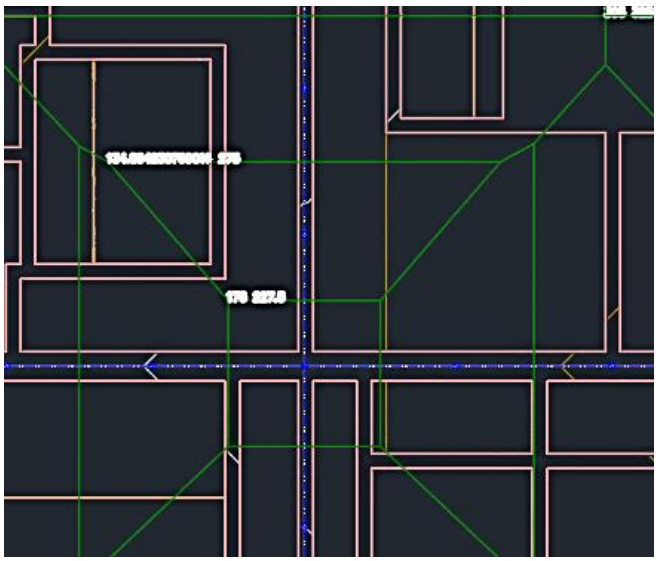


Рис. 5. Отстроенные зоны дренирования

На рисунке 5 видно, что каждый многоугольник имеет внутри себя один дождевой колодец. В рамках данных многоугольников попадают некие объекты поверхности, от которых зависит, какой расход воды будет попадать в данный дождевой колодец. И поэтому, дальнейшим шагом, будет создание алгоритма расчёта площадей водосбора, по которому будет известно, с каких площадей будет попадать вода в данный дождевой колодец. Данная реализация пока нигде не применяется в связке, и, поэтому, автоматизация таких расчётов ускорит работу проектных институтов в целом и минимизирует ошибки, которые возникали при ручном вычислении.

Литература

1. Скворцов А.В. Триангуляция Делоне и её применение. Томск: Изд-во Том. ун-та, 2002. 128 с.

© Туренко С.К., Рейхерт В.С., 2022

УДК 519.687.7
https://doi.org/10.36906/AP-2022/46

Слива М.В.
ORCID: 0000-0001-7915-2282, канд. пед. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия

ДИНАМИЧЕСКОЕ КОНФИГУРИРОВАНИЕ СОВМЕСТНОЙ РАБОТЫ RASPBERRY PI И ARDUINO В РАМКАХ УПРАВЛЕНИЯ УСТРОЙСТВАМИ

Аннотация. В статье рассматривается подход к совместной работе Arduino и миникомпьютера Raspberry Pi. Также показан способ динамического конфигурирования для автоматического распознавания устройств на основе Arduino. Примеры реализованы с использованием языка программирования Java. Для работы с USB используется библиотека JSSC.

Ключевые слова: Arduino; Raspberry Pi; динамическое конфигурирование; Java; USB.

Sliva M.V.
ORCID: 0000-0001-7915-2282, Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia

DYNAMIC CONFIGURATION OF RASPBERRY PI AND ARDUINO COLLABORATION WITHIN DEVICE MANAGEMENT

Abstract. The article discusses how Arduino and Raspberry Pi work together. A dynamic configuration method for automatic device recognition based on Arduino is also shown. Examples are implemented using the Java programming language. The JSSC library is used to work with USB.

Keywords: Arduino; Raspberry Pi; dynamic configuration; Java; USB.

Современные бюджетные реализации систем типа «Умный дом» достаточно легко (с учётом правильной настройки и написания соответствующего программного обеспечения) создаются средствами Arduino и Raspberry Pi (https://clck.ru/dWNyn; https://clck.ru/dWNzm). Поэтому совместное использование Arduino и Raspberry Pi для управления устройствами в рамках Интернета вещей является интересной практикой, например, для создания умного помещения [1].

В такой системе Raspberry Pi может выступать в качестве сервера, который отвечает за связь всей системы с внешним миром и обработку данных, а Arduino берет на себя роль контроллера устройств, для получения информации от устройств или датчиков и передачи данных на сервер (рис. 1).

CC BY

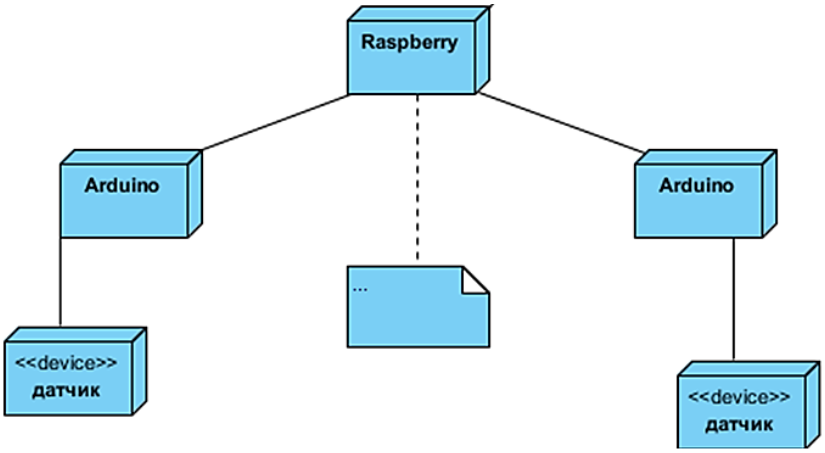


Рис. 1. Совместное использование Arduino и Raspberry Pi

Есть несколько вариантов подключения Arduino к Raspberry Pi: i2c, SPI, UART, Bluetooth, Wi-Fi, USB. Т.е. к одному серверу на основе Raspberry Pi можно подключить несколько Arduino, и к одному Arduino может быть подключено несколько датчиков или устройств. В этом случае каждое Arduino выступает в роли своеобразного промежуточного хаба для управления устройствами или передачи значений от сервера или клиента к нужному устройству или датчику. Причём для пользователя такая комплексная система, состоящая из разных подсистем, может выглядеть как монолитная структура.

При этом, в зависимости от возможностей изменения, можно разделить конфигурирование системы на статическое и динамическое.

Статическое – конфигурация системы (что куда подключено) устанавливается один раз и в процессе эксплуатации системы не меняется. Понятно, что это не всегда удобно, т.к. требует перекомпиляции всего проекта при изменении состава устройств на основе Arduino или порта подключения и не позволит определять изменение активности устройств, когда оно работает или отключено.

Динамическое – конфигурацию системы можно изменять в любой момент, т.е. можно подключать Arduino с разными устройствами, и, в зависимости от конкретного устройства, можно загрузить нужный интерфейс управления. Но для реализации подобного функционала необходимо добавить в систему возможность автоматического распознавания подключаемых устройств на основе Arduino, что-то типа Plug&Play в ОС Windows.

Для этого нужно сделать обмен данными при первом подключении Arduino к серверу, чтобы получать информацию о наличии устройств или датчиков в составе данной конфигурации.

Принцип такого обмена может быть следующим:

• в функции setup в скетче Arduino нужно добавить передачу на сервер информации о подключённых к данному узлу устройствах. Данная функция срабатывает всегда при включении или перезагрузке Arduino. В неё можно добавить ожидание ответа от сервера, чтобы дальнейшее функционирование устройства или датчика было согласовано с главным узлом. Также практичнее будет выделить передачу информации о подключённых к данному узлу устройствах в отдельный метод, вызов которого будет производиться в функции setup.

Такой подход к тому же позволит вызывать этот метод отдельно в процессе работы с данным Arduino, если понадобиться получить на сервере информацию о составе устройств или датчиков конкретного узла. Примерный участок кода для Arduino, реализующий описанный функционал, может быть следующим:

```
void setup(){// следующий код должен быть в самом конце функции setup
  while (!Serial) {
    ; // ожидаем подключение по USB (если используется именно USB), нужно для
      //некоторых моделей
  }
  sendDeviceInfo(); // вызываем функцию для передачи данных через USB
  while (initVal != '1') { // ожидаем приема от сервера уведомления
    if (Serial.available())
      initVal = Serial.read();
  }
}

void sendDeviceInfo() { // функция для передачи данных через USB
  Serial.println("Ardu 1 Devices:");  // отправляем перечень устройств,
  Serial.println("Servo");            //подключенных к данному Arduino
  Serial.println("DC_Motor");
  Serial.println("end devList"); // признак конца перечня
}
```

• на сервере нужно сделать проверку подключения к USB устройств с приемом данных от них и соответствию первого сообщения формату «Ardu 1 Devices:».

Для автоматизации последнего пункта лучше всего подойдёт создание специального потока, который будет с определённой периодичностью (например, раз в 2 секунды) проверять наличие нового подключения по USB. Также удобно было бы хранить текущий состав подключённых Arduino и устройств. И в зависимости от выбранного устройства или датчика можно подгружать нужный графический интерфейс для управления.

Диаграмма классов для реализации такого подхода на стороне Raspberry Pi с использованием языка Java может быть следующей (рис. 2):

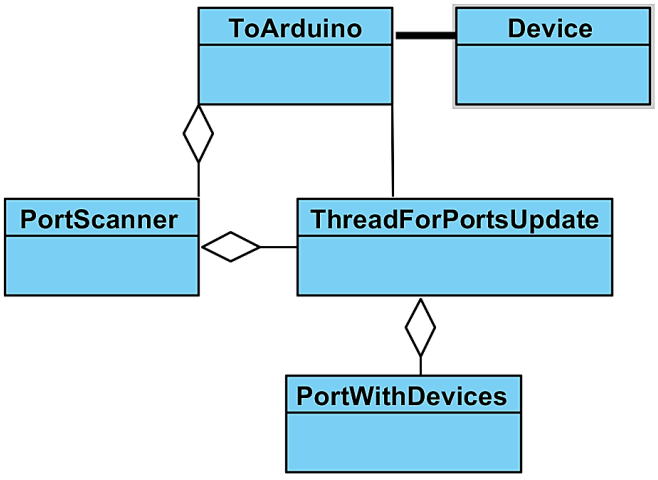


Рис. 2. Диаграмма классов для проверки подключения нового устройства

Действия для автоматического обнаружения подключённого по USB Arduino могут происходить по следующему алгоритму:

1. При запуске основной программы (класс ToArduino) создаётся объект класса PortScanner.

2. PortScanner запускает поток для периодической проверки портов USB (ThreadForPortsUpdate), этот поток становится доступен объекту класса ToArduino.

3. ThreadForPortsUpdate периодически (например, раз в 2 секунды) опрашивает порты на предмет наличия нового подключения и сохраняет все данные в объектах класса PortWithDevices.

4. При появлении нового подключения поток получает данные об устройствах данного Arduino и передаёт их основному классу, который выводит графический интерфейс для работы с подключённым устройством (класс Device).

И теперь для подключения нового устройства достаточно лишь чтобы скетч (программа для Arduino) содержал информацию об устройствах в нужном формате, а сервер имел соответствующий интерфейс управления данным устройством.

Для доступа к USB можно использовать jSSC (Java Simple Serial Connector) – библиотеку для работы с последовательными (COM) портами компьютера (https://clck.ru/dWP34). С ее помощью можно получать имена портов, читать и писать данные, получать события о произошедших событиях и т. д.

jSSC можно разделить на несколько основных частей:

• SerialNativeInterface – класс, который предоставляет доступ ко всем «нативным» методам jSSC.

• SerialPort – класс, с помощью которого непосредственно происходит работа с нужным портом (например, запись данных в нужный порт).

• SerialPortList – класс, представляющий список доступных COM-портов в виде строкового массива с именами портов (COM2, COM7 и т.д.).

• SerialPortEventListener – интерфейс, который необходимо реализовать, если нужно обрабатывать события порта (приём сообщений от устройств, подключённых через Arduino).

В итоге мы получаем динамически конфигурируемую систему для управления устройствами, которую можно использовать как для реализации концепции Интернета вещей в целом, так и для создания частного случая в виде умного помещения.

Литература

1. Слива М.В. Использование Node.js в качестве платформы для умного помещения // Традиции и инновации в образовательном пространстве России: мат-лы VII Всероссийской научно-практической конференции (г. Нижневартовск, 21 апреля 2018 года). Нижневартовск, 2018. С. 49-52.

© Слива М.В., 2022

УДК 004.912:004.896
https://doi.org/10.36906/AP-2022/47

Раскатова М.В.
канд. техн. наук
Челышев Э.А.
ORCID: 0000-0001-8417-8823
Национальный исследовательский университет «МЭИ»
г. Москва, Россия

ВЕКТОРИЗАЦИЯ ТЕКСТОВ В ЗАДАЧАХ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА: ИСТОРИЯ И РАЗВИТИЕ

Аннотация. В докладе рассмотрены история и развитие векторизации текстов как одного из важнейших этапов машинной обработки текстов на естественном языке. Представлены некоторые частотные алгоритмы векторизации: one-hot encoding, мешок слов, TF-IDF, а также word embedding модели векторизации. Кратко рассмотрены языковые модели как средство векторизации текстов. Описываются также достоинства и недостатки рассматриваемых подходов.

Ключевые слова: векторизация; one-hot encoding; мешок слов; TF-IDF; word embedding; языковая модель.

Raskatova M.V.
Ph. D.
Chelyshev E.A.
ORCID: 0000-0001-8417-8823
National Research University "Moscow Power Engineering Institute"
Moscow, Russia

VECTORIZATION OF TEXTS IN NATURAL LANGUAGE PROCESSING TASKS: HISTORY AND DEVELOPMENT

Abstract. The report examines the history and development of text vectorization as one of the most important stages of natural language texts machine processing. Some frequency vectorization algorithms are presented: one-hot encoding, a bag of words, TF-IDF, as well as word embedding vectorization models. Language models as a means of text vectorization are briefly considered. The advantages and disadvantages of the approaches under consideration are also described.

Keywords: vectorization; one-hot encoding; bag of words; TF-IDF; word embedding; language model.

CC BY

В последнее время популярность набирает использование машинного обучения для различных сфер хозяйственной и общественной жизни. Одним из возможных приложений машинного бучения является обработка естественного языка (англ. NLP, Natural Language Processing). Обработка естественного языка используется в системах электронного документооборота, чат-ботах, машинном переводе, в новостных агрегаторах и во многих других сферах [1, с. 254].

Обработка естественного языка развивается уже не первый год и включает в себя целый комплекс решений и алгоритмов. Одним из этапов подготовки текстовых данных является векторизация текстов, которая является предметом данного доклада.

Необходимость векторизации текстов при условии использования некоторых алгоритмов машинного обучения очевидна, так как эти алгоритмы способны работать исключительно с числовыми данными. Так как основной единицей языка выступает слово, возникает нетривиальная задача преобразования слов к числовой форме, которую и решает векторизация текстов.

Авторы не ставят перед собой цели раскрыть в данном докладе все тонкости и подходы к этой задаче. Ограничимся только рассмотрением основных вех истории развития векторизации и некоторых её алгоритмов.

Используемые термины

В дальнейшем будут использоваться следующие термины:

Терм – то же самое, что и слово.

Документ – совокупность термов, представляющая одну смысловую единицу (например, предложение, абзац, статья).

Корпус – набор документов.

Словарь – совокупность всех используемых термов.

Частотная векторизация

Наверно, самым простым и примитивным подходом к векторизации слов является подход, получивший в английском языке название **one-hot encoding** (OHE). Он предполагает, что формируется словарь, содержащий все термы используемого корпуса. Для каждого терма определяется его номер – позиция в словаре. Терму ставится в соответствие вектор, в котором все компоненты нулевые за исключением одной, соответствующей по порядку номеру терма.

Однако, такой подход крайне несовершенен по той причине, что корпус может содержать очень большое число термов, и, следовательно, размерность векторных представлений термов окажется также большой, что негативно повлияет на вычислительную сложность задачи их обработки. Причём, среди термов словаря будут как часто встречающиеся термы, так и те, которые встретить можно крайне редко. Согласно закону Ципфа, до 80% текста формируется с использованием 30% слов [4, с. 107]. Исключение редко встречающихся термов из словаря значительно уменьшает размерность векторов, полученных методом one-hot encoding, при этом потери информации на этапе векторизации незначительны.

CC BY

Чтобы сформировать векторное представление для целого документа, можно составить матрицу, каждая строка которой представляет отдельный терм. При большом числе термов в документе число строк в такой матрице может оказаться также достаточно большим, что опять же усложнит задачу обработки. Решением этой проблемы является метод **мешка слов** (англ. bag of words), в котором каждому документу ставится в соответствие сумма векторов термов, входящих в него. Тогда документу будет соответствовать вектор, каждая компонента которого равна числу упоминаний соответствующего терма в документе без учета порядка следования [6, с. 1001]. Несмотря на потерю информации о следовании термов, с использованием метода мешка слов документы можно сравнивать: для этого нужно определить расстояние между векторами.

TF-IDF

Использование метода мешка слов при сравнении двух документов имеет существенный недостаток. Некоторые термы могут встречаться в различных документах достаточно часто, хотя сами документы, могут относиться к разной тематике. Правильнее было бы сравнивать документы, учитывая в первую очередь более редкие по встречаемости термы, свойственные конкретной тематике. Для решения этой проблемы было бы полезным ставить отдельным термам в соответствие некоторые веса так, чтобы нечасто встречающимся, но характерным для конкретной тематики термам соответствовал высокий вес.

Статистическая мера **TF-IDF** (англ. term frequency – inverse document frequency, рус. частота слова – обратная частота документа).

TF – частота вхождения некоторого терма t в документ d. Данная мера определяется по формуле (1).

$$\mathrm{tf}(t,d) = \frac{n_t}{\sum_k n_k} \quad (1)$$

где n_i – количество вхождения i-ого терма в документ d.

IDF – мера, которая уменьшает вес часто встречающихся термов и определяется как логарифм величины, обратной частоте документов, в которых встречается некоторый терм t. Данная мера определяется с использованием формулы (2).

$$\mathrm{idf}(t,D) = \log\frac{|D|}{|\{d_i \in D \mid t \in d_i\}|} \quad (2)$$

где $|D|$ – количество документов в корпусе; $|\{d_i \in D \mid t \in d_i\}|$ – количество документов корпуса, которые содержат терм t.

Мера TF-IDF определяется по формуле (3) как произведение мер TF и IDF.

$$\mathrm{tf}-\mathrm{idf}(t,d,D) = tf(t,d) \times idf(t,D) \quad (3)$$

Таким образом, широко употребляемые термы имеют высокие значения меры TF, а термы, специфичные для небольшого количества документов корпуса, имеют высокое значение меры IDF. Выполненное с использованием меры TF-IDF сравнение документов оказывается гораздо более точным, чем при использовании метода мешка слов [5, с. 179-182].

Word Embedding

Все рассмотренные выше методы векторизации страдают от общего недостатка: больших размерностей векторных представлений документов. С ростом решаемых задач в

CC BY

области NLP это становилось вычислительно все более сложной задачей. Кроме того, рассмотренные выше методы не учитывают семантических, то есть смысловых отношений термов. Поэтому назрела необходимость в новых решениях для векторизации текстов.

Подход, при котором векторные представления термов оказываются значительно меньше размерности используемого словаря, получил в англоязычной литературе название **word embedding**. Русскоязычных аналогов для данного термина авторам неизвестно. Такие модели векторизации представляют собой искусственные нейронные сети. Первой моделью векторизации, основанной на идее word embedding, считается **word2vec**, разработанный в 2013 году чешским аспирантом Томашом Миколовым и предложенный им в [2].

Модель векторизации word2vec, вычисляет вероятность встретить некоторое слово по контексту. На этапе обучения word2vec получает на вход текстовый корпус большого размера, формирует словарь из полученного корпуса и строит векторные представления для его термов. При этом размерность таких векторов гораздо меньше, чем в предыдущих подходах.

Концепция word embedding предполагает два основных алгоритма построения векторного представления: **continuous bag of words** (CBOW, англ. непрерывный мешок слов) и **skip-gram**. При использовании CBOW для каждого терма определяется вероятность его появления в некотором контексте. При использовании алгоритма skip-gram все происходит наоборот: вычисляется вероятность контекста по некоторому терму.

Стоит заметить, что здесь, казалось бы, о семантических отношениях речи не идёт, только о статистических свойствах корпуса. Но за счёт учёта контекста и больших объёмов данных для обучения у word embedding моделей векторизации есть достоинство: они учитывают семантические отношения термов. Близким по значению термам ставится в соответствие близкие по расстоянию вектора.

Word2vec произвёл в своё время революцию в области NLP. Однако его недостатком можно считать фиксированный на этапе обучения словарь: модель не в состоянии выдать векторное представление для терма, ранее не присутствовавшего в словаре. Решить такую проблему может модель векторизации **FastText**, являющаяся модификацией wor2vec [3, с. 21]. В дальнейшем возникли другие модели (например, **GloV**e), но, несмотря на нововведения и улучшения, их принцип остаётся тем же, каким он был заложен Миколовым в 2013 году.

Языковые модели

Рассмотренные выше модели векторизации являются статическими. Несмотря на их популярность, они имеют недостаток: не учитываю контекстно-зависимую природу термов, многозначность. Одному терму, который может встречаться в совершенно разных контекстах, будет ставиться в соответствие единственный усреднённый вектор. Решить данную проблему позволяют **языковые**, или контекстуализированные, **модели** (англ. language model). Они способны определить, с какой вероятностью в тексте может встретиться некоторая последовательность слов и выдать векторное представление для терма в зависимости от текущего контекста [3, с. 21]. Подробно рассматривать языковые модели в данной работе мы не будем, ограничимся лишь поверхностным разбором.

CC BY

Первыми языковыми моделями являются **BERT** (англ. Bidirectional Encoder Representations from Transformers) и **ELMo** (англ. Embeddings from Language Models), появившиеся в 2018 году. Затем стали появляться и другие языковые модели. Так, например, у российской компании Яндекс также есть разработанная ими языковая модель YaLM (https://yandex.ru/lab/yalm-howto).

Безусловно, языковые модели более требовательны к вычислительным ресурсам, однако они незаменимы в целом ряде задач: исправление опечаток, машинный перевод, генерация текста и др.

Подводя итоги, можно заключить, что скорость развития решений в области NLP поражает, так как с 2013 года по сегодняшний день произошли сразу две революции в этой сфере: сначала появление концепции word embedding и создание модели векторизации word2vec, а потом появление языковых моделей.

Насколько известно авторам, со времени появления языковых моделей каких-то принципиально новых подходов в области векторизации текстов не появилось. Однако стремительное развитие NLP обещает, что в скором будущем можно ожидать новой революции в NLP.

Литература

1. Manning C.D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge, England: Cambridge University Press, 2008.

2. Mikolov T., Sutskever I., Chen K., Corrado G.S., Dean J. Distributed Representations of Words and Phrases and their Compositionality // Advances in neural information processing systems. 2013. Vol. 26.

3. Жеребцова Ю.А., Чижик А.В. Сравнение моделей векторного представления текстов в задаче создания чат-бота. // Вестник Новосибирского государственного университета . Серия: Лингвистика и межкультурная коммуникация. 2020. Т. 18. № 3. С. 16-34.

4. Кучерова С.В. Закон Ципфа и его приложения в области лингвистики // Некоторые вопросы анализа, алгебры, геометрии и математического образования. 2020. № 10. С. 107-108.

5. Оськина К.А. Оптимизация метода классификации текстов, основанного на TF-IDF, за счёт введения дополнительных коэффициентов // Вестник Московского государственного лингвистического университета. Гуманитарные науки. 2016. № 15(754). С. 175-187.

6. Савченко Т.Ю. Обработка естественного языка для использования в машинном обучении: частотная векторизация, TF-IDF, word2vec // Аллея науки. 2018. Т. 4. № 6(22). С. 1000-1002.

© Раскатова М.В., Челышев Э.А., 2022

УДК 004.896
https://doi.org/10.36906/AP-2022/48

Раскатова М.В.
канд. техн. наук
Челышев Э.А.
ORCID: 0000-0001-8417-8823
Национальный исследовательский университет «МЭИ»
г. Москва, Россия

ИНФОРМАЦИОННАЯ СИСТЕМА АВТОМАТИЧЕСКОЙ РУБРИКАЦИИ НОВОСТНЫХ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ МАШИННОГО ОБУЧЕНИЯ

Аннотация. В докладе рассмотрен процесс разработки информационной системы автоматической рубрикации новостных текстов. Представлена структура системы и схема взаимосвязи её компонентов. Подробно описан процесс подготовки текстовых данных для задачи классификации. Рассмотрен процесс построения и обучения классификаторов, а также оценки их обобщающей способности. Описан разработанный в рамках системы веб-сайт и его пользовательский интерфейс.

Ключевые слова: информационная система; рубрикация; классификатор; метрика; веб-сайт.

Raskatova M.V.
Ph. D.
Chelyshev E.A.
ORCID: 0000-0001-8417-8823
National Research University "Moscow Power Engineering Institute"
Moscow, Russia

INFORMATION SYSTEM FOR AUTOMATIC CATEGORIZATION OF NEWS TEXTS USING MACHINE LEARNING

Abstract. The report examines the process of developing an information system for automatic categorization of news texts. The structure of the system and the scheme of the relationship of its components are presented. The process of preparing text data for the classification task is described in detail. The process of constructing and training classifiers, as well as evaluating their generalizing ability, is considered. The website developed within the framework of the system and its user interface are described.

Keywords: information system; categorization; classifier; metric; website.

CC BY

В последнее время все большей востребованностью пользуются различные решения в области обработки естественного языка (англ. NLP, Natural Language Processing). Одним из возможных приложений машинного обучения в этой сфере является автоматическая рубрикация текстов, которая значительно опережает традиционную ручную рубрикацию в способности эффективно обрабатывать большие объёмы текстов. В понятиях машинного обучения по прецедентам (так называемое обучение с учителем) рубрикация сводится к задаче мультиклассификации текстов, то есть их классификации на некоторое число классов [2, с. 25].

В данном докладе рассматривается разработанная авторами информационная система, осуществляющая рубрикацию русскоязычных новостных статей на девять непересекающихся рубрик, посвящённым отдельным сторонам общественной жизни: Наука и техника, Политика, Экономика и бизнес и др.

Обзор информационной системы

Схема взаимосвязи компонентов информационной системы представлена на рисунке 1. Новостные статьи и информация о них хранятся в базе данных 1, работа с которой осуществлялась с использованием СУБД MySQL. Использование базы данных позволяет добавлять, удалять и изменять информацию о новостных статьях. Система автоматической рубрикации 2, реализованная с использованием языка программирования Python, содержит в себе обученный классификатор и с некоторым интервалом обращается к базе данных, считывает те статьи, для которых еще не определена рубрика и классифицирует их, относя к одной из рубрик. Веб-сайт обеспечивает для конечного пользователя возможность удобного взаимодействия с информационной системой и предоставляет интуитивно понятный пользовательский интерфейс.

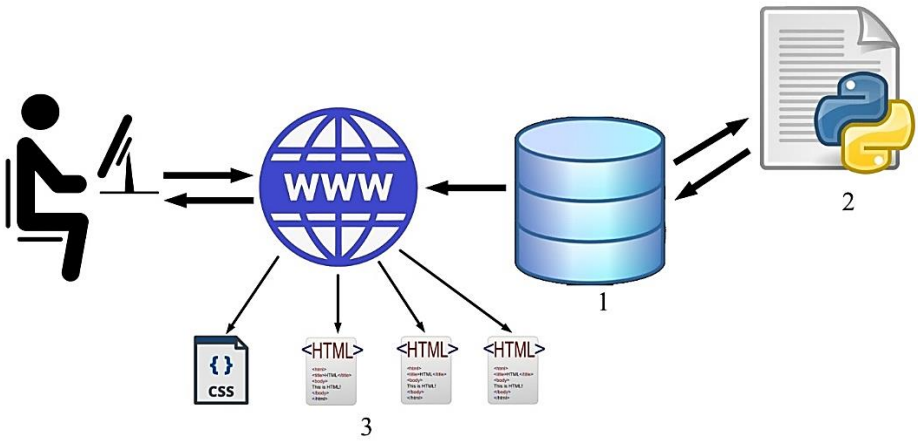


Рис. 1. Схема взаимосвязи компонентов информационной системы
(1 – база данных, 2 – система автоматической рубрикации, 3 – веб-сайт)

Подготовка данных

Для разработки системы автоматической рубрикации был подготовлен набор данных новостного интернет-портала **Lenta.Ru** (https://clck.ru/YcTK8). Для подготовки данных для задачи классификации был разработан программный модуль на языке программирования Python **prepare.py**.

CC BY

Первый этап подготовки данных, а именно приведение всех символов к общему регистру и удаление нерелевантных символов производит функция **preprocess_text** программного модуля. Данная функция осуществляет замену всех символов входной строки на строчные, а также заменяет символ «ё» на «е». Далее с использованием регулярных выражений удаляются присутствующие в тексте URL-ссылки, а также все небуквенные символы, исключая пробелы.

Вторым этапом подготовки данных является токенизация, то есть разбиение текста на отдельные текстовые единицы – токены. В данной работе токенизация осуществляется с использованием встроенного метода **word_tokenize** стандартной библиотеки **nltk.**

Следующим этапом подготовки данных является нормализация токенов. Для решение этой задачи имеется два подхода стемминг и лемматизация [3, с. 78]. В рамках данной работы для решения задачи нормализации токенов использовался метод лемматизации, для чего была реализована функция **lemmatize**, которая, получая на вход список токенов, нормализует их с использованием русскоязычного морфологического анализатора, реализованного в библиотеке **pymorphy2** [1, с. 322].

Для повышения качества подготовленных данных необходимо удаление стоп-слов [4, с. 448]. Для решения задачи удаления стоп-слов в программном модуле prepare.py реализована функция **delete_stop_words**. Получая на вход список токенов, она возвращает список, очищенный от стоп-слов.

Последним этапом подготовки данных является векторизация, в результате которой каждому токену ставится в соответствие вектор некоторого -мерного векторного пространства. Вектор, соответствующий целому тексту, можно определить, например, как среднее арифметическое векторов токенов, входящих в текст. В данной работе для векторизации использовалась предобученная модель векторизации для русского языка **FastText** (https://clck.ru/dWUPX).

В этой и аналогичных задачах в процессе обработки большого количества объёмных текстов в условиях ограниченных вычислительных ресурсов могут возникнуть некоторые проблемы. Одной из них может выступать нехватка оперативной памяти для хранения всего набора данных и проведения над ним некоторых операций. Одним из возможных решений данной проблемы является использование инструментария библиотеки **pandas** языка программирования Python, который позволяет обрабатывать набор данных частями, размер которых можно задавать опционально. Возможный программный код представлен далее.

```
import pandas as pd  # импорт пакета для работы с датафреймами
# импорт пакета для отслеживания времени выполнения обработки датафрейма
from tqdm import tqdm
import prepare  # импорт модуля с функциями подготовки данных
data = pd.read_csv(data_path,
                   sep=',',
                   error_bad_lines=False,
                   usecols=['text', 'topic', 'tags'],
                   chunksize=1000)
```

CC BY

```
# произведем подготовку с использованием функции
# prepare_text модуля prepare
# обработка датафрейма осуществляется частями
chunk_list = []
for chunk in data:
    # для каждой части датафрейма осуществляем обработку
    chunk['text'] = chunk['text'].progress_apply(prepare.prepare_text)
    chunk_list.append(chunk)
# объединяем обработанные части датафрейма
data = pd.concat(chunk_list)
```

Обучение и оценка классификаторов

Подготовленные описанным выше образом данные были разделены на обучающую и тестовую выборки. На обучающей выборке с использованием библиотеки **scikit-learn** языка программирования Python были обучены четыре классификатора: наивный байесовский классификатор (НБК), логистическая регрессия (ЛР), случайный лес решающих деревьев (СЛРД) и искусственная нейронная сеть (ИНС), архитектура которой описана в [5, с. 13-14]. Для данных классификаторов с использованием метода sklearn. методом решетчатого поиска были определены значения гиперпараметров, дающих наибольшие значения обобщающей способности. Оценка обобщающей способности проводилась с использованием скользящего контроля, или кросс-валидации.

В качестве метрик классификации при оценке обобщающей способности использовались метрики точность $precision$, полнота $recall$ и F_1-мера [5, с. 14-15]. Гистограмма средних взвешенных значений использованных метрик классификации представлена на рисунке 2. Заметно, что наибольшие значения метрик показывает классификатор на основе искусственной нейронной сети. Классификаторы на основе логистической регрессии и случайного леса решающих деревьев показывают меньшую обобщающую способность. Наихудшие показатели у наивного байесовского классификатора.

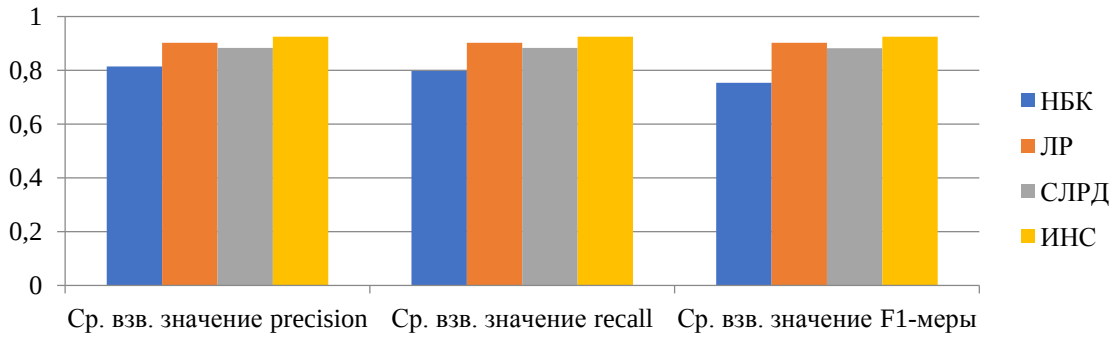


Рис. 2. Гистограмма средних взвешенных значений метрик для различных классификаторов

Веб-сайт

Веб-сайт в рамках информационной системы разрабатывался с использованием фреймворка Django языка программирования Python. Веб-сайт включает в себя ряд тематических веб-страниц, посвящённых той или иной рубрике в отдельности. Пользователь может удобным образом осуществлять навигацию по веб-сайту с использованием бокового

меню. При желании пользователь может перейти на страницу с оригинальной новостной статьёй, нажав на предложенную на веб-сайте новость. Пользовательский интерфейс веб-сайта представлен на рисунке 3.

Рис.

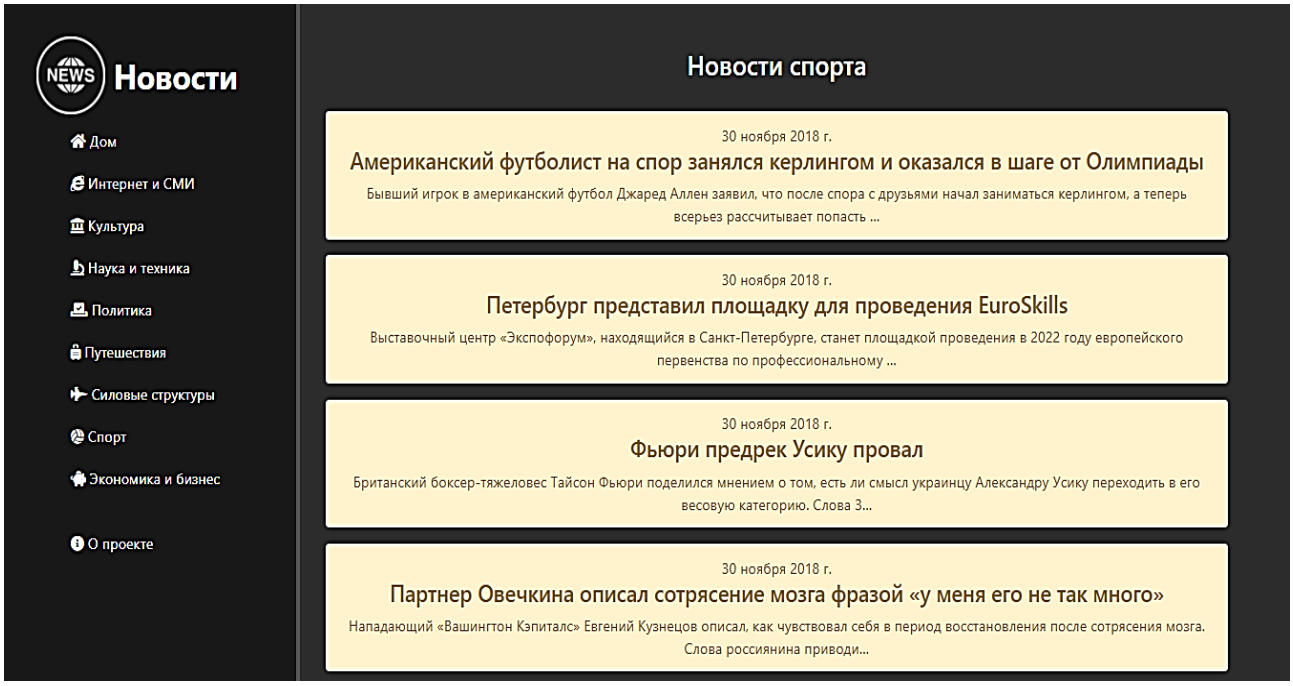


Рис. 3. Пользовательский интерфейс веб-сайта

Таким образом, рассмотренная выше информационная система решает задачу рубрикации новостных текстов с высокой точностью и предоставляет удобный для пользователя интерфейс. При этом система может быть доработана добавлением возможности сбора новостей в режиме реального времени из различных новостных изданий с дальнейшей группировкой новостных статей по тематике.

Литература

1. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts. 4th International Conference, AIST 2015. Communications in Computer and Information Science. Vol. 524. Springer, Cham, 2015. P. 320-332. https://doi.org/10.1007/978-3-319-26123-2_31

2. Агеев М.С., Добров Б.В., Лукашевич Н.В. Автоматическая рубрикация текстов: методы и проблемы // Учёные записки Казанского государственного университета. Серия: Физико-математические науки. 2008. Т. 150. № 4. С. 25-40.

3. Вершинин Е.В., Тимченко Д. К. Исследование применения стемминга и лемматизации при разработке систем адаптивного перевода текста // Наука. Исследования. Практика: Сб. избранных статей по мат-ам Международной научной конференции (г. Санкт-Петербург, 25 декабря 2019 года). Санкт-Петербург, 2020. С. 77-79.

CC BY

4. Мартынов В.А., Плотникова Н.П. Нормализация и фильтрация текста для задачи кластеризации // XLVIII Огарёвские чтения: Мат-лы научной конференции. В 3-х частях (г. Саранск, 06-13 декабря 2019 года). Саранск, 2020. С. 448-452.

5. Челышев Э.А., Оцоков Ш.А., Раскатова М.В. Разработка информационной системы для автоматической рубрикации новостных текстов // Международный журнал информационных технологий и энергоэффективности. 2021. Т. 6. № 3(21). С. 11-17.

© Раскатова М.В., Челышев Э.А., 2022

РОБОТОТЕХНИКА И ПРОГРАММИРОВАНИЕ

УДК 551.46.077
https://doi.org/10.36906/AP-2022/49

Аксенов Е.Н., Фурдуй О.М.
Приднестровский государственный университет им. Т.Г. Шевченко
г. Тирасполь, Приднестровская Молдавская Республика

ПРИМЕНЕНИЕ ПЛАТФОРМЫ *MICRO:BIT* ДЛЯ ПОВЫШЕНИЯ УРОВНЯ ЗАИНТЕРЕСОВАННОСТИ ШКОЛЬНИКОВ И СТУДЕНТОВ В ИЗУЧЕНИИ ЦИФРОВЫХ ТЕХНОЛОГИЙ

Аннотация. Рассматривается передовой мировой опыт в области повышения заинтересованности подрастающего поколения цифровыми технологиями на примере Великобритании. Описывается оригинальный подход в данной области и опыт его применения. Выделяются различные особенности, которые влияют на качественную интеграцию цифровых образовательных инструментов в аудитории. Приводятся результаты эксперимента по заинтересованности после ознакомления детей Великобритании с образовательной платформой *Micro:Bit* и сравнение с результатами аналогичного эксперимента у студентов среднего профессионального образования в ПГУ им. Т.Г. Шевченко.

Ключевые слова: платформа *Micro:Bit;* разработчики; плата; логика; студенты.

Aksenov E.N., Furduy O.M.
Pridnestrovian State University
Tiraspol, Pridnestrovian Moldavian Republic

APPLYING THE MICRO: BIT PLATFORM TO INCREASE THE LEVEL OF INTEREST OF PUPILS AND STUDENTS IN LEARNING DIGITAL TECHNOLOGIES

Abstract: The article considers the advanced world experience in the field of increasing the interest of the younger generation in digital technologies on the example of Great Britain. The original approach in this area and the experience of its application are described. Various features are highlighted that affect the high-quality integration of digital educational tools in the classroom. The results of an experiment on interest after acquaintance of children in Great Britain with the educational platform Micro: Bit and comparison with the results of a similar experiment among students of secondary vocational education at PSU are presented. T.G. Shevchenko.

Keywords: Micro: Bit platform; developers; board; logic; students.

CC BY

Почти сорок лет назад британская вещательная корпорация *BBC* задалась целью познакомить нацию с азами компьютерной грамотности. И запустила собственную серию бытовых компьютеров *BBC:Micro*. Производила их она не сама – этим по контракту занималась компания *Acorn Computers*.

Успех был полным. Серия *Micro* разошлась тиражом более чем миллион экземпляров, проникнув в большинство учебных заведений страны. Образовательным учреждениям покупку частично оплачивало государство. Оригинальный *BBC:Micro* стоил под 300 фунтов.

На них играли, на них учились, писали музыку, строили системы управления, даже организовали первую в своём роде национальную «мультимедийную» «вики-энциклопедию», на десятилетия опередившую время. Проект *BBC* оставил глубокий след в истории: фактически, вспоминая о 8-битных компьютерах, британцы вспоминают именно *BBC:Micro,* а не, к примеру, намного более успешный коммерчески британский *ZX Spectrum*.

В настоящее время выросли поколения, которые воспринимают технику как предметы быта и при этом перестают задумываться как именно это всё работает.

В последнее десятилетие Британия (как и весь мир) бьёт тревогу: дефицит цифровых профессионалов ширится, обещая в следующие годы разрастись в полномасштабный кризис. Предотвратить его надеется корпорация *BBC* всё тем же, уже проверенным приемом. Путем выпуска новой версии микрокомпьютера и раздачи его школьникам.

Рис. 1 BBC Micro

Идея заключалась в том, чтобы познакомить ребёнка с базовыми понятиями и концепциями цифровой техники (алгоритмы, управление памятью, внешними устройствами, коммуникации), сделав ставку более на детское любопытство, фантазию и жажду самовыражения, нежели на образование в традиционном смысле.

К идее многие отнеслись весьма критично. Но на некоторые вопросы критиков ответить достаточно легко. Почему, например, *BBC* не взяла на вооружение уже готовый и зарекомендовавший себя британский проект *Raspberry PI* или схожий с ним итальянский *Arduino*? Чтобы не загасить детский интерес в самом начале лишними сложностями! Начав с

CC BY

функционально простого *Micro:Bit,* а программировать его можно начинать сразу же после подключения к компьютеру, школьники позже сами смогут разобраться со «взрослой» *PC*-архитектурой.

Учтена и ошибка, допущенная в прошлом. В 80-х годах *BBC Micro* рассматривался некоторыми предпринимателями (в частности, Клайвом Синклером, разработчиком *ZX Spectrum*) как вмешательство государства в компьютерный бизнес. Как ещё можно было назвать субсидируемое государством производство отдельно взятой модели? *Micro:Bit* в силу своей примитивности ни с кем не конкурирует, но даже если бы и смог, выпущен будет ограниченной серией в один миллион экземпляров. Позже, если кто-то из частных компаний пожелает, то сможет продолжить производство, но уже своими силами. *BBC* ставит целью только разжечь интерес и предоставить тенденции развиваться самостоятельно.

Но с этим как раз связан второй вопрос, ответить на который гораздо труднее. Будет ли успешным сам проект *Micro:Bit*? Ведь он должен повторить успех 8-битных компьютеров, главными пользователями которых были дети. А вот нужно ли это сегодняшним детям?

Корпорация *BBC* не надеется организовать весь процесс в одиночку: в проекте уже более двух десятков участников, включая производителей и разработчиков самого высшего ранга (*ARM, Freescale, Microsoft, Samsung, Python Software Foundation*), СМИ, образовательные учреждения (*Open University*). Выпускаются занимательные телепередачи и сериалы, программное обеспечение и схемы. Проводятся даже посвящённые *Micro:Bit* уроки в школах. Инициативу в целом назвали *Make It Digital*, а *Micro:Bit* стал её центральным элементом.

В отличие от оригинального *BBC Micro, Micro:Bit* предоставляется каждому ребенку 11 или 12 лет по всей Великобритании бесплатно. Технические спецификации платы уже открыты вместе с исходным кодом, и некоммерческая компания будет следить за будущим развитием, что позволит сделать эти платы коммерчески доступными в Великобритании и на международном уровне. В настоящее время плату можно приобрести в китайских магазинах за 20 долларов.

Поступление *Micro:Bit* в школы началось в 2016 г., британским школьникам передан 1 миллион плат. По состоянию на октябрь 2018 г., *Micro:Bit* распространён в 50 странах мира общим числом 2 миллиона экземпляров.

Рассмотрим подробнее плату *Micro:Bit.*

CC BY

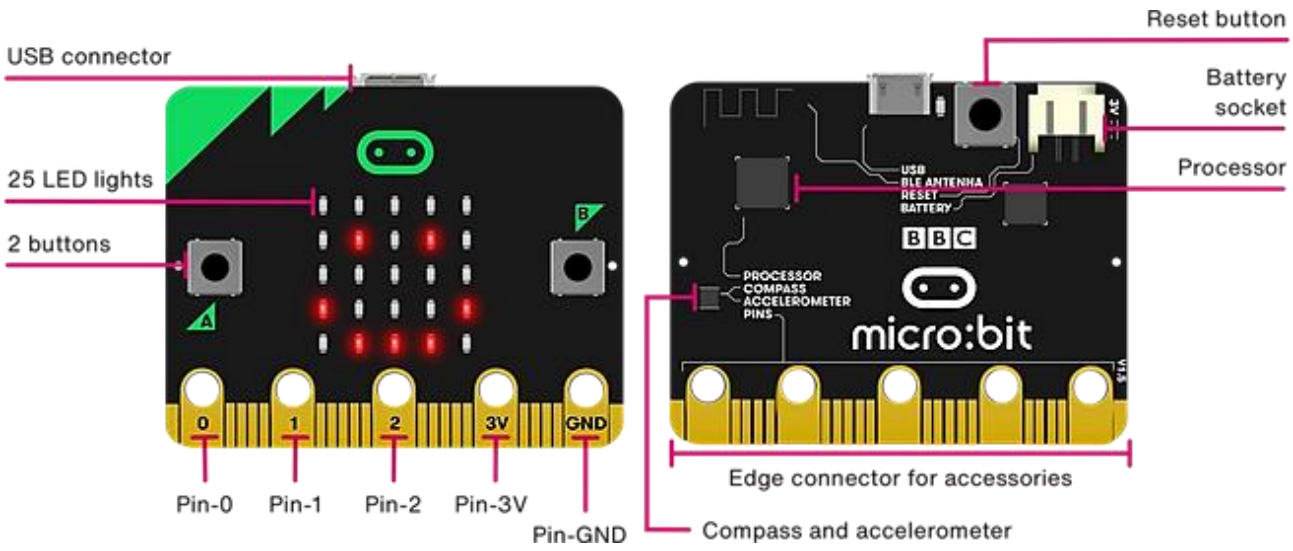


Рис. 2 Плата *Micro:Bit*

Универсальные процессорные платы, такие как *Arduino*, существенно сократили время на реализацию задумок и обучающих проектов, но, чтобы увидеть первые результаты своей деятельности, к ним всё равно приходится добавлять периферию: управление, индикацию или базовый набор датчиков. Возможностей плат множество, но пока все модули объединятся и заработают, интерес новичков уже рискует поугаснуть.

Разработчики обучающей платы *Micro:Bit* пошли иным путём: сама плата – это законченное устройство с набором базовых функций, которых вполне достаточно для лёгкого старта. На борту уже находятся две программируемые кнопки, «экран» в виде матрицы из красных 25 светодиодов, датчики ускорения и температуры, магнитометр. Это позволяет начинающему разработчику быстро реализовать простое устройство или приступить к учебному курсу, ведь всё необходимое уже под рукой.

Встроенные кнопки превращают плату *Micro:Bit* в портативное игровое устройство с возможностью питания от батареек.

Плата умеет обмениваться данными по *Bluetooth* со смартфонами и планшетами на базе *Android* и *iOS*. Результаты работы программы – например, показания встроенных датчиков – можно мгновенно вывести на светодиодную матрицу или наблюдать в мобильном приложении, а визуальная среда программирования *Microsoft MakeCode* с поддержкой *JavaScript* включает в себя онлайн-эмулятор устройства для удобной отладки кода.

Сама по себе среда *Microsoft MakeCode* - отличный инструмент для обучения детей навыкам кодирования и принципам компьютерного программирования.

Microsoft MakeCode имеет много встроенной логики для создания аркадных игр или «ретро-игр». Принципы программирования основаны на создании, настройке и соединении графических блоков. Дети могут даже попробовать инструмент преобразования кода, чтобы показать, как их блочный код меняется на официальный язык кодирования.

Дети могут работать через логику программирования, не зацикливаясь на техническом синтаксисе. Блочное кодирование позволяет сначала подумать о логике и попробовать различные варианты их решения.

CC BY

В результате работа с *Micro:Bit* проходит весьма наглядно и интерактивно, чтобы обучающиеся успели понять каждый шаг создания своего устройства и лучше усвоили материал.

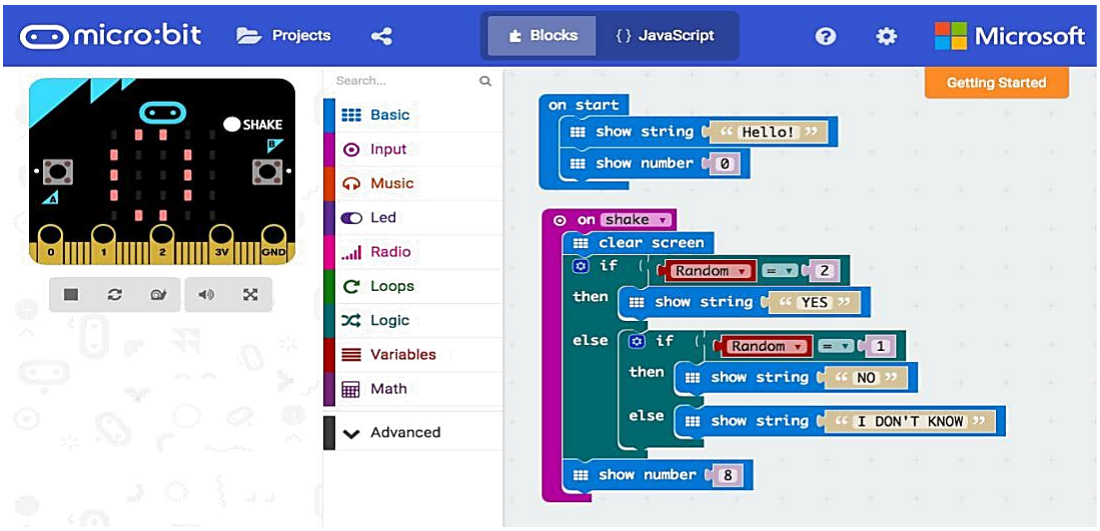


Рис. 3 Среда *Microsoft MakeCode*

Несмотря на то, что *Micro:Bit* ориентирована на школьный курс информатики, она подходит для создания проектов разного уровня сложности. На базе платы можно собрать и умные часы с фитнес-трекером, и мобильных роботов на колёсной платформе.

Исследования, проведенные в Великобритании обобщают впечатления учеников и преподавателей, когда они впервые столкнулись с *Micro:Bit* в классе. Устройство было представлено группе из 36 учеников 12 и 13 лет и 5 преподавателей из 4 стран. Затем после краткого ознакомительного курса все ученики ознакомились с платой, запрограммировав её самостоятельно [1].

Цель эксперимента состояла в том, чтобы выявить различные особенности, которые влияют на качественную интеграцию инструментов, включающих цифровое обучение в классе, и исследовать их ценность в учебном процессе. Простота использования - это фактор, который определяет мотивацию обучающихся к использованию инструмента для обучения. Также учитывалось использование инструмента для достижения образовательных целей. Таким образом, была предпринята попытка установить корреляцию между мотивационной и образовательной ценностью инструмента.

Участников поделили на команды и поощряли конкуренцию с другими командами. Целью было создать «самое веселое» решение внутри группы. Обучающиеся могли свободно проверять все остальные группы в любое время, и все обсуждать между участниками из других групп.

После занятий, продолжавшихся около 4 часов, каждому участнику выдавали анкету со следующими вопросами: возраст, пол, город, страна, что вы узнали о программировании *Micro:Bit*?, мне было весело программировать *Micro:Bit*?, трудно было программировать *Micro:Bit*?, мне раньше нравилось кодирование?. мне нравится кодирование сейчас?, я хотел бы узнать больше о *Micro:Bit*?

CC BY

Обучающиеся могли писать ответы в виде свободного текста на первые пять вопросов. По вопросам 6-10 они могли ставить оценку от 1 до 5, 1 - означает категорически не согласен и 5- означает решительно согласен. Результаты анкеты представлены в таблице 1.

Таблица 1

Распределение по гендерному признаку ответов на вопросы с 6 по 10 в Великобритании

Пол	Понравилось	Тяжело разбираться	Нравилось программировать раньше	Нравится программировать сейчас	Хочу узнать больше о Micro:Bit
Мальчики	4.0	3.4	3.7	3.7	4.1
Девочки	4.1	3.8	3.5	3.6	4.1
Среднее	4.1	3.6	3.6	3.7	4.1

Увеличение значения переменной «нравится программировать» с 3,6 до 3,7 всего за несколько часов также связано с высоким качеством опыта, полученного с помощью инструмента.

Студенты, которым интереснее, проявляют повышенный интерес к обучению кодированию. Студенты, заявившие о более низком качестве опыта (веселье), не проявляют никакого увеличения интереса к обучению кодированию.

Результаты свидетельствуют об эффективности инструмента *Micro:Bit* для повышения уровня заинтересованности.

В нашем учебном заведении был проведен аналогичный опыт с группой студентов среднего профессионального образования (СПО) 3 курса специальности «Компьютерные системы и комплексы». Группа – 14 человек. Из них 13 мальчиков и только одна девочка. Студентам были розданы платы Micro*:Bit* и после краткого обучения были поставлены задачи: разработать простые устройства – компас, несколько интерактивных игр и электронное пианино. После реализации проектов были розданы опросники с аналогичными вопросами. Результаты представлены в таблице 2.

Таблица 2

Распределение по гендерному признаку ответов на вопросы с 6 по 10 у студентов СПО

Пол	Понравилось	Тяжело разбираться	Нравилось программировать раньше	Нравится программировать сейчас	Хочу узнать больше о Micro:Bit
Мальчики	4.6	3.4	4.0	4.4	4.2
Девочки	5	3	5	5	4
Среднее	4.8	3.2	4.5	4.7	4.1

Результаты, приведенные в таблице 2, также подтверждают эффективность подхода в повышении заинтересованности. Более высокие оценки могут быть объяснены за счет более высокого среднего возраста (17 лет против 13 лет в исследовании Великобритании) и сознательного выбора специальности обучения.

Таким образом можно сделать вывод о том, что для повышения вовлеченности подрастающего поколения в цифровые технологии необходимо повысить доступность средств обучения (например, раздавать учебные микрокомпьютеры), обеспечить понятными и подробными инструкциями по использованию и проектированию новых устройств с

CC BY

использованием этих микрокомпьютеров и интегрировать платформы, подобные *Micro:Bit* в образовательные программы школ и СПО.

Литература

1. Videnovik M., Zdravevski E., Lameski P., Trajkovik V. The BBC micro: bit in the classroom: learning experience and first impressionshttps // 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET). – IEEE, 2018. Pp. 1-5.

© Аксенов Е.Н., Фурдуй О.М., 2022

УДК 004.896
https://doi.org/10.36906/AP-2022/50

Makhmadiev Kh.I.
Ataeva G.I.
ORCID: 0000-0003-4571-0028
Bukhara State University
Bukhara, Uzbekistan

HISTORY OF THE DEVELOPMENT OF ROBOTICS

Abstract. Robotics, as an interdisciplinary branch of science and technology, is associated with a variety of computer-controlled electromechanical structures designed for a variety of applications. A decade ago, robotics was a graduate-level research area in engineering and computer science offered by only a few major engineering institutes. It has now become an engineering discipline in its own right, with a small but growing number of universities offering an engineering bachelor's degree in robotics.

Keywords: robotics; mechanisms; programs; manipulators; android; nanorobots.

Махмадиев Х.И.
Атаева Г.И.
ORCID: 0000-0003-4571-0028
Бухарский государственный университет
г. Бухара, Узбекистан

ИСТОРИЯ РАЗВИТИЯ РОБОТОТЕХНИКИ

Аннотация. Робототехника как междисциплинарная отрасль науки и техники связана с множеством управляемых компьютером электромеханических конструкций, предназначенных для множества приложений. Десять лет назад робототехника была областью исследований в области инженерии и информатики, которую предлагали только несколько крупных инженерных институтов. Теперь это самостоятельная инженерная дисциплина, и небольшое, но постоянно растущее число университетов предлагает степень бакалавра инженерных наук в области робототехники.

Ключевые слова: робототехника; механизмы; программы; манипуляторы; андроид; нанороботы.

People have always sought to improve their lives and increase their opportunities. Therefore, people dreamed of inventing various tools, mechanisms, machines in order to facilitate their work, improve quality and productivity. Moreover, these dreams appeared long before society received the first ideas about the exact sciences.

CC BY

Robotics is a natural logical continuation of technology as a phenomenon. Now robots are gradually displacing a person from many areas of his activity, providing in return all new opportunities for efforts: watching movies, underwater diving, computer games, etc. Part of the universal labor spent by humanity on the production of means of production, and not the final product of consumption, is gradually increasing from 0%, obviously striving for 100%.

Robotics today is an intensively developing scientific and technical discipline that studies both the theory, methods of calculation and design of robots, their systems and elements, and the problems of complex automation of production (Already now the efforts of most of the best modern robots are aimed at the production of other machines: machine tools, cars, computers) and scientific research using robots. The subject of robotics is the creation and application of robots, other means of robotics and technical systems and complexes based on them for various purposes.

Robotics is an area with already huge economic impact, which has produced and will continue to produce several new products and markets. These markets require an engineering workforce educated and qualified to contribute effectively to robotics. Robotics has evolved from several different engineering disciplines and sciences. This requires a systematic approach to integrating concepts from the fields of mechanics, electrical engineering, computer engineering, software development and computer science.

While robotic manipulators used in manufacturing still represent the largest population of robots in the modern world, an increasing number of applications require various types of robots, including unmanned aerial vehicles, commonly called drones, unmanned underwater vehicles, unmanned ground vehicles, several types of biomimetic robots, including humanoid robots, as well as micro-robots and nanorobots, which are expected to have several applications, including in the healthcare sector, and these are just some of them. The automotive industry is currently in the final stages of development for the upcoming commercialization of driverless cars, which creates an urgent need to expand knowledge in the field of automatic and autonomous mechanisms and processes.

Robot is an indefinite concept to which any kind of machine can be attributed; the term is usually used for artistic effect or means that certain manipulative mechanisms are used in the machine, allowing the machine to manipulate (control) objects. An important property of robots is a certain degree of autonomy.

An android is a humanoid robot, i.e. an anthropomorphic, human-imitating machine that seeks to replace a person in any of his activities. An android must look and behave like a human.

Robotics is an applied science engaged in the development of automated technical systems. Robotics relies on such disciplines as electronics, mechanics, programming. There are construction, industrial, household, aviation and extreme (military, space, underwater) robotics.

Automation and robotization of production in the capitalist world began in the 50s of the XX century. It was at that time that the appearance of the first industrial robots can be attributed. They carried out the assembly of equipment, and the simplest monotonous operations.

The first such robot was developed by self-taught inventor George Devol in 1954. The robot arm weighed two tons and was controlled by a program recorded on a magnetic drum. The system

CC BY

was named Unimate, a patent was issued for a new device, and in 1961 the inventor founded the company Unimation.

The first robot was installed at the General Motors plant (at the foundry site) in 1961. Then the novelty was tested by Chrysler and Ford plants,

The Unimate system was used to work with cast metal parts that the manipulator extracted from the casting molds. The gripping device was controlled by a hydraulic drive. The robot had 5 degrees of freedom and a gripping device with two “fingers”. The accuracy of the work was very high up to 1.25 mm. And he was more efficient than a person - he worked faster and with less marriage.

In 1967, industrial manipulators came to Europe. They are already expanding their functionality, mastering the professions of welder, painter. The robot gets "technical vision" through video cameras and sensors, it learns to determine the dimensions of products and their location.

In 1982, IBM developed an official language for programming robotic systems. In 1984, Adept introduced the first Scara robot with an electric drive. The new design made the robots simpler and more reliable, while maintaining high speed.

In the 90s, a controller with an intuitive control interface appeared, which the operator could control, he could change parameters and adjust the operating mode. Since then, the control capabilities of robots and their functions have only developed, their complexity, speed, and number of axes have increased, various materials have been used, development and management capabilities have become wider, and several first confident steps towards artificial intelligence have been made.

At the same time, in the USSR, he was actually a leader in robotics. It all started back in the 30s. In 1936, a 16-year-old Soviet schoolboy Vadim Matskevich created a robot that could raise his right hand. To do this, he spent 2 years working in the lathes of the Novocherkassk Polytechnic. Earlier, at the age of 12, he created a small radio-controlled armored car that fired fireworks. The authorities paid attention to the “robot” Matskevich and in 1937 he represented him at the 1937 World Exhibition in Paris.

At the turn of the 30–40s of the XX century. automatic lines for processing bearing parts also appeared in the USSR, and at the end of the 40s of the XX century. for the first time in world practice, a complex production of pistons for tractor engines was created with automation of all processes – from loading raw materials to packaging finished products.

In 1966, a manipulator for laying metal sheets was invented in Voronezh, in 1968, an underwater robot “Manta” with a sensitive gripping device was developed in Leningrad – later it was improved.

In 1985, 40 thousand industrial robots were already used and several times exceeded the number used in the USA. Automated lines worked hard at AVTOVAZ in the 80s and were even attacked by “hacker” workers.

There were major military and space developments. A unique achievement at that time was the unmanned reconnaissance DBR-1, which was adopted by the USSR Air Force back in 1964. Such a device could perform reconnaissance missions over the entire territory of Western and Central Europe.

CC BY

One of the most notable achievements of Soviet robotics and science was the creation of a design bureau named after Lavochkin's “Lunokhod-1”. It was the Soviet spacecraft that became the world's first planetary rover that successfully completed its mission on the surface of another celestial body.

“Humanoid robots” also developed: in 1962, the first robot guide Rex appeared - he conducted excursions for children at the Polytechnic Museum. They say he still “works” there.

More than 100 thousand units of industrial robotics were produced in the Soviet Union. They replaced more than one million workers, but in the 90s these robots disappeared.

In the future, the development of robotics is progressing at a rapid pace, because key industries are developing – physics, chemistry, electrical engineering and, most importantly, electronics. Vacuum tubes were replaced by power electronics, later microchips, then microcontrollers. There are new materials, new ways of automation and programming methods.

First of all, development is taking place in the USA, in Southeast Asia and Western Europe.

Controlled robotic lines are being introduced in production facilities, robotic manipulators are used in all industries, in agriculture, medicine, in space and, of course, in everyday life.

In some industries, up to 50% of the work is done by industrial robots, for example, in the automotive industry, they can weld, paint, and move parts to another assembly site, where other robots will take care of them.

There are even 100% automated factories. There is a factory in Japan where robots assemble robots themselves. And they even prepare food for 2,000 people – the office center that serves this plant.

In the 90s, there was some decline. The introduction of robots using existing technologies at that time did not bring the expected profit to production and the financing of some large-scale projects was suspended. For a number of reasons – both economic and social – the expected boom did not happen, they remained as niche products for car assembly and a number of other industries.

A sharp jump occurred only in the mid-noughties and this development continues. First of all, because the military became interested in robotics.

It is already impossible to stop development, and all countries wishing to be at the forefront of global industry have to accept and catch up.

Robot design and robotics tasks

There are six general tasks of robotics:

1. Moving – moving in any environment
2. Orientation – be aware of your location
3. Manipulation – freely manipulate objects of the environment
4. Interaction – contact with your own kind
5. Communication – communicate freely with a person
6. Artificial intelligence – a robot must independently decide how to execute a human command

The most optimal movement of the robot on wheels and a tracked platform. It is these methods that provide the greatest stability and patency.

CC BY

It is more difficult for wheeled platforms with cross-country ability – the wheel cannot overcome an obstacle higher than its radius. Wheel schemes are constantly being improved, powerful servomotors are used, independent suspensions are being developed, tires with ground hooks are used.

Four-legged and insectomorphic robots are stable (this means in the form of insects, several “legs”, usually 6) Such devices are often used for military purposes.

The robot learned to walk on two legs for a very long time. Of all the existing ones, only the humanoid ASIMO from Honda copes well with this, he can not only walk steadily, but also climb the steps, the company has been developing it for more than 25 years.

Most of the humanoid robots are still moving on the platform.

In addition to walking on the ground, certain models can crawl, swim and fly.

The robot is oriented in space with the help of sensors, sensors, video cameras, has the ability to “see” in the infrared range, detect ultrasonic vibrations and perceive thermal radiation.

The operator can also control, he can be in the same room or several kilometers away.

The main tasks of robotics are being solved in one way or another. The robot becomes more perfect, it knows how to cooperate with other robots, learns to communicate with a person and understand him better.

An interesting training scheme for a space robot satellite, probably the same principle is used to configure other robotic systems. “Emotional learning”, as the developers call it. Its essence is that it lays the “apparatus of emotions”, which tells the satellite what is “good” for him and what is “bad”. Good - if it targets a specific specified object – it increases the score, bad –if it deviates from it – the score will be reduced. Well, so long as the device does not become a stable “good”.

For example, it can be useful for space telescopes. The training is conducted with the help of an operator and takes about 20 minutes, the result is displayed in the knowledge base.

Specifically, this described device can be thrown into outer space by an astronaut: the satellite will perform the rest of the actions itself.

Robotics of the future can independently collect new knowledge, analyze it and put it into practice.

© Makhmadiev Kh.I., Ataeva G.I., 2022

УДК 519.687.7
https://doi.org/10.36906/AP-2022/51

Изибаиров Ф.Ф.
ORCID: 0000-0003-4339-7012
Казанский государственный энергетический университет
г. Казань, Россия

ПРИМЕНЕНИЕ РОБОТОТЕХНИКИ В СОЦИАЛЬНОЙ СФЕРЕ

Аннотация. Статья посвящена анализу возможностей современной робототехники в среде человека. Проведён обзор готового продукта, используемого для решения определенной проблемы в социальной сфере. На основе данного примера рассмотрены все положительные стороны для удовлетворения потребностей человека. Предложен альтернативный продукт, с более расширенными возможностями. Проанализирован его функционал, решающий повышенное количество задач и проблем. Для наглядности нового продукта проведено сравнение с предыдущим. Также предложено использование сервисного робота по новому назначению.

Ключевые слова: робототехника, социальная сфера, сервисные роботы, спрос продуктов робототехники, роботизация.

Izibairov F.F.
ORCID: 0000-0003-1661-8860
Kazan State Power Engineering University
Kazan, Russia

APPLICATION OF ROBOTICS IN THE SOCIAL SPHERE

Abstract. This article is devoted to the analysis of the possibilities of modern robotics in the human environment. The review of the finished product used to solve a certain problem in the social sphere is carried out. Based on this example, all the positive aspects for meeting human needs are considered. An alternative product with more advanced features is proposed. Its functionality is analyzed, which solves an increased number of tasks and problems. For clarity, the new product is compared with the previous one. It is also proposed to use a service robot for a new purpose.

Keywords: robotics, social sphere, service robots, demand for robotics products, robotization.

Робототехника – область науки и техники, связанная с изучением, созданием и использованием принципиально нового технического средства комплексной автоматизации производственных процессов – робототехнических систем. При использовании этих систем можно почти целиком исключить ручной труд на вспомогательных, а также и на основных технологических операциях.

CC BY

Классификации робототехнических систем делится на следующие классы:

• манипуляционные робототехнические системы

• мобильные (движущиеся) робототехнические системы

• информационные и управляющие робототехнические системы.

Манипуляционные робототехнические системы являются одним из самых востребованных классов, они получили наибольшее развитие и практическое применение.

Мобильные (движущиеся) робототехнические системы-это некоторые платформы, перемещением которых управляет автоматика. Эти роботы имеют запрограммированную автоматическую адресацию цели, а также могут автоматически загружаться и разгружаться.

Информационные и управляющие робототехнические системы – это комплексы измерительно-информационных и управляющих средств. Они автоматически производят сбор, обрабатывают и передают информацию.

Однако современная робототехника помимо производственных направлений начала активно развиваться в социальной сфере. Некоторые домашние проблемы людей начали решать роботы, одним из таких простых примеров является робот-пылесос, который способен убирать пыль и грязь полов почти во всех комнатах помещениях.

Наше общество развивается, а с ним увеличивается и уровень условий жизни человека. В следствие неизбежным является рост социальных потребностей. Люди желают более тщательнее следить за здоровьем, чистотой дома. С помощью плодов современной робототехники человек способен облегчить эти трудоёмкие процессы, используя роботов, способных решать эти задачи.

Сервисные роботы, предназначенные для того чтобы помогать людям организовывать их личную жизнедеятельность, приводят к автоматизации личности, то есть к её самостоятельности. Это является большим прорывом в развитии человеческой цивилизации. Человек получает возможность уделять больше времени интересным ему занятиям и тратить меньше сил на мелкие однотипные задачи.

Рассмотрим один из самых популярных, домашних роботов помощников. Робот-пылесос – это пылесос на основе искусственного интеллекта, который производит уборку. Его работа основывается на заложенных в нем алгоритмам и датчиках. Этот аппарат очень прост в использовании, управление может осуществляться разными вариантами: команды могут поступать в него с помощью пульта ДУ, LCD-дисплея или голоса. В более оснащённых моделях появляется возможность использовать мобильное приложение для передачи задач в любое время в любом месте, что преимущественно увеличивает пользу от данного робота. В сравнении с обычными ручными пылесосами, этот аппарат гораздо эффективнее: он способен убирать мусор в самых труднодоступных местах. Робот-пылесос значительно экономит время его пользователя. Ещё одним плюсом является его многофункциональность: некоторые модели способны проводить влажную уборку. Также этот аппарат заботится о здоровье человека, встроенные фильтры удерживают самую мелкую пыль, не выпуская её в воздух. Преимуществом робота-пылесоса также является удобство, он бесшумно выполняет свою работу, что даёт возможность проводить уборку даже в ночное время.

CC BY

В сфере социального здравоохранения активно используют сервисных роботов. Робот тюлень Paro – робот на базе искусственного интеллекта, оснащённый датчиками, способными контролировать весь рабочий процесс. Данный продукт является своего рода профессиональным зоотерапевтом. Его внешний вид вызывает умиление людей. Разработчики позиционируют Paro как робота, предназначенного для оказания успокаивающего эффекта и вызова положительных эмоций у пациентов больниц, и жителей домов престарелых. Создатель данного робота утверждает, что он может вполне заменить собой собаку или кошку в качестве домашнего животного. Paro знает своё имя, способен различать грубость или похвалу, понимать и запоминать своё месторасположение. Робот формирует свои привычки и поведение при общении с пользователем. Невероятно широкий эмоциональный отклик данного аппарата позволяет использовать его в учреждениях здравоохранения и социального обслуживания. Своеобразная зоотерапия помогает поддерживать нормальный эмоциональный фон больного человека. Робот весьма хорошо реагирует на общение, с ним можно контактировать: его можно гладить, с ним можно разговаривать. Данные функции очень полезны в детских больницах и домах престарелых. Paro является самым лечебным роботом по мнению Книги рекордов Гиннеса.

Робототехника в здравоохранение улучшает качество обслуживания пациентов, эффективность работ в клинических условиях и обеспечивает безопасную среду как для пациентов, так и для медицинских сотрудников.

Медицинское обслуживание высокого качества. Роботы поддерживают медицинские процедуры, связанные с проникновением через естественные внешние барьеры организма, проводят мониторинг состояния пациентов с хроническими заболеваниями. Также они облегчают рабочий процесс медицинского персонала, что способствует более тщательному уделению времени пациентам и высокой степени сочувствия к ним.

Операционная эффективность. Обслуживающие роботы оптимизируют однообразные и утомительные задачи, способствуют более последовательным рабочим процессам, а также снижают физическую нагрузку персонала. Эти аппараты способны подсчитывать инвентарь, осуществляя это гораздо быстрее работников, заказывать расходные материалы, лекарства и оборудования, при их необходимости. Подготовку больничных помещений для новых пациентов ускорят мобильные чистящие и дезинфицирующие роботы, позволяющие делать это быстро.

Безопасная рабочая среда. Некоторые учреждения здравоохранения используют роботов, для обеспечения безопасности персонала, перемещая расходные материалы и белье в больницах, исключая воздействие патогенных микроорганизмов, что уменьшает риск заболеваний.

Существует множество разновидных роботов, предназначенных для очищения воздуха. Каждый из них обладает конкретными задачами и имеет свой функционал. Потребность людей дышать свежим воздухом в домашних условиях привела к притоку спроса разнообразных аппаратов, способных создавать здоровый микроклимат в помещении. На данный момент одним из таких продуктов является робот Ecovacs Deebot A330. Этот прибор

CC BY

является автономно движущимся очистителем воздуха. Фильтрация воздушной среды происходит при движении робота по помещению. Ecovacs Deebot A330 способен очищать воздух от посторонних запахов, загрязнений и бактерий в: домах, офисах, в медицинских, образовательных и торговых учреждениях. Данный прибор актуален для людей с заболеваниями органов дыхания и аллергиков различного рода. Робот очищает воздух, используя трёхступенчатую систему фильтрации, которая способна удалить опасные летучие органические соединения и частицы размером 0,3–0,6 микрон. При передвижении по помещению датчики Ecovacs Deebot A330 ищут источник загрязнения. Следующим этапом робот начинает очистку воздуха.

Однако очистка воздуха от опасных летучих органических соединений и частиц является не пределом современной робототехники. Проблемами микроклимата воздушной среды помещения являются недостаточная влажность воздуха, наличие в ней опасных микроорганизмов, избыточное содержание угарного и углекислого газов. Решением данной задачи хочу предложить создание нового альтернативного робота. В его функционал будут входить следующие возможности: автоматическое перемещение по помещению, анализ микроклимата воздушной среды, демонстрация характеристики воздуха пользователю для предотвращения аллергических болезней, фильтрация опасных летучих органических соединений и частиц, мобильная очистка и дезинфекция среды от микроорганизмов, повышение влажности путём испарения ароматических жидкостей, сообщение владельцу проветривания помещения с помощью голосового помощника или мобильного приложения связанного с устройством. Данный робот будет создавать здоровый микроклимат, пригодный для людей всех возрастных категорий.

Современные роботы применяются по задуманным назначениям и поставленным им задачам. Однако робототехника не отрицает использование конкретных аппаратов в абсолютно других направлениях. Роботы способны осуществлять разные деятельности, главным для этого является то, что они должны работать во благо человечества и хороших намерений. Приносить пользу всему окружающему, а не вредить среде, в которой они находятся.

Помимо основной задачи робота Paro, который выполняет свою деятельность в сфере социального здравоохранения, как профессиональный зоотерапевт, его можно использовать для обучения собак-поводырей, которые будут работать с людьми или же для обучения собак-ищеек, способных найти заложенную бомбу, тем самым спасая множество человеческих жизней. Для этого потребуется изменить облик тюленя на породу собаки, являющейся одной из самых легко обучаемых. Внести изменения в систему для работы с животными. Возможно придётся добавить какое- то количество новых датчиков в арсенал аппарата. Однако имеющегося функционала уже будет достаточно, чтобы анализировать повадки собак и привыкать к ним. Благодаря искусственному интеллекту, Paro сможет обучиться работе с животными, а затем начнёт обучать их сам. Собаки смогут контактировать с роботом и получать навыки, необходимые для работы с людьми, а также усваивать умения для поисков опасных объектов, использующихся недоброжелателями. Это приведёт к большому

CC BY

количеству профессионально обученных собак, способных работать с людьми, и множеству сбережённых жизней. Данный пример показывает нам многофункциональность сервисных роботов.

Таким образом, исходя из вышесказанного, можно сказать, что использование роботов в социальной сфере – огромный прорыв. Раньше людям приходилось самим заниматься рутинными делами, затрачивая на это свой физический ресурс. Благодаря таким роботам жизнь человека стала намного проще, люди начали экономить огромное количество времени, у них появилась возможность более тщательнее следить за своим здоровьем, лучше контролировать свой эмоциональный фон. Эти роботы способны помогать с психологическими проблемами, заботиться о дееспособных людях, приносить пользу окружающей среде сообщать человеку об угрозах. Плоды современной робототехники стали обыденностью в наши дни.

© Изибаиров Ф.Ф., 2022

УДК 681.5
https://doi.org/10.36906/AP-2022/52

Исмоилова М.Н.
Бухарский государственный университет
г. Бухара, Узбекистан

ЭТАПЫ РАЗВИТИЯ РОБОТОТЕХНИКИ

Аннотация. Робототехника – раздел машиностроения, электротехники и информатики, который занимается проектированием, конструированием, эксплуатацией и применением роботов, а также компьютерными системами для их управления, сенсорной обратной связи и обработки информации. Эти технологии имеют дело с автоматизированными машинами, которые заменяют людей в опасных средах или производственных процессах или похожи на людей по внешнему виду, поведению и восприятию. В статье представлена информация об этапах развития робототехники.

Ключевые слова: роботы; промышленные роботы; механическая рука; исследование; математическая модель.

Ismoilova M.N.
Bukhara State University
Bukhara, Uzbekistan

STAGES OF ROBOTICS DEVELOPMENT

Abstract. Robotics is a branch of mechanical engineering, electrical engineering and computer science that deals with the design, construction, operation and application of robots, as well as computer systems for their control, sensory feedback and information processing. These technologies deal with automated machines that replace humans in hazardous environments or production processes or are similar to humans in appearance, behavior and perception. The article provides information about the stages of robotics development.

Keywords: robots; industrial robots; mechanical arm; research; mathematical model.

С момента своего появления и по сегодняшний день около 10% существующих роботов используются в машиностроении, а около 70% – в механической обработке, штамповке и транспортировке. Это связано с тем, что эти процессы включают в себя простые операции, такие как извлечение и установка большого количества деталей. Для применения роботов в других отраслях необходимо совершенствовать их конструкцию. В 1985 году доля роботов, используемых в штамповке и механической обработке, была уменьшена до 60 процентов, а в сборочных работах-увеличена до 20 процентов. В 1990 году было запланировано довести количество роботов, используемых в сборке, сварке, окраске, до 33 процентов. Но этот план остался нереализованным.

CC BY

Вы когда-нибудь наблюдали, как люди работают перед сборочным конвейером на автомобильном заводе? По конвейеру, длина которого составляет около километра, начинает поступать кузов автомобиля. С двух его сторон по ходу движения, сверху на тележках, транспортёры собирают двигатели, колеса, болты, гайки, стекла, сиденья, ну и все члены машины идут по порядку один за другим. Кузов крепится и декорируется десятками, сотнями тысяч узлов и деталей, смотришь каждую минуту на готовую машину. Автомобиль за автомобилем.

Их собирает огромное количество людей – операторов. На конвейере все процессы специализированы. Весь процесс сборки строго распределён, разбит на отдельные операции. Каждая операция возлагалась на одного, а иногда и на двух операторов, с учётом её выполнения за одну минуту. «В минуту-автомобиль» – означает, что человек должен каждую минуту начинать и завершать порученную ему операцию. Одна операция каждую минуту. Два часа спустя десятиминутный перерыв, а затем ещё одна операция в минуту.

Тысячи людей с утра до вечера стоят перед машинами, станками, конвейерами. Однообразие выполняемой работы утомляет служащего, а также требует значительных усилий. Автоматизация производственных процессов в определенной степени освободила людей от ряда тяжёлых работ. Но эта проделанная работа невероятно скудна с точки зрения удовлетворения потребностей общества.

Машины становятся все более сложными, в то время как человеческий труд, который их использует, все ещё остаётся простым, неквалифицированным. По мере того, как производство становится все более механизированным и автоматизированным, увеличивается количество однородных, насыщающих человека видов труда. Как только опытный и знающий токарь приступает к изготовлению детали, он, в первую очередь, творчески подходит к работе, сам устанавливает сырье на станок, сам затачивает резак, ну и все работы выполняет сам, и, наконец, пусть рука ударит, создавая деталь. Такой квалифицированный токарь, безусловно, ценится во все времена. Однако бесчисленное количество машин и автоматов требуют такой же работы, другими словами, все интересные задачи ложатся на машины, а скучные-на людей.

Поскольку создаётся машина, которая может выполнять множество сложных и точных задач без вмешательства человека, нельзя ли оснастить её рядом других устройств и назначить ей другие задачи?

Например, если станок самообслуживания, он устанавливает заготовку на станок, берет готовое изделие и кладёт его на место. Почему бы не создать такой идеальный автомобиль самообслуживания?

Станки, токарные станки, различные устройства, сборочные операции кажутся простыми и лёгкими, если они выполняются с большими усилиями. Однако, если одна и та же работа автоматизирована с использованием традиционных методов и инструментов, система автоматизации становится более сложной и становится слишком узкой для адаптации к производству только одного однородного продукта. В результате универсальность машины ограничена, а производство такой машины очень дорогое.

CC BY

Управление машин требует от оператора ума, ловкости и определенных навыков. Чтобы автоматизировать его работу, потребуется создать целый набор специальных машин и роботов, которые смогут заменить человека.

Многие производственные процессы состоят из набора различных операций, которые повторяются несколько раз. Периодичность также можно назвать законом механизированного, автоматизированного производства. Робот, как и другие автоматы, предназначен для выполнения задач, которые всегда обратимы. В этом смысле робот похож на любую другую машину.

В настоящее время промышленный робот – это универсальная система различного назначения. Его можно использовать в работе различных машин и станков, в управлении различными технологическими процессами. Промышленный робот, оснащённый программной системой управления, как и другие машины, называется автоматическим манипулятором с программным управлением.

Рабочий орган робота-механическая рука, которая действует и выполняет работу, напоминает руку человека. Вот чем он отличается от других автоматов, и человек становится ближе к операторской деятельности. Получить представление о научно-техническом уровне современных промышленных роботов можно путём анализа рабочих параметров серийно выпускаемых роботов. К важнейшим рабочим параметрам относятся несущая способность, тип управления, степень подвижности, скорость перемещения, точность позиционирования, внешний вид передатчиков, надёжность, стоимость, вес, габариты и др.

Скорость промышленных роботов недостаточна. Сравнительный анализ роботов, производимых за рубежом и в нашей стране, показывает, что робототехника в нашей стране развивается в соответствии с общими тенденциями и требованиями.

В 1985 году в московских «Сокольниках» прошла выставка «Роботокомплексы 85». На этой выставке были представлены современные промышленные роботы, способные выполнять самые сложные операции.

На выставке на стендах более двухсот фирм из Болгарии, Чехословакии, США, Англии, Австрии, Японии, Финляндии, Франции, ФРГ, Швеции, Швейцарии были представлены не только отдельные модели роботов, но и созданные с их помощью робототехнологические комплексы, гибкие производственные модули и системы. Например, на немецкой экспозиции можно было познакомиться с робототехническим комплексом, изготавливающим детали из пластика и металла, а на болгарской-с роботами, обслуживающими токарные станки и обрабатывающие центры. Особый интерес у экспонентов вызвал робототехнический комплекс для сварки, который был представлен на чехословацкой экспозиции. Разнообразна и экспозиция, привезённая финнами. В частности, на стенде фирмы Nokia можно было ознакомиться с роботизированной автоматической системой с возможностью просмотра. На австрийской экспозиции был представлен робот, способный разгружать грузы, на японской-роботы, способные выполнять операции покраски, выполняемые фирмой «Санко Сейки», выполнять операции, связанные с контролем металлургического продукта.

CC BY

В рамках выставки состоялся научно-технический симпозиум, на котором специалисты из разных стран проанализировали проблемы дальнейшего развития робототехники. Печатный робот разработан для автоматизации стандартного процесса печати в электронной промышленности. Этот робот самодостаточен с заготовками, печатает на печатных платах различные рисунки и ставит готовую продукцию на конвейер. Операции запрограммированы, а производительность можно регулировать в больших масштабах. Использование печатного робота позволило оператору освободить территорию от опасности для здоровья.

Робот, созданный НПО «Силикат», избавил пять человек от тяжёлого ручного труда. Процесс, который обычно выполняется вручную, поручается этому роботу, который кладёт кирпичи на тележки, чтобы поместить их в кирпичную печь.

Машины успешно используются на нескольких кирпичных заводах по всей стране. Однако специалисты профсоюза не только повторили знаменитую петербургскую схему, но и значительно ее усовершенствовали.

Созданный в Ташкенте робот более компактен, его пневмодержащие «руки» работает эффективнее. За час он положит пять тысяч кирпичей. Промышленный вариант робота прошел испытания на Ташкентском экспериментальном заводе строительных материалов. На заводах республики до сих пор кирпичи делают из влажной глины. Это делает формованный кирпич более мягким, и машина может его повредить. Иная ситуация с линией Naked, которая производит 20 миллионов кирпичей в год. Именно поэтому здесь был запущен первый робот.

В течение 1980–1985 гг. На заводе Ташкентского тракторного производственного объединения создано и введено в эксплуатацию пять роботизированных технологических комплексов по шлифовке (РТК):

1. РТК для обработки мелких деталей кузова. Кузов крана 102В был принят как представитель этих деталей. Линия состоит из буровой установки модели 2Р135Ф2, управляемой 3 цифровыми программами, токарного полуавтомата модели NT300, токарного станка 2Г135 и 2 роботов RB 231;

2. Линия обработки алюминиевых колпачков трех типов состоит из 2 моделей модели 1N713, сверлильного станка модели 2G175 и робота RB 231;

3. Комплексная линия по обработке валов валов состоит из одного токарного станка модели MP71, двух токарных станков с ЧПУ 1717ФЗ, двух шлицевых станков МШ300 Т и двух роботов Р505;

4. Линия для обработки клиновых и пазовых валов шпинделя состоит из фрезерно-центрирующего станка MR71M, специального станка с токарным станком 1713FZ с управлением 2 цифровыми программами и 2 роботов R505;

5. Комплексная линия обработки валов с призматической шпонкой и пазами состоит из 2-х фрезерно-центровочных станков MR71M, 2-х токарных станков 1N713, 2-х специальных токарных станков и 3-х роботов R505.

Эти линии присоединяются к единому участку, образуя самостоятельную производственную структурную единицу. Всего на этой линии 14 роботов, 24 станка.

CC BY

Успех предприятий, их эффективное влияние на экономику, на все сферы жизни общества не могут быть обеспечены только усилиями научных работников, разумеется. Все большее значение приобретает вовлечение в этот процесс, имеющий огромное значение для народного хозяйства, всех работников общественного производства, всех звеньев хозяйственного механизма.

Развитие предприятий требует коренного изменения методов и методов хозяйственной деятельности, решительной борьбы с коллективизмом, искреннего уважения к науке, слаженной работы с людьми науки.

Далеко не все руководители производственных объединений и предприятий до конца осознали всю важность этого требования. Об этом свидетельствуют факты неудовлетворительного внедрения завершенных научно-исследовательских работ, пренебрежительного отношения к заключению хозяйственных договоров с научными учреждениями и высшими учебными заведениями, отказа в создании отраслевых научно-исследовательских лабораторий на базе институтов и высших учебных заведений, нежелания создавать на предприятиях опытно-экспериментальную базу. В последнее время применение на предприятиях роботизированных, роботизированных комплексов, гибких автоматизированных систем становится не только необходимостью жизни, но и модой.

Роботизированные, роботизированные комплексы, гибкие автоматизированные системы должны быть построены и внедрены, прежде всего, в условиях предприятий с всесторонним анализом, на основе надежного и высокопроизводительного оборудования. Автоматизация производства, в свою очередь, ставит перед собой задачу совместить автоматизированную систему проектирования с гибкой автоматизированной системой.

Разработка системы автоматизированного проектирования заключается в оснащении предприятия современной вычислительной техникой и оснащении её средствами на этой основе. Создание гибких систем автоматизированного проектирования, основанных на передовой технологии, представляет собой сложную проблему, требующую сочетания теоретических исследований и производства.

Можно с уверенностью сказать, что широкое внедрение кибернетических методов остается одним из главных направлений научно-технического прогресса. Важнейшим из кибернетических методов является моделирование. Конструкторы изучают его характеристики в различных моделях, прежде чем строить какую-либо машину.

С помощью уравнений на языке математики можно описать процессы, относящиеся к различным объектам, и получить ответы, когда они выражены в числах. Такая постановка исследуемого процесса называется математическим моделированием. Математические модели, используемые в современной исследовательской работе, настолько сложны, что их невозможно проанализировать без современных высокопроизводительных компьютеров.

Исследователи, многократно проверяя процессы, происходящие на объекте, на математической модели, получают полную информацию о состоянии процесса, а также возможность спроектировать и эффективно управлять оптимальным вариантом конструкции.

CC BY

История внедрения математической модели началась с решения такой высокой задачи, как ядерная энергия и освоение космического пространства. В настоящее время важное практическое применение этих методов порождает проблемы управления технологическими процессами, разработки роботизированных комплексов и гибких автоматизированных систем. Выражается в виде формул и признается физическим законом только в случае проведения точных количественных расчетов.

Некоторые другие науки еще не достигли этого уровня, поэтому их стремительный рост находится на грани краха. Взять, к примеру, управление технологическими процессами, робототехнический комплекс, гибкие автоматизированные системы. Отдавая должное огромному труду технологов и машиностроителей, необходимо твердо признать, что машиностроение еще не является наукой с точными и количественными расчетами. Законы развития задач металлообработки, управления технологическими процессами, робототехнических комплексов, гибких автоматизированных систем еще не полностью отражены в уравнениях и формулах. Многие исследователи и инженеры, техники и ученые винят компьютерные технологии. Однако основной причиной этого является не только компьютер, но и отсутствие разработки жизненно важных математических моделей. Поэтому жизнь требует разработки математических моделей.

Часто говорят, что задача математиков – разработать математическую модель и провести на её основе вычислительные эксперименты. Немногие люди понимают эту модель и уравнение профессионально. Каждый профессионал должен в полной мере использовать свой опыт для описания процессов в своей области. Эти проблемы касаются почтовой службы в целом.

Лучшая модель создается только тогда, когда мы работаем вместе с математиками, физиками, химиками, биологами и технологами, им нужно понимать друг друга и находить общий язык. Итак, теперь вы можете услышать термины техническая кибернетика, медицинская кибернетика, сельскохозяйственная кибернетика, химическая кибернетика, кибернетика машиностроения и так далее. Именно поэтому в средних школах, лицеях и профессиональных колледжах, вузах на учебных курсах рекомендуется изучать кибернетику, компьютеризацию. Одна из следующих задач - углубленный анализ процессов почтовой связи и создание ее математических моделей.

Промышленных роботов принято делить на три поколения в зависимости от функции и сложности услуги (как компьютер). Первое поколение промышленных роботов, которые широко используются сегодня, имеют электромеханическую систему, программное обеспечение и память и не имеют возможности обеспечивать обратную связь. Определение координат выполняется с помощью механических устройств и высокоточных потенциометров. Роботы первого поколения способны толкать грузы весом от нескольких граммов до нескольких тонн, а программа на запоминающем устройстве может выполнять до 1024 движений.

Большинство роботов этого поколения имеют три уровня скольжения, а также рукоятку рычага. Устройство рычажной ручки промышленных роботов первого поколения в

CC BY

эксплуатации имеет пять уровней перемещения: вращение в горизонтальной плоскости, линейное перемещение в вертикальной плоскости и вдоль оси, вращение вокруг собственной оси и изгиб в вертикальной плоскости. Во втором поколении промышленных роботов последовательность операций и память координат осуществляется с помощью электронных схем, размер памяти позволяет записывать координаты в виде чисел. Это, в свою очередь, позволяет создать сложную программу операций. Эти роботы более маневренны, чем промышленные роботы первого поколения, и могут хранить сложные программы на запоминающем устройстве. Некоторые из этих промышленных роботов имеют фотоэлектрические и сенсорные устройства, которые могут медленно принимать форму и состояние детали. Роботы второго поколения могут работать с очень высокой точностью в тех же условиях, что и базовое оборудование. Высокий уровень надежности цифровых программно-управляемых систем обеспечивает их бесперебойную работу в течение длительного времени.

Факторы, определяющие целесообразность использования роботов на конкретных рабочих местах, включают:

1. Возможность изготовления деталей с изменяемой номенклатурой и высокой степенью модерации технологического процесса;

2. Повышение производительности труда. Повышение продуктивности коктейлей следует анализировать в течение длительного периода времени (не менее одного месяца). Поскольку скорость, с которой движется робот, намного медленнее, чем скорость, с которой движется человек, успех в математике может быть достигнут только за счет неэффективного использования рабочего времени;

3. Стабилизировать технологический процесс и улучшить качество продукции за счёт исключения человеческого фактора при однородных и опасных операциях.

© Исмоилова М.Н., 2022

УДК 004.5
https://doi.org/10.36906/AP-2022/53

Никонова Е.З.
ORCID: 0000-0003-4416-2629, канд. пед. наук
Лемиш Д.В.
ORCID: 0000-0003-4142-7441
Нижневартовский государственный университет
г. Нижневартовск, Россия

СЕМАНТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ БЕСКОНТАКТНОГО УПРАВЛЕНИЯ ЭВМ

Аннотация. В статье рассматривается процесс моделирования системы бесконтактного управления ЭВМ, необходимый для выявления основных проблем и разработки стратегии реализации будущей системы. Раскрывается проблема взаимодействия пользователей с ОВЗ с персональным компьютером. Рассматриваются инструментальные средства разработки системы, а также алгоритм работы системы в нотации EPC.

Ключевые слова: система; бесконтактное управление; ЭВМ; ограниченные возможности здоровья.

Nikonova E.Z.
ORCID:0000-0003-4416-2629, Ph.D.
Lemish D.V.
ORCID: 0000-0003-4142-7441
Niznevartovsk State University
Nizhnevartovsk, Russia

SEMANTIC MODELING OF THE SYSTEM OF CONTACTLESS CONTROL AN ELECTRONIC COMPUTER

Abstract. The article discusses the process of modeling a contactless computer control system, which is necessary to identify the main problems and develop a strategy for the implementation of the future system. The problem of interaction of users with HIA with a personal computer is revealed. The system development tools are considered, as well as the system operation algorithm in EPC notation.

Keywords: system; contactless control; computers; limited health opportunities.

Текущий уровень развития информационных технологий способствует активному внедрению ассистивных технологий во все сферы деятельности человека. Данные технологии предназначены для предоставления равных возможностей использования современных технических средств между рядовыми пользователями и лицами с ограниченными

CC BY

возможностями здоровья. На сегодняшний день ассистивные технологии являются одним из приоритетных направлений развития страны, что подтверждается разработкой государственной программы «Доступная среда», реализация которой запланирована на период с 2011 по 2025 год. Целью данной программы является создание условий для интеграции лиц с ограниченными возможностями здоровья в общество и повышение их качества жизни. Кроме того, использование ассистивных технологий регулируется различными государственными нормативно-правовыми документами. Так, например, с целью обеспечения равных условий получения качественного образования, Федеральный закон № 273 – ФЗ «Об образовании в Российской Федерации» определяет условия обучения и развития обучающихся с ограниченными возможностями здоровья, включающие в себя использование специализированных образовательных программ и методом обучения, учебных материалов и специальных технических средств. Совокупность данных фактов обосновывает высокую актуальность темы данной работы.

Цель работы заключается в моделировании системы бесконтактного управления электронно-вычислительной машиной (ЭВМ), в частности персональным компьютером (ПК), которое позволит выявить основные проблемы и разработать стратегию реализации будущей системы.

Объектом исследования являются информационные технологии в области прикладного программного обеспечения и искусственного интеллекта. Предметом исследования является ассистивная технология бесконтактного управления ЭВМ.

В процессе формирования и анализа требований к будущему программному продукту, в первую очередь были изучены особенности устройства ЭВМ и её органов управления, а также возможности взаимодействия человека с ЭВМ.

Как известно, персональный компьютер является комплексом технических (уровень архитектуры), аппаратных (уровень комплектующих) и программных средств (уровень программного обеспечения), предназначенных для автоматической обработки информации. Ключевым элементом любой современной ЭВМ на аппаратном уровне являются так называемые периферийные устройства, предназначенные для ввода и вывода информации. К таким устройствам относятся: клавиатура, компьютерная мышь, монитор, сканер, принтер, видеокамера и т. д.

Периферийные устройства являются основным интерфейсом взаимодействия человека с ЭВМ. Они получают информацию посредством физического воздействия пользователя на устройство и преобразуют данные воздействия в инструкции, которые, впоследствии, передаются процессору при помощи контроллера устройства и операционной системы.

Таким образом, основным условием взаимодействия человека с ЭВМ является возможность его физического воздействия на периферийные устройства. Однако, данное условие может стать невыполнимым вследствие различных нарушений опорно-двигательного аппарата пользователя.

CC BY

Для осуществления возможности взаимодействия пользователя с ЭВМ, независимо от данного условия, разрабатываемая система должна предоставить человеку возможность использования ЭВМ посредством бесконтактного управления.

По данным открытого источника «Ain», наиболее популярными операционными системами для персональных компьютеров являются:

– операционная система Microsoft Windows, которая составляет 60% от общего числа пользователей ПК;

– операционная система MacOS, которая составляет 5% пользователей;

– операционная система Linux, которая составляет 2,33% пользователей.

Исходя из данного факта, для предоставления обширных возможностей использования персонального компьютера людям с ограниченными возможностями здоровья, разрабатываемая система также должна удовлетворять условию кроссплатформенности, т. е. должна поддерживаться всеми популярными операционными системами.

На сегодняшний день существуют аналогичные системы бесконтактного управления, однако, помимо высокой стоимости, они требуют наличия дополнительного оборудования и определенных знаний по его инсталляции. Кроме того, такие системы обладают узкой направленностью. Так, например, контроллер Tobii Eye Tracker 5 (примерная стоимость двадцать тысяч рублей) предназначен лишь для поддержки пользователя в прохождении компьютерных игр.

Для решения указанных проблем разрабатываемая система должна быть независима от оборудования, автоматизирована и доступна.

Таким образом, в результате формирования и анализа требований была составлена спецификация требований, которая представлена в таблице.

Таблица

№	Описание
Спецификация функциональных требований	
1	Система должна предоставлять пользователям возможность бесконтактного управления ЭВМ.
2	Система должна быть совместима с такими операционными системами, как: Microsoft Windows, MacOS, Linux.
3	Система должна быть независимой от оборудования. Это означает, что любое оборудование должно быть совместимо с системой.
4	Система должна быть автоматизированной. Это означает, что необходимо свести к минимуму число действий пользователя, необходимых для использования системы.
5	Система должна быть доступна. Это означает, что стоимость конечного продукта должна быть посильной практически всем пользователям. Из данного требования следует целесообразность использования свободно распространяемых технических и программных средств разработки программных продуктов.
Спецификация нефункциональных требований	
1	Система должна отвечать всем современным требованиям концепции пользовательского опыта
2	Система должна быть удобной для использования
3	Система должна предусматривать различные подходы к управлению ЭВМ, людьми, с различными особенностями развития
4	Система должна обеспечивать возможность полноценного управления ЭВМ абсолютно без физического контакта пользователя с периферийными устройствами. Это означает, что

CC BY

	система должна предусматривать все возможные сценарии использования ЭВМ и содержать в себе вспомогательный функционал, заменяющий привычные методы взаимодействия пользователя с ЭВМ.

Наиболее удачным выбором для разработки данной системы является технология Java Runtime Environment – минимальная реализация виртуальной машины, необходимая для исполнения программного кода, разработанного с использованием языка программирования Java. На сегодняшний день данная технология реализована в большинстве известных операционных систем, включая Microsoft Windows, MacOS, Linux.

Более того, язык программирования Java не зависит от платформы. Это означает, что использование данной технологии позволит значительно сократить время и стоимость разработки системы, а также упростит её перенос на различные платформы.

Для разработки также понадобятся прикладные программные интерфейсы (API) указанных операционных систем. Данные интерфейсы содержат в себе реализацию функционала целевой операционной системы (классы, а также их свойства и методы), необходимые для разработки приложения.

Для реализации графического пользовательского интерфейса (GUI) программного продукта необходимо использовать различные графические редакторы векторного и растрового типа. Среди растровых графических редакторов стоит отметить Paint.Net и GIMP – бесплатные редакторы, обладающие полным набором функций, необходимых для реализации GUI. Среди векторных графических редакторов стоит отметить редактор Gravit Designer – бесплатный векторный графический редактор, который реализован в большинстве известных платформ.

Рассмотрим алгоритм работы системы, реализованный в нотации EPC (см. рис.).

Согласно данному алгоритму после запуска операционной системы служба автоматически запускается служба бесконтактного управления ЭВМ. В случае, если служба была успешно запущена, происходит автоматический поиск и подключение периферийного устройства – видеокамеры. После того, как была найдена и подключена видеокамера, система циклически получает с неё изображение и передаёт его службе бесконтактного управления. Данная служба, в свою очередь, посредством обученной нейронной сети выполняет поиск силуэта человека (пользователя). Если пользователь был успешно найден, нейронная сеть считывает состояние пользователя и ожидает от неё управляющих жестов (силуэтов). В случае успешного определения управляющего силуэта служба формирует определенные команды и отправляет их на исполнение операционной системе.

Таким образом, благодаря автоматическому запуску и поиску нужного периферийного устройства, количество действий пользователя, необходимых для использования системы, будут сведены к минимуму, а использование обученной нейронной сети открывает обширные возможности разработки новых интерфейсов взаимодействия человека с ЭВМ, исключающих их физический контакт. Также, благодаря использованию простейших периферийных устройств, система будет доступна большому числу пользователей.

CC BY

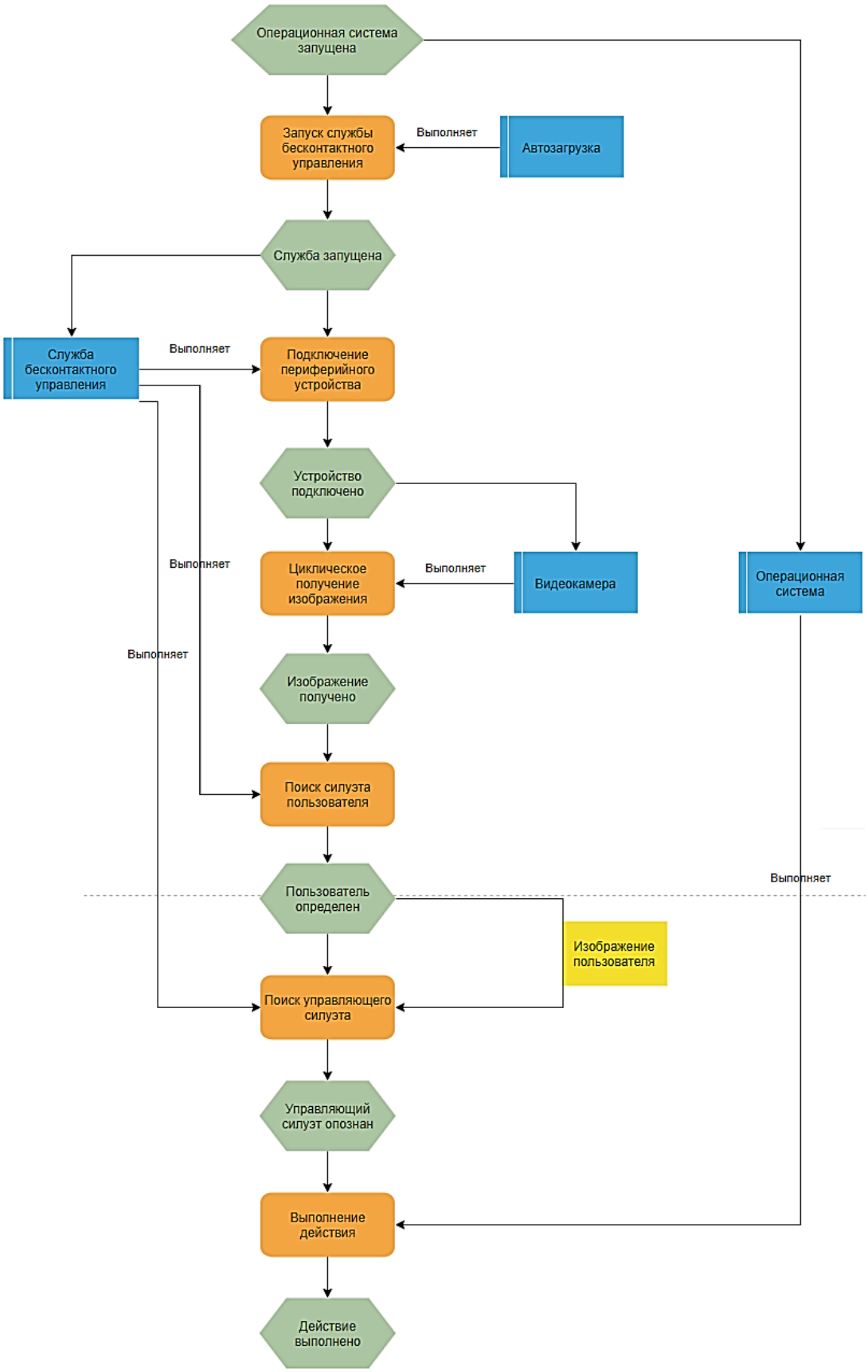


Рис. Алгоритм работы системы в нотации EPC

© Никонова Е.З., Лемиш Д.В., 2022

УДК 371.31
https://doi.org/10.36906/AP-2022/54

Виштак О.В.
Мамедов Р.Р.
Балаковский инженерно-технологический институт –
филиал Национального исследовательского ядерного университета «МИФИ»
г. Балаково, Россия

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПРЕДПРИЯТИЯ МАЛОГО БИЗНЕСА

Аннотация. В статье представлен обзор типовых задач предприятия малого бизнеса. Предложено решение по автоматизации работы с клиентами, с помощью информационных технологий. Представлено описание системы для работы с клиентами, её задач и функций.

Ключевые слова: автоматизация бизнеса; система для работы с клиентами; CRM-система; ИТ-решения для малого бизнеса.

Vishtak O.V.
Mammadov R.R.
Balakovo Institute of Engineering and Technology –
branch of the National Research Nuclear University «MEPhI»
Balakovo, Russia

DEVELOPMENT OF A WEB APPLICATION FOR A SMALL BUSINESS ENTERPRISE

Abstract. The article presents an overview of typical tasks of a small business enterprise. A solution for automating work with clients using information technology is proposed. The description of the system for working with clients, its tasks and functions is presented.

Keywords: business automation; customer service system; CRM system; IT solutions for small businesses.

На первый взгляд может показаться, что малому бизнесу необязательно нужны инструменты автоматизации. Но на самом деле все наоборот: чтобы рационально использовать ограниченные трудовые и финансовые ресурсы и выполнить все заказы в срок и с высоким качеством необходимы эффективные инструменты автоматизации,

В данной статье рассматривается предприятие малого бизнеса «Barbershop b13». Данное предприятие оказывает разовые парикмахерские и косметологические услуги потребителям. Ежедневно это предприятие сферы услуг сталкивается с рядом типовых задач:

– регистрация и отслеживание заявок на обслуживание клиентов;
– кассовый учёт доходов за оказание розничных услуг;
– расчёт заработной платы сотрудников;

CC BY

– закупка расходных материалов, необходимых для оказания услуг;

– ведение расписания заявок клиентов, для мастеров компании;

– рекламная деятельность;

– ведение статистики прибыли и издержек за определенные периоды времени.

Согласно ФЗ № 402-ФЗ «О бухгалтерском учёте», предприятия малого бизнеса могут не иметь бухгалтерского отдела, при условии, что они самостоятельно ведут учёт доходов и расходов для учёта налогов.

Для разработки информационной системы предприятия проводится анализ бизнес-процессов предприятия в нотации IDEF0, исходя из принципа «AS-IS» [1]. На диаграмме описан процесс выполнения услуги барбершопа, со стороны клиента.

На вход бизнес процесса поступает клиент, которые заинтересован в услуге предприятия, на выходе – клиент, которому оказали интересующую его услугу. Данный бизнес процесс регламентируется порядком предоставления услуг, расписанием мастеров-парикмахеров, и списком цен на услуги.

Действующими лицами бизнес-процесса являются клиент, и парикмахер, которые используют в работе парикмахерские инструменты, телефон для записи, расходуемые косметические средства, и деньги за оказанную услугу.

На рисунке 1 представлена контекстная диаграмма бизнес-процесса «Обслуживание клиента в Барбершопе b13».

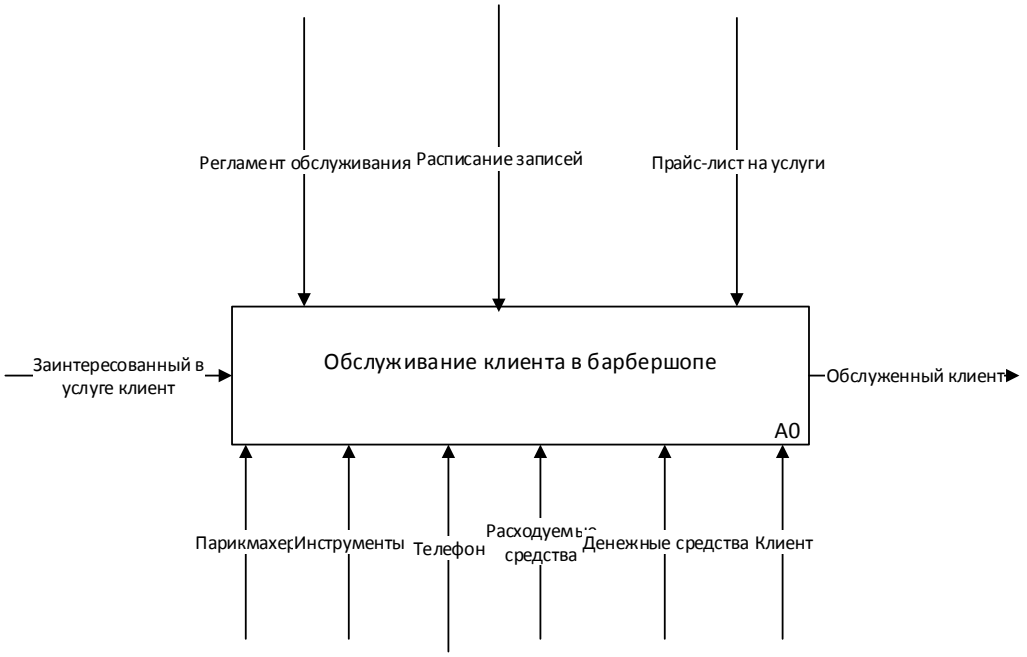


Рис. 1. Контекстная диаграмма бизнес-процесса «Обслуживание клиента в Барбершопе b13e»

Далее бизнес-процесс был разбит на подпроцессы, из которых была составлена диаграмма декомпозиции.

Всего выделено 4 этапа: запись клиента по телефону, выбор интересующей его услуги (модели стрижки, бороды) в соответствии с прайс-листом предприятия, оказание услуги мастером-парикмахером клиенту, и оплата оказанной услуги, в соответствии с прайс-листом.

CC BY

На вход, выход, управление, и механизм диаграммы поступают те же интерфейсные дуги, что и на контекстной диаграмме данного бизнес-процесса.

На рисунке 2 представлена диаграмма декомпозиции бизнес-процесса «Обслуживание клиента в Барбершопе b13».

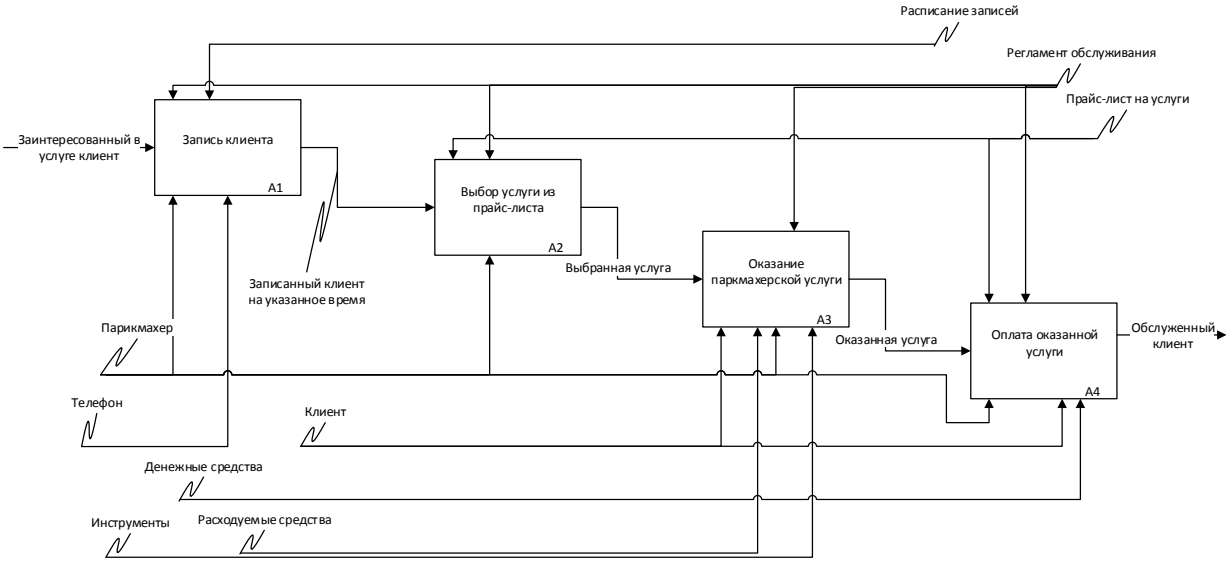


Рис. 2. Диаграмма декомпозиции бизнес-процесса «Обслуживание клиента в Барбершопе b13»

Далее проводится анализ программных решений, автоматизирующих рутинные задачи. На российском рынке существует несколько популярных решений для малого бизнеса. «Битрикс24» – полноценный корпоративный портал, подходит как малому бизнесу, в том числе стартапам, так и гигантам рынка (https://clck.ru/XaDTz). Что немаловажно для молодого бизнеса, система абсолютно бесплатна для компаний, штат которых – не более 12 человек. При этом предоставляются следующие возможности:

– ведение и управление базой данных клиентов;

– доступ к облачному хранилищу данных;

– полное управление проектами и задачами;

– управление продажами, в т. ч. отслеживание заказов и оплат, а также складской учёт;

– аналитика (графики, воронки продаж);

– планирование дел – таск- и тайм-менеджмент для сотрудников;

– работа с документами и финансами;

– интеграция с сайтом, социальными сетями, популярными веб-приложениями, а также 1С (сразу отображается вся информация по счетам);

– email-трекер – сохранение истории переписки внутри системы;

– мобильные приложения для ОС iOS и Android.

Недостатки:

– бесплатная версия ограничена по количеству сотрудников организации;

– система аналитики слишком сложная и перегруженная;

– нету функции автоматической рассылки.

CC BY

«Мегаплан» – корпоративная облачная система преимущественно для малого и среднего бизнеса (https://clck.ru/XaDTz).

– полное управление продажами (планирование многоэтапных продаж, отслеживание каналов сбыта, формирование аналитических отчётов);

– точная база данных по клиентам (сохранение истории отношений с клиентами);

– внутренний менеджмент – выставление задач для сотрудников, анализ их продуктивности, контроль сроков выполнения работы;

– интеграция с телефонией, почтовыми сервисами и 1С.

Недостатки:

– бесплатная версия ограничена по функционалу;

– нету функции автоматической рассылки.

«Простой Бизнес» – отечественный CRM-проект для автоматизации бизнеса (https://clck.ru/XaDTz). Предлагает тариф без оплаты для компаний с пятью сотрудниками, для крупных организаций существует общая платная лицензия без оплаты за каждого работника.

– учёт клиентов (ведение полной клиентской базы, сохранение контактов и звонков, учёт документов по сделкам);

– управление проектами (task-менеджер, календари дел, диаграммы, графики продаж);

– шефство над персоналом организации (time-менеджмент, управление удалёнными сотрудниками, отслеживание личной эффективности, внутренний чат и телефония);

– ведение классического документооборота (шаблоны для оформления документов, отправка документации по email, ЭЦП);

– ведение бухгалтерии и учёта по складу (движение денежных средств, электронные отчёты, проводки, управление складом);

– наличие встроенных коммуникаций (интеграция с телефонией и почтовыми мессенджерами);

– современная сквозная аналитика бизнеса (анализ эффективности рекламных затрат, аналитика по параметрам: ROI, продажи, конверсия, средний чек, прибыль).

Недостатки:

– бесплатная версия ограничена по количеству сотрудников организации;

– нету функции автоматической рассылки.

Trello – система, которая заслужила популярность у стартаперов благодаря простоте, интуитивно понятному интерфейсу и наличию бесплатной версии (freemium-модель, но ограничения не по тестовому периоду, а по количеству интеграций и объёму загружаемых файлов) (https://clck.ru/XaDTz). Организована по принципу японских канбан-досок: последовательный контроль за всеми этапами выполнения задач и проектов. Под каждый проект система выделяет доску, разделённую на списки с карточками. Карточки имеют массу возможностей: указание задачи, голосование, комментирование, загрузка файлов и т. д. (https://clck.ru/XaDTz).

Недостатки:

– бесплатная версия ограничена по тестовому периоду;

CC BY

– автоматизирует внутренние задачи компании, но не подходит для взаимодействий с клиентами;

– нет функции автоматической рассылки.

Вышеописанные программные решения, так или иначе автоматизируют задачи общения с клиентами, осуществляют контроль транзакций, что относит такие системы к классу CRM.

Система управления взаимоотношениями с клиентами (CRM, CRM-система, сокращение от англ. Customer Relationship Management) – прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов. Они помогают вести учёт сделок и контрагентов, напоминают о запланированных встречах, важных звонках и письмах, автоматизируют рутинные процессы и в конечном счёте приводят к росту прибыли.

Все вышеописанные системы обладают одним или несколькими недостатками. Так, все системы имеют бесплатную демо-версию, сильно ограниченную по функционалу, времени использования, или же по количеству сотрудников в компании. Разрабатываемый продукт планируется сделать бесплатным для использования, без ограничения функционала. Кроме того, не во всех представленных системах удобная и информативная система подсчета статистики продаж. Также ни в одной системе нету автоматической рассылки писем рекламного характера по базе «горячих» клиентов.

Разрабатываемый продукт будет включать в себя все лучшие качества существующих CRM-систем, и устранит вышеописанные недостатки.

Таким образом, разработка CRM-системы для предприятия малого бизнеса целесообразна.

Функционал разрабатываемой CRM-системы

– хранение информации в базе данных – CRM-система собирает и хранит данные о клиентах;

– контроль сотрудников – CRM-система фиксирует все задачи, напоминает о сроках и уведомляет о просроченных делах сотрудников;

– автоматическое переназначение ответственного сотрудника, а также по каждому клиенту ставит задачи на каждом этапе продажи;

– минимизация ошибок из-за человеческого фактора

– массовые рассылки рекламного характера, с продуманными персонализированными шаблонами разных целевых аудиториях позволяют поддерживать постоянный контакт с потенциальными и существующими клиентами. В CRM есть возможность создавать обычные текстовые письма или формировать яркие письма в фирменном стиле компании в формате HTML.

– аналитика и отчётность – В интерфейсе CRM владелец компании может видеть, какие клиенты приносят ему наибольшую прибыль и насколько быстро менеджеры реагируют на

CC BY

поступающие запросы. Также система позволяет легко увидеть, какие рекламные каналы были наиболее эффективными, а какие не принесли результатов. При это владелец видит результативность отдельных сотрудников или отделов в целом и динамику выполнения плана продаж.

– управление отделами – CRM помогает различным отделам действовать как единое целое и максимально упростить общение с клиентом на каждом этапе. Любой сотрудник может получить полную информацию о клиенте, истории взаимоотношений с ним и понять суть его запроса.

В разрабатываемую CRM-систему будет входит несколько модулей:

– модуль регистрации и авторизации;

– модуль просмотра предоставляемых услуг;

– модуль оформления заказа услуги;

– модуль анализа работы с клиентами;

– модуль администрирования.

Таким образом, разрабатываемое программное обеспечение для предприятия малого бизнеса «Barbershop b13» позволит автоматизировать рутинные процессы по работе с клиентами, что повысит удобство и качество сервиса. Клиенты барбершопа смогут дистанционно выбрать интересующую их услугу, и оформить заказ на любое удобное время. Кроме того, система позволит рассылать рекламные предложения для клиентов, однажды воспользовавшихся услугами предприятия, тем самым превращая их в постоянных.

Литература

1. Черемных С.В., Семенов И.О., Ручкин В.С. Моделирование и анализ систем. IDEF-технологии: практикум. М.: Мир, 2020. 192 с.

© Вишгак О.В., Мамедов Р.Р., 2022

УДК 681.5
https://doi.org/10.36906/AP-2022/55

Пащенко О.И.
ORCID: 0000-0002-3248-1107, канд. пед. наук
Нижневартовский государственный университет
г. Нижневартовск, Россия
Рысаева Т.И.
Школа программирования для детей «Кодики»
г. Нижневартовск, Россия

ОСОБЕННОСТИ ФОРМИРОВАНИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ ПРИ ИЗУЧЕНИИ ПРОГРАММИРОВАНИЯ В УСЛОВИЯХ СМЕШАННОГО ОБУЧЕНИЯ

Аннотация. В современном обществе актуальным является вопрос внедрения инновационных форм получения образования, которые соответствуют потребностям цифрового общества. Развитие алгоритмического мышления становится одним из необходимых условий для успешной самореализации личности в цифровом обществе, в условиях динамично изменяющегося и развивающегося мира. Поэтому проблема формирования и развития алгоритмического стиля мышления учащихся очень актуальна в современном образовательном процессе, тем более в условиях смешанного обучения. В статье раскрыты характерные особенности формирования алгоритмического мышления при изучении программирования, представлен анализ платформ, предоставляющих возможности блочного программирования, описаны особенности использования платформы PencilCode. Материал статьи дополнят существующие подходы к формированию алгоритмического мышления, а также расширят знания в вопросах использования блочного программирования для формирования алгоритмического мышления.

Ключевые слова: алгоритмическое мышление; смешанное обучение; программирование; блочное программирование; платформа PencilCode.

Paschenko O.I.
ORCID: 0000-0002-3248-1107, Ph.D.
Nizhnevartovsk State University
Nizhnevartovsk, Russia
Rysaeva T.I.
Programming school for children "Kodiki"
Nizhnevartovsk, Russia

THE PECULIARITIES OF THE FORMATION OF ALGORITHMIC THINKING IN THE STUDY OF PROGRAMMING IN CONDITIONS OF MIXED LEARNING

Abstract. In modern society, the issue of introducing innovative forms of education that correspond to the needs of the digital society is relevant. The development of traditimic thinking becomes one of the necessary conditions for the successful self-realization of the individual in a digital society, in a dynamically changing and developing world. Therefore, the problem of the formation and development of traditimic style of thinking of students is very relevant in the modern educational process, especially in the conditions of mixed learning.

The article discloses the peculiarities of the formation of algorithmic thinking when studying programming, presents an analysis of platforms that provide block programming capabilities, describes the features of using the PencilCode platform. The material of the article will complement the existing approaches to the formation of traditimic thinking, as well as expand knowledge in the use of block programming to form traditimic thinking.

Keywords: traditimic thinking; mixed training; programming; block programming; platform PencilCode.

В настоящее время современному образованию в условиях пандемии требуется подходы, технологии, методы, позволяющие организовать процесс обучения с включением дистанционных технологий [7].

Смешанное обучение, как форма организации образовательного процесса обладает большим ресурсом для решения проблем, с которыми сталкивается современное образование [3].

Смешанное обучение – образовательный подход, совмещающий обучение с учителем с применением образовательной деятельности в онлайн-формате и предполагающий элементы самостоятельного выбора учеником пути, места, времени, темпа обучения, а также интеграцию опыта обучения с учителем и онлайн. Учебный процесс при смешанном обучении представляет собой последовательность фаз электронного и традиционного обучения, которые чередуются во времени [4].

Важной составляющей интеллектуального развития человека является алгоритмическое мышление. Под способностью алгоритмически мыслить в широком смысле понимается умение решать задачи различного происхождения, требующие составления плана действий для достижения желаемого результата.

Любую человеческую деятельность можно представить в виде процесса решения тех или иных задач, а именно учебных, познавательных, трудовых и т. п. В связи с этим, весьма важными являются умения решать эти задачи, разрабатывать стратегию их решения, прогнозировать результаты своей деятельности, анализировать и находить рациональные способы решения задач, то есть те умения, которые характеризуют алгоритмическое мышление [1]. Особую значимость они приобретают на современном этапе развития общества, когда новые информационные технологии внедряются во все сферы человеческой деятельности.

Таким образом, развитие алгоритмического мышления становится одним из необходимых условий для успешной самореализации личности в информационном обществе,

в условиях динамично изменяющегося и развивающегося мира. Поэтому проблема формирования и развития алгоритмического стиля мышления учащихся очень актуальна в современном образовательном процессе [5].

Алгоритмическое мышление – множество мыслительных операций, а также приёмов, направленных на решение заданий, в результате которых формируется алгоритм, являющийся особым продуктом человеческой деятельности. Такого рода метод мышления выделяется от иных формальностью, логичностью, ясностью, возможностью выразить любую абстрактную мысль в виде ряда инструкций, пошаговое исполнение которых, воплощает эту идею в действительность [2]. Именно подобное мышление поспособствует удачному освоению программирования.

Программирование – один из наиболее интересных, но в тоже время сложных разделов в содержании предмета информатика в современной школе [6]. Формирование алгоритмического мышления на уроках информатики при изучении программирования в общеобразовательной школе является актуальной и важной задачей.

Информатика и ИКТ имеет высокую значимость как школьная дисциплина, в рамках которой изучаются средства для формирования общеучебных умений использования новейших технологий в процессе обучения, формирования операционального (алгоритмического) мышления обучающихся. Поэтому актуальными остаются создание и обоснование способов эффективного изучения темы алгоритмизации и программирования.

Одна из трудностей, с которыми встречаются учителя информатики на занятиях – это обучение теме «Основы алгоритмизации», а также «Начало программирования». При изучении этой темы, следует составлять последовательность инструкций, пошагово осуществлять выполнение созданной программы, но из-за того, что у учеников зачастую не сформировано алгоритмическое мышление, у них постоянно возникают трудности в решении подобных задач. Именно поэтому особенно важно использовать современные, проверенные и интересные методы обучения программированию детей и подростков.

Одним из методов, несомненно, эффективных для понимания алгоритмов, развития алгоритмического мышления, является блочное программирование. Благодаря простой организации и минимизации лексической составляющей, отсутствию специфических слов и яркому, похожему на конструктор дизайну многих программ, созданных для развития данного направления, оно становится мощным помощником, неготовых мгновенно принять и впитать огромное количество информации по изучаемому языку и общим правилам программирования.

Для формирования алгоритмического мышления при изучении формирования навыков программирования необходимы наглядные примеры, практика. Уже с появлением первых языков программирования, разработчики и педагоги стремились к тому, чтобы максимально упростить обучение. Создавали простые языки, достаточные для изучения базовых понятий: Basic, Pascal. Потом начали визуализировать текст: Logo и Squeak Etoys. Блочное программирование самая последняя разработка.

CC BY

Процесс программирования стал больше похож на собирание конструктора, где каждая деталь имеет свои имя и назначение. Если конструктор собрать правильно, то получится настоящий рабочий код.

Почему именно блочное программирование?

1. Не надо сразу погружаться в сложности с синтаксисом.

В изучении программирования одной из сложностей является то, что необходимо запоминать огромное количество слов-команд, структуру кода. Для русскоговорящих детей запоминать это вдвойне сложно, так как все названия, ключевые слова пишутся на английском. А в блочном программировании цветные блоки всегда подсказывают, что с помощью них можно выполнить. Обучающимся не нужно думать над отдельными словами, он сосредоточен именно на процессе программирования, создании алгоритмов.

2. Уменьшение когнитивной нагрузки.

Программирование одновременно творческий и логический процесс. С чем новичкам справится нелегко. Применение блоков минимизирует когнитивную нагрузку до восприятия считанного числа цветных деталей. Обучающиеся больше сосредотачиваются на том, как именно их расставить правильно.

3. Количество ошибок.

Ошибку в коде бывает нелегко найти. Просто понять, как тяжело ребёнку, когда его программа не работает только потому, что он забыл поставить точку с запятой в конце строки. Множество мелких ошибок часто отбивает желание у новичков продолжать программировать. Блоки же уменьшают количество подобных ошибок. Достаточно лишь логически думать, как правильно соединить их друг с другом.

Использование специализированных средств реализации алгоритмов для исполнителей, включая среды блочного программирования, позволяет использовать игровые методы обучения. Таким образом, обучающиеся познают базовую структуру языков программирования, не делая упор на изучение пунктуации и лексикона, а наличие ярких, понятных блоков помогает лучше запомнить материал.

В настоящее время существует множество платформ, предоставляющих возможности блочного программирования. К наиболее известным относятся такие платформы как Scratch, MiencraftEDU, PencilCode, Co-paces,Blockly, MicrosoftMakeCode, MitAppinventorTynker.

Некоторые из вышеперечисленных платформ позволяют использовать их в процессе обучения программированию в онлайн формате. Это становится важным при смешанном обучении, ставшим особо актуальным в период пандемии, когда происходит сочетание традиционных форм аудиторного обучения с элементами электронного обучения, в котором используются специальные информационные технологии. Ниже приведён сравнительный анализ вышеперечисленных платформ, предоставляющих возможности блочного программирования (табл.).

CC BY

Таблица

Анализ платформ, предоставляющих возможности блочного программирования

Платформы / Критерии	Scratch	Miencraft EDU	PencilCode	Co-Spaces	Blockly	Microsoft MakeCode	MitAppinventor	Tynker
Поддерживаемые языки	Более 50, включая русский	71 язык	Английский	Русский, английский	Русский, английский	Английский	Английский	Английский
Вес программы	9,2 МБ	438 МБ	-	-	-	-	-	-
Необходимость скачивания	Нет	Да	Нет	Нет	Нет	Нет	Нет	Нет
Наличие сайта	Да	Нет	Только сайт	Только сайт	Только сайт	Только сайт	Только сайт	Только сайт
Возможность переключиться на язык программирования	Только при скачивании доп.контента	Да, JavaScript	Да, лого	Да, JavaScript	Да. JavaScript, Python, PHP, Dart, XML	JavaScript, Phyton	Нет	Да, JavaScript, HTML, CSS, Phyton.
Системные требования		Windows 10 или выше, 32 или 64-битный процессор. Mac OSX ElCapitan. 2 ГБ ОЗУ. 8.8 ГБ свободного места на жёстком диске.	-	Windows 8 и выше	-	-	Windows: Windows XP, Windows Vista, Windows 7+. GNU / Linux: Браузеры: App Inventor неподдерживает Microsoft Internet Explorer..	-
Совместимость с ОС	Windows 98 (и выше), Mac OS X 10.3 (и выше), UbuntuLinux	Windows, Mac OS	Windows, Mac OS	Windows, Mac OS	Windows, Mac OS	Windows	Windows, Mac OS	Windows, Mac OS

CC BY

Бесплатность	Полностью бесплатная платформа	Доступна очень ограниченная пробная версия	Полностью бесплатная платформа	Можно как и пользоваться бесплатной версией с ограничением, так и взаимодействовать с 30-дневной бесплатной.	Полностью бесплатная платформа	Бесплатно	Полностью бесплатная платформа	Полностью бесплатная платформа
Особенности	База примеров, множество возможностей, расширения под Lego, прочие платформы для робототехники.	Совмещение кода с популярнейшей у детей игрой	Возможность плавного перехода от блочного программирования к текстовому, их комбинирования.	Возможность создания учебного класса, выдавать задания как по отдельности, так и группам. Лучшая особенность при работе с детьми: возможность удалённо подключиться к проекту ребёнка, напрямую вносить изменения, просматривать код.	Игра-квест для лучшего понимания блочного программирования и математики, логики. Также имеется собственный редактор кода.	Возможность создавать программы в редакторе кода от Microsoft, а после – на одну из поддерживаемых платформ.	Позволяет путём блочного программирования создавать игры и программы на телефон.	Можно создавать учебные классы, большое разнообразие шаблонов для разных языков программирования, красочный интерфейс.
Возможность проведения уроков в онлайн режиме	Да	Только, если у всего класса мощные компьютеры	Да	Да	Да	Да	Да	Да

CC BY

Возможность публиковать свои игры, поддерживаемые платформы	Только с друзьями и по ссылке	Только с друзьями, отдельным файлом	Только с друзьями и по ссылке	С друзьями по ссылке, при приобретении и полной версии – возможность поделиться в галерее	Только скачанным файлом	Только скачанным файлом	Можно поделиться в галерее, по QR-коду или же ссылкой	Нет
Наличие галереи с работами других	Да	Нет	Да, но небольшая	Да, очень большая. Также есть возможность копирования проектов себе	Да	Нет	Да	Да
Рекомендуемый возраст	7+	7+	10+	8+	9+	10+	10+	8+

Проведя анализ множества платформ по разнообразным критериям, таким как поддерживаемые языки, вес программы, системные требования, бесплатность, возможность проведения онлайн-занятий, нами определена в качестве максимально подходящей, интересной и отвечающей всем требованиям – платформа PencilCode.

Платформа основывается на языке CoffeScript. Этот язык был специально разработан для обучения детей, а также лег в основу PencilCode. Изначально с помощью языка Лого в программе-прародителе дети управляли большим роботом-черепахой, отсюда и главный герой PencilCode – черепаха. В данной платформе можно совмещать блочное программирование и строковое.

Помимо ранее перечисленных преимуществ, платформа имеет возможность переходить от блочного программирования к написанию скриптов на упрощённом языке программирования, что позволяет мягко перевести внимание обучающихся на привычные, изучаемые в школах языки программирования.

Также данная платформа имеет сайт, а, следовательно, не возникает проблем с проведением онлайн-уроков или использованием платформы в качестве дополнительного инструмента при обучении программирования в школе даже в условиях смешанного обучения. На рисунке 1 и рисунке 2 приведён пример программы и её выполнение.

CC BY

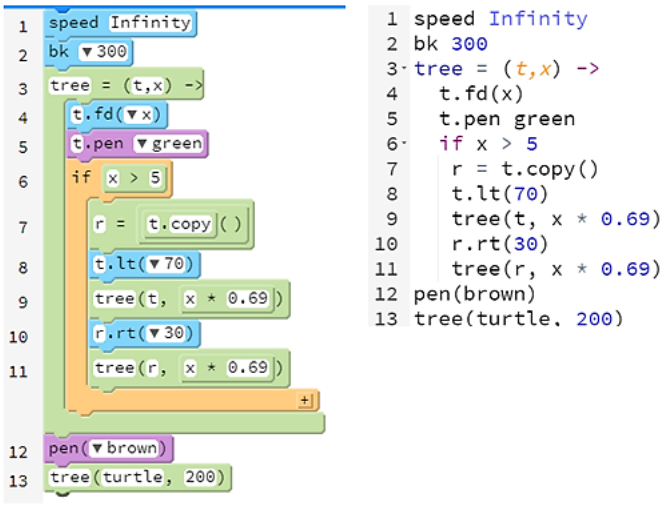


Рис. 1. Пример программы

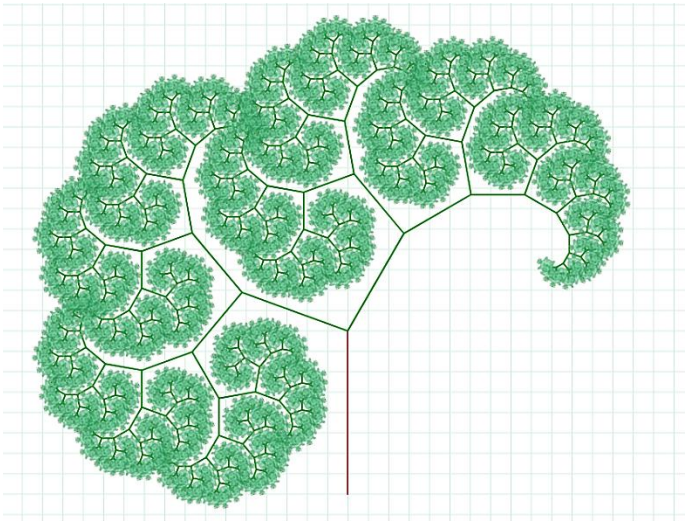


Рис. 2. Выполнение программы

Подводя итог необходимо отметить, что нами рассмотрен один из подходов к формированию алгоритмического мышления при изучении программирования, который можно использовать в условиях смешанного обучения.

Литература

1. Гаврилова И.В. Критерии сформированности уровней алгоритмического мышления // Педагогическая информатика. 2018. № 3. С. 3-8.

2. Кнут Дональд Э. Алгоритмическое мышление и математическое мышление. М.: Изд. иностр. лит-ры, 1999. 110 с.

3. Кравцов В.В., Савельева Н.Н., Черных Т.В. Смешанное обучение как ответ на вызовы современному образованию // Образовательные технологии и общество. 2015. Т. 18. № 4. С. 659-669.

CC BY

4. Нагаева И.А. Смешанное обучение в современном образовательном процессе: необходимость и возможности // Отечественная и зарубежная педагогика. 2016. № 6(33). С. 56–67.

5. Никонова Е.З., Пащенко О.И. Определение способностей к программированию как условие эффективной подготовки к профессиональной деятельности // Интернет-журнал «Мир науки». 2017. Т. 5. № 2. https://clck.ru/dWjmW

6. Пащенко О.И., Скороход Е.Ю. Особенности обучения языкам программирования в профильной школе // Культура, наука, образование: проблемы и перспективы: Мат-лы V Международной научно-практической конференции (г. Нижневартовск, 09-10 февраля 2016 года). Нижневартовск, 2016. Ч. II. С. 173-176.

7. Пащенко О.И. Принципы и характерные особенности дистанционного обучения информатике и ИТ на современном этапе развития системы образования // Современное программирование: Мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 287-291. https://doi.org/10.36906/AP-2020/56

© Пащенко О.И., Рысаева Т.И., 2022

УДК 004.4
https://doi.org/10.36906/AP-2022/56

Смирнов А.Р., Виштак О.В.
Балаковский инженерно-технологический институт –
филиал Национального исследовательского ядерного университета «МИФИ»
г. Балаково, Россия

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ В ПРОЦЕССЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Аннотация. В статье рассматривается разработка информационной системы поддержки учебной деятельности обучающихся в дистанционном формате, включая дополнительное образование. Рассматриваются функциональные возможности разрабатываемой системы, проведён анализ с существующими системами.

Ключевые слова: дистанционное обучение; программное обеспечение; учебная деятельность.

Smirnov A.R., Vishtak O.V.
Balakovo Institute of Engineering and Technology –
branch of the National Research Nuclear University «MEPhI»
Balakovo, Russia

FUNCTIONAL FEATURES OF THE INFORMATION SYSTEM FOR EDUCATIONAL SUPPORT IN THE DISTANCE EDUCATION

Abstract. The article discusses the development of an information system to support the educational activities of students in a distance format, including additional education. The functional capabilities of the system being developed are considered, the analysis with existing systems is carried out.

Keywords: distance learning; software; educational activities.

С появлением персональных компьютеров и развитием сетевых технологий стало возможным дистанционное обучение (ДО). Оно позволяет независимо от времени суток и местоположения изучать новые дисциплины, то есть каждому обучающемуся самому формировать свою индивидуальную технологию обучения [1; 2].

В России и многих других странах дистанционные формы обучения до недавнего времени не применялись в широком масштабе из-за ряда объективных причин - в основном из-за недостаточного развития и широкого распространения технических средств новых информационных и телекоммуникационных технологий. В настоящее время созданы

CC BY

технические предпосылки для широкого использования дистанционного обучения в образовании. Более того, наметилось отставание реализации идей ДО от возможностей, предоставляемых техническими средствами.

Успешное внедрение электронного обучения основывается на правильном выборе программного обеспечения, соответствующего конкретным требованиям, целям и задачам, предъявляемыми к нему организацией [3; 4; 5; 6]. Ниже представлена схема взаимодействия преподавателя и студентов посредством системы дистанционного обучения (рис.).

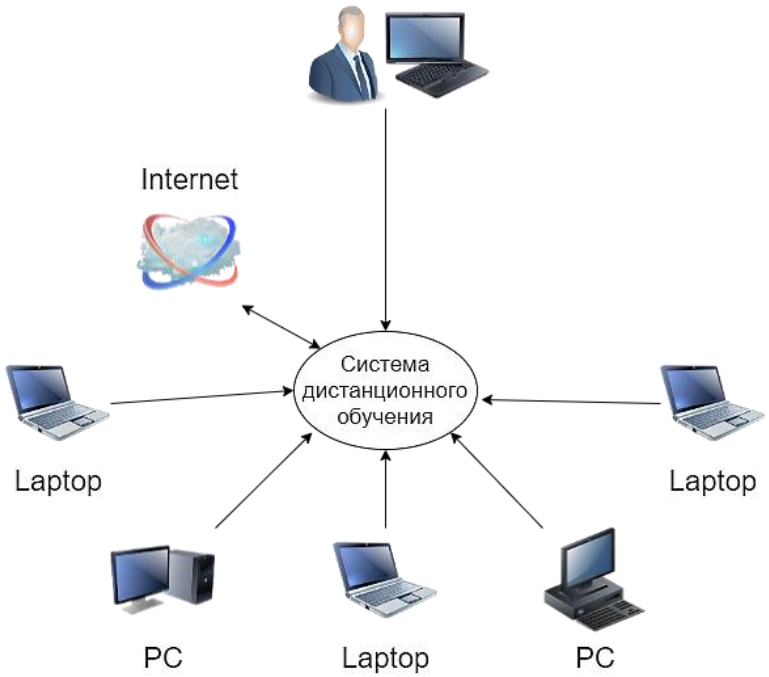


Рис. Схема взаимодействия

Система дистанционного обучения (СДО) выделяем следующие подсистемы: администраторскую и пользовательскую [7; 8]. Администраторская часть включает в себя рабочие места администратора системы и преподавателя. Преподаватель может поддерживать постоянную связь с учащимися посредством аудио- и видеосвязи, и чата. Пользовательская часть используется студентами для поддержки связи с преподавателем, они могут самостоятельно изучать материалы, рекомендованные преподавателем (презентация с экрана, либо документом).

Ниже представлена сравнительный анализ наиболее популярных систем дистанционного обучения (табл.) [3; 4; 6].

Таблица

Сравнение СДО

Lync	Zoom	Skype
Обмен мгновенными сообщениями	Обмен текстовыми и графическими сообщениями;	Мгновенные сообщения
Возможность голосовых и видеозвонков	Организация видео- или аудиосвязи	Голосовые звонки и видеосвязь
Проведение собраний	Коллективный чат	Проведение собраний

CC BY

Предоставление ведущему сведений о присутствующих	Хранение данных	Звонки на телефоны
Совместный доступ к файлам	Совместный доступ к файлам	Совместный доступ к файлам
Возможность участникам демонстрировать свой экран	Рассылку приглашений на онлайн-конференцию	Демонстрация экрана
Интерактивные презентации с возможностью рисовать на доске	Слияние со всеми операционными системами.	Запись звонков и автоматические субтитры
	Запись и расшифровка	Приватные беседы
	Встроенные инструменты коллективной работы	Перевод голоса и текста
	Демонстрация слайдов в качестве виртуального фона	Сетевые статусы
		Предварительный просмотр ссылок

Проанализировав характеристики СДО, был выделен ряд недостатков:

– Skype – очень «тяжёлое» приложение для старых ноутбуков: оно «тянет» много ресурсов, подвисает и тормозит работу других программ, а также тяжело транслирует видео и звук, затрачивая много ресурсов компьютера;

– что касается Zoom, то для его для полноценной работы, необходима подписка, поскольку функционал без подписки ограничен, как минимум, длительность звонка составляет 40 минут, для учебных организаций этого недостаточно;

– в Microsoft Lync стоит ограничение по символам в чате, нет возможности управлять аккаунтами сотрудников, частые проблемы с соединением.

Соответственно для обеспечения информационной поддержки учебной деятельности обучаемых в дистанционном формате было принято решение по разработке приложения (ИСП УД), по минимуму ограничивающее действия пользователей и обеспечивающее комфортные условия их работы. Разрабатываемое приложение будет иметь следующий функционал:

– обмен текстовыми и графическими данными;

–организация видео и -аудиосвязи;

– коллективный чат;

– совместный доступ к файлам;

– интерактивные презентации с возможностью рисовать на доске;

– демонстрация слайдов в качестве виртуального фона.

ИСП УД разрабатывается с учётом минимизирования всех рассмотренных недостатков СДО. Также преимуществом данного приложения является то, что даже самый неопытный пользователь может легко и быстро его освоить. Особенность ИСП УД заключается в том, что в ней реализуются все необходимые для функционирования модули, которые позволяют использовать данную систему в её полном объёме, поскольку не требуется дополнительная подписка для расширения функциональности приложения.

Внедрение ИСП УД позволит обеспечить эффективное взаимодействие между преподавателем и студентами в дистанционном формате, поскольку является универсальной автоматизированной СДО, позволяющей реализовать образовательную программу любого уровня, включая дополнительное образование, позволяющей осуществлять на практике

CC BY

возможность самостоятельной деятельности с различными источниками информации, представленные преподавателем, групповую работу, а также оперативное взаимодействие обучающего с преподавателем.

Литература

1. Виштак О.В. Использование технологии дистанционного обучения в ВУЗе // Педагогика. 2005. № 1. С. 51-56

2. Виштак Н.М., Штырова И.А., Грицюк С.Н. Методы и формы дистанционного обучения в дополнительном профессиональном образовании // Современные наукоемкие технологии. 2016. № 6-1. С. 107-110.

3. Спулок Б. Три основные проблемы Lync // Windows IT Pro/Re. 2014. № 12. https://clck.ru/dWmp9

4. Виштак О.В., Штырова И.А. Сервисы удалённого доступа для организации учебного процесса в магистратуре // Векторы развития магистратуры будущего в условиях глобальных вызовов: Сб. статей участников Международной конференции «Магистратура 2025 – содружество стейкхолдеров. Работая вместе, строим будущее» (г. Нижний Новгород, 15-31 октября 2020 года). Н. Новгород, 2020. С. 30-34.

5. Виштак Н.М., Калистратов М.И. Системы взаимодействия участников учебного процесса. // Актуальные проблемы и пути развития энергетики, техники и технологий: Сб. трудов VI Международной научно-практической конференции (г. Балаково, 23 апреля 2020 года). Балаково, 2020. С. 181-185.

6. Качков М.С., Виштак Н.М. Мессенджеры: сравнительный анализ // Современные технологии и автоматизация в технике, управлении и образовании: Сб. трудов II Международной научно-практической конференции (г. Балаково, 18 декабря 2019 года). Балаково, 2020. С. 183-190.

7. Можаева Г.В., Рыльцева Е.В., Скрипка В.И. Автоматизированная система дистанционного обучения «Электронный университет» // Открытое и дистанционное образование. 2008. № 3(31). С. 68-73.

8. Власенко А.А. Разработка адаптивной системы дистанционного обучения в сфере информационных технологий: Дисс. … канд. техн. наук. Воронеж, 2014. 123 с.

© Смирнов А.Р., Виштак О.В., 2022

УДК 007.52
https://doi.org/10.36906/AP-2022/57

Шакиров Д.Р.
ORCID: 0000-0003-1661-8860
Казанский государственный энергетический университет
г. Казань, Россия

СОВРЕМЕННАЯ РОБОТОТЕХНИКА И ЕЁ ПЕРСПЕКТИВЫ

Аннотация. Робототехника – одно из передовых направлений на сегодняшний день, которое начало стремительно развиваться в 20 веке. Уже сейчас роботы используются в промышленности, медицине, быту, исследованиях. Развитие робототехники происходит с каждым годом, и можно уверенно сказать, что оно продолжится и дальше. Уже сейчас ученые говорят, какими будут роботы в будущем. Например, что роботы благодаря искусственному интеллекту смогут учиться всему сами, роботы будут помогать людям с бытовыми проблемами и широко применяться в промышленности. Но несмотря на то, что роботы выгодны заводам из-за своей точности и дешевизне по сравнению с человеком, есть минус – количество рабочих мест для людей может значительно уменьшиться.

Ключевые слова: робототехника; перспективы; будущее; роботы; прогнозы; промышленность.

Shakirov D.R.
ORCID: 0000-0003-1661-8860
Kazan State Power Engineering University
Kazan, Russia

MODERN ROBOTICS AND ITS PROSPECTS

Abstract. Robotics is one of the leading directions today, which began to develop rapidly in the 20th century. Already, robots are used in industry, medicine, everyday life, and research. The development of robotics is happening every year, and we can confidently say that it will continue in the future. Scientists are already talking about what robots will be like in the future. For example, thanks to artificial intelligence, robots will be able to learn everything themselves, robots will help people with everyday problems and will be widely used in industry. But despite the fact that robots are beneficial to factories because of their accuracy and low cost compared to humans, there is a minus – the number of jobs for humans can significantly decrease.

Keywords: robotics; prospects; future; robots; forecasts; industry.

Робототехника является одним из важнейших направлений технического прогресса. Сейчас робототехника – одна из передовых отраслей. Роботы могут быть запрограммированы

CC BY

на определенные команды человеком, но также они могут развиваться благодаря искусственному интеллекту.

В наше время робототехника развивается стремительными темпами. Например, всего лишь 70 лет назад все машины собирались полностью вручную, но в наши дни это все стало автоматизировано и значительно быстрее: роботы занимаются сваркой, окраской, механообработкой, литьём, штамповкой и сборкой. Это значительно ускоряет производство, уменьшает расход материалов, а значит увеличивает прибыль для многих заводов. Поэтому робототехника является очень перспективным направлением, которое способно совершить радикальные экономические и социальные перемены в нашем мире.

Главная задача робототехники – создание(конструирование) роботов и использования их в различных сферах человеческой деятельности.

Основные преимущества роботов – многофункциональность, скорость выполнения задачи, универсальность.

В 20 веке робототехника начала развиваться стремительными темпами: в 1960-х годах на производстве General Motors впервые начали использоваться механизированные манипуляторы от компании Unimation; в 1970-х годах создали робота Shakey, который умел анализировать свои действия; в 1987 году создали Международную федерацию робототехники, которая была предназначена для содействия в проведениях исследований и разработок в области робототехники по всему миру; в конце 1990-х годов создали первый марсоход, который внёс огромный вклад в изучение далёкого Марса.

На сегодняшний день роботы используются практически во всех сферах человеческой деятельности. В медицине роботы могут самостоятельно проводить сложные операции, диагностировать заболевания и даже ухаживать за пациентами. В быту существуют роботы-пылесосы, которые убирают пыль без помощи человека. В образовании используют в школах и высших научных заведениях для «обучении на проектах». В промышленности производят детали, занимаются сборкой. В транспорте автопилоты, которые могут водить автомобиль без помощи человека. В безопасности пожарные роботы, которые могут без помощи человека найти очаг возгорания и потушить его. В исследованиях роботы исследуют опасные места, например, где есть радиация, низкая температура, в космосе и т. д. В сельском хозяйстве сбор урожая, распыление удобрений, стрижка овец. Также, их используют в военном деле: танки, самолёты и т. д.

Роботов можно разделить на два класса: манипуляторы и мобильные роботы. Манипуляторы – роботы, которые имитируют руку человека и не способны перемещаться самостоятельно. Мобильные роботы – роботы, способные перемещаться самостоятельно. Мобильных роботов разделяют на три класса: воздушные, наземные и морские. Мобильные роботы способны перемещаться благодаря колёсам, гусеницам, полёту, ползать, плавать или ходить.

С каждым годом рынок роботов растёт. Объем промышленных роботов в 2018 году составил 16,5 млрд долларов. В настоящее время лидером по закупка промышленных роботов является Китай. Объем сельскохозяйственных роботов в 2018 году составил 2,4 млрд

CC BY

долларов. В 2018 году в медицине объем составил 2,8 млрд долларов. А уже в 2019 году объем мирового рынка роботов достиг 34 млрд долларов. Причём на автомобильную промышленность приходится самая большая доля рынка – около 32%. Это значит, что инвестирование робототехники очень большое, быстро растёт с каждым годом и интересует многих. Поэтому развитие робототехники идёт каждый год.

Уже сейчас люди создали невероятных и сложных роботов, о которых даже и не могли представить всего 15 лет назад:

Автомобиль Тесла может называться роботом, так как он оснащён автопилотом. Это изобретение является невероятной революцией в наше время. Сложно даже представить, что автомобиль способен ехать без помощи человека. Автомобиль оснащён множеством датчиков и камер, которые отслеживают дорогу и окружающую среду. Благодаря им машина может избежать различных опасных ситуаций и даже аварию.

Autonomous mobile robots(AMR) – робот, который оснащён системой навигации, и благодаря беспилотным роботизированным транспортным системам он может отделить некоторые детали из конвейерной ленты.

Существует известная компания «Boston Dynamics», которая уже более 20 лет создаёт различных роботов. Одни из последних роботов, созданных компанией: Handle – 2-х метровый робот, которые умеет погружать коробки благодаря присоскам; Atlas – робот, который может передвигаться на двух конечностях и переносить предметы на руках; WildCat – робот, похожий на кошку, имеет 4 механические лапы, умеет передвигаться с рекордной для роботов скоростью 45 км/ч.

Робот-манипулятор Hadrian X, созданный компанией Fast Brick Robotics, может построить дом из кирпича за 2 дня, а также он способен выполнять 3D печать. Он намного эффективнее рабочих.

Также очень важным событием было создание марсохода Perseverance, который был доставлен на Марс 18 февраля 2021 года. Он предназначен для исследования кратера Езеро в рамках экспедиции НАСА «Марс-2020».

Сейчас очень популярны роботы-компаньоны. Они предназначены для одиноких людей: миллионов пожилых или умственно отсталых. Благодаря таким роботам, такие люди реже впадают в депрессии. Например, робот «Buddy» способен разговаривать с человеком на двух языках (французский и английский), способен определить пожар в доме, совершать телефонные звонки.

Во многих передовых больницах уже сейчас используют роботов для проведения сложных операций. Ведь у таких роботов не могут «дрожать руки». Также роботов используют для определения диагноза у больного. По статистике, робот может поставить правильный диагноз даже тогда, когда этого не видит опытный специалист.

Для людей с физическими проблемами придумали экзоскелеты и протезы. Например, уже разработали экзоскелет для людей, у которых нарушена опорно-двигательная система. Только благодаря ему они могут передвигаться. Также существуют экзоскелеты для

CC BY

облегчения нагрузки у человека. А благодаря протезам люди без какой-либо части тела могут жить, как обычный человек.

Все эти роботы являются невероятно сложными и приносят огромную пользу для людей.

Именно поэтому сейчас очень востребованы специалисты в области робототехники и программирования. Также аналитики прогнозируют рост востребованности специалистов в следующие несколько десятков лет. Уже сейчас востребованы следующие профессии: проектировщик промышленной робототехники, проектировщик домашних роботов, инженер-композитчик, оператор многофункциональных робототехнических комплексов и множество других.

В России также активно используются роботы во многих сферах деятельности. Наши учёные с каждым годом придумывают новых роботов, которые не отстают от разработок передовых стран мира. Однако в России мало роботов. Число промышленных роботов составляет менее 1% от мирового. В будущем в России роботы буду использоваться во многих предприятиях.

Очень популярными стали соревнования по робототехнике, в которых могут участвовать даже школьники. В них участникам необходимо за определённое время сделать своего робота. Управляя им, они должны выполнить определённую задачу. Например, есть очень известное соревнование First Robotics Competition. Благодаря таким соревнованиям, робототехника популяризируется в мире и среди молодёжи.

За роботами будущее. Ведь благодаря роботам жизнь людей становится значительно проще.

Уже сейчас многие страны заинтересованы в применении роботов в военных целях. Армия США начинает использовать роботов для доставки экипировки и боеприпасов на поле боя. США использует четвероногого робота Spot для разведывательных целей. Также есть роботы, которые охраняют территории. Поэтому можно сказать, что роботы будут все чаще и чаще использоваться в армиях многих стран мира.

С каждым годом появляются новые роботы, которые становятся лучше. Многие компании и различные аналитики сказали, какими будут роботы через пару лет:

По прогнозу экспертно-аналитического отчёта РВК в 2022 году роботы научатся лучше понимать эмоции, поведение человека и реагировать на него.

Международная компания Boston Consulting Group предполагает, что в 2025 году из-за использования роботов в промышленности может произойти уменьшение затрат на оплату труда на 16%.

A Roadmap for U.S. Robotics: From Internet to Robotics предполагают, что в 2028 году появятся первые медицинские микророботы, способные перемещаться в организме пациента самостоятельно.

По мнению Namura Research Institute в 2035 году роботы освоят до 49% специальностей.

Уже в 2021 году в Татарстанском университете Иннополис придумали робота «Гагарин», который способен «считывать» эмоции с человеческого лица и имитировать некоторые из них.

CC BY

В будущем роботы смогут более точно определять эмоции у человека. Например, они смогут замечать мелкие детали, для определения лжи у человека. Учёные говорят, что такие роботы – будущее. Они смогут взаимодействовать с человеком и даже работать учителями, продавцами.

Также учёные говорят, что в будущем произойдёт:

Массовое внедрение роботов в производстве; использование роботов в качестве водителя; роботы смогут охранять жилье, готовить еду, самостоятельно стирать вещи; произойдёт актуализация роботов-питомцев; появятся роботы-курьеры, которые смогут ходить в магазин за продуктами вместо человека; роботы смогут заниматься полной уборкой дома.

Таким образом, исходя из всего вышесказанного, можно сделать вывод, что робототехника – действительно быстро развивающееся и перспективное направление. Со временем, роботы появятся во многих сферах жизни человека и станут обыденностью. Будущее развитие робототехники имеет огромные плюсы, однако есть некоторые недостатки. Например, благодаря роботам, на производстве увеличивается скорость и качество выполненной работы, жизнь людей становится значительно проще. Однако из-за увеличения количества роботов на производстве может уменьшится количество рабочих мест для людей.

© Шакиров Д.Р., 2022

УДК 004.622
https://doi.org/10.36906/AP-2022/58

Яблоков Е.Н.
ORCID: 0000-0001-9044-2053, канд. техн. наук
Санкт-Петербургский государственный университет
аэрокосмического приборостроения
г. Санкт-Петербург, Россия

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ БЛОКА КОДИРОВАНИЯ СООБЩЕНИЙ ДЛЯ ГРАЖДАНСКОЙ АВИАЦИИ

Аннотация. Данная статья посвящена информации об обмене данными между гражданскими самолётами и наземными станциями. Представлены базовые данные о работе наземных служб, представлены алгоритмы кодирования сообщений, а также приведена реализация алгоритмов на языке высокого уровня для транспондеров гражданской авиации.

Ключевые слова: циклические коды; кодирование; режим-s; гражданская авиация.

Yablokov E.N.
ORCID: 0000-0001-9044-2053, Ph.D.
Saint Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia

SOFTWARE IMPLEMENTATION OF THE CODING BLOCK FOR CIVIL AVIATION

Abstract. This article is devoted to information about the exchange of data between civil aircraft and ground stations. Basic data on the operation of ground services are presented. The message coding algorithms are considered, and the implementated algorithms using the high-level language is presented.

Keywords: cycle codes; coding; mode-s; civil aviation.

В настоящее время наблюдение за воздушным движением осуществляется с помощью первичного обзорного радара и вторичного обзорного радара [1, с. 2]. Первичный радар работает с отражёнными волнами, радар передаёт электромагнитную волну и в общем случае измеряет направление и время между переданным и отражённым сигналом от самолёта. Первичные радары определяют наличие воздушного объекта, наклонное расстояние до летящего объекта, а так же его азимут, но данный вид радара не может произвести идентификацию объекта. Альтернативой подобному радару является технология ADS-B. ADS-B или Automatic Dependent Surveillance-Broadcast [2, с. 2] – технология, при которой самолёт периодически осуществляет передачу нескольких своих параметров, а именно – своё

CC BY

текущее положение местоположение, путевую скорость самолёта, его высоту, а так же вертикальную скорость и личный идентификатор. Кроме перечисленных параметров могут быть переданы и другие параметры. В режиме S для идентификации воздушного судна используется SSR. SSR – представляет собой радарную систему наблюдения, требующее, чтобы воздушное судно было оборудовано транспондером (передатчиком и ответчиком). Режимы транспондера разделяются на военные и гражданские, функционирующие по схожим, но все таки различающимся протоколам. В отличие от SSR, PR не предъявляет таких условий, так как он работает независимо на принципах эхолокации [3, с. 40]. SSR может опрашивать воздушное судно в нескольких режимах. Режим A для получения идентификационного кода самолёта, представляющего из себя 4-значный восьмеричный идентификационный код. Часто режим A сопрягается с режимом C, который предоставляет информацию о барометрической высоте воздушного судна. Данные режимы позволяют передать короткую информацию. Третий и последний режим получил код S и позволяет осуществлять общение между радаром и самолётом в расширенном режиме. В 2009 году режим S был расширен дополнительным набором команд для общения и представляет собой удобный механизм для обмена информация с самолётом. При задействованном режиме S SSR выборочно опрашивает воздушное судно по его 24-битному адресу, который выделяется каждому воздушному судну после регистрации (адрес ИКАО). Все передаваемые сообщения от радара во всех трёх режимах передаются на частоте 1030 МГц (рис. 1), как описывается в документах ICAO.

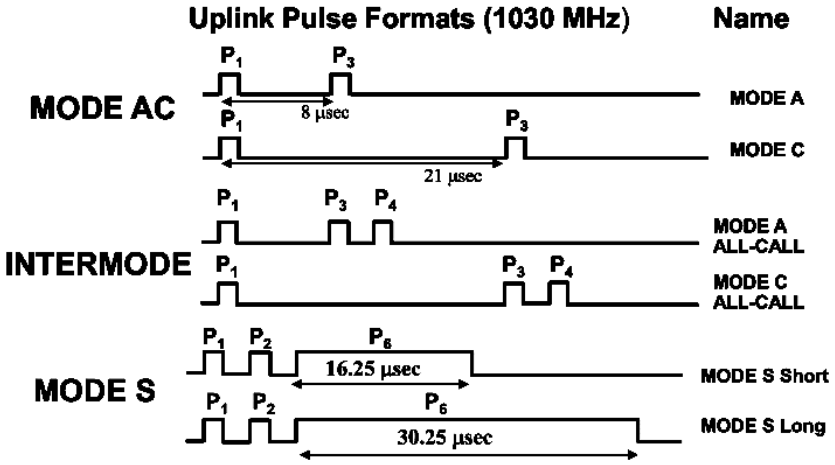


Рис. 1. Варианты и длительность передачи сообщения в различных режимах SSR

Ответные сообщения посылаются транспондером воздушного судна на частотах 1090. Необходимо отметить то факт, что не все самолёты оснащены транспондером, оборудованном режимом S, в этом случае на запросы режима транспондер с поддержкой только режима A/C не будет ответа (рис. 2). Формат и размер сообщений остаются одинаковыми. Дополнительно можно отметить, что транспондеры не являются обязательным оборудованием на всех воздушных судах.

Uplink	Mode-AC transponder	Mode-S transponder
MODE A	MODE A	MODE A
MODE C	MODE C	MODE C
MODE A ALL-CALL	MODE A	No Reply
MODE C ALL-CALL	MODE C	
MODE S Short	No Reply	MODE S if addressed or All-Call
MODE S Long		

Рис. 2. Ответные сообщения при различных запросах

В отличии от режимов A/C, в сообщения режима S добавляется дополнительные 24 бита, позволяющая проверить достоверность принимаемого сообщения. Каждое сообщение режима S содержит 56 или 112 бит, из которых последние 24 бита составляют контрольную сумму.

Для побитного кодирования контрольной суммы используется простая схема умножения, представленная на рисунке (2). Данная схема позволяет довольно быстро побитно перемножить поступающее сообщения для получения контрольной суммы.

Несмотря на общий принцип кодирования сообщений, который используется при передаче и приёме сообщений от транспондера к радару и обратно – надо отметить, что при различном направлении передачи используется немного другой принцип кодирования контрольной суммы. Если передача идёт от радара к транспондеру, то непосредственное сообщение состоит из самого сообщения (32 бита) и адреса (24) кодируются с помощью полинома и результат этого кодирования записывается в биты контрольной суммы. В случае с обратной передачей – кодируется только само сообщение, а в 24 бита контрольной суммы записывается побитное «или» с нужным адресом, как указано в формуле (2).

$$P(x) = M(x)/G(x)$$
$$P'(x) = P(x) \oplus \{ICAO, AA, IC\} \quad (2)$$
$$T(x) = M(x) + P'(x)$$

С точки зрения аппаратной реализации – для кодирования сообщений в радаре и декодирования на воздушном судне используются одинаковые схемы кодирования, а вот для обратной передачи декодер радара и кодер транспондера делаются по разным схемам. В общем случае, при передаче информации от транспондера к радару сначала происходит кодирование сообщения, после чего происходит операция побитного сложения с адресом.

При декодировании сообщение последовательно умножается на обозначенный формулой (1) полином и после преобразования осуществляется проверка контрольной суммы.

Изначально радар не обладает информацией о номерах самолётов, поэтому он периодически узнает о номерах самолётов в воздухе с помощью специальных широковещательных запросов All-call. Так как изначально самолёты не знают идентификатор радара, который обращается к ним, то для вычисления контрольной суммы радар использует

CC BY

специальный адрес FFFFFF, который обозначает широковещательный запрос. В этом запросе радар посылает свой идентификатор в одном из полей сообщения. В ответ на широковещательный запрос принявшие самолёты предоставляют ответ, в котором ответно передаётся 24 битный адрес объекта по ИКАО. В качестве адреса для контрольной суммы каждое воздушное судно использует адрес радара, который был им получен при широковещательном запросе. Сообщения, которые отвечают за запрос от радара и ответ от воздушного судна соответственно обозначаются как запросы UF11 и DF11 (рис. 3).

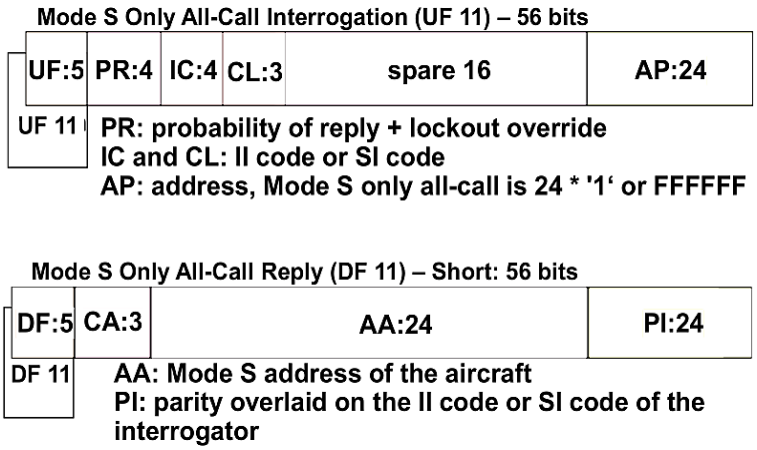


Рис. 3. Формат широковещательного запроса и ответа от воздушного судна

При большом скоплении самолётов радар может отправить специальное сообщение, которое сообщает самолёту с определенным адресом не отвечать на широковещательные запросы. Таким образом, послав несколько широковещательных запросов – радар будет обладать полной информацией о самолётах, которые находятся в зоне видимости радара.

В соответствии с документами ИКАО – при отправке сообщений от радара к воздушному судну – ошибка не корректируется, таким образом – любое ошибочное сообщение отбрасывается самолётом. В то время как радар может быть дополнен устройством, которое будет не только детектировать ошибки, но и корректировать принятые сообщения.

По сравнению с протоколом передачи в режиме кодирование передаваемых сообщений в режиме A/C, режим S обеспечивает для исправления ошибок. Можно разработать усовершенствованный алгоритм, использующий преимущества циклических кода, используемых при кодировании контрольной суммы. Основные усилия при исправлении ошибок направлены на корректировку так называемых «пакетных ошибок». Данные ошибки возникают из-за наложения ответа в режиме A/C в результате наложения ответа режима A/C при приёме ответа режима S. Из-за этого ошибка часто происходит не водном, а в нескольких битах подряд, что может привести как к искажению передаваемого сообщения, так и к искажению контрольной суммы.

Как говорилось выше – при передаче сообщения от радара к самолёту с помощью полинома кодируется и само сообщение (32 или 88 бит) и адрес самолёта (24 бита). В результате, если при проверке сообщения оказывается, что полученный адрес не совпадает с

CC BY

адресом самолёта – такое сообщение просто отбрасывается и корректировка не осуществляется.

Если же ошибка произошла при приёме ответа от самолёта радаром – то корректировка ошибки возможна. Самолёт, как и радар, использует свой адрес для вычисления контрольной суммы. Поэтому, если при приёме ответа в результате проверки адрес самолёта не совпадёт с ожидаемым – это будет являться признаком ошибки. В этом случае – можно задействовать одну из схем корректировки (или их совокупность).

Для ускорения определения места ошибки – радар сообщает информацию о так называемых битах доверительности. Биты доверительности делятся на две группы – биты с сильным доверием и со слабым. Биты со слабым доверием являются битам, где вероятность ошибки наиболее высока. Эта информация передаётся в блок корректировки ошибок.

Такой метод может оказаться полезным в районах с большой загруженностью РЧ-каналов, в частности в тех случаях, когда приёмник режима S используется в сочетании со всенаправленной антенной или антенной с широкой диаграммой направленности.

Учитывая эту дополнительную информацию от радара возможны 3 схемы корректировки ошибки – классическую, модифицированную и с полным перебором.

Теория корректирования ошибок в данном случае базируется на цикличности кодов, которые использовались для кодирования. Полином для кодирования был выбран таким образом, чтобы гарантировано исправлять ошибки, которые идут подряд, из-за наложения сигналов. Документы ИКАО пишут о минимум 12 идущих подряд битах, которые могут быть исправлены. Практические эксперименты показывают возможность исправления до 24 подряд идущих бит, в которых была ошибка. Исправление так называемых пакетных ошибок возможно благодаря использованию циклических кодов [3, с. 135] при кодировании проверочной информации.

Схема декодирования базируется на принципе, который позволяет учитывать так называемый «синдром ошибки». Синдром ошибки – это получившийся остаток при попытке проверить сообщение, который получился в результате получения контрольной суммы. Путем умножения на получившийся остаток с поправкой на полином возможно скорректировать произошедшую ошибку. Эту схему коррекции использует как, классический метод коррекции, так и модифицированный.

С помощью данного синдрома путём обратного умножения на изначальный полином производится пересчёт контрольной суммы. Классический метод не учитывает информацию о битах с низким доверием и последовательно проверяет каждые 24 бита на наличие ошибок. Таким образом – коррекция ошибки произойдёт при самом худшем случае и самом длинном сообщении за 88 циклов умножения на первоначальный полином, умножения на получившуюся последовательность и проверки результата.

Метод полного перебора позволяет декодировать ошибку, проверив все возможные варианты ошибки во всех битах со слабой степенью доверительности. Данный метод использует только информацию от радара о местах ошибок и осуществляет проверку контрольной суммы после каждого варианта перебора. Самым главным недостатком такого

CC BY

метода является очень большое время, которое необходимо затратить на перебор всех вариантов возможных ошибок. Плюсом данного метода является возможность коррекции любых малых ошибок по длительности в разных местах сообщения, которые не попадают в 24 битовое окно коррекции классического метода.

Самым быстрым и относительно простым методом для реализации является модифицированный метод. Он учитывает как информацию, которую предоставляет радар о битах с низким уровнем доверия, так и «синдром ошибки», который получается в результате декодирования.

Алгоритм такой коррекции состоит из следующих пунктов:

- получаем синдром ошибки и биты доверительности от радара, и начинаем провыерку с конца сообщения;
- проверяем синдром ошибки и биты с низким уровнем доверия от радара;
- если ВСЕ единицы текущего синдрома ошибки совпадают с битами, которые имеют низкий уровень доверительности, то ошибка локализована, переходим к коррекции ошибки;
- если 3-й пункт не выполнен, то модифицируем синдром ошибки домножая его на базовый полином;
- сдвигаем на один разряд влево и снова переходим к пункту 2.

Коррекция ошибки предполагает простую операцию сложения по модулю 2 с текущим синдромом ошибки, начиная с текущей позиции сдвига. Рассмотрим простой пример корректирования ошибки на сообщении в 56 бит (рис. 4).

```
int n_poly = 0xFFFA04;
for (i=0; i<56; i++)
  {
    if (((ndov&0xFFFFFF) | ersyndr) == (ndov&0xFFFFFF))
    break;

    bit = ersyndr&1;
    ersyndr =ersyndr>>1;

    if (bit) ersyndr=ersyndr ^ n_poly;

    ndov=ndov>>1;
    out+=(mes&1)*bt;
    bt=bt<<1;
    mes=mes>>1;
  }
  mes = mes^ersyndr;
  mes = mes<<i;
  mes = mes ^ out;
```

Адрес:
0xdab505
Оригинал:
0x5008f50fe8206
Получено:
0x5008a05fe8206
Информация от радара
0x0000FFF000000
(1 - недостоверные биты)
Синдром ошибки:
0xca9407

№	Биты доверия	Синдром	Сообщение
0	0	ca9407	0
1	800000	9ab007	2
2	c00000	b2a207	6
3	e00000	a6ab07	6
4	f00000	acaf87	6
5	f80000	a9adc7	6
6	fc0000	ab2ce7	6
7	fe0000	aa6c77	6
8	ff0000	aacc3f	6
9	ff8000	aa9c1b	206
10	ffc000	aab409	206
11	ffe000	aaa000	206

Рис. 4. Простой пример реализации алгоритма на языке С и результат выполнения

CC BY

В результате выполнения данного алгоритма ошибка будет декодирована за один проход по полученному сообщению. После нахождения места ошибки и синдрома достаточно сложить синдром ошибки и остатка сообщения и итоговое сообщение будет восстановлено.

Подводя итог можно отметить тот факт, что выше представленный алгоритм коррекции информации в радаре легко реализовывается аппаратно. В ряде случаев, для коррекции единичных бит он может быть дополнен алгоритмом полного перебора, однако, учитывая, что подавляющая группа ошибок составляют так называемые «пакетные» ошибки - это может оказаться нецелесообразным.

Литература

1. Orlando V.A., Drouilhet P.R. ATC-42 Mode S Beacon System: Functional Description (Rev D) // Lincoln Laboratory. 1986.

2. Huy V., Sun J., Ellerbroek J., Hoekstra J. ADS-B and Mode S Data for Aviation Meteorology and Aircraft Performance Modelling. Thesis for: MSc. Aerospace Engineering. https://doi.org/10.13140/RG.2.2.29307.36646/1

3. Белоус А.И., Мерданов М.К., Шведов С.В. СВЧ-электроника в системах радиолокации и связи. Книга 2. М.: Техносфера, 2018. 703 с.

© Яблоков Е.Н., 2022

ВОПРОСЫ МЕТОДОЛОГИИ ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ И ВЗАИМОДЕЙСТВИЯ С ОБЩЕОБРАЗОВАТЕЛЬНЫМИ ШКОЛАМИ

УДК 004.023
https://doi.org/10.36906/AP-2022/59

Абрамян Г.В.
ORCID: 0000-0002-5440-1411, д-р пед. наук
Российский государственный педагогический университет им. А.И. Герцена
Государственный университет морского и речного флота им. адмирала С.О. Макарова
г. Санкт-Петербург, Россия

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ И РИСКИ ПЕРСОНАЛИЗАЦИИ ОБУЧЕНИЯ В УСЛОВИЯХ ЦИФРОВИЗАЦИИ ОБРАЗОВАНИЯ

Аннотация. В статье рассматриваются современные особенности, тенденции, риски и тренды российского образования в условиях цифровизации. Обосновывается необходимость учёта рисков использования зарубежных систем электронного обучения на основе LMS и LXP-систем.

Ключевые слова: цифровизация образования; персонализация обучения; персонализация учебного опыта; особенности; тенденции; риски; тренды; LXP-системы.

Abrahamyan G.V.
ORCID: 0000-0002-5440-1411, Ph.D.
The Herzen State Pedagogical University of Russia
Admiral Makarov State University of Maritime and Inland Shipping
Saint-Petersburg, Russia

MODERN TRENDS AND RISKS OF PERSONALIZATION OF EDUCATION IN THE CONDITIONS OF DIGITALIZATION OF EDUCATION

Abstract. The article examines modern features, tendencies, risks and trends of Russian education in the context of digitalization. The necessity of taking into account the risks of using foreign e-learning systems based on LMS and LXP-systems is substantiated.

Keywords: digitalization of education; personalization of education; features; trends; risks; trends; LXP systems.

CC BY

В условиях цифровизации экономики и периодически вводимых пандемических ограничений в мире российская система высшего образования функционирует и развивается в особом режиме, при котором цели, задачи и направления развития высшего, среднего профессионального (СПО) и общего (ОО) образования изменяются в соответствии с требованиями времени. Так например, в системе СПО в ближайшее время планируется реализовать мероприятия по: 1) ускоренному двухгодичному выпуску обучаемых, в том числе укорачивая уже существующие образовательные программы, проводя акселерацию и ускоренную подготовку по рабочим профессиям, развивая краткосрочные курсы, 2) практико-ориентированное содержание образования, 3) разработке и развитию новых форм и моделей обучения, в том числе на основе дуальных форм обучения при которых: 3.1) работодатели принимают более активное участие в создании учебных планов и программ, 3.2) теоретическая подготовка обучаемых осуществляется в учреждении образования, а практическая подготовка ведётся на предприятии и конкретном рабочем месте. В статье выделяются основные тенденции, условия и риски процессов цифровизации образования.

Тенденция 1. В новых условиях цифровое образование, персонализация данных и экспериментальная дистанционная/удалённая работа в вузах, системе СПО и школах все чаще становится нормой. Все ступени образования теперь можно получать в любом месте и в любое время (в вузах, СПО и школах) при наличии соответствующего ИКТ-оборудования в том числе и в личном пользовании у обучаемых, однако сам учебный процесс все шире и глубже сопровождается сбором образовательных и индивидуальных данных о характере прохождения учебных маршрутов. Получаемые персональные данные об обучаемых, как правило, собираются образовательным учреждением и поступают на экспертизу в региональные и федеральные отделы образования, профильные министерства и ведомства для принятия организационных и управленческих решений об эффективности учебных процессов.

В условиях пандемии, безусловно, цифровые удалённые формы обучения как и телемедицина это более безопасные, качественные и дешёвые формы оказания услуг обучения/лечения населению. Однако реальное или онлайн цифровое социальное общение/обучение и диалог для обучаемых из регионов уже в ближайшее время может стать доступным только для людей обладающими достаточными средствами и возможностями. Например, в условиях пандемии за дополнительные «домашние» занятия с репетитором, обучаемым (студентам и школьникам) или их родителям было необходимо заплатить в среднем полторы-две тысячи рублей в час, занятия с использованием видеосервисов стоило около 500 рублей в час, а подписка на использование различных образовательных платформ в среднем стоила 200-300 рублей в месяц, в сети Интернет представлены многочисленные и бесплатные или более бюджетные открытые электронные ресурсы. Электронные и цифровые формы образования в условиях пандемии позволили с одной стороны обеспечить государству образовательный процесс, преподавателям реализовать профессиональную деятельность, а обучаемым и их родителям получить удалённые образовательные услуги «на дому». В новых условиях изменились и функции ППС и учителей – они все больше становились тьюторами создающими, использующими и контролирующими актуальный аудио/видео контент,

CC BY

параллельно занимающимися всеми организационными процессами и профессиональными задачами. В этой связи электронное/цифровое образование при сохранении качества предоставляемых услуг можно рассматривать как среду поддерживающую распараллеливание задач персонального мониторинга и управления образовательными процессами.

Тенденция 2. В современных условиях целью обучения, например в системах СПО и вузах становится уже не столько обучение, развитие и воспитание, сколько создание и поддержание системы персональной подготовки молодёжи к эффективной продаже на свободном рынке услуг своих уже сформированных и формируемых компетенций, навыков, знаний, умений и навыков. В этой связи талантливый молодой человек – выпускник СПО или вуза возможно уже не будет восприниматься просто как социально самоценная личность, но он будет ценен своими персональными практико-ориентированными компетенциями, знаниями, умениями и навыками, в том числе и адаптивными навыками SOFT SKILLS, опытом разработки и реализации творческих проектов, навыками виртуализации и анимации деятельности, технологиями и алгоритмами цифровой деятельности, виртуально-анимированными моделями услуг на глобальном рынке, дизайнерскими идеями и пр. Для этого необходимо, чтобы ещё и на стадии обучения в вузе и системе СПО обучаемый начал формировать свою персонализированную модель компетенций/навыков/знаний, параллельно выстраивая свою карьеру в системе обучения и обществе, становясь при этом частью «живого» товарного рынка компетенций, генерируя при этом в цифровую среду свой цифровой образ. В зависимости от уровня развития, глубины, связей и отношений модели компетенций и талантливости обучаемый будет стоить тем дороже, чем более развитой будет его компетентностная модель и соответственно, тем он как будущий специалист выше будет ценится и дороже продаваться на рынке глобальных услуг.

Модели персональных компетенций/навыков/знаний можно будет оперативно подстраивать под актуальные, например региональные или федеральные задачи собирая и агрегируя данные, например из HR-платформ, текстов вакансий, содержимого задач. Для этого рекомендуется использовать таксономию образовательных результатов, полученную в результате машинного анализа большого количества персональных данных и текстов. При этом обучаемые реализуя свои индивидуальные учебные проекты параллельно могут отражают ход своей деятельности в tasktracker. Для этого например можно воспользоваться сервисом для мониторинга командной работы Trello, который позволяет планировать и публиковать текущие учебные задачи, систематизировать их и следить за исполнением.

Тенденция 3. Траектории персонального развития и обучения позволяют дифференцировать обучаемых и тем самым формировать группы обучаемых ориентированные на решение творческих задач (ГООТЗ) с одной стороны, а с другой стороны формировать более прикладные/профильные группы обучаемых - будущих функциональных исполнителей «заданных» производственных процессов и алгоритмов (ФИЗППА).

В условиях цифровизации электронного обучения в вузах, СПО и школах параллельно идут процессы дифференциации системы образования на академическую/фундаментальную и прикладную/профильную составляющие образования. При этом система

CC BY

академического/фундаментального образования постепенно становится все более закрытой, элитарной и доступной либо для избранного контингента обучаемых, либо для обеспеченных семей, которые в состоянии компенсировать расходы на обучение своих детей. Именно для реализации в первую очередь ФИЗППА образования по нашему мнению и необходима система цифрового/электронного обучения, для управления которой необходимо в первую очередь создавать искусственный интеллект. Пандемия и связанный с ней резкий отток трудовых мигрантов, в том числе из РФ показал, что за реализацией интеллектуального ФИЗППА образования на основе персонализированного цифрового/электронного мониторинга и обучения, безусловно, большое будущее.

Тенденция 4. При всех положительных составляющих персонализированного цифрового/электронного обучения и возможности интеллектуализации и автоматизации ФИЗППА образования, нельзя забывать о проблемах, которые могут возникать при подготовке молодёжи. В частности поиск и обработка готовой информации в цифровой/электронной среде и связанный с этим набор текстов подменяет важные для развития мозга обучаемых процессы письменного изложения мыслей, которые совместно тактильными операциями формируют навыки/умения и как результат развития происходит рост нейронных связей и систем между левым и правым полушариями мозга. При использовании обучаемыми манипуляторов (клавиатуры/мыши) или традиционных ручки/карандаша при вводе/обработке информации используются совершенно различные тактильные зоны, и в первом случае используются лишь «подушечки» пальцев. Поэтому необходимо проводить глубокие психофизиологические исследования, как и каким образом, использование и соответственно развитие «новых» тактильных зон вместо традиционных «зон» будет сказываться на развитии когнитивных и умственных способностей, сознания и интеллекта. Необходимо также учитывать, что в процессе персонализированного электронного обучения происходит значительное увеличение временных периодов ввода/набора/редактирования текстов, вставки объектов, «протаскивания» мыши и др. Для изучения этих явлений предлагается разработать перечень типовых электронных операций «электронного» обучаемого, которые он осуществляет с использованием: 1) клавиатуры, 2) тактильных ручных манипуляторов (мышь, джойстик, игровые пульты и пр.), 3) виртуальных нательных тактильных манипуляторов и датчиков (виртуальный шлем, очки и пр.), 4) подкожных электронных чипов.

Тенденция 5. В процессе персонализированного онлайн обучения в РФ в настоящее время используются в основном зарубежные программы и информационные системы: 1) глобальные информационные сервисы видеоконференций - Zoom, Skype, MicrosoftTeams, CiscoWebexMeetings, Slack, GoogleHangoutsMeet, FacebookMessenger, GoToMeeting, JoinMe, CyberLink U Meeting, BlueJeans, Lifesize, FreeConference, Starleaf, Trueconf, 2) системы электронного обучения (СЭО) - Canvas от Instructure, AdobeCaptivatePrime от Adobe, iSpringSuite от Ричмедиа, Академия-Медиа от Издательский центр «Академия», АнтиТренинги от Интернет Университет, АнтиТренинги от Интернет Университет, iSpringPage от Ричмедиа, iSpringPage от Ричмедиа, eFront от Epignosis, myQuiz от WaveAccess,

CC BY

Система тестирования INDIGO от IndigoSoftwareTechnologies, Electude от ElectudeInternational, Collaborator от Davintoo, ATutor, eLearningServer 4G от Гиперметод, Teachbase от Интернет Школа, Emdesell от ИП Соколов М.А., GetCourse от Система Геткурс, iSpringMarket от Ричмедиа, , Mirapolis LMS от Mirapolis, StartExam от Милдсофт, CourseLab от WebSoft, OnlineTestPad, Eliademy от CloudBerryTec, Testand от Тестанд, Testand от Тестанд, AdobeCaptivate от Adobe, Uchi.pro от Учи.Про, OpenedX от edX, MyLMS от WebSoft, MyLMS от WebSoft, Google Класс от Google, Google Класс от Google, ArticulateStoryline 360 от ArticulateGlobal, Ё-Стади от Седьмое Небо, iSpringLearn от Ричмедиа, Edmodo от Edmodo, Эквио от Эквио, Gurucan от Гурукэн, Moodle от Moodle, Teachable от Teachable, ZenClass от ДзенКласс, LearnWorlds от LearnWorlds, Moco от Technomatix, TalentLMS от Epignosis, Unicraft от Юникрафт, eTutorium LMS от Етуториум, TotaraLearn от TotaraLearningSolutions, Eduardo от Лекториум, EduTerra.PRO от Территория Образования, 1С:Электронное обучение, Корпоративный университет от 1С, Thinkific от Thinkific, JustClick от ДжастКлик что приводит к: 1) рискам возможного полного или частичного ограничения доступа к образовательным системам, 2) возможности отключения сервисов для определенных категорий пользователей, в том числе государственных учреждений, вузов, школ, а также например находящихся в так называемых «санкционных» списках, 3) утечке персональных данных граждан РФ в глобальную сеть [6, с. 8; 11, с. 297; 12, с. 4].

Персонализация учебного опыта при обучении в настоящее время осуществляется с использованием алгоритмов машинного обучения на основе как правило зарубежных инструментов навигации Learning Experience Platform (LXP-систем). Например зарубежная система: 1) Valamis позволяет персонально учитывает и «знает» лично каждого обучаемого (цели, образовательные траектории и пр.) и непрерывно подсказывает действия необходимые в данный момент, 2) Edcast: 2.1) формирует так называемые скиллсеты – наборы навыков, которые обучаемые должны изучить, чтобы освоить определенную предметную область или раздел, например, стать, графическим дизайнером определенного уровня. Под конкретные навыки и подбираются рекомендации, 2.2) содержит систему экспертной оценки навыков, которая является важнейшим инструментом и источником информации для системы формирования рекомендаций по обучению, 3) открывания и мониторинга контента Degreed, которая позволяет следить, что происходит у других обучающихся/преподавателей и составлять персональную траекторию развития. [7, с. 7]

Использование зарубежных LXP-систем предполагает, что образовательные организации должны быть глубже интегрированы в сетевые совместные программы. Таким образом, каждый сетевой вуз/колледж/школа полностью формирует только часть образовательных активностей по наиболее глубоко проработанной профильной тематике. А далее сетевые вузы/колледжи/школы обмениваются контентом реализуя совместную логику партнёрства, но не конкуренции. Это позволяет каждому вузу/образовательной организации больше внимания уделить качеству контента. Однако данная деятельность также приводит к утечке персональных данных в глобальную сеть.

CC BY

Однако если для реализации сетевой программы используются зарубежные сервисы видеоконференций, СЭО и LXP-систем, то в условиях импортозамещения программного обеспечения параллельно российское образование становится зависимым от работы данных сервисов и как результат система отечественного образования уже перестанет быть суверенной и независимой [3, с. 363; 8, с. 92]. Причем сбор образовательных персональных данных для управления контентом [2, с. 261] и интерфейсами формирования профессиональных компетенций осуществляется в настоящее время во многом без научного обоснования [1, с. 255; 5, с. 251], с неясными целями и возможными негативными последствиями как для обучаемых и их родителей, так и для самой системы образования (учителя, ППС, управленческие кадры) [9, с. 32; 10, с. 6; 13, с. 218; 18, с. 126].

Тенденция 6. Использование современных систем персонализированного электронного обучения предполагает наличие дома у обучаемых высокоскоростных каналов связи и дорогостоящей компьютерной техники и внешних устройств, что приводит с одной стороны к зависимости доступности образовательных услуг и качества образования от: 1) материальной обеспеченности обучаемых, их родителей и учебного заведения в котором установлена СЭО школы, 2) наличия ИТ-компетенций у обучаемых и их родителей, преподавателей, руководящего и обслуживающего персонала учреждения образования [15, с. 18; 16, с. 125; 17, с. 100]. В реальных условиях РФ это может привести к принудительному разделению обучающихся, родителей, ППС, обслуживающего ИТ-персонала учреждения образования по территориальным, материальным, финансовым и как результат этническим, национальным и религиозным признакам. Цифровые средства (аппаратные, программные, организационные и пр.) в соответствии с законодательством РФ должны быть наполнены лицензионным программным обеспечением, поэтому приобретая продукты компаний Microsoft, Apple и др. сотрудники образовательных учреждений и родители обучаемых должны помнить о том, что Microsoft, Apple, Intel, AMD являются американскими компаниями и в доктрине безопасности США РФ по настоящее время является одним из главных врагов США. Исходя из этого руководству образования необходимо принимать адекватные организационные решения с точки зрения информационных рисков защиты цифровых и электронных персональных данных обучаемых в условиях реализации реального суверенитета РФ [4, с. 414; 14, с. 69].

Литература

1. Абрамян Г.В. HIGH-HUME методология и алгоритмы реализации HIGH-TECH управления контурами естественнофизиологических, электронноцифровых и гибридных интерфейсов формирования профессиональных компетенций выпускников вузов // Современное программирование: Мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 255-260. https://doi.org/10.36906/AP-2020/50

2. Абрамян Г.В. Вербальные, визуальные и паралингвистические невербальные компоненты HIGH-HUME/HIGH-TECH цифрового управления подготовкой выпускников

вузов с учетом региональных фонетических, фонологических, морфологических, лексикологических и синтаксических конструкций и форм организации IT-коммуникаций // Современное программирование: Мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 261-266. https://doi.org/10.36906/AP-2020/51

3. Абрамян Г.В. Модели развития учебно-образовательных, научно- исследовательских и промышленно-производственных ИТ-технологий, сервисов и процессов в России и странах ближнего зарубежья в условиях импортозамещения программного обеспечения // Информатика: проблемы, методология, технологии. Информатика в образовании: мат-лы XVIII Международной школы-конференции (г. Воронеж, 08-09 февраля 2018 года). 2018. С. 363-368.

4. Абрамян Г.В. Риски и потенциальные угрозы компьютерных систем и технологий электронного обучения на платформе WINDOWS научно-образовательной среды Российской Федерации // Региональная информатика и информационная безопасность: Сб. трудов. Выпуск 1. СПб., 2015. С. 414-416.

5. Абрамян Г.В. Формирование профессиональных компетенций выпускников вузов в цифровой HIGH-HUME образовательной среде на основе HIGH-TECH суггестивнолингвистического анализа и управления профессиональной деятельностью, коммуникациями и контентом учебных каналов // Современное программирование: мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 251-254. https://doi.org/10.36906/AP-2020/49

6. Гладилина И.П., Ермакова И.Г. Цифровая трансформация образования: зарубежный и отечественный опыт // Современное педагогическое образование. 2021. №3. С. 8-12.

7. Жедигеров Д.Ж., Абрамян Г.В. Система учебной навигации студентов и преподавателей университета на основе спутниковой системы GLONASS и облачных сервисов NAVITEL // Информационно-телекоммуникационные системы и технологии: мат-лы Всероссийской научно-практической конференции (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 213.

8. Кицела К.И., Соколов М.А., Тенишев Р.Д., Абрамян Г.В. Технология отбора мобильных программ-клиентов для обмена личными сообщениями в информационно-образовательной среде вуза // Информационно-телекоммуникационные системы и технологии: мат-лы Всероссийской научно-практической конференции (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 92.

9. Копыльцов А.А., Копыльцов А.В. Обработка слабо формализованной информации, поступающей от технических систем // Вестник Нижневартовского государственного университета. 2013. №1. С. 32-36.

10. Копыльцов А.В. Оценка качества программного обеспечения. СПб.: ЛГОУ, 2000. 80 с.

CC BY

11. Кусаинова А.С. Инструменты и методы оценки функционирования IT-инфраструктуры высшего учебного заведения // Вестник университета Туран. 2016. №4 (72). С. 297-300.

12. Мынбаева А.К. Обзор новейших теорий образования: Педагогика 2.0, Образование 3.0 и Хьютагогика (Эвтагогика) // Вестник Казахского национального университета. Серия Педагогические науки. 2019. Т. 61. №4. С. 4-16.

13. Савельев С.Д., Абрамян Г.В. Информационная модель распределенной автоматизированной информационной системы кадрового учёта в финансовом университете при Правительстве РФ на основе удалённого вызова процедур «Тонкий клиент» // Информационно-телекоммуникационные системы и технологии: мат-лы Всероссийской научно-практической конференции (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 218.

14. Сафиуллин А.Р., Александров А.А. «Умный город»: ключевые характеристики в условиях четвертой промышленной революции // Цифровая экономика и Индустрия 4.0: новые вызовы: Труды научно-практической конференции с международным участием (г. Санкт-Петербург, 02-04 апреля 2018 года). СПб., 2018. С. 69-80.

15. Ситдиков А.А., Буснюк И.Ю., Тупий Е.О., Абрамян Г.В. Информационная модель оптимизации инфокоммуникаций в вузе на основе интерактивной системы взаимодействия студентов и преподавателей // Информационно-телекоммуникационные системы и технологии: (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 18.

16. Шагай М.А., Абрамян Г.В. Современные тенденции и особенности управления качеством обучения в системе среднего образования Ленинградской области в информационной среде // Информационно-телекоммуникационные системы и технологии: (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 125.

17. Шумкова И.Г., Абрамян Г.В. Особенности, значение и принципы формирования информационно-образовательного пространства вуза на основе интернет-ресурса «СПБ ГИПИСР» // Информационно-телекоммуникационные системы и технологии: (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 100.

18. Янкевичус А.А., Абрамян Г.В. Система оптимизации штатов на основе дифференциации сотрудников технических служб сервиса и консультирования пользователей персональных компьютеров в педагогическом университете // Информационно-телекоммуникационные системы и технологии: (г. Кемерово, 16-17 октября 2015 года). Кемерово, 2015. С. 126.

© Абрамян Г.В., 2022

УДК 378.14.015.62
https://doi.org/10.36906/AP-2022/60

Алексеева В.Н., Комарова О.В., Орлова Л.А.
Приднестровский государственный университет им. Т.Г. Шевченко
г. Тирасполь, Приднестровская Молдавская Республика

ФОРМИРОВАНИЕ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ СТУДЕНТОВ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ СПО НА ОСНОВЕ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ И ПРОИЗВОДСТВЕННЫХ ПРАКТИК

Аннотация. В условиях современного динамически развивающегося общества ощущается потребность в инновационных методах и технологиях обучения. Статья посвящена обзору образовательных технологий, применяемых при формировании профессиональных компетенций студентов системы среднего профессионального образования. Рассмотрены технология формализованных образовательных ситуаций, технология решения профессионально-ориентированных задач (задачный подход) и кейс-технология. В статье рассмотрен вопрос о значении производственных практик, как инструмента обеспечения занятости молодёжи в современных условиях.

Ключевые слова: студент; образовательное учреждение; организация практики; профессиональные компетенции; обучение; профессиональная деятельность; технология; формализованные образовательные ситуации; задания; кейс-технология.

Alekseeva V.N., Komarova O.V., Orlova L.A.
Pridnestrovian State University
Tiraspol, Pridnestrovian Moldavian Republic

FORMATION OF PROFESSIONAL COMPETENCIES OF STUDENTS OF EDUCATIONAL ORGANIZATIONS OF VOCATIONAL EDUCATION ON THE BASIS OF THE ORGANIZATION OF INDEPENDENT WORK AND PRODUCTION PRACTICES

Abstract. In the conditions of a modern dynamically developing society, there is a need for innovative teaching methods and technologies. The article is devoted to the review of educational technologies used in the formation of professional competencies of students of secondary vocational education. The technology of formalized educational situations, the technology of solving professionally-oriented tasks (task approach) and case technology are considered. The article considers the question of the importance of industrial practices as a tool for ensuring youth employment in modern conditions.

CC BY

Keywords: student; educational institution; organization of practice; professional competencies; training; professional activity; technology; formalized educational situations; tasks; case technology.

Главная цель профессионального образования любого уровня заключается в подготовке компетентного специалиста, мобильного, активного, инициативного, готового к эффективному самостоятельному решению профессиональных задач в любых условиях. Компетентный специалист характеризуется наличием сформированных общекультурных и профессиональных компетенций. Несмотря на достаточно широкое исследование способов, форм и методов организации самостоятельной работы студентов, единого решения задач формирования профессиональных компетенций нет. Преподавательская деятельность позволила выделить ряд объективно существующих противоречий:

– между необходимостью формирования профессиональных компетенций у студентов и недостаточной профессиональной направленностью дисциплин, в том числе дисциплин математического и естественнонаучного цикла;

– между необходимостью динамичности, вариативности формирования профессиональных компетенций, как условия эффективного профессионального труда в дальнейшем и статичностью предметной подготовки в аудиторном формате образовательного процесса;

– между увеличением доли учебного времени в образовательном процессе колледжа на самостоятельную работу студентов и неготовностью студентов рационально использовать это время для формирования профессиональных компетенций;

– между необходимостью самостоятельного личностного и профессионального развития студентов в период профессиональной подготовки в учебном заведении и недостаточным уровнем мотивации у студентов и умений организации самостоятельной деятельности, и прежде всего, учебной;

– между необходимостью ориентации учебных дисциплин на формирование профессиональных компетенций и недостаточной обеспеченностью (предметной, технологической, методической) процесса самостоятельной работы студентов в рамках дисциплин [1].

Новые условия в сфере обучения оказывают непосредственное воздействие на цели преподавания и подготовки в области СПО.

Педагогические технологии, используемые в учебном процессе для формирования профессиональных компетенций, отличаются большим разнообразием, проистекающим из вариативного сочетания традиционных и инновационных способов, методов, приёмов и средств обучения. Так, например, при обучении учебным дисциплинам применяются модульные, проблемные, исследовательские, информационные и компьютерные технологии. Широко используются при проведении групповых занятий активные методы обучения (деловые и ролевые игры, анализ конкретных ситуаций и др.).

CC BY

При использовании различных педагогических технологий нельзя забывать о психологической компоненте любого процесса общения, о взаимопонимании и «взаимопроникновении». Общение – непрерывная деятельность, любая передача информации предполагает взаимодействие и потребность убедить других в тех или иных идеях, мнениях, решениях. Общение необходимо для взаимопонимания. Это важно учитывать, чтобы добиваться практических результатов. Современная профессиональная деятельность человека, практически, вся построена на взаимодействии для обмена информацией, т.е. на коммуникативной компетентности (беседы, переговоры, дискуссии, речи, совещания, презентации или виртуальное общение по Интернету и т. д.). Для того чтобы приобрести необходимые навыки общения, специалисты (учителя, врачи, менеджеры, юристы, банкиры, торговые представители, специалисты по связям с общественностью, диспетчеры, налоговые служащие, секретари и т.д.) посещают различные тренинги.

Современная система профессионального образования главное внимание уделяет не столько становлению личности студента и его внутреннего мира, сколько развитию его интеллекта и эрудированности, освоению и переработке знаковой информации из окружающего внешнего мира, приобретению профессиональных компетенций.

Имитационные игры, по свидетельству разных специалистов, позволяют организовывать интенсивное межличностное взаимодействие и осваивать не только коммуникативную компетентность, но и нравственные нормы, принципы и ценности, развивать духовные потребности, обучать сотрудничеству, партнёрским взаимоотношениям, пониманию друг друга и себя самого, корректировке своего потенциала для успеха в личных и профессиональных отношениях [5].

Рассмотрим технологию формализованных образовательных ситуаций при формировании профессиональных компетенций студентов технического колледжа и её преимущества. Спроектированные формализованные образовательные ситуации должны отражать профессиональную деятельность и готовить студента применять приобретённые знания при решении профессиональных задач. При этом у студентов будет формироваться проектное мышление, аналитические способности, стремление к самообразованию, что в дальнейшем обеспечит успешность личностного и профессионального роста. Наличие такой практической составляющей в содержании образовательного процесса основывается на освоении профессиональной деятельности и, в дальнейшем, на приобретении профессиональных компетенций. Цель разработанных заданий должна заключаться в развитии способности студента самостоятельно приобретать новую информацию и анализировать её. В этой ситуации происходит смена образовательных приоритетов. При этом важно не столько приобретение уже готового знания, сколько инициатива и собственные усилия и умения студента в данной сфере деятельности [3].

Следующая технология - решение профессионально-ориентированных задач (задачный подход), также направлена на развитие у студентов навыков самостоятельной работы. Представление учебного материала с целью освоения профессиональных знаний, умений, навыков должно осуществляться поэтапно, посредством применения задач различного уровня

CC BY

сложности. В процессе изучения дисциплины студент может видеть логику перехода от одного уровня к другому, овладевать способами действий, присущими разным уровням профессиональной деятельности. Сущность данного проектирования содержания формализованных заданий состоит в том, что студенту предлагается чётко структурированная информация, в которой указан объем знаний и уровень его усвоения. С помощью грамотно разработанных формализованных заданий профессиональной направленности можно дать оценку уровню и качеству усвоения материала, а также уровню сформированных в ходе изучения дисциплины дескрипторов, т.е. уровню сформированности части профессиональных компетенций.

Следует отметить перспективность кейс-технологии среди методов обучения в технических учебных заведениях СПО. Суть кейс-технологии, или кейс-метода (англ. Case study), состоит в том, что усвоение знаний и формирование умений есть результат активной самостоятельной деятельности студентов по разрешению противоречий, в результате чего и происходит творческое овладение профессиональными знаниями, навыками, умениями и развитие мыслительных способностей. Метод case-study, или метод конкретных ситуаций (от английского «case» – случай, ситуация), – метод активного проблемно-ситуационного анализа, основанный на обучении путем решения конкретных задач-ситуаций (решение кейсов). Кейс – описание конкретной реальной ситуации, которая предназначена для обучения студентов анализу разных видов информации, её обобщению, навыкам формулирования проблемы и выработки возможных вариантов её решения в соответствии с установленными критериями. Совместный разбор жизненных ситуаций – универсальный способ [4].

В последние десятилетия метод кейсов стал применяться и в инженерном образовании. Кейсовая технология – обучение действием. Овладение профессиональными компетенциями становится результатом активной самостоятельной деятельности студента по разрешению противоречий в предложенных ситуациях. Целью кейс-технологии является научить студентов, как индивидуально, так и в составе группы следующему:

– развитию навыкам анализа и критического мышления;

– соединению теории и практики;

– представление примеров принимаемых решений;

– демонстрации различных позиций и точек зрения;

– формированию навыков оценки альтернативных вариантов в условиях неопределённости.

Дидактическими задачами кейс-технологии является:

– принятие верных решений в условиях неопределённости;

– овладение навыкам исследования ситуаций;

– разработка плана действий, ориентированных на намеченный результат;

– разработка алгоритма принятия решения;

– применение полученных теоретических знаний при решении практических задач.

Помимо вышеперечисленного при применении анализа ситуаций достигаются и дополнительные эффекты. Поскольку кейс-метод ориентирован на самостоятельную

CC BY

индивидуальную и групповую деятельность студентов, при его решении у них развиваются коммуникативные умения. При решении общей проблемы полезным оказывается технологическое сотрудничество, которое позволяют всем студентам полностью осмыслить и усвоить учебный материал, дополнительную информацию, а главное, – научиться работать совместно и самостоятельно, развить так называемые «мягкие навыки». Подводя итог, необходимо отметить, что наибольшего эффекта можно достичь при разумном сочетании традиционных и инновационных технологий обучения, когда они взаимосвязаны и дополняют друг друга.

Как известно, фундамент будущего эффективного трудоустройства закладывается, в том числе, в рамках обучения в виде производственных практик студентов, которые способствуют освоению и усвоению материала на практике, что в свою очередь позволит получить необходимый опыт для дальнейшего трудоустройства.

Если на базе образовательных учреждений, будет достаточное количество мест, в которых возможно проходить практику, то, соответственно, будет повышаться рейтинг образовательного учреждения. Чем выше рейтинг образовательного учреждения, тем ценнее выпускники, и тем легче им в дальнейшем будет трудоустроиться [2]. Если все выпускники будут иметь опыт работы по специальности, то общество станет более конкурентоспособным в целом.

Организация практик на всех этапах образовательного процесса должна быть направлена на обеспечение непрерывности и последовательности овладения студентами широкого спектра видов профессиональной деятельности в соответствии с требованиями к уровню подготовки выпускника, в том числе сформулированных непосредственными профильными организациями.

Литература

1. Георге И.В. Формирование профессиональных компетенций студентов образовательных организаций высшего образования на основе организации самостоятельной работы. Тюмень: ТИУ, 2016. 143 с.

2. Ермакова Ж.А., Никулина Ю.Н. Качество образования с позиции потребителей образовательных услуг вуза // Креативная экономика. 2017. Т. 11. № 7. С. 725-734. https://doi:org/10.18334/ce.11.7.38162

3. Синкина Е.А. Технология формирования профессиональных компетенций через проектирование формализованных заданий // Вестник Пермского национального исследовательского политехнического университета. Машиностроение, материаловедение. 2014. Т. 16. № 1. С. 101-105.

4. Голованова И.И., Асафова Е.В., Телегина Н.В. Практики интерактивного обучения. Казань: Казан. ун-т, 2014. 288 с.

5. Панфилова А.П. Психология общения. М.: Академия, 2013. 368 с.

© Алексеева В.Н., Комарова О.В., Орлова Л.А., 2022

УДК 371.321.3
https://doi.org/10.36906/AP-2022/61

Басов В.А.
ORCID: 0000-0002-1986-148X, канд. физ.-мат. наук
Московский государственный технический университет гражданской авиации
г. Москва, Россия

О СОДЕРЖАТЕЛЬНОМ И ТЕХНОЛОГИЧЕСКОМ АСПЕКТЕ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ ВЫСШАЯ МАТЕМАТИКА В УСЛОВИЯХ ЦИФРОВОЙ ТРАНСФОРМАЦИИ

Аннотация. Рассматривается актуальный подход к содержанию дисциплины Высшая математика с акцентом на информатизацию образования. Междисциплинарный комплексный характер и реализация принципа научности достигаются путём включения в содержание дисциплины практических приложений из базисных алгоритмов Data Science. Проанализированы предпосылки для изменения структуры представления учебного материала и реализации дидактических возможностей средств ИКТ. Предложен вариант инструментального обеспечения проведения лабораторных занятий. Приведены подходы к компьютерной визуализации учебной информации об изучаемых объектах.

Ключевые слова: высшая математика; цифровая трансформация; jupyter notebook; data science.

Basov V.A.
ORCID: 0000-0002-1986-148X, Ph.D.
Moscow State Technical University of Civil Aviation
Moscow, Russia

ABOUT THE CONTENT AND TECHNOLOGICAL ASPECT OF TEACHING THE HIGHER MATHEMATICS DISCIPLINE IN THE CONTEXT OF DIGITAL TRANSFORMATION

Abstract. The current approach to the content of the Higher Mathematics discipline with an emphasis on informatization of education is considered in the article. The interdisciplinary complex nature and the implementation of the scientific principle are achieved by including practical applications from the basic algorithms of Data Science in the content of the discipline. The prerequisites for changing the structure of the presentation of educational material and the implementation of didactic capabilities of ICT tools are analyzed. A variant of instrumental support for conducting laboratory studies is proposed. Approaches to computer visualization of educational information about the objects under study are presented.

Keywords: higher mathematics; digital transformation; jupiter notebook; data science.

CC BY

Сегодня можно с уверенностью констатировать, что цифровая трансформация стала трендом. Активно внедряемые цифровые технологии, трансформируют множества горизонтальных и вертикальных бизнес-процессов, оптимизируют операционные процедуры, изменяют модели и форматы взаимодействия в различных отраслях экономики. В этих условиях существенным образом возрастают требования к развитию компетенций выпускников, повышению их культуры работы с данными и различными цифровыми решениями, что делает актуальной проблему повышения требований к содержательному аспекту преподавания целого ряда дисциплин.

Совершенно понятно, что это касается в первую очередь, дисциплин так называемого IT блока: информатика, программирование, информационные технологии в профессиональной деятельности. Вместе с тем, если рассмотреть базис алгоритмов Data Science, то оказывается, что его составляют «классические» математические дисциплины: Линейная алгебра, Математический анализ, Теория вероятностей и математическая статистика. При этом в большинстве университетских курсов Высшей математики эти факты освещены слабо или не фигурируют вовсе. Практические приложения рассматриваемой теории на инженерных специальностях ВУЗов, как правило, ограничиваются физическими примерами. Например, механические приложения определенного интеграла. Такой консервативный подход, ограничивающий знакомство обучающихся с ролью высшей математики в современных информационных технологиях, существенным образом снижает возможности воспитания у обучающихся прикладной математической культуры, формирования интуиции и эрудиции в вопросах приложений Высшей математики.

Конечно, в решении задач актуализации образовательных программ можно идти путем включения в образовательную программу отдельных специальных дисциплин. Так по широко растиражированной СМИ информации в МГУ с 2021 года для всех студентов введены курсы по искусственному интеллекту. Это положительный шаг, но все же более рациональным видится более раннее рассмотрение некоторых основ в рамках содержания математических дисциплин, в частности дисциплины Высшая математика.

Проведённый содержательный анализ алгоритмов Data Science позволяет предложить следующий примерный перечень приложений, рассмотрение которых можно дополнительно включать в рабочую программу дисциплины Высшая математика:

Линейная алгебра:

– обработка изображений;

– векторное представление слов;

– предварительная обработка и преобразование цифровых наборов данных.

Математический анализ:

– метод градиентного спуска;

– метод Ньютона.

Теория вероятностей и математическая статистика:

– логистическая регрессия;

– анализ временных рядов.

CC BY

В рамках рассмотрения технологического аспекта преподавания дисциплины Высшая математика трудно обойти процесс информатизации образования. «Информатизация образования рассматривается как целенаправленно организованный процесс обеспечения сферы образования методологией, технологией и практикой создания и оптимального использования научно-педагогических, учебно-методических и программно-технологических разработок, ориентированных на реализацию возможностей информационных и коммуникационных технологий, применяемых в комфортных и здоровье сберегающих условиях» [2, с. 9].

Наиболее широкими дидактическим возможностям здесь обладают лабораторные занятия, проводимые в форме компьютерных практикумов, реализация которых, создаёт предпосылки для интенсификации процессов изучения разделов Высшей математики.

Традиционно применяемым программным обеспечением здесь выступает табличный процессор Microsoft Excel (https://clck.ru/RRKyB), в отдельных случаях Maple (https://clck.ru/bczwH) и MathCad (https://clck.ru/bczwj). Например, [1]. Вместе с тем, этот подход имеет существенные ограничения при попытке организации образовательного процесса с помощью дистанционных образовательных технологий (ДОТ), когда взаимодействие обучающихся и педагогических работников реализуется в основном с применением информационно-телекоммуникационной сети Интернет. Актуальность ведения образовательного процесса с помощью ДОТ обусловлена текущими условиями эпидемиологической ситуации, связанной с распространением коронавирусной инфекции (COVID-19). Основная трудность здесь возникает не только в связи с тем, что обучающиеся осваивают образовательные программы в полном объёме независимо от своего места нахождения, но и профессорско-преподавательский состав так же может работать вне стен оборудованных аудиторий ВУЗа. Здесь возникает проблема не только обеспечения техническими средствами, но и соответствующим лицензионным программным обеспечением, под которое разрабатывалось все действующее учебно-методическое обеспечение дисциплины.

Технологический анализ показывает, что для целей организации компьютерного практикума по Высшей математике, наиболее оптимальным является применение дистрибутива Python Anaconda (https://anaconda.org) и блокнотов Jupyter Notebook (https://jupyter.org) (https://anaconda.org). Собственно, синергетический эффект такого подхода заключается в двух положениях:

1) Python имеет достаточно развитую систему библиотек, и прежде всего математических. В частности, это open-source модуль NumPy (Numeric Python: https://numpy.org). Данная библиотека предоставляет в виде функций общие математические и числовые операции, обеспечивая функционал, который можно сравнить с функционалом MatLab. Здесь предоставляются базовые методы для манипуляции с большими массивами и матрицами. Ещё одна библиотека – SciPy (Scientific Python: https://scipy.org), существенным образом расширяет функционал NumPy, добавляя огромную коллекцию прикладных математических техник: минимизация, преобразование Фурье, регрессия, и другие.

CC BY

2) Jupyter Notebook обеспечивает объединение кода, визуализации, повествовательного текста, математических уравнений и другие мультимедиа в единый документ, который можно использовать в качестве отчёта. Этот документ легко экспортируется в файл, который можно передать, например, для аттестации результатов выполнения лабораторной работы. Важно, что данное программное обеспечение с открытым исходным кодом, бесплатно распространяется на основе модифицированной лицензии BSD. Лицензии этого типа являются одними из самых популярных лицензий для свободного программного обеспечения и используются для многих программ. Кроме того, непосредственная работа осуществляется в браузере, что так же снижает издержки на приобретение, установку и эксплуатацию дополнительного программного обеспечения.

Отдельно стоит остановиться на библиотеках, обеспечивающих визуализацию данных. Графики, диаграммы, тепловые карты и другие приёмы визуализации данных помогают сделать отчёты по лабораторным работам более понятными, убедительными и удобными для восприятия. Строить различные типы графиков и настраивать параметры их отображения удобно с помощью библиотек Matplotlib (https://matplotlib.org) и Seaborn (https://seaborn.pydata.org). Библиотека Plotly может применяться для создания интерактивных графиков и отчётов (https://plotly.com/python). Описания, представленные на сайте, в достаточной мере подробные и снабжены большим количеством примеров. Это даёт возможность для преподавателя творчески подойти к формированию учебных заданий.

Естественно, процесс разработки и внедрения в курс Высшей математики лабораторных занятий сопряжён с множеством учебно-методических проблем, которые ещё только предстоит разрешить в рамках научных исследований в области информатизации образования. При этом, трансформировать содержание дисциплины можно поэтапно, ограничившись элементами Линейной алгебры, а так же Векторной алгебры и Аналитической геометрии. Это с одной стороны позволяет выработать необходимые навыки доведения решения задачи до практически приемлемого результата – числа, графика, точного качественного вывода с применением информационных технологий. С другой, обеспечивает у обучающихся достаточный уровень практических навыков работы, для самостоятельного применения математического аппарата в различных вычислительных средах.

В целом, описанные подходы, обеспечивают реализацию практико-ориентированного обучения. Для студентов, особенно инженерных специальностей первого курса, необходимо максимально создавать условия, чтобы они могли погрузиться в профессиональную сферу, реализовать себя творчески и проявить свою самостоятельность в каждой дисциплине своего учебного плана. Реализация практико-ориентированного подхода при преподавании дисциплины Высшая математика наиболее оптимальным, естественным образом может осуществляться в формировании фундаментальной научной базы знаний у студентов с параллельным получением цифровых компетенций. Такой подход в обучении, в конечном счёте, не только обеспечит эффективность реализуемых образовательных программ ВУЗа, но и будет способствовать формированию грамотных конкурентоспособных выпускников.

CC BY

Литература

1. Мороз О.В., Потапова Н.В. Компьютерный практикум как форма организации практических занятий студентов при изучении курса математики // Общество: социология, психология, педагогика. 2019. № 11. https://doi.org/10.24158/spp.2019.11.26

2. Роберт И.В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты). М.: БИНОМ. Лаборатория знаний, 2014. 398 с.

© Басов В.А., 2022

УДК 004.822
https://doi.org/10.36906/AP-2022/62

Лозикова И.О.
ORCID: 0000-0001-6177-0541
Валеева Д.Р.
ORCID: 0000-0002-2594-162X
Тюменский индустриальный университет
г. Тюмень, Россия

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ ДЛЯ ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ

Аннотация. В статье рассмотрены актуальные вопросы формирования профессиональных компетенций. Представлено решение с использованием онтологического моделирования базы знаний для интеллектуальной системы поддержки принятия решения (СППР) по выбору индивидуальной образовательной траектории. Описаны методы и технологии онтологического моделирования. Представлен процесс построения модели и продемонстрирован вариант применения модели для построения ИОТ. Сделаны выводы о преимуществах онтологического подхода.

Ключевые слова: система поддержки принятия решения; компетенции; индивидуальная образовательная траектория; семантические технологии; онтология.

Lozikova I.O.
ORCID: 0000-0001-6177-0541
Valeeva D.R.
ORCID: 0000-0002-2594-162X
Industrial University of Tyumen
Tyumen, Russia

DECISION SUPPORT SYSTEM FOR THE FORMATION OF PROFESSIONAL COMPETENCIES OF UNIVERSITY GRADUATES IN IT AREAS

Abstract. The article deals with topical issues of the formation of professional competencies. A solution is presented using ontological modeling of a knowledge base for an intelligent decision support system (DSS) for choosing an individual educational trajectory. Methods and technologies of ontological modeling are described. The process of constructing a model is presented and a variant of using the model for constructing an IOT is demonstrated. Conclusions are made about the advantages of the ontological approach.

CC BY

Keywords: decision support system; competence; individual educational trajectory; semantic technologies; ontology.

Увеличение темпов индустриализации общества в России требует подготовку высококвалифицированных специалистов в IT-области. Каждый год выпускается большое количество выпускников, но уровень подготовки выпускников в большей мере не соответствует требованиям рынка труда. Отсюда возникает потребность формирования профессиональных компетенций в данном направлении. Одним из решений проблемы является индивидуализация образовательного процесса, то есть построение индивидуальной образовательной траектории (ИОТ).

Массовый характер образования и его непрерывность, ориентация на личность обучающегося, подчёркивают важность организации образовательного процесса по индивидуальным образовательным траектория. Сложность и комплексный характер проблемы требует разработки такой технологии, которая на протяжении всего учебного процесса поможет выстраивать студенту свою образовательную траекторию, учитывая его индивидуальные характеристики и потребности. Самостоятельный выбор ИОТ и принятие решения обучающимся поможет выполнить система поддержки принятия решения (СППР, Decision Support System, DSS), ядром которой будет база знаний.

Онтологический подход к моделированию строит формальное описание знаний предметной области и логические отношения между ними, применяя семантические технологии. Технологическими стандартами разработки семантических моделей являются формальные системы RDF, RDFS, OWL и SWRL. Онтология знаний состоит из набора терминологических аксиом TBox, набора утверждений об индивидах ABox и правил получения логических выводов на множестве утверждений [1]. Использование специализированного программного обеспечения позволит преобразовать модель в базу знаний, которая будет использована для разработки системы поддержки принятия решения (рис. 1).

CC BY

Рис. 1. Технология преобразования модели в базу знаний

Предметная область онтологии СППР выбора ИОТ – образовательная программа профессионального стандарта по направлению [2]. В данной онтологии представление индивидуальной образовательной траектории (Траектория) – перечень дисциплин (Дисциплина), согласно индивидуальному выбору обучающегося (Студента), для формирования универсальных, общепрофессиональных, профессиональных компетенций (Компетенция) соответственно целевому профессиональному стандарту (ПС) направления/специальности (Специальность). Для тестирования компетентности онтологической модели составим список целевых вопросов разрабатываемой онтологии:

• При выборе в текущем Семестре этой Дисциплины как изменится список результирующих Специальностей индивидуальной Траектории?

• Траектория Специальности какие *содержит* Дисциплины?

• Какие Компетенции *формируют* выбранную Специальность?

• Текущий список Дисциплин индивидуальной Траектории Студента, *стремится* к каким Специальностям?

• Если Студент выбрал эту Дисциплину, то какие индивидуальные Компетенции он *вырабатывает*/формирует?

• Какие Компетенции, в каких Дисциплинах Студенту необходимо получить, чтобы результатом была заданная Специальность?

Основные сущности предметной области и отношения между ними отражает модель TBox (рис. 2).

CC BY

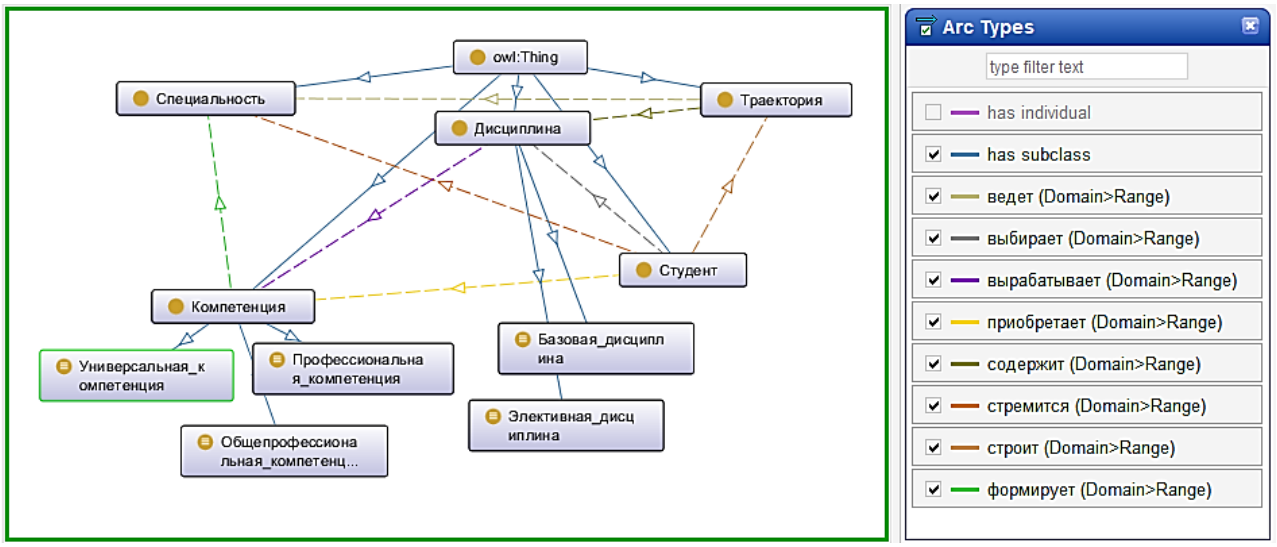


Рис. 2. Модель TBox

В модели ABox представлены индивидуальные экземпляры сущностей (рис. 3).

Правила логического вывода отвечают на целевые вопросы онтологии и разработаны языке SWRL и SPARQL.

Реализация онтологии и оценка её компетентности происходит для ядра направления ИВТ (09.00.00), фрагменты реализации модели представлены на рисунках 4–6.

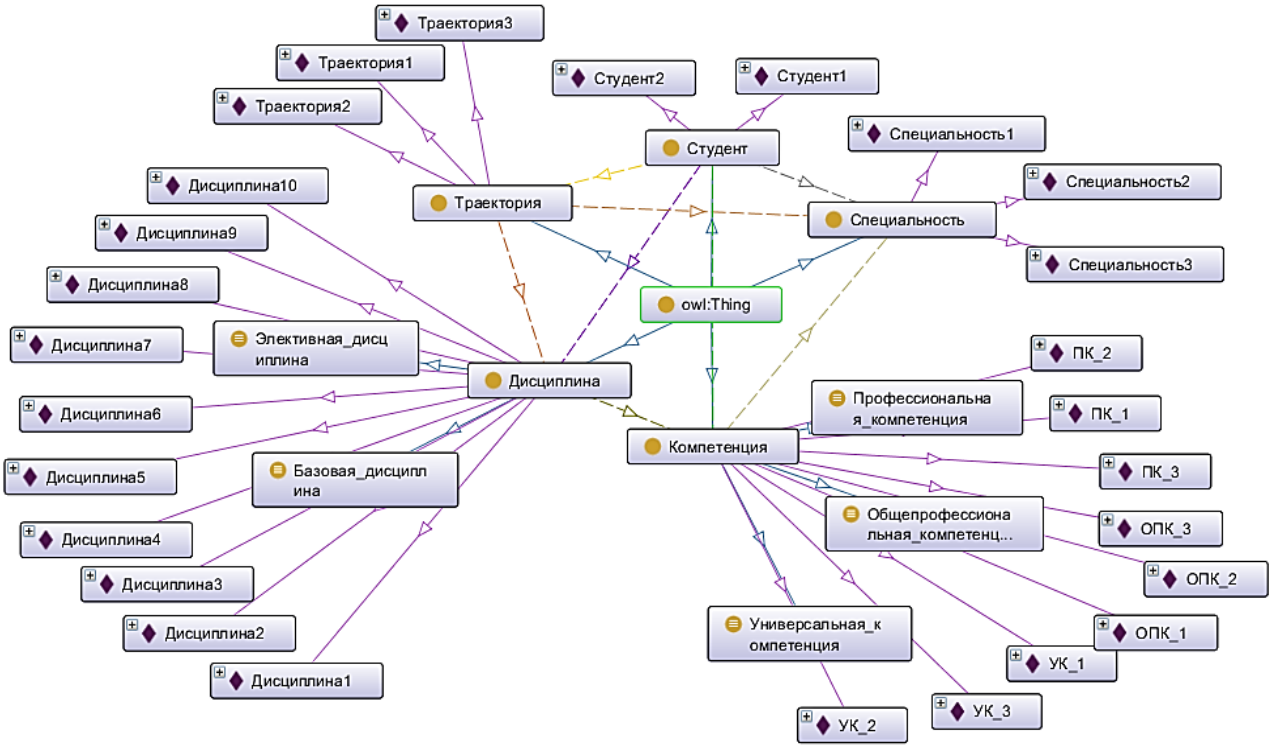


Рис. 3. Модель ABox

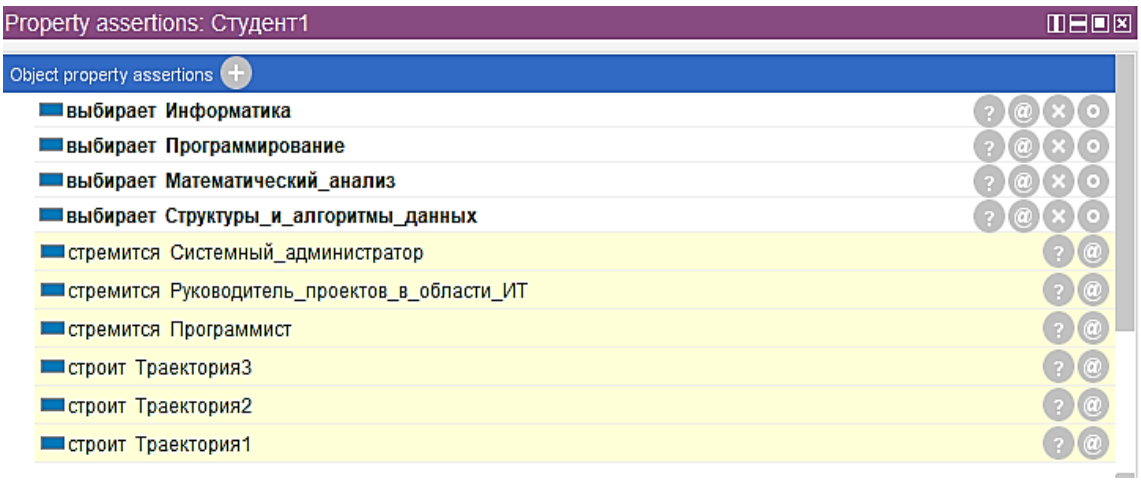


Рис. 4. Фрагмент ядра направления

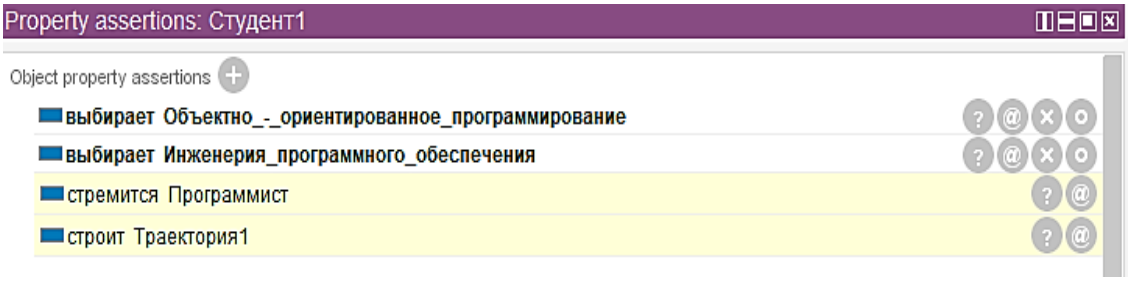


Рис. 5. Результирующая траектория Студента1

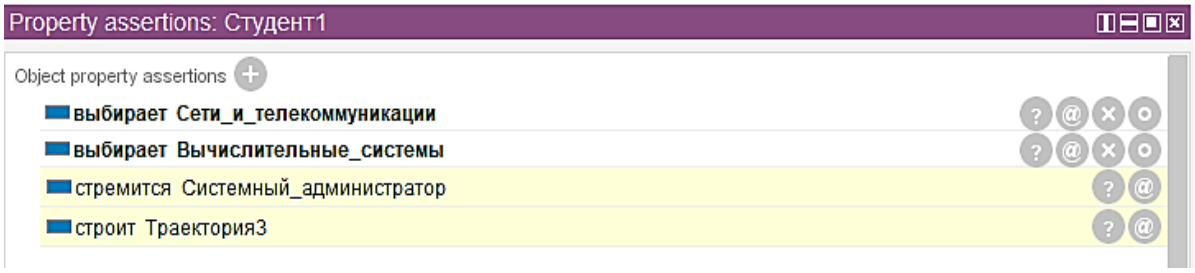


Рис. 6. Обновлённая результирующая траектория Студента1

Общая архитектура построения такой системы предполагает, что СППР – это интегрированное приложение корпоративной информационной системы образовательного учреждения, а программная реализации СППР может быть WEB-приложением, мобильным приложением и др.

Модель демонстрирует следующие преимущества онтологического подхода к разработке:

– использование в модели терминов предметной области и логических рассуждений, понятных пользователям, экспертам и разработчикам;

– добавляя новые факты и правила в онтологию, легко масштабировать и изменять систему, расширяя границы представленных знаний и правил логического вывода.

Сочетание технологий онтологического моделирования с возможностями быстрой обработки огромных объёмов информации программным приложением делает

онтологический подход эффективным методом разработки системы поддержки принятия решений в области индивидуальных образовательных траекторий.

Литература

1. Горшков С.В., Кралин С.С., Муштак О.И., Гумеров С.З., Мирошниченко М.Г., Гребешков А.Ю., Шебалов Р.Ю. Онтологическое моделирование предприятий: методы и технологии. Екатеринбург: Изд-во Урал. ун-та, 2019. 236 с.

2. Сиволапова А.К., Гильмулина Т.П. Инструменты построения индивидуальной образовательной траектории студента // Профессиональное образование в России и за рубежом. 2016. № 3(23). С. 79-81.

© Лозикова И.О., Валеева Д.Р., 2022

УДК 372.8
https://doi.org/10.36906/AP-2022/63

Гибадуллин А.А.
ORCID: 0000-0003-2545-9355
Нижневартовский государственный университет
г. Нижневартовск, Россия

КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИГР В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

Аннотация. Статья посвящена использованию компьютерных игр в качестве объекта разработки. В контексте исследования это подразумевает их реализацию при обучении языкам программирования. Данная методика способствует пониманию и развивает умение применять алгоритмические конструкции на практике. Графическое сопровождение обеспечивает их наглядность. Это позволяет разнообразить образовательный процесс творческими заданиями с возможностью их адаптации под уровень учащихся.

Ключевые слова: интеллектуальные игры; языки программирования; компьютерная игра; игровой процесс; интеллектуализация; геймификация.

Gibadullin A.A.
ORCID: 0000-0003-2545-9355
Nizhnevartovsk State University
Nizhnevartovsk, Russia

COMPUTER IMPLEMENTATION OF INTELLECTUAL GAMES IN LEARNING PROGRAMMING

Abstract. The article is devoted to the use of computer games as an object of development. In the context of research, this implies their implementation in teaching programming languages. This technique develops understanding and the ability to apply algorithmic constructions in practice. Graphic support provides their clarity. It allows you to diversify the educational process with creative tasks with the ability to adapt them to the level of students.

Keywords: mind games; programming languages; computer game; game process; intellectualization; gamification.

Повышение эффективности образования, его соответствие изменениям в жизни и деятельности общества относится к актуальным направлениям развития педагогической науки. Это требует определенных инноваций и внедрения новых технологий. Исследователи отмечают объективную необходимость геймификации образовательной деятельности [9]. Данный термин стал активно использоваться только после 2010 года. Несмотря на то, что

CC BY

дидактические игры как средства обучения известны давно, их внутренний потенциал не исчерпан. А с приходом современных информационных технологий их применение становится практически безграничным, ведь их компьютерные реализации соответствуют тенденциям глобальной информатизации различных сфер деятельности человечества и модернизации мышления новых поколений.

В отличие от игр как таковых геймификация не предполагает развлекательной компоненты. При этом она обладает существенными преимуществами, если сравнивать ее с традиционными образовательными технологиями. Её отличает высокая степень мотивации и заинтересованности у учащихся, их самостоятельность и активная включенность в освоение материала. Мы отмечаем возможности для адаптации под уровень конкретного школьника или студента, что приводит к большей персонификации обучения в соответствии с личностно-ориентированной парадигмой. Однако есть и недостатки. Среди них мы можем отметить необходимость специальной подготовки преподавательского состава.

Дистанционное обучение, потребность в котором вызвана сложившимися в настоящее время обстоятельствами, также не является исключением. Оно предусматривает большие возможности для геймификации образовательного процесса [10]. Здесь стоит отметить, что многие игровые методы имеют аналоги в учебной системе: специфические задания, уровни, механизмы оценивания. Мы рассматриваем именно интеллектуальные игры как объект разработки ввиду их потенциала для развития мышления [3]. На их примере мы демонстрируем работу алгоритмов, программных структур, особенности языков программирования [4]. Уже есть определенный опыт их использования для дистанционного обучения другими авторами [8].

Как правило, изучение языков программирования включает в себя множество разделов. Это линейные структуры, условные операторы, циклы, массивы, функции, знакомство с графикой и программированием движущихся объектов. Стоит отметить, что достаточно большое количество интеллектуальных игр требует их для своей реализации. Помимо всего прочего именно они позволяют наглядно и в движении представить работу всех составляющих.

Следует добавить, что к популярным программным продуктам относятся именно компьютерные игры. Разработка некоторых из них достаточно проста и соответствует уровню школьников и студентов. В настоящее время существует огромный накопленный опыт разработки интеллектуальных игр, применения соответствующих алгоритмов и методов. Причём такие игры могут отличаться по видам и жанрам. Программы подобного рода были одними из первых и появились в середине прошлого века. На данный момент существует множество платформ и движков для их создания.

Одно из важных свойств компьютерных игр – моделирование определенных ситуаций и явлений с целью выработки алгоритмов принятия решений, поиска выхода из трудных ситуаций [7]. Часто это означает симуляцию действительности, моделирование реального пространства-времени в рамках игровых пространства и времени. На самом деле происходит визуализация из цифровой составляющей, программного кода.

CC BY

В сложных имитационных играх требуется обеспечение большей правдоподобности. Это необходимо, чтобы они копировали важные аспекты окружающей нас реальности. При этом нам необходимо реализовать непрерывное пространство и время при помощи функций, то есть аналитически. Как правило, в игровом мире моделируют гравитацию, наличие препятствий. Возникает необходимость визуализации сцены в реальном времени. В общем случае её реализация делится на две компоненты: создание каркаса и наполнение его деталями.

Порядок действий при разработке сцены выглядит следующим образом.

• Создание сцены и фоновых объектов;

• Добавление динамических элементов;

• Программирование реакций на действия пользователя;

• Обработка критических ситуаций и коллизий.

Целесообразно использование объектно-ориентированного подхода. Для того чтобы обеспечить единство интерфейса, нужно создать базовый класс. Это избавит от необходимости менять весь проект для изменения либо добавления новых объектов, а также облегчит их взаимодействие между собой. Все остальные классы будут происходить от базового, наследуя его свойства. Возникает необходимость симуляции взаимодействия динамических объектов. В таком случае геометрическое представление находится в зависимости от физических свойств имитируемых объектов.

Для эффективного управления ресурсами необходимо объединять схожие объекты в группы, чтобы не расходовать каждый раз лишние объёмы памяти. Самый простой способ в реализации – объявление массива. После этого на объекты можно будет ссылаться по их индексам. Группировка элементов позволяет организовать их иерархию, при которой поведение объектов нижнего уровня зависит от характеристик объектов верхнего уровня.

В рамках исследования мы рассматриваем целый класс игровых программ, к которым относятся: крестики-нолики, пятнашки, «2048», шашки, шахматы. Их объединяет наличие двухмерного поля с клетками. Происходит его наполнение разнотипными объектами: числами определенного диапазона, фигурами и т. д. Они могут добавляться в процессе или инициализироваться изначально, меняя своё положение в зависимости от действий пользователя и в соответствии с определенными правилами. Некоторые из этих игр просты в своей реализации, содержат незначительное количество правил и объектов. Представляется возможным просчитать все сценарии и возможные результаты на дереве принятия решений. Примером служат крестики-нолики. Другие же содержат настолько большое количество вариантов, что требуют применения искусственного интеллекта, эвристик, математического аппарата теории игр и методов оптимизации [6]. В качестве такого примера приводят классическую игру в шахматы [1]. Её дидактический потенциал изучен уже достаточным образом [5]. Она позволяет знакомить с современными технологиями даже школьников [11]. Исследователи отмечают её роль в развитии мышления и творческих способностей при интегрированном обучении [2].

На практике мы отмечаем следующие особенности разработки.

CC BY

Реализация сцены происходит при помощи инициализации двумерного массива. Мы создаём сетку (grid) или таблицу (тег <table> в html). Игровые объекты, как правило, представлены различными типами данных: строковыми, числовыми, содержащими ссылки на графические файлы. Их расположение может быть задано изначально, а может осуществляться случайным образом в зависимости от конкретной игры.

Работа с циклами нужна для обновления игровой сцены после каждого хода. Пространство и время в таких играх представляется дискретным, достижение цели осуществляется за конечное число шагов. Проверка результата и сверка с искомым расположением объектов происходит после любого изменения. В некоторой мере это способно замедлять выполнение программы.

Использование ветвлений и условных операторов необходимо для отработки реакций компьютера на действия пользователя, а также для обработки коллизий и непредвиденных ситуаций. Они подходят для проверки окончания игры и для вывода сообщений о победе одной из сторон либо же ничьи. В крестиках-ноликах это расположение в одну линию одинаковых объектов, в пятнашках – определенная строго заданная ситуация, в шашках – отсутствие шашек противника, в «2048» – достижение одной из ячеек некоторого числового значения, в шахматах – мат или пат.

Функции и подпрограммы применяются для реализации перемещений и изменений в соответствии с игровым процессом. Может возникнуть иллюзия свободы воли пользователя и компьютера, однако происходит жёсткая регламентация всех действий алгоритмически. Никто не может выйти за рамки игровых правил и потенциально возможных сценариев.

Таким образом, происходит на практике ознакомление учащихся с основными аспектами программирования. Они осваивают различные типы данных, алгоритмические конструкции, линейные и условные операторы, циклы и ветвления, функции, графику, интерактивное взаимодействие с пользователем не по отдельности, а в виде цельной системы. Гораздо эффективнее происходит осмысление ими материала. Ведь они участвуют в создании полноценной программы, в которой демонстрируется функционирование всех её компонент.

Автор отмечает следующие возможные направления для дальнейшей работы:

– раскрытие взаимосвязей между компонентами разработки.

– изучение механизмов адаптации интеллектуальных компьютерных игр под способности обучающихся.

Можно заключить, что реализация рассматриваемого в данном исследовании класса игр демонстрирует определенный дидактический потенциал для преподавания алгоритмов и программирования. Возможно их использование в качестве практических заданий в соответствии с учебной программой. Это позволяет разнообразить образовательный процесс творческими наглядными упражнениями.

Литература

1. Батырь П.А., Сапожников С.В. Применение методов оптимизации при программировании шахмат // Энергия-2018: Тринадцатая международная научно-техническая

CC BY

конференция студентов, аспирантов и молодых учёных: Материалы конференции. В 6-ти томах. (г. Иваново, 03–05 апреля 2018 года). Иваново, 2018. С. 80.

2. Быкова, А.Р. Разработка компьютерных игр и тренажёров обучающимися при интегрированном обучении программированию и шахматам // Научный руководитель. 2018. № 2(26). С. 27-34.

3. Гибадуллин А.А. Программирование компьютерных интеллектуальных игр как метод обучения информатике // Педагогика: Мат-лы 59-й Международной научной студенческой конференции (г. Новосибирск, 12-23 апреля 2021 года). Новосибирск, 2021. С. 15-16.

4. Гибадуллин А.А. Разработка интеллектуальных игр в процессе обучения программированию // Педагогика: Мат-лы 58-й Международной научной студенческой конференции (г. Новосибирск, 10-13 апреля 2020 года). Новосибирск, 2020. С. 9-10.

5. Куев А.И. Модификация шахматных программ как процесс развития способностей искусственного интеллекта // Вестник науки Адыгейского республиканского института гуманитарных исследований имени Т.М. Керашева. 2019. № 19(43). С. 125-134.

6. Кушнир Н.В., Кушнир А.В., Каширин А.А., Тимченко А.В. Игровой искусственный интеллект // Научные труды Кубанского государственного технологического университета. 2016. № 2. С. 149-158.

7. Лагун Д.А., Гродель Ю.В. Эволюционный подход к игровому искусственному интеллекту // Приоритетные направления развития науки и образования. 2014. № 3(3). С. 165-167.

8. Митрошин А.А, Мороз В.Р. Дистанционное обучение программированию в процессе реализации алгоритмов игры в шашки // Современные технологии в науке и образовании – СТНО-2017: сб. трудов II Международной научно-технической и научно-методической конференции: в 8 т. (г. Рязань, 01-03 марта 2017 года). Рязань, 2017. С. 199-200.

9. Носков Е.А. Технологии обучения и геймификация в образовательной деятельности // Ярославский педагогический вестник. 2018 № 6. С. 138-144.

10. Титов С.А. «Геймификация» дистанционного обучения // Образование – путь к успеху. Международный форум «YEES 2012»: сб. научных трудов (г. Москва, 15 мая 2012 года). М., 2012. С. 206-208.

11. Швечков А.Е. Знакомство с основами искусственного интеллекта на уроках информатики в начальной школе // Научный альманах. 2020. № 2-1(64). С. 150-153.

© Гибадуллин А.А., 2022

УДК 371.3
https://doi.org/10.36906/AP-2022/64

Казиахмедов Т.Б.
ORCID: 0000-0002-5937-1514, канд. пед. наук
Симурзина Е.А.
ORCID: 0000-0001-8850-493X
Нижневартовский государственный университет
г. Нижневартовск, Россия

МЕЖДИСЦИПЛИНАРНЫЙ ПОДХОД ПРИ ИЗУЧЕНИИ ТИПОВ И СТРУКТУР ДАННЫХ В ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

Аннотация. В представленной работе описан междисциплинарный подход при изучении типов и структур данных на межязыковой основе. В качестве среды разработки использованы C++, PascalABC, PyCharm, Java Script. В работе рассматривается изучение типов и структур как математических структур с особенностями их реализации в разных языках высокого уровня. Именно использование разных языков программирования даёт полное представление об их реализации в языках программирования. Результатом работы является элементы методической системы междисциплинарного обучения типам и структурам данных, организации совместимости этих структур при разработке многоязычных программ.

Ключевые слова: классификация типов и структур данных; математические структуры; многоязычные проекты и приложения.

Kaziakhmedov T.B.
ORCID: 0000-0002-5937-1514, Ph.D.
Simurzina E.A.
ORCID: 0000-0001-8850-493X
Nizhnevartovsk State University
Nizhnevartovsk, Russia

INTERDISCIPLINARY APPROACH TO STUDYING DATA TYPES AND STRUCTURES IN PROGRAMMING LANGUAGES

Abstract. The presented work describes an interdisciplinary approach to the study of data types and structures on an interlanguage basis. C ++, PascalABC, PyCharm, Java Script were used as a development environment. The paper deals with the study of types and structures as mathematical structures with the peculiarities of their implementation in different high-level languages. It is the use of different programming languages that gives a complete picture of their implementation in programming languages. The result of the work is the elements of a methodological system for

CC BY

interdisciplinary teaching of data types and structures, the organization of the compatibility of these structures in the development of multilingual programs.

Keywords: classification of data types and structures; mathematical structures; multilingual projects and applications.

Обучение IT бакалавров разработке ПО требует формирования у них понимания важности структурирования данных и их использования для решения задач автоматизации деятельности различных отделов предприятий. Порой сталкиваемся с тем, что данные собранные в одних программах, студенты с трудом используют в других, с трудом представляют их в разных форматах. Поэтому обучение типам и структурам данных нужно рассматривать как один из основных элементов обучения программированию. Причём управление потоками данных в разных языках программирования различно по своим возможностям [1]. Работа по эффективному использованию структур данных и потоков данных требует их всестороннего анализа с целью не допущения создания лишних копий.

Программируя решение любой задачи, необходимо выбрать уровень абстрагирования. Иными словами, определить множество данных, представляющих предметную область решаемой задачи. При выборе следует руководствоваться проблематикой решаемой задачи и способами представления информации. Здесь необходимо ориентироваться на те средства, которые предоставляют системы программирования и вычислительная техника, на которой будут выполняться программы. Во многих случаях эти критерии не являются полностью независимыми.

Изучая программирование, студенты должны иметь представление о возможностях разных языков в формировании математических структур данных. Говоря о математических структурах, прежде всего –это составные данные: векторы, последовательности, множества, отображения, матрицы, тензоры, графы, деревья и т. д. (рис. 1.).

Можно сказать, что структура данных – кейс, хранящий данные в определенном макете, который позволяет структуре данных быть эффективной в некоторых операциях и неэффективной в других.

Полноценный подход в формировании всех этих структур можно реализовать на многоязычном программировании. О многоязычии при изучении программирования в первом курсе по направлению ИВТ сказано в [2].

Массивы – одна из самых широко используемых и самых простых структур данных. Задачи на массивах важны для изучения самих языков. В математике массивы- это вектора, матрицы, n-мерные массивы. В программировании важно, как и где хранить данные в этих структурах. Поэтому говоря о последовательностях следует обратить внимание на их связность или не связность. Однонаправленные, двунаправленные, кольцевые, мультисписки очень важны для формирования различных моделей памяти для хранения данных. Использование этих структур, которые в ЭВМ представляются в динамической (Heap) памяти позволяют решить задачи на графах, деревьях и т. д.

CC BY

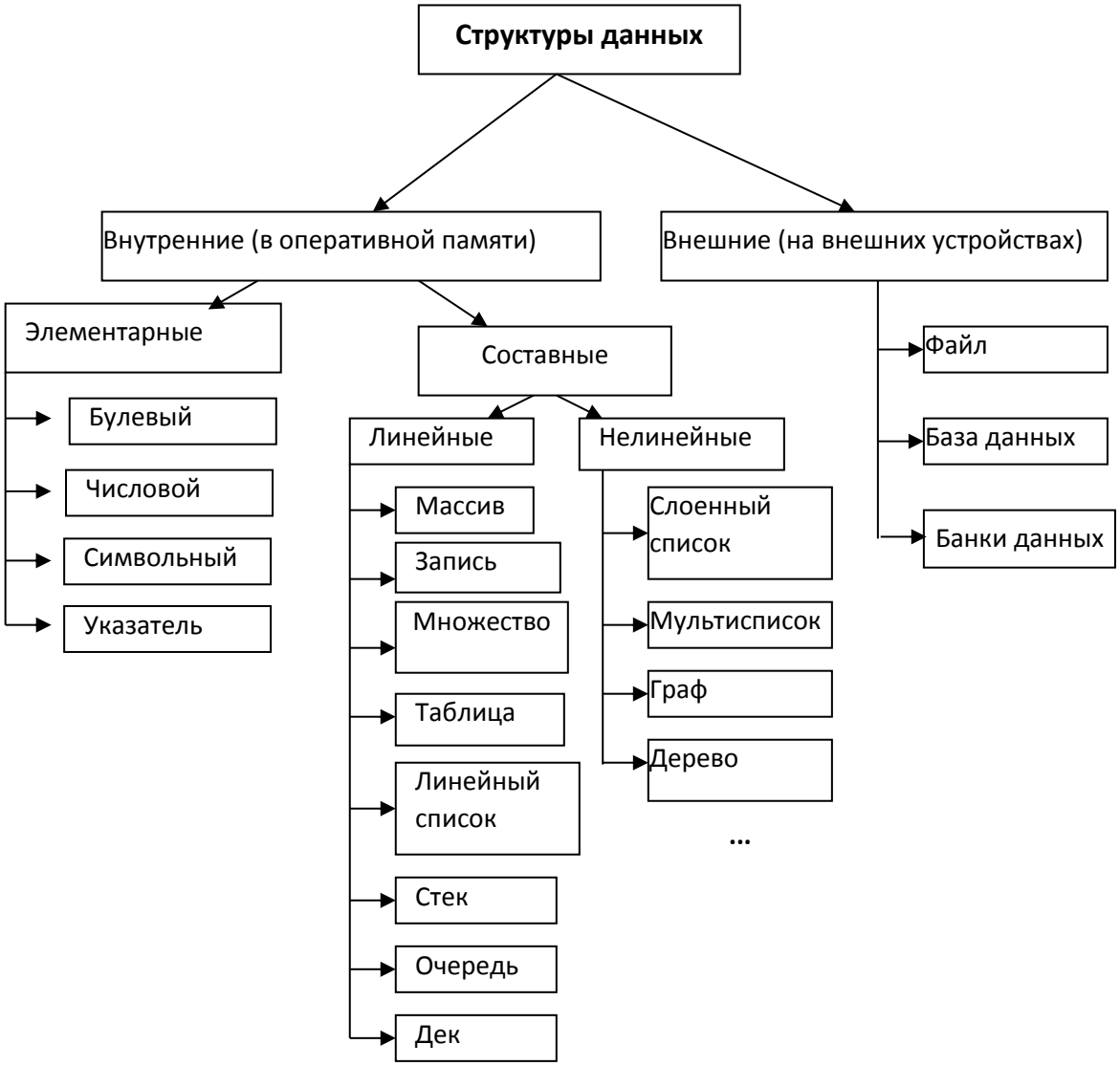


Рис. 1 Структуры данных

Наш опыт позволяет нам выполнить следующие сравнения языков высокого уровня (табл.).

Таблица

Сравнение методов реализации составных структур

Тип данных и структур	Java Script	Python	C++
Вектор	Одномерный массив	Список[]	Одномерный массив, шаблон vector
Множества, Мультимножества	Представление в виде массива	Множество {}	Шаблоны Set,MultiSet
Отображения, отображения с дубликатами	Программная реализация	Список пар	Программная реализация, шаблоны map,multimap
Двумерный массив	Двумерный массив	Список списк ов	Двумерный массив
Списки однонаправленные, двунаправленные, циклические	На основе массивов	Списки	На основе указателей, шаблон List

Стек	Программная реализация	Класс stack	Программная реализация, шаблон List
Очередь	Программная реализация	Класс queue	Программная реализация через указатель *, шаблон queue
Дек	Программная реализация	Класс deque	Программная реализация через указатель *, шаблон deque
Таблица	Массив записей	Список кортежей	Массив записей, в том числе и динамический
Запись	Встроенный тип	Список	Встроенный тип(struct)
База данных	таблицы	Список картежей	таблицы
Граф	Программная реализация(матрица смежности, матрица инцидентности, список вершин, список рёбер)	Вложенные списки	Программная реализация (матрица смежности, матрица инцидентности, список вершин, список рёбер и др)
Дерево	Программная реализация (методы матрица смежности, матрица инцидентности, список вершин, список рёбер, двунаправленный список)	Вложенные списки	Программная реализация (методы матрица смежности, матрица инцидентности, список вершин, список рёбер, двунаправленный список)

Составные структуры нужно реализовать как абстракции, т. е. как данные + методы работы с данными.

Следовательно, при абстрагировании мы рассматриваем составные типы как математические структуры, объясняя связь реализуемых методов именно для каждой структуры с математикой: алгебра логики, теория множеств, матричная алгебра, теория графов, алгоритмы на графах и деревьях, алгоритмы поиска на графах и деревьях [3].

В программировании хранение данных в оперативной памяти выполняется как в статической, так и динамической памяти (Heap-область). Изучение особенностей использования динамической памяти лучше демонстрировать на примере безразмерных массивов разных типов. Ниже приводится код работы с динамическими массива на C++:

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
struct ms{
string fio;
int voz;
};
int* a,**bb;  float *b; ms *mas;
  int n;
void main(){
system("chcp 1251 >null");
```

CC BY

```
cout<<"Введите  N"<<endl;
cin>>n;
//a=new int[n];
mas=new ms[n];
a=(int *)calloc(n,sizeof(int));
a[1]=7;
for (int i=0;i<n;i++)a[i]=i;
for (int i=0;i<n;i++)cout<<a[i]<<endl;
b=new float[n];
//b=(float *) calloc(n,sizeof(float));
for (int i=0;i<n;i++)*(b+i)=float(a[i]);
for (int i=0;i<n;i++)printf("\n%f",(*(b+i)));
mas[0].fio="Иванов";
mas[0].voz=30;
 mas[1].fio="Петров";
mas[1].voz=28;
    cout<<" "<<endl;
 for (int i=0;i<2;i++){cout<<mas[i].fio<<" "<<mas[i].voz<<endl;}
 getch();
}
```

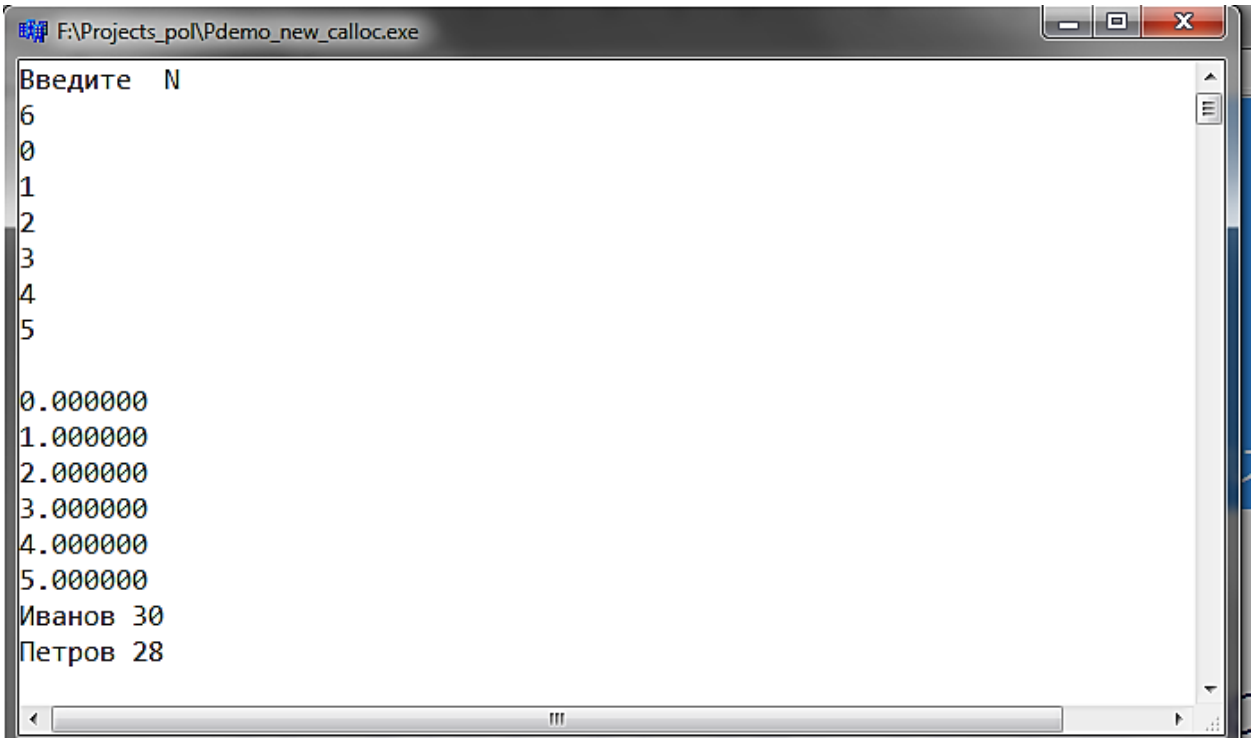


Рис. 2 . Экран выполненной программы

Основной структурой для полного понимания как хранятся данные в памяти является связанный список. В отличии от массивов связанные списки имеют структурную гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями. Реализация списков формирует и представление о программировании всех видов списков:

• однонаправленные: каждый узел хранит адрес или ссылку на следующий узел в списке и последний узел имеет следующий адрес или ссылку как NULL (1->2->3->4->NULL);

• двунаправленные: две ссылки, которые связаны с каждым узлом, одним из опорных пунктов на следующий узел и один к предыдущему узлу (NULL<-1<->2<->3->NULL);

• круговые: все узлы соединяются и образуют круг. В конце нет NULL. Циклический связанный список может быть одно или двукратным циклическим связанным списком (1->2->3->1).

Для списков реализуется следующие основные операции:

• InsertAtEnd – вставка заданного элемента в конец списка;

• InserAtHead – вставка элемента в начло списка;

• Delete – удаляет заданный элемент из списка;

• DeleteAtHead – удаляет первый элемент списка;

• Search – возвращает заданный элемент из списка;

• IsEmpty – возвращает true, если связанный список пуст.

С использованием структур данных при разработке программ более подробно можно ознакомиться в [4].

Графы и деревья – математические структуры, программирование которых ускоряет задачи поиска и сортировки, особенно задачи поиска в больших объёмах данных. Граф как математический объект представляет собой совокупность двух множеств – множество самих объектов (вершин) и множество их парных связей (рёбер). Графы находят широкое применение во многих современных сферах. Они используются и в социальных науках (например, социологии) и в естественных науках (физике и химии), но наибольшее применение графы получили в информатике и сетевых технологиях.

Пример программы реализации графов на C++:

```
//Обход графа в ширину.
#include <iostream>
#include <queue> // очередь
using namespace std;
int main()
{
 queue<int> Queue;
 int mas[7][7] = { { 0, 1, 1, 0, 0, 0, 1 }, // матрица смежности
 { 1, 0, 1, 1, 0, 0, 0 },
 { 1, 1, 0, 0, 0, 0, 0 },
```

CC BY

```
  { 0, 1, 0, 0, 1, 0, 0 },
  { 0, 0, 0, 1, 0, 1, 0 },
  { 0, 0, 0, 0, 1, 0, 1 },
  { 1, 0, 0, 0, 0, 1, 0 } };
  int nodes[7]; // вершины графа
  for (int i = 0; i < 7; i++)
    nodes[i] = 0; // исходно все вершины равны 0
  Queue.push(0); // помещаем в очередь первую вершину
  while (!Queue.empty())
  { // пока очередь не пуста
    int node = Queue.front(); // извлекаем вершину
    Queue.pop();
    nodes[node] = 2; // отмечаем ее как посещенную
    for (int j = 0; j < 7; j++)
    { // проверяем для нее все смежные вершины
      if (mas[node][j] == 1 && nodes[j] == 0)
      { // если вершина смежная и не обнаружена
        Queue.push(j); // добавляем ее в очередь
        nodes[j] = 1; // отмечаем вершину как обнаруженную
      }
    }
    cout << node + 1 << endl; // выводим номер вершины
  }
  cin.get();
  return 0;
}
```

Дерево – иерархическая структура данных, которая состоит из узлов (вершин) и рёбер (дуг). По сути деревья – это связанные графы без циклов. Важно, чтобы студенты программировали деревья следующих видов:

- N- деревья
- Сбалансированные деревья
- Бинарные деревья
- Дерево бинарного поиска
- AVL дерево
- 2-3-4 деревья

Одно из самых распространённых – бинарное дерево.

Имеется три способа обхода деревьев:

1. В прямом порядке (сверху-вниз) - префиксная форма.
2. В симметричном порядке (слева-направо) – инфиксная форма
3. В обратном порядке (снизу-вверх) – постфиксная форма

CC BY

Пример реализации дерева на C++:

```
#include <stdlib.h>
#include <iostream>

using namespace std;
struct node {
  int data;
struct node *left;
  struct node *right;
};
// Создание нового узла
struct node *newNode(int data) {
  struct node *node = (struct node *)malloc(sizeof(struct node));
  node->data = data;
  node->left = NULL;
  node->right = NULL;
  return (node);
}
// Прямой обход дерева
void traversePreOrder(struct node *temp) {
  if (temp != NULL) {
    cout << " " << temp->data;
    traversePreOrder(temp->left);
    traversePreOrder(temp->right);
  }
}
```

IT индустрия ставит серьёзные проблемы перед Вузами и государством в условиях информатизации всех сфер экономики государства [5].

Таким образом, подготовка IT специалистов становится одной из главных задач при подготовке кадров для цифровой экономики.

Литература

1. Казиахмедов Т.Б. Яламов Г.Ю. Содержание и методы обучения программированию бакалавров по направлению «Информатика и вычислительная техника» // Педагогическая информатика. 2019. № 4. С. 47-58.

2. Казиахмедов Т.Б., Симурзина Е.А. Междисциплинарный подход в обучении программированию бакалавров ИВТ // Современное программирование: Мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 246-250. https://doi.org/10.36906/AP-2020/48

CC BY

3. Казиахмедов Т.Б. Информатика как профессионально-ориентированная дисциплина в системе подготовки бакалавров // Современное программирование: Мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 19-24. https://doi.org/10.36906/AP-2020/03

4. Казиахмедов Т.Б., Елыкомов Р.Н. Этапы разработки автоматизированных информационных систем удалённого тестирования обучающихся // Информатизация образования – 2020: Международная научно-практическая конференция (г. Орел, 29-31 октября 2020 года). Орел, 2020. С. 329-334.

5. Казиахмедов Т.Б. Опережающее обучение в области индустрии информационных технологий в условиях развивающейся экономики и перманентных реформ высшего образования // Педагогическая информатика. 2014. № 4. С. 62-72.

© Казиахмедов Т.Б., Симурзина Е.А., 2022

УДК 371.39
https://doi.org/10.36906/AP-2022/65

Абильдинова Г.М.
ORCID: 0000-0002-4262-6197, канд. пед. наук
Каратаева Ж.Р.
ORCID: 0000-0003-2560-5498
Евразийский национальный университет им Л.Н. Гумилева
г. Нур-Султан, Казахстан

ОСОБЕННОСТИ ДИСТАНЦИОННОГО ОБУЧЕНИЯ В ПЕРИОД САМОИЗОЛЯЦИИ

Аннотация. Одним из актуальных вопросов 2020 года является влияние пандемии на все сферы жизни, включая образование. В данной статье описываются использования дистанционного образования, особенности дистанционного образования. Достоинства и недостатки данной формы обучения.

Ключевые слова: дистанционное обучение; онлайн обучение; платформа; образование; информационные технологии; цифровизации.

Abildinova G.M.
ORCID: 0000-0002-4262-6197, Ph.D.
Karataeva Zh.R.
ORCID: 0000-0003-2560-5498
L.N. Gumilyov Eurasian National University
Nur-Sultan, Kazakhstan

FEATURES OF DISTANCE LEARNING DURING SELF-ISOLATION

Abstract. One of the pressing issues of 2020 is the impact of the pandemic in all areas, including education. This article describes the use of distance education, the features of distance education. Advantages and disadvantages of this form of training.

Keywords: distance learning; online learning; platform; education; information technology; digitalization.

Дистанционная форма обучения за последние годы и так приобретала все больше и больше соратников, а вот с приходом пандемии, в целях замедления нарастающей волны угрозы здоровью населения и закрытием учебных заведений дистанционное обучение стало неотъемлемой частью образования. А активное использование информационных технологий в образовании позволило с лёгкостью взаимодействовать студенту и преподавателю.

Дистанционное образование - это возможность получения полноценного образования без отрыва от привычной обстановки и выбор удобного времени, отражающее все присущие

CC BY

учебному процессу компоненты (цели, методы, средства обучения). Более того, это продвинутый вид образования, возникший в результате информационной революции [2].

Выделяют следующие преимущества дистанционного образования, такие как:

– в условиях пандемии, уменьшение риска заболеваемости как учащихся так и преподавательского состава [5].

– увеличение степени вовлеченности при помощи технологий VR;

– возможность учиться удалённо с любой точки континента при наличии гаджетов и в любое время [1];

– возможность просмотра видеолекций в любое время и тот отрезок который менее понятен;

– возможность получения равного образования для реализации прав человека на образование и информацию невзирая на физическую способность;

– возможность получения непрерывного образования без потери актуальности и новизны информации для студента;

– влечёт прямую экономическую выгоду.

Однако, выделяют и другую сторону дистанционного образования.

Главным недостатком отмечают увеличение риска плагиата, возможности снижение навыков логического и творческого мышления, упадок коммуникативных навыков и ухудшения основных параметров здоровья, сложность в оценивании ученика и учёта переработки учителя [4].

Более того, согласно [2], на пути реализации и развития дистанционного образования является относительная неразвитость ИТ инфраструктур в России. Данная проблема широко охвачена и в Казахстане, не во всех уголках страны имеется широкополосный интернет, ну или сбои в работе интернета и отсутствие компьютеров [3].

В целом рассмотрев как положительные так и отрицательные стороны дистанционного образования можно резюмировать что, дистанционное образование требует от человека высокой степени цифровой грамотности. Что по сути является естественным отбором, для результата которой как минимум требуется жёсткая самодисциплина, более того, зависит от самостоятельности и сознательности учащегося, от его мотивации, сформированности умений и навыков самостоятельной работы.

Таким образом, можно сказать что дистанционное образование может и должно оставаться, и развиваться параллельно с традиционными формами, для повышения квалификации, самообразования, сопровождения студентов с особыми образовательными потребностями, обучающихся по индивидуальным планам, работающих на производстве и наверное никогда не сможет заменить традиционную форму обучения, однако высокая рентабельность и низкая цена образовательной услуги является отличной мотивацией для её развития в целом.

CC BY

Литература

1. Леонтьев Е.Д. Особенности дистанционного образования в условиях цифровой трансформации // Международный журнал гуманитарных и естественных наук. 2021. №5-1(56). С. 211-215. https://doi.org/10.24412/2500-1000-2021-5-1-211-216

2. Гаспарян А.В. Проблемы и перспективы дистанционного образования // Педагогика, психология, общество: актуальные исследования: сб. мат-ов Всероссийской научно-практической конференции (г. Чебоксары, 19 марта 2021 года). Чебоксары, 2021. С. 101-104.

3. Уцын Г.Е., Гришаева Н.Ю. Проблемы дистанционного образования // Современное образование: повышение конкурентоспособности университетов: мат-лы международной научно-методической конференции (г. Томск, 28-29 января 2021 года). Томск, 2021. С. 168-171.

4. Морозов А.В., Самборская Л.Н. Особенности электронного образования в условиях цифровизации // Управление образованием: теория и практика. 2020. № 2(38). С. 62-68.

5. Наркузиева Г.З. Дистанционное образование: плюсы и минусы //Academic Research in Educational Sciences. 2021. Т. 2. № 4. С. 1020-1025.

© Абильдинова Г.М., Каратаева Ж.Р., 2022

УДК 371.39
https://doi.org/10.36906/AP-2022/66

Барышева И.В.
Институт информационных технологий, математики и механики
Национального исследовательского Нижегородского
государственного университета им. Н.И. Лобачевского
г. Нижний Новгород, Россия
Козлов О.А.
Институт стратегии развития образования Российской академии образования
г. Москва, Россия
Малкина Е.В.
Институт информационных технологий, математики и механики
Национального исследовательского Нижегородского
Государственного университета им. Н.И. Лобачевского
г. Нижний Новгород, Россия

УРОВЕНЬ ПОДГОТОВКИ АБИТУРИЕНТОВ ПО ПРОГРАММИРОВАНИЮ И ЕГО ВЛИЯНИЕ НА ВЫБОР ПРОФЕССИИ

Аннотация. В статье рассматривается проблематика подготовки будущих студентов по программированию, описываются практические задачи, помогающие абитуриентам успешно сдать экзамен в рамках информатики.

Ключевые слова: обучение; информатика; программирование; подготовка абитуриентов.

Barysheva I.V.
Institute of Information Technologies, Mathematics and Mechanics
Lobachevsky State University of Nizhny Novgorod
Nizhny Novgorod, Russia
Kozlov O.A.
Institute of Education Development Strategy of the Russian Academy of Education
Moscow, Russia
Malkina E.V.
Institute of Information Technology, Mathematics and Mechanics
Lobachevsky State University of Nizhny Novgorod
Nizhny Novgorod, Russia

THE LEVEL OF TRAINING OF APPLICANTS IN PROGRAMMING AND ITS IMPACT ON THE CHOICE OF PROFESSION

Abstract. The article discusses the problems of training future students in programming, describes practical tasks that help applicants to successfully pass the exam in computer science.

Keywords: education, computer science, programming, preparation of applicants.

Школьный курс «Информатика и КТ» появился в средине восьмидесятых годов прошлого века. Название курса в те времена мало соответствовало содержимому, которое практически сводилось к элементам программирования в разных вариантах в зависимости от компетенции учителя. Однако предмет, независимо от названия, всегда вызывал интерес школьников, частично доказательством этого факта можно считать победы наших школьников на международных олимпиадах. При этом понимание будущей профессии, связанной с данным предметом также было очевидным.

Впервые информатика, как вступительный экзамен в ННГУ, появилась в 1995 году на юридическом факультете, чуть позже на экономическом, финансовом и факультете ВМК. Содержимое экзамена – четыре задания:

• системы счисления;

• логические уравнения;

• составление программы содержательной задачи с числовыми данными;

• составление программ для задачи с преимущественно строковыми данными.

Например. Первое задание предполагает знания основных элементов позиционных систем счисления, взаимные переводы из одной системы в другую, выполнение арифметических действий в различных системах.

Выполнение второго задания требовало знания логических операций, умения строить таблицы истинности, использовать законы Моргана. В качестве ответа необходимо представить решения уравнения, а не их количество.

Успешное решение двух первых заданий обеспечивало получение минимальной положительной оценки. Решение задач по составлению программ можно разбить на этапы, каждый из которых в зависимости от сложности оценивался разным количеством баллов. Обязательным пунктом был пункт разработки структуры данных и именование объектов. окончательная оценка формировалась из суммарного количества баллов за все этапы. Сложность задач варьировалась в зависимости от факультета.

Появление экзамена на юридическом факультете оправдывалось чёткой системой выставления оценок с одной стороны, с другой стороны, в процессе обучения программированию задач с легендой, прежде всего, формируется умение структурировать информацию, умение представлять сложную проблему как совокупность менее сложных задач, подчас уже имеющих решение, умение выявлять зависимости между объектами.

Вариант задания для экзамена на юридическом факультете от 12.03.2006 г. [1].

Задание 1. *Найти шестнадцатеричное решение уравнения*

*2*X = M + 17*L – K – T + P +X,*

где 17 – десятичное число,

M - десятичное число *M=36 480,*

CC BY

P - восьмеричное число *P=25 007,*

L - восьмеричное число *L= 7 241,*

K - двоичное число *K= 1101 1100 0010 0110,*

T - шестнадцатеричное число *T= FAAA.*

Задание 2. *Найти решение булевского уравнения*

$$\overline{(x1+x3)\cdot\overline{(x2+x4)}} = \overline{x3+x2+x1\cdot x4}$$

Задание 3. *Дан список студентов и список наименований предметов, по которым студенты сдавали экзамены. Один из предметов – экономика. Известны оценки каждого студента по каждому предмету.*

Разработать ***алгоритм,*** *дать его* ***описание*** *(общую схему и назначение используемых переменных), привести* ***программу*** *определения предмета, по которому наименьшее количество студентов получили свои наилучшие оценки. Указать преобладающую оценку в группе по каждому предмету. Вывести таблицу оценок в порядке убывания среднего балла студентов, при равных средних баллах – по убыванию оценок по экономике. Предметы в таблице расположить в порядке возрастания преобладающей оценки в группе по каждому предмету.*

КОНТРОЛЬНЫЙ ПРИМЕР

Исходные данные: *Кол-во студентов – 7, кол-во предметов – 3.*

Фамилия	Экономика	Информатика	Математика
Борисова	5	5	4
Берегов	3	2	2
Герасимов	3	3	2
Горнова	4	4	4
Доронина	3	3	4
Мартынов	4	5	5
Якушина	5	5	5

Решение:

Наименьшее количество студентов (4) получили свои наилучшие оценки по предмету ***математика.***

Преобладающие оценки: по экономике – 3, по математике - 4, по информатике -5.

Таблица оценок

Фамилия	*Экон-ка*	*Мат-ка*	*Инфор-ка*
Якушина	*5*	*5*	*5*
Борисова	*5*	*4*	*5*
Мартынов	*4*	*5*	*5*
Горнова	*4*	*4*	*4*
Доронина	*3*	*4*	*3*
Герасимов	*3*	*2*	*3*
Берегов	*3*	*2*	*2*

CC BY

Задание 4. *Несколько текстовых отрывков, составленных из предложений, были разделены на части–фрагменты, которые произвольным образом перемешаны в наборе строк одинаковой длины. Фрагменты отделены друг от друга символами '|'. Разделение на фрагменты происходило по позициям внутри слов. В оба фрагмента дополнительно включались соседние к месту разделения участки слова, чтобы в конце фрагмента–предшественника и в начале фрагмента-продолжения содержалась общая часть разделённого слова длиной в три символа.*

Разработать АЛГОРИТМ, дать его ОПИСАНИЕ (общую схему и назначение используемых переменных), привести ПРОГРАММУ, восстанавливающую исходные тексты высказываний и печатающую их в порядке возрастания количества предложений, а при их равенстве – по возрастанию количества слов. Признаком конца предложения является точка. В тексте могли использоваться знаки препинания: запятая, точка, дефис, двоеточие.

КОНТРОЛЬНЫЙ ПРИМЕР

Исходные данные:

Количество строк текста – 3.

Строки текста:

Должен знать твой враг. Артур Шопенгауэр.| тупков. Георг Вильгельм Фридрих Гегель.| ругу того, что не долж | иное, как ряд его поступ | Не говори друг | Человек есть не что ино

Результат:

1) Не говори другу того, что не должен знать твой враг. Артур Шопенгауэр.

2) Человек есть не что иное, как ряд его поступков. Георг Вильгельм Фридрих Гегель.

Для абитуриентов факультетов, где программирование составляет значительную часть изучаемых предметов, задачи давались более сложные. Например, предложенные на экзамене по информатике 17.07.2006 г:

Задание 1. *Найти шестнадцатеричное решение уравнения*

*4*X = M + L - K + T + 3*X ,*

где

M – десятичное число M=53 256,

L – восьмеричное число L= 7 254,

K – двоичное число K= 1011 1011 0010 1000,

T – шестнадцатеричное число T= F77D.

Задание 2. *Решить булевское уравнение* $\overline{(x1 + x2) * x3 + \overline{x5})} = x2 + \overline{x1} * x2$.

Задание 3. *Для расчёта квартплаты за 1 месяц используется таблица, содержащая следующие поля: название улицы, № дома, № квартиры, фамилия квартиросъёмщика, площадь квартиры, количество проживающих, дата оплаты (день и месяц в формате dd/mm.). Даты даны в пределах одного високосного года. Второй таблицей задана информация о наличии льгот (либо ветеран войны, либо ветеран труда) у некоторых из квартиросъёмщиков. Значение тарифов оплаты коммунальных платежей определяется третьей таблицей:*

CC BY

Тариф № 1 за 1 кв.м (в руб.)	20
Тариф №2 за 1 кв.м (в руб.)	25
Льгота для ВОВ (в %)	70
Льгота для ветеранов труда (в %)	50

Если на 1 человека приходится более 15 кв. м, то оплата начисляется по тарифу №2. Пени (0.2% от суммы оплаты за каждый день просрочки) начисляются, если оплата произведена после 10 числа каждого месяца.

*Разработать **алгоритм**, дать его **описание** (общую схему и назначение используемых переменных), привести **программу**, печатающую фамилии и адреса всех жильцов, внёсших максимальную квартплату за январь. Фамилии расположить в алфавитном порядке. Подсчитать общую заплаченную сумму пени за январь.*

Контрольный пример

Исходные данные:

Улица	Дом	Кварт.	Фамилия	Площ.	Кол. прож.	Дата оплаты
Мира	1	1	Иванов	50	3	09/02
Труда	2	2	Петров	30	2	10/02
Труда	2	3	Сидоров	62.5	5	31/01
1 Мая	3	1	Иванов	30	1	15/02

Фамилия	Улица	Дом	Кварт.	Тип льгот
Иванов	1 Мая	3	1	Ветеран войны

Решение:

Максимальную квартплату – 1250 руб. внесли:

Иванов ул. Мира д.1 кв.1,

Сидоров ул. Труда д.2 кв.3.

Сумма пени -2 руб. 25 коп.

***Задание 4.** Билеты с вопросами для экзамена упакованы в строках одинаковой длины, строка обрывается на том символе, на котором достигнута заданная длина, все слова отделены ровно одним пробелом. Номера билетов не совпадают с порядком их следования в тексте. Известно, что каждый билет имеет два вопроса, сформулированные каждый в своём предложении. Необходимо преобразовать текст так, чтобы:*

- ✓ *Порядок следования билетов совпадал с их нумерацией;*
- ✓ *Слово «Билет» стояло посредине строки;*
- ✓ *Вопросы в билете пронумерованы и прижаты к левому краю.*

Разработать АЛГОРИТМ, дать его ОПИСАНИЕ (общую схему и назначение используемых переменных), привести ПРОГРАММУ решения задачи.

CC BY

КОНТРОЛЬНЫЙ ПРИМЕР

Исходные данные:

Билет 3 Теорема Пифагора. Свойства равнобедренного треугольника. Билет 1 Условия равенства треугольников. Трапеция, определение, свойства, площадь трапеции. Билет 2 Теорема о трёх перпендикулярах. Средняя линия треугольника, определение, свойства.

Результат:

Билет 1

1. Условия равенства треугольников.

2. Трапеция, определение, свойства, площадь трапеции.

Билет 2

1. Теорема о трёх перпендикулярах.

2. Средняя линия треугольника, определение, свойства.

Билет 3

1. Теорема Пифагора.

2. Свойства равнобедренного треугольника.

В основе решения всех заданий по программированию лежит фиксированное число простых алгоритмов, которые применялись в течение многих лет, задачи при кажущейся сложности не являются олимпиадными, так как существует методология их решения. Выпускники школ, которые готовились к экзамену по информатике, понимали, что программирование это важная составляющая будущей профессии, понимали сложность данной науки, сопоставляли свои возможности, свои склонности к данной науке.

Интенсивное развитие информатики как науки привело к размыванию границ школьного предмета «Информатика», постепенно программирование вытеснялось темами, может быть, более простыми, привлекательными, модными, наконец. Если посмотреть на задания ЕГЭ, можно оценить разнообразие тем, некоторые из них сложно отнести собственно к информатике. Программирование представлено, одной стороны простейшими задачами, решение которых не требует умения составлять алгоритмы. С другой стороны, фигурируют такие понятия как функции без объяснения аппарата формально-фактических параметров, или рекурсия, являющаяся одним из сложнейших понятий в программировании. Эти понятия используются без какого-либо объяснения, а школьники должны просто запомнить последовательность действий. Однако, целью вступительного экзамена является, в конечном итоге не получение решений конкретных задач, а отбор наиболее талантливых абитуриентов, способных логически мыслить, структурировать поставленную проблему, применять полученные знания. Когда на решение 27 заданий даётся 3 часа 55 минут, трудно предполагать наличие серьёзных вопросов, требующих вдумчивого решения, скорее натасканных действий.

Уровень абитуриентов пришлось учитывать и на экзаменах, проводимых университетом. Задания все больше становились ЕГЭ-подобными. Действия над числами в различных системах счисления все проще, логические задачи элементарными.

CC BY

Вариант задания для экзамена 20.07.2021

Задание 1.

Вычислить значение арифметического выражения

$$745_8 + A0C3_{16} + 1111000011_2$$

Задание 2.

Вычислить в шестнадцатеричной системе счисления разность двух шестнадцатеричных чисел

$$ABFCD_{16} - AEDC_{16}$$

Задание 3.

Определить систему счисления p, в которой выполняется равенство

$$107_p + 17_p = 86_{10}$$

Задание 4.

Определить минимальное целое значение X, для которых логическое выражение истинно

$$(X>250)\&(X<=340)$$

Задание 5.

Доказать, что логическое выражение

$$(X \vee Y) \vee (\neg(X \vee Y)$$

Истинно при любых значениях переменных X, Y

Задание 6.

Найти все решения логического уравнения для переменных X,Y,Z,W, перевести каждое решение как двоичное число в шестнадцатеричную систему счисления, полученные цифры записать в качестве ответа без пробелов.

$$(X + Z)\neg\ (Z + W) + XY = \neg(\ \neg(Z + Y) + W\,X)$$

Задание 7.

Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
var k, s: integer;
begin
s:=0;
k:=1;
while k < 11 do begin
s:=s+k;
k:=k+1;
end;
write(s);
end.
```

Задание 8.

CC BY

Из общего списка участников олимпиады сформировать список соискателей, которым рекомендовано льготное обучение на подготовительном факультете. В этот список включаются участники олимпиады, набравшие баллы более среднего балла всех претендентов и не попавшие в число победителей. Список победителей считать заданным.

Представить описание переменных и словесное описание алгоритма, написать программу на алгоритмическом языке для решения задачи.

Задание 9.

Задана таблица, которая содержит сведения о полученных медалях на спортивной Олимпиаде. Эти сведения включают: название страны, количество золотых, серебряных и бронзовых медалей, а также число мировых рекордов, принадлежащих спортсменам данной страны. Разработать алгоритм, дать описание всех переменных, написать программу, определяющую три страны, имеющие наивысший рейтинг. Рейтинг равен сумме произведений количества каждого типа медалей на соответствующий коэффициент плюс число мировых рекордов. Коэффициент золотой медали равен 1, серебряной – 0.75, бронзовой – 0.5.

Представить описание переменных и словесное описание алгоритма, написать программу на алгоритмическом языке для решения задачи.

Контрольный пример.

Страна	Золото	Серебро	Бронза	Мировые рекорды
Китай	10	3	5	1
Австралия	4	0	1	3
Россия	1	7	5	7
Япония	5	6	8	2

Ответ: Китай.

Задание 10.

В заданной строке найти самое длинное слово, включающие буквы «а» и «н» на заданную букву, кроме этого вывести в качестве результата полученные слова.

Представить описание переменных и словесное описание алгоритма, написать программу на алгоритмическом языке для решения задачи.

Задание с номерами 3 и 6 среди абитуриентов считалось повышенной сложности. Исключение составляют абитуриенты, сдававшие экзамен онлайн, особенно те, кто участвовал в августе, сентябре и подавали документы на платное или очно-заочные отделения. Последние три задания, в которых требовалось написать программу, попробовали начать решать 19 из 54 участвовавших в экзамене. Полностью завершить хотя бы одно из трех заданий не удалось никому. Если на алгоритмизацию в школьной программе отведено 5-8 часов, сложно предполагать, что школьники понимают особенности профессии программиста.

Существует мнение, что только на уроках информатики школьник может и должен усвоить основные нормы и правила информационного взаимодействия, в том числе и в сетях, как с другими пользователями, так и с интерактивными информационными ресурсами, и оградить свою личностную информационную сферу от возможного агрессивного внешнего воздействия. Кроме того, ученика только на уроках информатики можно обучить основным

CC BY

приёмам выявления легитимной информации, в том числе путём опознания её по отличительным признакам. Обучение приёмам и мерам защиты от некачественной информации также является прерогативой курса информатики, так как поиск информации по существенным признакам включает приёмы критического отношения к содержательному компоненту информации. При информационном взаимодействии с другими пользователями ученика на уроках информатики следует обучать форматам сетевого общения, как технологическим, так и этическим, позволяющим пресекать агрессивное отношение к другим пользователям или бесцельные информационные «вбросы» [2].

Безусловно, все это имеет место быть, и школьный курс информатики должен все затронутые вопросы освещать. Современный человек должен обладать всеми перечисленными компетенциями. А где программирование? От уровня квалификации наших программистов зависит само существование и интерактивных информационных ресурсов, и сетевое общение, и облачные хранилища и многое другое, а главное, информационная независимость и безопасность страны. Чем более ориентированными в профессии придут абитуриенты в высшие учебные заведения, тем более профессиональных программистов получится подготовить в итоге. Отсутствие программирования как самостоятельного предмета, но растворенного в аморфной массе «Информатики» ставит под вопрос осознанного выбора будущей профессии школьниками.

Литература

1. Барышева И.В., Козлов О.А. Проектный метод в изучении основ программирования студентами профильных специальностей // Педагогическая информатика. 2016. № 4. С. 78-83.

2. Роберт И.В., Мухаметзянов И.Ш., Козлов О.А., Поляков В.П., Шихнабиева Т.Ш., Касторнова В.А. Актуализация содержания предметной области «информатика» основной школы в условиях научно-технического прогресса периода цифровых технологий // Наука о человеке: гуманитарные исследования. 2019. № 3(37). С. 58-69.

© Барышева И.В., Козлов О.А., Малкина Е.В., 2022

УДК 378
https://doi.org/10.36906/AP-2021/67

Коледин В.В.
ORCID: 0000-0002-8410-918X, канд. физ.-мат. наук
Филиал Тюменского индустриального университета в г. Нижневартовске
г. Нижневартовск, Россия
Зверева Е.А.
ORCID: 0000-0001-7684-2499, канд. пед. наук
Филиал Южно-Уральского государственного университета в г. Нижневартовске
г. Нижневартовск, Россия

МЕТОДЫ РАЗВИТИЯ МАТЕМАТИЧЕСКОЙ КОМПЕТЕНТНОСТИ БАКАЛАВРОВ НАПРАВЛЕНИЯ «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Аннотация. В данной работе рассматривается актуальность развития математической компетентности у обучающихся по направлению подготовки «Программного обеспечения» как важной составляющей его профессиональной компетентности. В качестве методов её развития предлагается формирование геометрических образов. В статье представлен пример создания математического образа, в данном случае единичной окружности, для того чтобы используя его сформировать у обучающихся понимание: что такое синус, косинус, чему равны углы от синуса, косинуса и т.д. Создать на основе образа единичной окружности такие понятия, как основное тригонометрическое тождество и синус двойного угла и т. д.

Ключевые слова: математическая компетентность; программная инженерия; тригонометрия; математический образ.

Koledin V.V.
ORCID: 0000-0002-8410-918X , Ph.D.
Branch of Industrial University of Tyumen in Nizhnevartovsk
Nizhnevartovsk, Russia
Zvereva E.A.
ORCID: 0000-0001-7684-2499, Ph.D.
South Ural State University Branch in Nizhnevartovsk
Nizhnevartovsk, Russia

METHODS OF DEVELOPMENT OF MATHEMATICAL COMPETENCE OF BACHELORS IN THE DIRECTION OF "SOFTWARE ENGINEERING"

Abstract: This paper examines the relevance of the development of mathematical competence among students in the field of training "Software Engineering" as an important component of its professional competence. As methods of its development, the formation of geometric images in

CC BY

mathematics is proposed. The article presents an example of creating a mathematical image, in this case a single circle, in order to use it to form students' understanding: what is the sine, cosine, what are the angles from the sine, cosine, etc. Create concepts such as the basic trigonometric identity, the sine of a double angle, etc. based on the image of a unit circle.

Keywords: mathematical competence; software engineering; trigonometry; mathematical image.

Становление современного разработчика программного обеспечения невозможно без качественной математической подготовки, являющейся основой для решения множества задач в области его профессиональной деятельности. Например, при изучении предметной области автоматизации возникает необходимость в математической формализации. На основе математических моделей разрабатывается соответствующее программное обеспечение, реализующее математическую модель объекта и математические методы, позволяющие найти оптимальное решение (https://clck.ru/bd3w3). Для решения задач интеллектуального анализа данных (Data Mining), которые заключаются в выявление скрытых закономерностей в большом объёме накопленной информации, необходимы знания методов теории вероятностей и математической статистики. Математическая подготовка нужна также в областях, связанных с машинным обучением, нейронными сетями. То есть чем больше системных задач решается программистом, чем больше нужно знание математической части. «Недостаточный уровень математической подготовки в процессе обучения специалистов приводит к тому, что при изучении специальных дисциплин происходит простое накапливание информации без глубокого понимания сущности происходящих процессов и их логической взаимосвязи» [2, с. 3]. Таким образом, умение результативно использовать математические знания в профессиональной деятельности является неотъемлемым показателем конкурентоспособности выпускника, обучающегося по направлению подготовки «Программная инженерия». Следовательно, развитие математической компетентности, являющейся важной составляющей профессиональной компетентности у обучающихся данного направления является профессионально значимой и актуальной.

«Компетентность, специалиста любого профиля формируется на протяжении всей его жизни и на всех ступенях его образования. Задачей профессионального образования является обеспечение преемственности в её развитии, которое заключается в определении уровня её сформированности у абитуриента, коррекции и формировании её в процессе обучения в вузе у студента в рамках освоения дисциплин учебного плана, стимулировании и ориентации выпускника на её совершенствование» [4, с. 41].

Но зачастую, на практике на занятиях по высшей математике в вузе мы сталкиваемся с очень низким уровнем базовой математической компетентности у «вчерашних школьников», а ныне студентов первокурсников, особенно студентов заочной формы обучения. Например, казалось бы, задавая элементарный вопрос студентам заочного или очно-заочного отделения, а порой даже очного отделения «Что такое синус и чему равен косинус 60 градусов?», мы получаем либо неправильный ответ, либо ответа вообще нет. Таким образом, помимо тем,

CC BY

предусмотренных рабочей программой дисциплины, преподавателю приходится объяснять ещё школьную программу студентам. Но часы, отводимые на дисциплину, ограничены, поэтому приходится придумывать новые идеи и подходы, чтобы развить математическую компетентность, заинтересовать обучающихся, привить им любовь к математике. В рамках данной статьи мы предлагаем в качестве такого подхода рассмотреть метод формирования геометрического образа, на примере развития тригонометрических знаний.

За основу были взяты основные пробелы студентов, которые ярко наблюдаются при повторении основ тригонометрии. Будем брать такие понятия как синус, косинус (тангенс, котангенс можно вывести из их определения, зная синус и косинус). Умение быстро находить «хорошие» углы от тригонометрических функций. Проекция, основное тригонометрическое тождество, синус двойного угла.

Методика находить «хорошие» углы заключается в следующем. Записываем три числа $\frac{1}{2}, \frac{\sqrt{2}}{2}, \frac{\sqrt{3}}{2}$. Даём задание студентам записать эти числа приближённо в десятичных дробях по возрастанию 0,5; 0,7; 0,9. На доске изображаем декартову систему координат. Рисуем в её начале окружность единичного радиуса (рис. 1). Даём задание студентам повернуть радиус окружности примерно на угол в 30^0, считая поворот радиуса против часовой стрелки для положительных углов и отсчитывая угол также от положительного направления оси x.

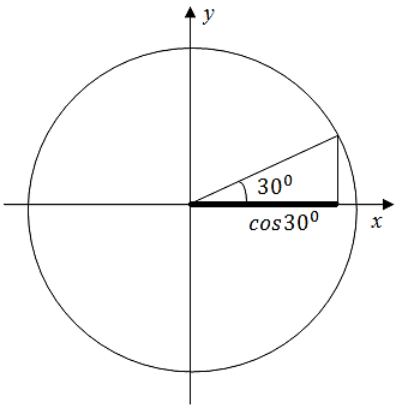


Рис. 1. Единичная окружность

Отложим перпендикуляр на ось x. И договоримся называть расстояние от 0 до точки пересечения перпендикуляра с осью x - косинусом x и обозначаем cos(α). Также называем это расстояние проекцией радиуса на ось x. Подписываем ровно под этой проекцией название $\cos(30^0)$. Задаём вопрос аудитории, чему примерно равно это расстояние из трех чисел $\frac{1}{2}, \frac{\sqrt{2}}{2}, \frac{\sqrt{3}}{2}$. Любой здравомыслящий человек понимает, что длина проекции в данном случае из трех предложенных чисел приблизительно равна $\frac{\sqrt{3}}{2} = 0{,}9$. Следовательно, $\cos(30^0) = \frac{\sqrt{3}}{2}$.

Теперь опустим перпендикуляр на ось y. Назовём расстояние от 0 до точки пересечения перпендикуляра с осью y синусом y и обозначим sin(α). Рядом с этой проекцией подписываем название $\sin(30^0)$. Задаём вопрос аудитории, чему примерно равно это расстояние из трех чисел $\frac{1}{2}, \frac{\sqrt{2}}{2}, \frac{\sqrt{3}}{2}$. Из аудитории говорят, что длина проекции в данном случае приблизительно равна $\frac{1}{2} = 0{,}5$. Следовательно, $\sin(30^0) = \frac{1}{2}$.

CC BY

Данный метод позволяет находить синус или косинус «хороших» углов от 0^0 до 360^0. Практика показывает, что данный метод вызывает интерес и быстрое нахождение студентами синус или косинус того или иного угла, не пользуясь таблицей, и освобождает их от её выучивания.

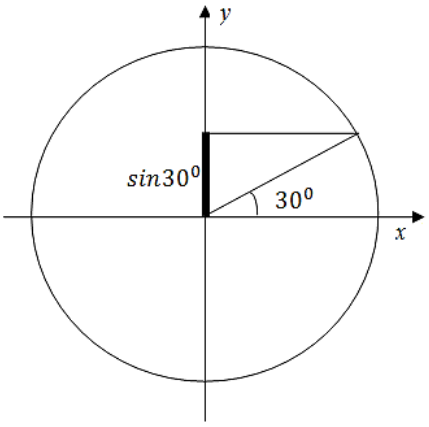


Рис. 2. Единичная окружность

Далее на занятиях ответ о синусах или косинусах «хороших» углов необходимо выдавать буквально автоматически. Конечно, каждый раз рисовать окружность и указывать примерно, где находятся те или иные углы, а также пользоваться всей выше проделанной методикой для ответа на вопрос уже будет нецелесообразно. Но после некоторого количества повторений, для каждого человека разного, у студента формируется образ окружности и вся выше проделанная методика. В данном случае образ помогает тренировать также и память. В последствие, когда у человека сформирован данный навык, он может сам автоматически создавать образы для тех или иных объектов и, поскольку «отследить» за мышлением каждого человека невозможно, да и не нужно, «рождаются» новые образы, которые помогают понимать, о чем идёт речь, и развиваться около этих понятий [1; 3; 5].

Что касается понятия синуса и косинуса, то на данном этапе они также практически полностью были объяснены. Стоит только для примера рассмотреть прямоугольный треугольник со сторонами x и y. Поскольку заранее мы уже сказали, что x=cos(α). Единственное, что нужно добавить, то что определение косинуса есть cos(α)=x/R, а определение синуса sin(α)=y/R. Так как для единичной окружности R=1, то cos(α)=x, а sin(α)=y. Так мы в самом начале и определяли проекции радиуса окружности на оси.

Основное тригонометрическое тождество введём, рассмотрев частный случай. Пусть также дана окружность единичного радиуса. Предложим студентам для начала нарисовать её и ответить на вопрос: «Чему равен $\sin(30^0)$ и $\cos(30^0)$?». Далее рассмотрим прямоугольный треугольник и зададимся вопросом, как найти по этим данным гипотенузу. Многие, как правило, не зная теорему Пифагора, сразу отвечают на вопрос, что $c = 1$, говоря, что гипотенуза в данном случае это радиус единичной окружности. Вместе заключаем, что для абсолютно любого угла, гипотенуза везде равна 1. Ну и поскольку ранее мы условились называть стороны треугольника x= cos(α) и y=sin(α), зная теорему Пифагора, находим $c =$

CC BY

$\sqrt{(\sin(\alpha))^2 + (\cos(\alpha))^2} = \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} = 1.$ Отсюда выводим и основное тригонометрическое тождество как: $1 = (\sin(\alpha))^2 + (\cos(\alpha))^2.$

Введём понятие синуса двойного угла. Возьмём для примера $\sin(60^0)$, который можно записать как $\sin(60^0) = \sin(2 \cdot 30^0)$. Многие уже ответят автоматически, что $\sin(60^0) = \frac{\sqrt{3}}{2}$. Теперь найдём $\sin(30^0) = \frac{1}{2}$ и $\cos(30^0) = \frac{\sqrt{3}}{2}$. Поставим вопрос: «Как связаны $\sin(60^0)$ с $\sin(30^0)$ и $\cos(30^0)$?». Запишем эту связь в виде чисел $\frac{\sqrt{3}}{2} = x \cdot \frac{1}{2} \cdot \frac{\sqrt{3}}{2}$, откуда $x = 2$. То есть $\sin(60^0) = 2 \cdot \sin(30^0) \cdot \cos(30^0)$. Обозначив $\alpha = 30^0$, можно переписать последнее соотношение как $\sin(2 \cdot \alpha) = 2 \cdot \sin(\alpha) \cdot \cos(\alpha)$, которое и называется синусом двойного угла. Что дает этот вид доказательства формулы синуса двойного угла? Оно даёт создание понятия без изначальных формул и букв, чего «боятся» и как следствие не понимают студенты заочного отделения.

Данная методика введения геометрических образов в обучение позволила повысить уровень математической компетентности обучающихся направления «Программная инженерия». Что подтверждает проведённое исследование у студентов, обучающихся по направлению «Программная инженерия» в рамках дисциплины «Математический анализ».

Литература

1. Баранец Н.Г., Верёвкин А.Б. Образы математики. Советские математики о науке. Ульяновск, 2015. 328 с.

2. Битнер Г.Г. Формирование математической культуры в системе подготовки инженеров-приборостроителей: Дисс. ... канд. пед. наук. Казань, 2005 203 с.

3. Далингер В.А. Обучение математике на основе когнитивно-визуального подхода // Вестник Брянского государственного университета. 2011. № 1. С. 299-305.

4. Зверева Е.А. Формирование ИКТ-компетентности бакалавров направления «Приборостроение» в процессе производственной практики: Дисс. … канд. пед. наук. Чебоксары, 2020. 180 с.

5. Исаева З.И., Баймурадова А.А. Деятельностный подход в обучении математике // Известия Чеченского государственного педагогического института. 2018. Т. 21. № 3(23). С. 135-137.

© Коледин В.В., Зверева Е.А., 2022

УДК 378.1
https://doi.org/10.36906/AP-2022/68

Махмутова М.В.
канд. пед. наук
Шевцова М.А.
Магнитогорский государственный технический университет им. Г.И. Носова
г. Магнитогорск, Россия

РАЗРАБОТКА КУРСА ГРАФИЧЕСКОГО ДИЗАЙНА В РАМКАХ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

Аннотация. В настоящее время в основе стратегии современного образования стоит возможность предоставить каждому обучающемуся проявить свой творческий потенциал, раскрыть свою индивидуальность. Дополнительное образование позволяет школьникам получить новые знания, умения и навыки в какой-либо сфере, способствует самореализации и самоопределению ребёнка. В статье рассмотрена программа разработки курса изучения графического дизайна «Диджитал-Арт», которая направлена на формирование и развитие творческих способностей учащихся через изучение основ графического дизайна, работы в сервисах и разработку индивидуальных творческих проектов.

Ключевые слова: дизайнерское мышление; самореализация; дополнительное образование; графический дизайн.

Makhmutova M.V.
Ph.D.
Shevtsova M.A.
Nosov Magnitogorsk State Technical University
Magnitogorsk, Russia

DEVELOPMENT OF A GRAPHIC DESIGN COURSE IN THE FRAMEWORK OF ADDITIONAL EDUCATION

Abstract. Currently, the basis of the strategy of modern education is to provide an opportunity for each student to show their creative potential, to reveal their individuality. Additional education for schoolchildren allows them to gain new knowledge, skills and abilities in any area, contributes to the self-realization and self-determination of the child. The article discusses the program for the development of a course in the study of graphic design "Digital - Art", which is aimed at the formation and development of creative abilities of students through the study of the basics of graphic design, work in services and the development of individual creative projects.

Keywords: design thinking; self-realization; additional education; graphic design.

CC BY

Главная цель современной системы дополнительного образования детей – это создание методических и технических возможностей для получения знаний, а также условий для творческой самореализации личности ребёнка. В настоящее время можно наблюдать большое количество методик и программ дополнительного образования, но популярность с каждым годом набирают авторские программы развития от молодых профессионалов, которые содержат инновационные подходы и формы проведения занятий [6].

Система дополнительного образования формирует у обучающихся ощущение ценности личности, самостоятельность, креативность, способность мыслить в различных ситуациях и находить выход из них. В рамках дополнительного образования учащиеся могут развивать свои способности, заниматься тем, что ему действительно интересно, а также усваивать материал в том ритме, который удобен ему, основываясь на индивидуальные особенности каждого школьника [7].

Так же следует отметить, что система дополнительного образования помогает выстраивать:

– межличностные отношения между обучающимися;

– положительную эмоциональную атмосферу;

– положительную динамику развития самореализации и самоопределения обучающихся.

Все эти факторы в совокупности влияют на развитие творческого потенциала детей, так как в большей степени ориентировано на удовлетворение их интеллектуальных потребностей и решения задач их творческого становления за пределами школьного учебного времени.

Основной целью курса «Диджитал-Арт» по изучению основ графического дизайна является формирование у детей творческого мышления. Таким образом, в рамках этого курса школьники научатся проводить исследование, смогут развить свои творческие навыки, проявить своё воображение и креативность, а также разработать собственные проекты [1].

Основные задачи данного курса:

– повысить графическую грамотность учащихся, способствовать формированию творческого, технического мышления и пространственных представлений, творческого воображения, эстетического восприятия и вкуса;

– ознакомить с основными методами художественного проектирования;

– ориентировать подростков в сферу рынка труда современных творческих профессий (графический дизайнер, веб-дизайнер и т. п.);

– сформировать определенные умения в художественном конструировании;

– сформировать умения работы в графических программах;

– сформировать умение разработки индивидуальных проектов и защиты их.

Следует отметить, что новизна данного курса заключается в том, что основной идеей курса является формирование у учащихся дизайнерского мышления, а также работа над индивидуальным проектом с получением комментариев и обратной связи, разработкой собственных кейсов [2].

Актуальность разработки данного курса формируется на основе таких аспектов современного общества как:

CC BY

– направленность на индивидуальный подход в образовании;

– формирование новой, современной культуры;

– решение проблемы самоопределения и самореализации школьников;

– оформление новых областей материального производства и современных технологий, новых актуальных профессий творческой направленности.

Основные формы и методы программы: данный курс рассчитан на теоретические и практические занятия. Они строятся на основе обучающей методики. В основу реализации методики положен индивидуальный подход. Образовательно-воспитательный процесс ориентирован на выявление творческих способностей учащихся, их самореализацию и самоопределение [3].

Форма организации деятельности обучающихся на занятии: фронтальная (беседа, объяснение), групповая форма используется при изучении нового материала, его обсуждении, а также индивидуальная форма обучения, которая будет использована в процессе выполнения домашних заданий, работой над собственными проектами [4].

Наряду с традиционными методами (объяснительно-иллюстративным, репродуктивным, частично-поисковым) в рамках курса также применяется метод проекта, так как практически в каждом разделе учащимся придётся заниматься разработкой индивидуальных проектов [5].

Также следует отметить, во-первых, графический дизайн чрезвычайно динамичен по своей природе: проблемы, темы, идеи, задачи, с которыми сталкивается дизайнер, меняются очень быстро в других видах дизайна. Таким образом, в процессе обучения учащиеся должны получить навыки взаимодействия человека с миром. Во-вторых, графический дизайн – язык коммуникаций, который предполагает, что дизайнер способен быстро адаптироваться в сложных и новых для себя ситуациях [8].

К каждому разделу в курсе будут приложены видеоматериалы, записанные с помощью технологии скринкастинга, презентации с авторскими разработками, а также материалы для самостоятельного изучения. После выполнения практического задания учащиеся получают обратную связь от преподавателя. По окончании курса обучающимся необходимо защитить свой проект, который состоит из выполненных работ и кейсов.

Для наглядности представления разделов курса графического дизайна «Диджитал-Арт» была разработана ментальная карта. Схема разделов курса представлена на рисунке 1.

Возраст детей, участвующих в реализации данной образовательной программы – 14–16 лет.

Программа рассчитана на один год – 144 часа. Режим занятия строится следующим образом: занятия проводятся два раза в неделю по два академических часа.

CC BY

Рис. 1. Схема структуры курса

Содержание программы:

I. Подготовительный модуль. Искусство дизайна:

1. Теория. Графический дизайн как понятие;
2. Теория. Виды графического дизайна;
3. Теория. Графический дизайн в современном мире.

II. Основы графического дизайна:

4. Теория. Композиция в дизайне. 10 правил композиции;
5. Практика. Отработка правил композиции;
6. Теория. Основы цветоведения. Главные характеристики цвета. Колористика;
7. Теория. Шрифтовой дизайн;
8. Практика. Составление цветовых сочетаний.

III. Бренд-дизайн и фирменный стиль:

9. Теория. SMM-стратегия;
10. Практика. Создание SMM-стратегии;
11. Теория. Бренд-стратегия;
12. Практика. Разработка бренд-стратегии;
13. Теория. Контент-стратегия;
14. Практика. Работа над разработкой контент-стратегии;
15. Теория. Упаковка;
16. Практика. Создание упаковки будущего аккаунта;
17. Практика. Работа над визуалом аккаунта.

CC BY

IV. Работа в онлайн-сервисе «Canva»:

18. Теория. Базовые принципы работы в онлайн-сервисе «Canva». Обзор основных инструментов и функционала;

19. Практика. Оформление социальных сетей;

20. Практика. Работа над индивидуальным проектом.

V. Работа в Adobe Photoshop:

21. Теория. Базовые принципы «Adobe Photoshop», обзор основных инструментов и функционала;

22. Практика. Работа с интерфейсом и функционалом программы;

23. Практика. Создание коллажа;

24. Практика. Разработка фирменного логотипа.

VI. Работа в Figma:

25. Теория. Основы работы в «Figma»;

26. Практика. Работа с интерфейсом и функционалом программы. Основные компоненты;

27. Практика. Создание интерфейса сайта.

VII. Инфографика:

28. Теория. Основы построения инфографики;

29. Практика. Обзор основных средств и инструментов для создания инфографики;

30. Практика. Создание инфографики в рамках индивидуального проекта.

VIII. Насмотренность в дизайне:

31. Теория. Что же такое насмотренность в дизайне;

32. Практика. Работа с референсами;

33. Практика. Работа с мудбордами.

IX. Разработка портфолио. Защита проектов:

34. Практика. Разработка идеи;

35. Практика. Создание портфолио. Сбор материалов;

36. Практика. Презентация портфолио. Коллективная оценка. Защита проекта.

Оценочные материалы: основным видом контроля результатов освоения учащимися дополнительных образовательных программ является итоговый контроль в виде защиты проекта, который разрабатывается в течение всего времени прохождения курса [9].

Критерии оценивания проектов учащихся:

– владение компьютерными дизайн-программами;

– аккуратность подачи;

– структурность и логичность;

– оригинальность проекта;

– стилистика проекта;

– композиционное решение;

– полнота раскрытия выбранной тематики.

CC BY

Баллы выставляются преподавателем и членами комиссии. Максимальный балл за выполнение критерия составляет 10 баллов, минимальный – 1 балл.

Используемые формы при оценивании результатов обучения:

– творческие выступления;

– открытое занятие для преподавателя и родителей, выполненное в форме творческой реализации.

Таким образом, дизайн-мышление необходимо школьнику для развития способности быстро находить рациональное, творческое или новаторское решение поставленных задач или в сложных для него ситуациях. Курс «Диджитал-Арт» спроектирован в рамках художественного направления и направления технического творчества в системе дополнительного образования. В настоящее время около 45% всех школьников занимаются разными видами образовательной деятельности в свободное время. Дополнительное образование в сфере графического дизайна помогает раскрыть и развивать творческий потенциал учащихся, так как оно ориентировано на удовлетворение их интеллектуальных потребностей и решения задач их творческого становления и развития за пределами школьного учебного времени.

Литература

1. Адамс Ш. Словарь цвета для дизайнеров. М.: КоЛибри, 2018. 272 с.

2. Алексеев А.Г. Дизайн-проектирование. М.: Юрайт, 2020. 91 с.

3. Ваулина О.Ю., Постовалова А.С., Щукин А.Д., Махмутова М.В. Информационные технологии в образовании // Интеллектуальные информационные системы: тенденции, проблемы, перспективы: сб. научных статей 8-й Международной научно-практической конференции «ИИС-2020» (г. Курск, 18 декабря 2020 года). Курск, 2020. С. 41-43.

4. Лаврентьев А.Н. Цифровые технологии в дизайне. История, теория, практика. М.: Юрайт, 2020. 209 с.

5. Литвина Т.В. Дизайн новых медиа: учебник для вузов. М.: Юрайт, 2019. 181 с.

6. Махмутова М.В. Непрерывное образование как условие развития современной цифровой экономики // Актуальные проблемы современной науки, техники и образования. тезисы докладов 76-й Международной научно-технической конференции (г. Магнитогорск, 16-20 апреля 2018 года). Магнитогорск, 2018. С. 462.

7. Махмутова М.В., Сеничева Е.И., Акимова О.А. Технология разработки и применения электронных образовательных ресурсов в учебном процессе вуза // Открытое образование. 2019. Т. 23. № 6. С. 50-58.

8. Павловская Е.Э. Графический дизайн. Современные концепции. М.: Юрайт, 2020. 120 с.

9. Павловская Е.Э. Основы дизайна и композиции: современные концепции. М.: Юрайт, 2020. 120 с.

© Махмутова М.В., Шевцова М.А., 2022

УДК 378.1
https://doi.org/10.36906/AP-2022/69

Махмутова М.В.
канд. пед. наук
Давлеткиреева Л.З.
канд. пед. наук
Алчинова Д.С.
Магнитогорский государственный технический университет им. Г.И. Носова
г. Магнитогорск, Россия

ПРОЦЕСС СОЗДАНИЯ ФИЛЬТРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ НА ПРИМЕРЕ «SPARK AR STUDIO» ДЛЯ ПРОДВИЖЕНИЯ КОМПАНИИ В СОЦИАЛЬНЫХ СЕТЯХ

Аннотация. В статье рассматриваются вопросы создания фильтра дополненной реальности (AR) на примере создания «маски» для официального аккаунта студенческого спортивного клуба «Стальные сердца» в социальной сети «Instagram», привлечения внимания целевой аудитории к бренду путем внедрения технологии дополненной реальности в процесс продвижения организации. В процессе создания фильтра использована техника отслеживания положения головы человека в пространстве на основе анализа видеокадров, и последующее наложение текстур на выделенную область в реальном времени. Предложенные в статье методы создания и настройки AR-фильтров позволили увеличить количественные показатели активности (количество просмотров stories аккаунта, число отметок в stories, количество новых подписчиков) аккаунта в социальной сети «Instagram».

Ключевые слова: инструкция; продвижение бренда; маркетинг; социальные сети; тренды; виртуальная реальность; современные технологии.

Makhmutova M.V.
Ph.D.
Davletkireeva L.Z.
Ph.D.
Alchinova D.S.
Nosov Magnitogorsk State Technical University
Magnitogorsk, Russia

THE PROCESS OF CREATING AN AUGMENTED REALITY FILTERON THE EXAMPLE OF «SPARK AR STUDIO» FOR COMPANY PROMOTION ON SOCIAL MEDIA

Abstract. The article discusses the create an augmented reality (AR) filter based on the example of creating a “mask” for the official account of the student sports club “Steel Hearts” in the social

CC BY

network “Instagram”, attracting the attention of the target audience to the brand by introducing augmented reality technology in the process of promoting the organization. In the process of creating the filter, a technique was used to track the position of a person's head in space based on the analysis of video frames, and the subsequent imposition of textures on the selected area in real time. The methods for creating and configuring AR-filters proposed in the article allowed to increase the quantitative indicators of activity (the number of stories views of the account, the number of stories marks, number of new subscribers) of the Instagram account.

Keywords: instruction; brand promotion; marketing; social networks; trends; virtual reality; modern technologies.

Социальные сети являются распространённой площадкой для коммуникации с целевой аудиторией [6]. Современные компании ведут активную работу в социальных сетях. Формат «Инстаграма» на сегодняшний день является наиболее актуальным, а сама платформа ежегодно предоставляет все больше возможностей.

«Инстаграм» – интернет-платформа, с помощью которой можно делиться со своей целевой аудиторией фотографиями и короткими видеозаписями [2]. Однако часто SMM-отдел организации не в полной мере использует возможности платформы, из-за чего теряет большое число потенциальных клиентов, которых можно было бы привлечь активным продвижением в сети Интернет и поддержкой актуальных трендов.

В мае 2017 года «Инстаграм» ввёл в оборот маски-фильтры для своих пользователей. Вскоре эта функция стала платформой для коллаборации с известными брендами и артистами, благодаря чему стала популярным способом продвижения в сети Интернет. На сегодняшний день маски – тренд, привлекающий новых подписчиков, а как следствие – увеличение клиентуры и чувство «причастности» уже имеющихся подписчиков.

Для решения ситуации неэффективного использования социальной сети «Инстаграм» в качестве платформы для продвижения бренда, предлагается разработать фирменный AR-фильтр. Под AR-фильтром подразумевается фильтр-«маска» с использованием дополненной реальности (англ. «augmented reality», AR) – результат введения в поле восприятия сенсорных данных с целью дополнения сведений об окружении и улучшения восприятия информации [3].

Для создания, настройки и тестирования фильтров выбрана «Spark AR Studio» – платформа дополненной реальности для операционных систем «Mac» и «Windows», которая позволяет создавать AR-эффекты в мобильной камере.

Целью создания и публикации AR-фильтров является привлечение новых клиентов, повышение активности постоянных клиентов и привлечение внимания к бренду [8]. Используя «маску», пользователи смогут записать видео или сделать фото с ними и выложить на своей странице. При публикации материалов с её использованием подписчики пользователя смогут увидеть автора маски, перейти на его аккаунт или тоже воспользоваться фильтром. Маска будет доступна всем желающим пользователям, открывшим аккаунт автора

CC BY

маски в «Инстаграм» [9]. Найти AR-фильтр также будет возможно через поиск по фильтрам в официальном приложении социально сети.

Рассмотрим подробнее процесс создания AR-фильтра на примере маски для аккаунта студенческого спортивного клуба. Маска будет представлять собой логотипы спортивных сборных клуба, сменяющиеся по нажатию на экран. Фильтр включает в себя 3 слоя: слой ретуши, слой коррекции и слой с логотипом. Слой ретуши предназначен для разглаживания кожи, слой коррекции отвечает за увеличение глаз и сужение подбородка. Все изменения имеют исключительно естественный косметический характер. Каждый из слоёв создавался и настраивался вручную.

Необходимое программное обеспечение:

1. Платформа дополненной реальности «Spark AR Studio» версии 86 или выше, скаченная с официального сайта продукта;

2. Пакет настроек «Face Distortion and Retouching» от «Spark AR Studio», скаченный с официального сайта продукта;

3. Пакет настроек «Face reference assets classic» от «Spark AR Studio», скаченный с официального сайта продукта;

4. Графический редактор «Adobe Photoshop» версии CS6 или выше (или другой графический редактор, поддерживающий работу с PNG форматом);

5. Приложение «Instagram» на мобильном устройстве, скаченное с App Store или Google Play (в зависимости от платформы телефона).

Перед началом работы необходимо скачать и установить «Spark AR Studio» и графический редактор на персональный компьютер в соответствии с инструкцией, предложенной при запуске установочного файла программы. Пакеты настроек для «Spark AR Studio» рекомендуется разместить в папке с программой. Пакеты необходимо разархивировать. Пакеты не требуют установки и необходимы для дальнейшей работы в программе.

Этап 1. Создание проекта. Запустите «Spark AR Studio». Нажмите «Blank project» для создания пустого проекта. Для сохранения проекта нажмите «File» > «Save». Выберете путь для сохранения, сохраните под понятным для Вас именем. Не забывайте сохранять проект после его изменения, нажав «File» > «Save».

Этап 2. Настройка рабочей области. Рабочую область программы можно условно разделить на 3 части. С левой стороны находится дерево проекта, где будут отображать созданные автоматически и добавленными нами настройки маски. Справа находится область настройки. По середине – рабочее пространство. Для демонстрации работы маски в рабочей области находится симулятор камеры телефона. При необходимости можно изменить настройки симулятора для просмотра маски на на конкретном девайсе. Также можно изменить внешность человека в симуляторе, что помогает смоделировать ситуацию исопльзования готовой маски на различных типах лица и кожи.

Этап 3. Создание основы для маски. Нажав правой кнопкой мыши на дерево проекта выберите функцию «Add», затем из выпадающего списка «Face Tracker». Как понятно из

CC BY

названия объекта, он необходим для отслеживания положения лица пользователя в камере [4]. После добавления «Face Tracker» в рабочей области проекта можно заметить, что на лице человека из симуляции появилась система координат.

Этап 4. Базовая ретушь лица. Для того, чтобы усовершенствовать маску необходимо, чтобы маска, при использовании, частично «убирала» недостатки кожи пользователя. Нажмите правой кнопкой мыши на созданный ранее «Face Tracker» > «Add» > «Face Mesh». Убедитесь, что созданный объект в дереве проекта находится под «Face Tracker». Если объект успешно создан, то на симуляции Вы увидите «маску» с шахматной раскраской. Для удобства работы с объектами рекомендуется давать им интуитивные названия, отображающие цель создания. Нажав дважды на созданный «Face Mesh» (или нажав на него правой кнопкой и выбрав пункт «Rename») дайте объекту название «Ретушь». Выберите объект «Ретушь» в дереве проекта. В настройках объекта (правая часть экрана) нажмите на «+» возле пункта «Material». Таким образом мы на созданную ранее маску «надели» материал, который выбрался по умолчанию и отобразился в настройках объекта и на симуляции.

Все используемые материалы отображаются в левой нижней части экрана в разделе «Assets» в папке «Materials». Для удобства переименуем материал также, как переименовали. Выбрав материал «Ретушь» просмотрим настройки материала в правой верхней части экрана. Для пункта «Shader Type» установим значение «Retouching». После смены типа шейдера маска станет «прозрачной». Теперь в правой части стал доступен пункт «Skin smoothing». Данная настройка отвечает за «сглаживание» кожи пользователя. Установите необходимый процент сглаживания, просматривая изменения на симуляторе. Рекомендуемый процент для естественной ретуши лица – 15–30%.

Этап 5. Коррекция лица. Ещё одним обязательным пунктом актуальных масок является коррекция лица [1]. Для этого создадим ещё один «Face Mesh», аналогично пункту ранее, и назовём его «Коррекция». Новый объект в дереве должен также находиться под «Face Tracker». Порядок масок между собой не важен. В настройках объекта «Коррекция» нажмите «+» рядом с пунктом «Deformation». В появившемся окне откройте файл «faceDistortionPack.fbx», находящийся по следующему пути: Face Distortion and Retouching\Face_Distortion_Finished\objects\faceDistortionPack. После применения пакета настроек маска станет прозрачной, а в настройках появятся новые пункты. Меняя процент появившихся пунктов настройки, можно изменить размер глаз, носа, рта, а также откорректировать форму лица. Рекомендуемые для естественной коррекции лица проценты: глаза (eye_l_scale, eye_r_scale) +10%, рот (mouth_scale) +5%, овал лица (triangle) +10%.

Этап 6. Тестирование фильтра. Проведите предварительное тестирование фильтра для дальнейшей работы. В дальнейшем тестирование можно применять на любой стадии разработки для проверки работоспособности фильтра. Для тестирования необходимо создать ссылку доступа к проекту. В левой нижней части нажмите на значок телефона, далее нажмите «Send» у пункта «Instagram Camera». Спустя некоторое время программа сгенерирует ссылку, перейдя по которой с мобильного телефона, фильтр автоматические откроется через приложение «Instagram». Доступ к тесту может получить любой пользователь, имеющий

CC BY

ссылку. Ссылка не имеет ограничений по времени или количеству переходов. При необходимости, после тестирования фильтра на себе, Вы можете вернуться к редактированию созданных ранее «face Mesh» и изменить настройки.

Этап 7. Создание новой текстуры для фильтра. Для следующего этапа необходимо в графическом редакторе открыть файл «faceFeminine» или «faceMasculine». Файл находится по следующему пути: Face-reference-assets-classic\FaceAssets\Textures. Данная текстура представляет собой «развёрнутую» текстуру лица, которая в дальнейшем будет «надета» на следующий «Face Mesh». Откройте картинку, которую хотите отобразить на лице пользователя в графическом редакторе. Картинка должна быть в формате PNG, не должна иметь фона и иметь хорошее качество. При необходимости фон можно удалить инструментом «Волшебный ластик». Перетащите выбранную картинку на текстуру и разместите в нужном месте лица. Например, на щеке пользователя под правым глазом. Область для размещения была выбрана таким образом, чтобы пользователь мог рассмотреть логотип, но при этом он не сильно перекрывал лицо. Данная область лица чаще всего используется на платформе «Инстаграм» для подобных масок. Одновременно на одной текстуре можно разместить не ограниченное количество объектов в т.ч. текстового формата.

Перед сохранением новой текстуры необходимо скрыть слой с лицом. Для этого в правой части нажмите на изображение глаза рядом со слоем «Фон». Убедитесь, что на текстуре в рабочей области графического редактора остался только добавленный Вами объект. Нажмите вкладку «Файл» в левом верхнему углу, в меню выберите «Сохранить как…». Дайте новой текстуре соответствующее название, выберете тип файла PNG и сохраните (желательно сохранять текстуры в папке с проектом).

Этап 8. Наложение новой текстуры. Для добавления созданной текстуры необходимо добавить ещё один «Face Mesh», разместить его под «Face Tracker» и дать новое имя. Выбрав новый «Face Mesh» в дереве проекта нажмите «+» > «Create New Material». В разделе «Assets» появится новый материал, дайте ему такое же имя, как у нового «Face Mesh». В настройках материала выберите шейдер «Face Paint». Маска станет «полупрозрачной». Нажмите «Add Asset» в нижней части раздела «Assets» > «Import From Computer…» и загрузите новую текстуру. Текстура отобразится в папке «Textures». В настройках материала в разделе «Texture» выберете нужную текстуру. Простая маска готова. Протестируйте её и, при необходимости, внесите изменения. Если картинка оказалась «не на месте» – зайдите в графический редактор и поменяйте расположение объекта на текстуре.

Этап 9. Смена изображение по нажатию. Для того, чтобы разнообразить готовый фильтр, можно сделать несколько изображений в одном фильтре, сменяющиеся по нажатию пользователя на экран. Данный шаг облегчит поиск нужного AR-фильтра пользователем, а загрузка и распространение готовой маски на платформе «Инстаграм» займёт меньше времени. Проделайте шаги 7–8 и создайте столько «Face Mesh» с Вашими текстурами, сколько хотите добавить в фильтр. Желательно не перегружать фильтр и не добавлять более 5 масок (не считая ретуши и коррекции, т.к. они не будут меняться и всегда останутся «включены»).

CC BY

При создании новых текстур они будут «накладываться» одна на другую. Для корректного отображения и удобства работы снимите галочку видимости у всех текстур, кроме «Ретушь», «Коррекция» и той текстуры, с которой вы работаете в данный момент. Когда добавлены все текстуры и настроено их отображение нужно добавить область работы с патчами. Для этого в меню выберем пункт «View» > «Show/Hide Patch Editor». В нижней части экрана появится новая область Patch Editor». Нажмите на ней правой кнопкой мыши. В появившемся окне вбейте в поисковой строке «Screen Tap» и добавьте его в область нажатием кнопки «Add Patch». Данный патч будет отвечать за нажатие пользователем на экран.

Для того, чтобы фильтр смог подсчитать число нажатий на экран, необходимо добавить счётчик нажатий. Аналогично предыдущему патчу ищем и добавляем патч «Counter». Для смены рисунка на маске необходимо добавить ещё один патч, ответственный за смену опций. Найдите и добавьте патч «Option Sendler». Патчи «Screen Tap» и «Counter» соединились автоматически. Для того, чтобы соединить их с «Option Sendler» нажмите на стрелку на выходе «Counter» и перетащите её к «Option Sendler». Для счетчика установите максимум (Maximum Count), равный количеству «Face Mesh» (не считая «Ретушь» и «Коррекция»). Для патча «Option Sendler» установите логический тип. Для этого нажмите на патч и выберете из появившегося внизу выпадающего списка тип «Boolean». Поставьте галочку напротив «Value». Т.к. маска имеет только 2 значения – вкл/выкл, следовательно именно логическое значение будет передаваться фильтру.

В настройках каждого «Face Mesh» (не считая «Ретушь» и «Коррекция») рядом с отметкой видимости нажмите стрелку. Строка станет жёлтой, а объект появится в виде патча в области патчей. Появившиеся патчи необходимо соединить с «Option Sendler». От последовательности соединения будет завесить порядок отображения масок при нажатии. Убедитесь, что галочка видимости стоит только на одном патче, чтобы избежать наложений. Маска со сменой изображений готова. Сохраните изменения и протестируйте фильтр. Представим эффективность решения на конкретном примере. AR-фильтр для студенческого спортивного клуба был протестирован и опубликован на аккаунте клуба в социальной сети «Инстаграм» [7].

Используя встроенную статистику официального приложения «Инстаграм» через 7 дней после внедрения фильтра, были увеличены количественные показатели [5], а именно:

– среднее количество взаимодействий с stories (ответы, посещение профиля, нажатия на наклейки) увеличено на 37% (с 50 до 80);

– среднее число аккаунтов, охваченных stories увеличено на 15% (с 850 до 1000);

– число взаимодействий с контентом увеличено на 30%;

– число подписчиков увеличено на 25%;

– число stories с отметкой аккаунта увеличено на 70% в первые 3 дня и на 45% в последующие 4 дня.

Высокая активность аккаунта и приток новых подписчиков повышает внимание к бренду и продвигает аккаунт в социальной сети, увеличивая вероятность попадания в рекомендации целевой аудитории организации.

CC BY

Таким образом, описанный в статье процесс по созданию и тестированию фильтров дополненной реальности для социальной сети «Инстаграм» и ряд включённых в неё рекомендаций протестирован на реальном примере. Применение AR-фильтра привело к привлечению внимания к бренду, повысило активность на странице компании и увеличило число новых подписчиков. В связи с популярностью платформы и высокой вариативностью использования фильтров с дополненной реальностью, описанные в статье процессы и рекомендации могут стать частью развития ИТ-инфраструктуры любой современной организации.

Литература

1. Leaver T., Highfield T., Abidin C. Instagram: Visual social media cultures. Cambridge, UK; Medford, MA: Polity, cop. 2020. XIII, 264.

2. Анурина Т.М., Остахова Ю.А. Визуальное изменение культуры в эпоху глобализации на примере приложения INSTAGRAM // Культура, наука и искусство – современные векторы развития вуза культуры: мат-лы Международной научно-практической конференции (г. Орел, 16 октября 2019 года). Орел, 2019. С. 72-74.

3. Белова О.П., Коткина М.В., Казнин А.А., Ананьин А.В., Софронова Т.В. 3D-моделирование и дополненная реальность. Архангельск: САФУ, 2018. 89 с.

4. Дубова Ю.И., Заруднева А.Ю. Брендинг как маркетинговый инструмент повышения конкурентоспособности компании. Волгоград: ВолгГТУ, 2019. 76 с.

5. Костиков В.Ю. Медиакоммуникации в продвижении бренда спортивного клуба: автореф. дисс. … канд. филол. наук. Воронеж, 2018. 26 с.

6. Kurvaeva L.V., Gavrilova I.V., Mahmutova M.V., Chichilanova S.A., Povituhin S.A. Development of knowledge base of intellectual system for support of formal and informal training of IT staff // Journal of Physics: Conference Series. IOP Publishing. 2018. Vol. 1015. No. 4. P. 042013.

7. Махмутова М.В., Махмутов Г.Р. Приоритет бизнеса в управлении ИТ на промышленном предприятии // Актуальные проблемы современной науки, техники и образования: сб. тезисов докладов 77-й международной научно-технической конференции (г. Магнитогорск, 22-26 апреля 2019 года). Том 1. Магнитогорск, 2019. С. 435.

8. Халилов Д. Маркетинг в социальных сетях. М.: Манн, Иванов и Фербер, 2017. 228 с.

9. Юров С.С., Дымова С.С. Бренд-стратегия. М.: Институт бизнеса и дизайна, 2020. 208 с.

© Махмутова М.В., Давлеткиреева Л.З., Алчинова Д.С., 2022

УДК 005+004.4’2
https://doi.org/10.36906/AP-2022/70

Меметова Ф.С.
ORCID: 0000-0002-8770-4830, канд. пед. наук
Крымский инженерно-педагогический университет им. Февзи Якубова
г. Симферополь, Россия

ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Аннотация. В статье предложены возможные формы и методы обучения программной инженерии. В работе проанализированы работы учёных, которые также рассмотрели и предложили теоретико-методологические компоненты системы обучения по программной инженерии. Также в статье рассмотрены и проанализированы общепрофессиональные компетенции по дисциплине «Программная инженерия». Рассмотрена форма виртуального общения в рамках дистанционного управления курсом «Программная инженерия». В работе предложены инструменты и средства обучения, используемые на занятиях по дисциплине «Программная инженерия».

Ключевые слова: формы; методы; обучение; студент; программная инженерия; дисциплина; компетенции.

Memetova F.S.
ORCID: 0000-0002-8770-4830, Ph.D.
Crimean Engineering and Pedagogical University the named after Fevzi Yakubov
Simferopol, Russia

FORMS AND METHODS OF TEACHING SOFTWARE ENGINEERING

Abstract. The article suggests possible forms and methods of teaching software engineering. The work analyzes the work of scientists who also considered and proposed theoretical and methodological components of the software engineering training system. Also, the article discusses and analyzes general professional competencies in the discipline “Software Engineering”. The form of virtual communication in the framework of remote control of the course “Software Engineering” is considered. The paper proposes tools and teaching aids used in the classroom on the discipline “Software Engineering”

Keywords: forms; methods; teaching; student; software engineering; discipline; competencies.

Под *методом* обучения понимается упорядоченный способ организации совместной деятельности субъектов образовательного процесса [преподавателя и обучающегося или группы обучающихся], направленный на усвоение содержания образования, общее и профессиональное развитие личности будущего специалиста. Метод обучения

CC BY

характеризуется тремя признаками: обозначая цель обучения, способ усвоения, характер взаимодействия субъектов обучения. Методы обучения разделяют на [1]:

1) объяснительно-иллюстративный, или информационно-рецептивный (рассказ, школьная лекция, объяснение, работа с учебником, демонстрация и др.) – учитель сообщает материал, ученики воспринимают;

2) репродуктивный (воспроизведение знаний и способов действий, деятельность по алгоритму , программе и т. д.) – ученик выполняет действия по образцу, предоставленному учителем;

3) проблемное обучение – учитель ставит перед учащимися проблему и демонстрирует пути её решения; ученики следят за логикой решения проблемы, получают образец развёрнутого познания;

4) частично-поисковый или эвристический – учитель разделяет проблему на части, ученики осуществляют отдельные шаги по решению подпроблем;

5) исследовательский, метод проектов – поисковая творческая деятельность учащихся на решение новых для них проблем.

Работы многих учёных посвящены рассмотрению и применения различных форм и методов обучения при проведении занятий по программной инженерии, в том числе работы А.И. Безверхого, Е.Ю. Андиева, С.В. Борисова, Е.В. Полуэктовой [1-3]. На основе образовательного стандарта (ФГОС 3++, приказ МО №922 от 19.09.2017) дисциплина «Программная инженерия» включена в учебный план направления подготовки 09.03.03 Прикладная информатика, формирующая общепрофессиональные компетенции (ОПК), описанные в таблице.

Таблица

Схема компетенций по программной инженерии

Дисциплина	Компетенции	Формулировка
Программная инженерия	ОПК-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.
	ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью.
	ОПК-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.
	ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения.
	ОПК-8	Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.

На основе формируемых компетенций для методики обучения программной инженерии студентов направления подготовки 09.03.03 Прикладная информатика были выбраны следующие методы:

Объяснительно-иллюстративный – проведение лекций, семинаров, самостоятельных работ.

CC BY

С учётом целей и места в учебном процессе различают лекции вводные, установочные, текущие, обзорные и заключительные.

Методическая разработка учебной *лекции* по дисциплине «Программная инженерия» включает: 1) Тема; 2) Цель; 3) Содержание (план лекции, материал лекции); 4) Средства (MS Power Point, Moodle); 5) Дидактические материалы (слайд-шоу); 6) Источники.

Процесс проведения лекции делится на три основных этапа: 1) ознакомление с темой и планом лекции (10 мин.); 2) выступление лектора, с использованием слайд-шоу, Moodle(для связи через интернет в реальном времени) (55 мин); 3) вопросы и обсуждение материала лекции (15 мин).

Методическая разработка *семинарского занятия* по дисциплине «Программная инженерия» включает: 1) Тем; 2) Цель; 3) Содержание (план семинарского занятия); 4) Средства (MS Power, Moodle, Интернет); 5) Дидактические материалы (слайд-шоу); 6) Источники.

Процесс проведения семинарского занятия делится на четыре этапа: 1) ознакомление с темой занятия (10 мин); 2) выступление преподавателя – представление студенческих проектов аудитории (10 мин); 4) выступление студента (студентов) – защита проектов с использованием слайд-шоу, Moodle (для связи с другими студентами команды в режиме реального времени), Moodle (если требуется показать материалы выложенные на сайте) (45 мин) ; 4) обсуждение представленных проектов (15 мин).

Репродуктивный – выполнение лабораторных практикумов. Методическая разработка *лабораторного занятия* по дисциплине «Программная инженерия» включает: 1) Тема; 2) Цель; 3) Содержание (методические указания по выполнению лабораторной работы); 4) Средства (Moodle, Интернет, IBM Rational Architect); 5) Дидактические материалы (учебное пособие «Проектирование и разработка программного обеспечения»); 6) Источники.

Процесс проведения лабораторного занятия делится на три этапа: 1) ознакомление с темой занятия (10 мин); 2) выполнение сквозного проекта (40 мин.) – студент выполняет задание с использованием средств разработки IBM Rational Architect, Java и выкладывает по необходимости результаты в систему Moodle в курс «Программная инженерия» для проверки результатов преподавателем; 3) выполнение индивидуального проекта (аналогично выполнению сквозного проекта) (30 мин.).

Метод проектов – выполнение индивидуальных, командных, сквозных проектов. Также при обучении студентов технологии разработки программного обеспечения, используются *слуховой, зрительный и кинетический методы*.

1) Слуховой метод даёт наименее желаемые результаты. Когда студент слышит концепцию программирования и проектирования, он в наименьшей степени усваивает материал. Применение слухового метода при обучении студентов используется в сочетании с визуальными методами.

2) Использование проектора, слайд-шоу и других визуальных методов помогут сделать занятия более приемлемым и понятным для студентов. К наиболее широко используемым визуальным методам визуального является использование учебника. Однако, с помощью

CC BY

учебников студенты не могут получать обратную связь или составить практические шаги, необходимые для решения проблемы.

3) Кинетический метод, является третьим подходом, который может использоваться для обучения студентов программированию и проектированию. При применении кинетического обучения студенты принимают активное участие в процессе обучения и общий уровень интереса студентов увеличивается. Повышение уровня интереса приводит к повышению качества обучения. Поэтому, при сочетании теории и практики во время обучения объектно-ориентированного языка Java, студенты обучаются программированию проще и быстрее. Также обнаружили, что лекции без испытания на практике дают более низкие результаты при выполнении практических, домашних заданий и экзаменов.

Интернет-ориентированное преподавание – разработанная методическая система кроме использования традиционных форм обучения: *лекция, лабораторное занятие, семинар,* основана на использовании виртуального общения, что позволит сделать получение образования интересным занятием.

Системы онлайнового обучения (e-learning systems) могут зависеть от двух составляющих – места и времени. В зависимости от значений, которые принимают эти переменные, возможны четыре варианта реализации таких систем:

1. В одном и том же месте, в одно и то же время возможно взаимодействие, называемое «лицом к лицу» (face-to-face – F2F); этот вариант требует применения системы групповой поддержки для решения задач на занятии под руководством преподавателя/инструктора; также в этом случае предполагается взаимодействие инструктора со студентами, студента со студентом в режиме реального времени;

2. Синхронное взаимодействие пользователей, находящихся в разных местах, посредством программного обеспечения; примером использования этого варианта может быть видеотрансляция по сети лекции или проведение практического занятия со студентами, распределёнными по нескольким аудиториям, возможно в разных географических местах;

3. Асинхронное взаимодействие пользователей, находящихся в одном и том же месте; использование в классе системы групповой поддержки в любое удобное для студента время;

4. Асинхронное взаимодействие пользователей, находящихся в разных местах, с использованием необходимого программного обеспечения; в данном случае подразумевается, что учебный материал доставляется в разное время, возможно, по требованию студента.

Выше рассмотренное форма виртуального общения в обучении в пространственно-временных вариантах с использованием инструментов для методический системы по программной инженерии представлена на рисунке 1.

Понятие «Синхронизированное распределённое взаимодействие» (Одно и то же время+Разные места) подразумевает, что может использоваться на выбор одна из онлайн оболочек для проектирования(UML Progect), держать связь с инструктором с помощью Moodle для изучения новой темы, а также Moodle служит для получения задания проверки уровня знаний, Google Groups/Moodle для отправки/получения задания преподавателю.

CC BY

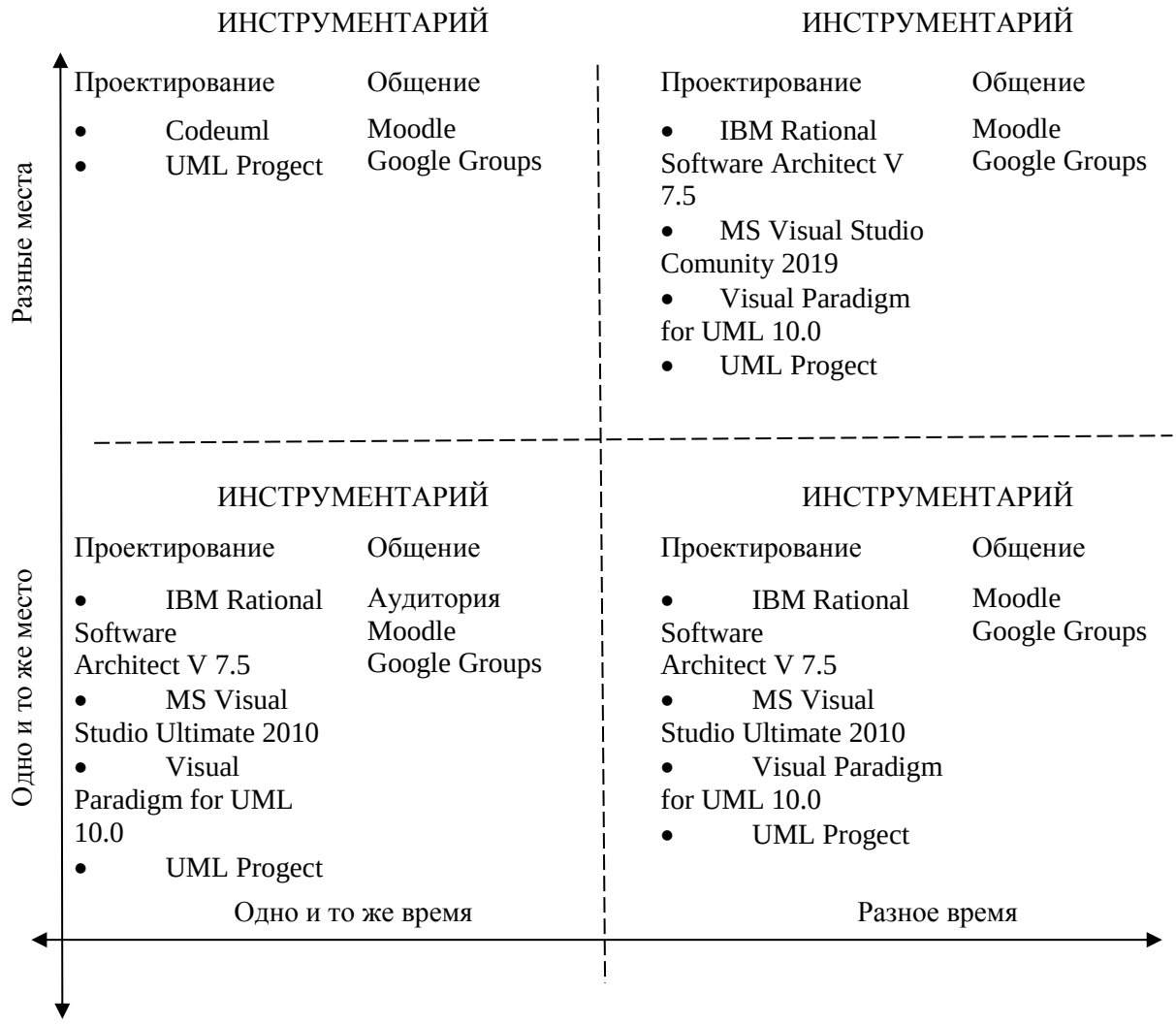


Рис. 1. Пространственно-временные варианты реализации виртуальной среды для методической системы

Понятие «Асинхронное распределённое взаимодействие» (Разное время+Разные места) подразумевает, что студенты используют, на выбор, весь доступный перечень инструментов для проектирования. Связь с преподавателем держат через Moodle и Google Groups для отправки/получения задания, проверки уровня знаний, самостоятельного изучения новой темы.

Понятие «Взаимодействие «лицом к лицу» (Одно и тоже время+Одно и то же место) подразумевает проведение занятия в одной аудитории и прямое общением с преподавателем, также как помощь может использоваться Moodle для проверки знаний и Moodle для отправки задания.

Понятие «Асинхронное взаимодействие» (Разное время+Одно и то же место) подразумевает, что студенты используют, на выбор, весь доступный перечень инструментов для проектирования. Связь с преподавателем держат через Moodle и Google Groups/Moodle для отправки/получения задания, проверки уровня знаний, самостоятельного изучения новой темы.

CC BY

В рамках предложенной методической системы был создан электронный методический курс в системе Moodle «Программная инженерия» (http://e.kipu-rc.ru/course/view.php?id=5351) с использованием сервиса Google Sites.

Структура и содержание сайта включают разделы: 1) О дисциплине; 2) Новости; 3) Вопросы и задания; 4) Критерии оценивания; 5) Лабораторный практикум; 6) Литература; 7) Программа дисциплины; 8) Программное обеспечение; 9) Методические материалы лабораторных занятий.

Электронный курс разработан для преподавателя и студентов, с помощью которого студенты могут самостоятельно готовиться к занятию, решать разного уровня задания, оценивать и сравнивать уровень своих знаний без помощи инструктора (преподавателя), выбирать необходимый инструментарий. На рисунке 2 представлена главная страница сайта.

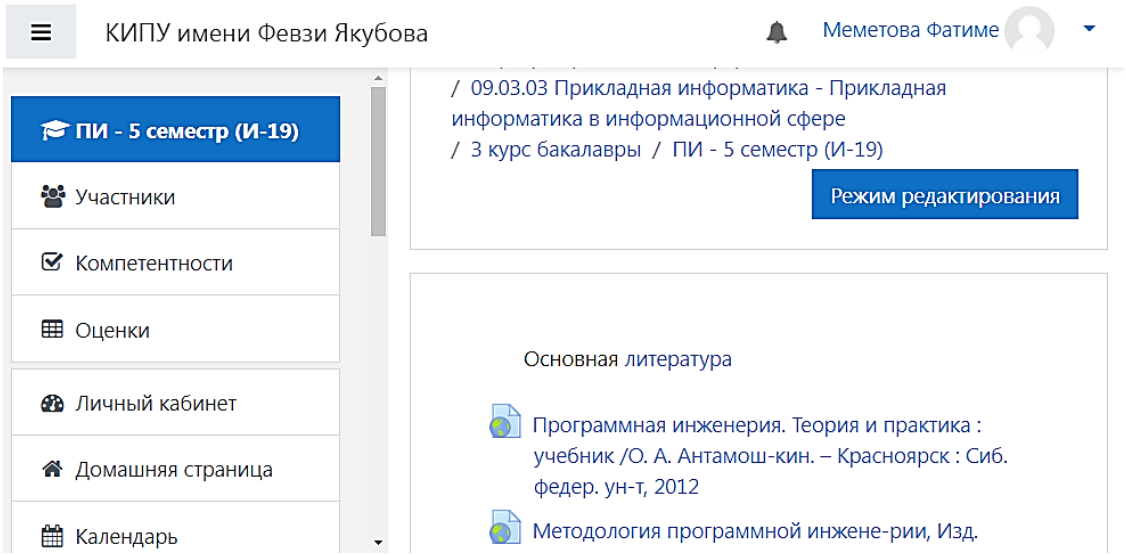


Рис. 2. Главная страница сайта

Студенту доступны все разделы курса (Новости, Вопросы и задания, Критерии оценивания, Лабораторный практикум, Литература, Программа дисциплины, Программное обеспечение, Тематический план лекций), по которому он может осуществить поиск, просмотр, скачивание.

Таким образом, в ходе анализа особенностей обучения студентов разработки программного обеспечения были: 1) представлены и сформулированы компетентности, необходимых для обучения программной инженерии; 2) составлена структура и содержание учебной дисциплины «Программная инженерия»; 3) предложены формы и методы проведения занятий по дисциплине «Программная инженерия».

Литература

1. Безверхий А.И. Проектное обучение программной инженерии // Академический форум корпорации EMC // Сб. тезисов докладов участников академической секции (г. Ялта, 23-28 сентября 2013 года). Симферополь, 2013. С. 71-72.

2. Андиева Е.Ю. Мультимедийный учебно-методический комплекс для дистанционного обучения по дисциплине «Введение в программную инженерию» // Хроники объединённого фонда электронных ресурсов наука и образование. 2015. Т. 1. № 1(68). С. 21.

3. Борисов С.В., Полуэктова Е.В. Выбор инструментария для начального обучения студентов по направлению подготовки «Программная инженерия» // Инженерный вестник. 2015. № 11. С. 16.

© Меметова Ф.С., 2022

УДК 378.147
https://doi.org/10.36906/AP-2022/71

Миронова Л.И.
ORCID: 0000-0002-3675-6008, д-р пед. наук, канд. техн. наук
Уральский федеральный университет им. первого Президента России Б.Н. Ельцина
г. Екатеринбург, Россия

ВАРИАНТ ВЗАИМОДЕЙСТВИЯ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ СО ШКОЛАМИ В ПРОЦЕССЕ ЦИФРОВОЙ ТРАНСФОРМАЦИИ ОБРАЗОВАНИЯ

Аннотация. В статье проведён анализ подготовки студентов IT-направлений. Показано, что их профессиональная подготовка основана на совместном изучении фундаментальных и прикладных информационных основ. Это позволяет им разрабатывать разного рода программное обеспечение. Предложен вариант создания студенческого научного кластера, в рамках которого студенты IT-направлений под руководством опытных университетских педагогов смогут разрабатывать для школ, лицеев и гимназий новые, актуальные цифровые учебные материалы, что является одной из главных задач процесса цифровой трансформации образования.

Ключевые слова: профессиональная подготовка студентов IT-направлений; студенческий научный кластер; цифровые образовательные ресурсы.

Mironova L.I.
ORCID: 0000-0002-3675-6008, Ph.D.
Ural Federal University named after the first President of Russia B.N. Yeltsin
Yekaterinburg, Russia

VARIANT OF UNIVERSITY GRADUATES IN IT AREAS INTERACTION WITH SCHOOLS IN DIGITAL TRANSFORMATION OF EDUCATION PROCESS

Abstract. The article analyzes the training of IT students. It is shown that their professional training combines the study of both fundamental and applied information bases. This allows them to develop all sorts of software. A variant of creating a student scientific cluster has been proposed, within which IT students, under the guidance of qualified university teachers, will be able to develop new, relevant digital educational materials for schools, lyceums and gymnasiums, which is one of the main tasks of the digital transformation of education.

Keywords: professional training of IT students; student scientific cluster; digital educational resources.

Традиционно подготовка выпускников вузов по IT-направлениям сочетает в себе два подхода: изучение фундаментальных информационных основ (Computer Science), что

CC BY

соответствует алгоритмическому подходу в обучении, и изучение прикладных информационных аспектов (Computer Engineering), что соответствует инженерному подходу в обучении. Такой симбиоз двух направлений подготовки будущих «айтишников» складывался в последние десятилетия, при этом у них, одновременно с навыками оценки, отбора, упорядочения и обработки информации, формировался творческий подход к созданию новых методов обработки информации и средств информатизации [8]. В ходе их подготовки используются как традиционные формы обучения студентов, регламентированные образовательным стандартом (лекции, семинары, лабораторные и практические занятия, контролирующие мероприятия, консультативные занятия, самостоятельная работа), так и электронные образовательные ресурсы (или интерактивные средства обучения (Ваграменко Я.А. [3], Касторнова В.А. [6]); цифровые образовательные ресурсы (Демкин В.П. [5]).

В данной статье мы будем употреблять термин «Электронный образовательный ресурс» (ЭОР), согласно результатам исследований, проведённым в рамках научной школы Роберт И.В.: под электронным образовательным ресурсом будем понимать электронное средство учебного назначения, обеспечивающее информирование студентов о методических особенностях преподавания той или иной дисциплины посредством удалённого интерактивного взаимодействия с пользователем; регламентацию самостоятельной работы студентов и предоставление учебно-методического контента на базе технологий мультимедиа, гипертекста, гипермедиа; автоматизацию контроля знаний и умений студентов по конкретной дисциплине [12]. Данное определение наиболее полно отвечает требованиям, предъявляемым к педагогической продукции, функционирующей на базе средств ИКТ.

Обучение будущих «айтишников» с использованием на различных этапах учебного процесса грамотно разработанных с методической точки зрения ЭОР позволяет сформировать у них следующие компетенции: умение самостоятельно приобретать знания, пользуясь разнообразными источниками информации; умение с этой информацией работать, используя различные способы познавательной деятельности; опыт использования ЭОР в удобное для студента время. При этом самостоятельное приобретение знаний студентом и его познавательная деятельность носит активный характер, приобретённые знания он применяет на практике для решения реальных проблем. Организация самостоятельной (индивидуальной или групповой) деятельности студентов предполагает использование новейших педагогических технологий, адекватных специфике данной формы обучения и стимулирующих раскрытие внутренних резервов каждого студента. Наиболее удачными в этом отношении являются исследовательские и проблемные методы (обучение в сотрудничестве, метод проектов и т. д.). Обучение с использованием ЭОР предусматривает активное взаимодействие как с преподавателем, так и со студентами в группе в ходе осуществления познавательной и творческой деятельности, а контроль носит систематический характер и строится как на основе оперативной обратной связи (предусмотренной в структуре учебного материала), так и текущего автоматического контроля (через системы тестирования) или отсроченного контроля (например, при очном тестировании).

CC BY

Таким образом, подготовка студентов IT-направлений, учебные курсы которой строятся в сочетании традиционных форм обучения с использованием ЭОР, способствует развитию самостоятельной, поисковой, научно-исследовательской деятельности студентов, а также повышению их интереса к будущей профессии.

Согласно государственным образовательным стандартам, в соответствии с которыми осуществляется подготовка студентов IT-направлений, объектами их профессиональной деятельности являются математические и алгоритмические модели, программы, программные системы и комплексы, методы их проектирования и реализации, способы производства, сопровождения, эксплуатации и администрирования в различных областях, в том числе в междисциплинарных. По окончанию учёбы для будущих «айтишников», в соответствии с образовательным стандартом, предусмотрена государственная аттестация в виде защиты выпускной квалификационной работы. Государственный экзамен проводится по усмотрению вуза. Новоиспечённый выпускник IT-направления как программист «…предназначен для работы в научных, проектных, конструкторских, технологических организациях, коммерческих структурах, в банках и на промышленных предприятиях, а также должен быть способен преподавать в школах и средних учебных заведениях» (https://clck.ru/dYXVW).

Одновременно выпускник IT- направления должен обладать компетенциями, которые охватывают наиболее распространённые сферы информационных технологий, а именно: аппаратную конфигурацию персонального компьютера, операционные системы, устройства сетевого взаимодействия, серверные операционные системы, сетевую безопасность.

В области аппаратной конфигурации персонального компьютера студентов IT-направления должен быть способен: установить операционную систему (ОС) на компьютер согласно требованиям пользователя и производителя; определить системные требования для установки ОС; конфигурировать ОС по требованию пользователя; установить программы, драйверы и необходимые приложения; произвести апгрейд компьютерных программ по требованию пользователя; разделить и форматировать жёсткий диск; установить обновлённое программное обеспечение для улучшения работы систем; создать резервные копии важной информации.

В области операционной системы выпускник IT-направления должен быть способен: использовать различные ОС для выполнения стандартных операционных заданий; распознавать имена, задачи и контент файловых систем; демонстрировать возможности использования командных функций и утилит для управления ОС, включая опции переключения и синтаксис; распознавать базовые концепты и процедуры для создания, просмотра и управления дисками, директориями и файлами; управлять различными операционными системами с помощью инсталляции, конфигурирования и апгрейда для обеспечения функциональных потребностей клиентов; определять необходимые для установки ОС процедуры и их выполнение; определять последовательности загрузки и методов, включая пошаговое создание экстренного загрузочного диска; определять необходимые процедуры, включая загрузку, добавление и конфигурирование драйверов и требуемых программ; определять действия, необходимые для оптимизации операционных

CC BY

систем и подсистем; устанавливать, конфигурировать и осуществлять апгрейд ОС; распознавать и определять общие ошибки и загрузочные сообщения от BIOS (Basic Input/Output System – базовая система ввода/вывода); определять действия, необходимые для исправления проблем с загрузкой; адекватно применять общие загрузочные инструменты и устройства; определять необходимость использования общих диагностических утилит и инструментов; определять общие операционные и пользовательские проблемы и их решение; конфигурировать IP-адреса, маски подсетей и маршрутов по умолчанию.

В области устройств сетевого взаимодействия выпускник IT- направления должен быть способен: подключать маршрутизатор с использованием консольного кабеля и удалённого доступа; конфигурировать пользовательский режим, устанавливать привилегии и пароли для удалённого доступа; конфигурировать интерфейс Ethernet с IP-адресом, маской подсети и описанием интерфейсов; конфигурировать сетевых клиентов (хостов) для создания сети; соединять компьютеры пользователей в сеть с помощью кабелей; проверять связь с использованием ping (время реакции компьютера на запрос), traceroute (служебная программа для определения маршрутов следования данных в сетях TCP/IP) и telnet (сетевой протокол для реализации текстового интерфейса по сети); конфигурировать маршрутизаторы с использованием сервисов и протоколов маршрутизации; управлять конкретной файловой системой; конфигурировать маршрутизатор и загрузку софта для операционной системы IOS (мобильная операционная система компании Apple) через запоминающие устройства с серверов TFTP (Trivial File Transfer Protocol – простой протокол передачи файлов); создавать резервные копии софта для IOS, а также конфигураций на сервере TFTP; управлять списками доступа для маршрутизатора конкретной файловой системы с целью обеспечения сетевой безопасности; проверять работоспособность сети; обнаруживать сетевые неполадки с использованием утилит ping, traceroute и telnet; определять основные параметры для конфигурирования беспроводных сетей; конфигурировать компоненты беспроводных сетей, протоколы в коммутаторах конкретной файловой системы, режимы коммутации для обеспечения работы высокоскоростной вычислительной сети, а также конкретную файловую систему для обеспечения высокой скорости; конфигурировать и проверять VLAN (Virtual Local Area Network- логическая – «виртуальная» – локальная компьютерная сеть), в коммутаторах конкретной файловой системы; конфигурировать маршрутизацию inter-VLAN на маршрутизаторе конкретной файловой системы, трансляцию сетевых адресов (NAT) на маршрутизаторе конкретной файловой системы, подинтерфейсы передачи данных, а также протоколы динамического распределения адресов DHCP (Dynamic Host Configuration Protocol) и осуществлять проверку DHCP.

В области серверных операционных систем выпускник IT-направления должен быть способен: создавать разрешения для файловых систем; создавать и управлять веб-серверами; управлять процедурой создания резервных копий; управлять удалёнными файлами; сохранять конфигурации; устанавливать образ ОС; создавать сервер сетевого протокола.

В области сетевой безопасности выпускник IT-направления должен быть способен: создавать закрытую беспроводную сеть и списки доступа для проверки подлинности адресов;

CC BY

конфигурировать безопасность коммутируемой сети и защиту от широковещательного шторма.

В ходе современной вузовской подготовки студенты IT-направлений осваивают и фундаментальные, и прикладные информационные основы. При этом специфической особенностью их подготовки является то, что средства ИКТ для них являются и объектом изучения, и средством обучения [8]. К компьютерным средствам познания относят языки программирования, прикладные профессиональные пакеты, а также компьютерные средства общего назначения для использования в профессиональной деятельности. К компьютерным обучающим средствам относят компьютерные средства, в структуре которых реализованы те или иные модели или их элементы для изучения тех или иных учебных дисциплин (педагогические программные средства (ППС), инструментальные программные средства учебные программные средства) [7]. Анализ учебно-методических материалов, используемых при подготовке студентов IT-направлений, показал, что около 80% учебных модулей осваиваются студентами с использованием пакетов прикладных программ и других средств ИКТ. Оставшиеся 20% учебных модулей не используют программные продукты в явном виде в силу специфики излагаемого материала, но при этом средства ИКТ успешно применяются в организации самостоятельной работы студентов, для консультационного общения с преподавателями в процессе подготовки курсовых работ, выпускных квалификационных работ, в ходе учебной, производственной и преддипломной практик. В таблице представлены базовые программные продукты и средства ИКТ, используемые при подготовке выпускников IT-направлений.

Проведённый анализ подготовки выпускников, обучающихся по IT-направлениям показал, что она обладает высоким потенциалом, позволяющим формировать у них необходимую профессиональную компетентность, которая может быть успешно использована в процессе цифровой трансформации образования в рамках развития цифровой инфраструктуры образовательных организаций, в процессе разработки цифровых сервисов и инструментов для всех видов аттестационных процедур, систем сбора, хранения и обмена данными успешности участников образовательного процесса (как педагогов, так и обучаемых всех возрастов и ступеней обучения), а также в рамках развития общедоступных цифровых коллекций учебно-методических материалов для организации образовательного процесса [16].

Взаимосвязь со школами может быть установлена при наличии в структуре вуза, который занимается подготовкой выпускников по IT-направлениям, организации, способной мотивировать студенческую молодёжь заниматься исследовательской деятельностью в рамках цифровой трансформации образования.

Цель такой организационной структуры – выявление, объединение, обучение, техническая, коммуникационная, информационная, профессионально-ориентированная поддержка студентов, реализация которой позволит решать профориентационные в плане цифровой трансформации образования задачи, способствующие саморазвитию и

CC BY

самоопределению личности студента, оказывающие существенное влияние на личностно-профессиональное становление будущего IT-специалиста.

Таблица

Базовые программные продукты и средства ИКТ,
используемые при подготовке выпускников IT-направлений [8; 13, 16; 17]

Дисциплина	Используемые программные продукты и средства ИКТ
Информатика и программирование (теория и практика)	Microsoft Office: MS Word, MS PowerPoint, MS Excel, алгоритмический язык Object Pascal
Методы вычислений (теория и практика)	MathCAD, MathLab, Бэйсик, Паскаль, СИ, СИ++, СИ#
Программирование (Object Pascal, Делфи, СИ++ Builder, Java, WEB, WEB-дизайн, Ассемблер, Параллельное программирование, Рекурсивно-логическое программирование, Паттерны программирования, Программирование для Андроид	Среда программирования Object Pascal, Среда визуального программирования Delphi, Среда визуального программирования C++Builder, Язык программирования Java, Язык программирования Ассемблер,HTML, PHP, C#, C/C++, библиотека OpenMP, High Performance Fortran, Язык логического программирования Пролог, язык проектирования UML; системы работы с UML-диаграммами ModelMaker и IBM XDE Rational Rose, Среда Eclipse для ОС Android
Информационная безопасность (правовые основы и методы защиты информации)	Программы PGP, Zone Alarm
Сети (Администрирование информационных систем, Сетевые информационные технологии, Интернет-Интранет, Корпоративные информационные сети)	Виртуальная машина Microsoft Virtual PC 2007, Операционная система Microsoft Windows Server 2003, Internet, электронная почта, ICQ, Skype, Mail Agent, MS Media Player, Win Amp, WinDVD, zplayer, ACDSee, Язык гипертекста HTML, язык виртуальной реальности VRML, язык программирования Java, браузеры: Internet Explorer, Mozilla Firefox, Opera, Chrome, расширения Plug-In и ActiveX, Web-серверы: Microsoft Internet Information Server, NetWare Web Server, Netscape Enterprise Server, Netscape FastTrack Server, Apache
Базы данных (СУБД, практикум)	MySQL, MS Access, СУБД MySQL, СУБД MS Access
Операционные системы (Операционные системы, среды и оболочки, ОС Minix)	Операционные системы: MS-DOS, UNIX, WINDOWS, Операционная система Minix
Технология разработки ПО, мобильных приложений, символьные вычисления	HTML, CSS, JavaScript, библиотека jQuery, PHP, Среда Eclipse для ОС Android, ОС Bada, ОС Ubuntu, Пакеты Mathcad, Maple
Компьютерное моделирование (теория и практика)	MathCAD, MathLab, Бэйсик, Паскаль, СИ
Системы искусственного интеллекта	Язык логического программирования Пролог
Компьютерная графика	MS PowerPoint, Open Impress, Графический редактор Adobe PhotoSHOP, Графический редактор CorelDraw, Среда Adobe Flash CS4

CC BY

Задачи такой организации должны быть следующие:

1. Активизация познавательной деятельности студентов, актуализация и интеграция теоретических и практических знаний в области ИКТ в конкретную целостность, применение которой направлено на цифровую трансформацию образования.

2. Развитие проективных и прогностических умений у IT-студентов (целеполагание, выдвижение научной гипотезы и т. п.).

3. Контроль качества разработанных студентами цифровых программных продуктов ведущими преподавателями IT- кафедр.

4. Формирование опыта обмена и научных дискуссий, свободы конструктивной критики.

5. Формирование у студентов уверенности в себе, осознание ими значимости выполненной исследовательской работы.

6. Развитие навыков общения с аудиторией, формирование речевой коммуникации, умение подготовить презентационные материалы, связанные с проведёнными разработками, способность публично демонстрировать результаты своей деятельности на конференциях разного уровня.

7. Создавать делегации студентов, показавших наилучшие результаты в области разработки ресурсов в рамках цифровой трансформации образования, для участия в крупнейших студенческих конференциях России и стран ближнего зарубежья.

8. Формирование устойчивого желания в дальнейшем заниматься программистской деятельностью в области цифровой трансформации образования.

Одним из путей вариантов создания обозначенной организации в структуре вуза может быть создание студенческого научного кластера (СНК). *Кластер* (англ. *cluster* – скопление [14]) – объединение нескольких однородных элементов, которое может рассматриваться как самостоятельная единица, обладающая определенными свойствами. Тогда под *студенческим научным кластером* будем понимать систему обучения, взаимообучения и инструментов самообучения в системе «наука – технологии – бизнес» [9; 10]. Главная цель всех участников образовательного кластера – подготовка высококвалифицированных IT-выпускников, способных решать задачи, связанные с цифровой трансформацией образования.

Организацию студенческих научных кластеров целесообразно создавать на базе институтов, входящих в структуру университета, которые занимаются подготовкой IT-выпускников. Если каждая институтская IT-кафедра организует свои представительства в школах, гимназиях, лицеях, то будет установлена вертикаль взаимодействия школы и вуза, наличие которой, помимо решения профориентационных задач школьников, позволит решать силами студентов-айтишников под руководством ведущих преподавателей в области информационных технологий задачи цифровой трансформации образования.

Студенческий научный кластер, реализация которого создаёт условия для формирования интереса будущих IT-выпускников к научным и прикладным исследованиям, одновременно, позволит во время учёбы в вузе развивать и укреплять его, а также формировать у студентов знания основ методологии науки [1]. Приобщение к занятиям исследовательской деятельностью в рамках студенческого научного кластера направлено на формирование у

CC BY

студентов навыков самообразования и повышения квалификации, способность к самоанализу и принятию нестандартных решений, что является одной из задач подготовки современных конкурентоспособных выпускников, которая реализуется средствами личностно развивающего профессионального образования [16] и соответствующими технологиями и организацией обучения [11; 15] за счёт активизации обучения в вузе [4] и повышении мотивации студентов [2].

Реализация предлагаемого в рамках статьи подхода в структуре подготовки будущих программистов носит взаимовыгодный характер: благодаря этой деятельности будущие программисты повышают своё профессиональное мастерство за счёт создания цифровой образовательной среды в школах, гимназиях и лицеях, которая обеспечивается новыми, актуальными цифровыми материалами, что является одной из главных задач процесса цифровой трансформации образования.

В заключении можно вспомнить прозорливое выражение академика Ершова А.П., высказанное им в конце 60-х годов прошлого века, основной мыслью которого было то, что «Программист – солдат второй промышленной революции и как таковой должен обладать революционным мышлением и мужеством». Мы являемся свидетелями того, как программисты разных направлений подготовки (в области системного и прикладного программирования, а также Web-программирования и программирования мобильных приложений) становятся главными действующими лицами в эпоху цифровизации. Поэтому насколько «умным» будет наш цифровой мир в ближайшем будущем и как интенсивно будет протекать процесс цифровой трансформации образования в большой степени будет зависеть от профессиональной компетентности выпускников IT-направлений.

Цифровизация экономики – очередной вызов времени и научно-технического прогресса. Цифровая трансформация образования – адекватный ответ на этот вызов.

Литература

1. Анисимов О.С. Методология: функции, сущность, становление. М.: ЛМА, 1996. 380 с.

2. Бакшаева Н.А., Вербицкий А.А. Психология мотивации студентов. М.: ЛОГОС, 2006. 184 с.

3. Ваграменко Я.А. Информатика: образовательный аспект. М.: ИИО РАО, 2011. 120 с.

4. Вербицкий А.А. Активное обучение в высшей школе: контекстный подход. М.: Высшая школа, 1991. 207 с.

5. Дёмкин В.П., Можаева Г.В. Классификация образовательных электронных изданий: основные принципы и критерии // Открытое и дистанционное образование. 2003. № 3. С. 1-6.

6. Касторнова В.А. Электронное пространство знаний: назначение и пример реализации // Вестник Череповецкого государственного университета. 2009. № 3. С. 19-24.

7. Майер В.Р. Методическая система геометрической подготовки учителя на основе новых информационных технологий. Красноярск: РИО КГПУ, 2001. 386 с.

8. Миронова Л.И. Инновационная основа формирования компетенций бакалавров – технологии продуктивного обучения на базе средств ИКТ // Управленец. 2011. № 9-10 (25- 26). С. 18-21.

CC BY

9. Миронова Л.И., Язовцев И.А. Стратегические вопросы создания образовательного облачного сервиса для триады взаимодействия «бизнес-власть-образование» // Известия Уральского государственного экономического университета. 2012. № 2(40). С. 147-152.

10. Набойченко С. Соболев А., Богатова Т. К реализации стратегии партнёрства высшей школы и бизнеса // Высшее образование в России. 2007. № 1. С. 3-10.

11. Полат Е.С. Новые педагогические и информационные технологии в системе образования. М.: ACADEMIA, 2000. 268 с.

12. Роберт И.В. Теория и методика информатизации образования: психолого-педагогический и технологический аспекты. М.: Бином. Лаборатория знаний, 2014. 398 с.

13. Гришина Е.А. Современный словарь иностранных слов: ок. 20000 слов. М.: Русский язык, 1992. 740 с.

14. Бордовская Н.В. Современные образовательные технологии. М.: КНОРУС, 2010. 432 с.

15. Сумцова Н.В., Едронова В.Н. Обеспечение качества образования: пути повышения эффективности учебного процесса // Высшее образование сегодня. 2003. № 6. С. 30-36.

16. Уваров А.Ю. Цифровая трансформация и сценарии развития общего образования. М.: НИУ ВШЭ, 2020. 108 с.

17. Шапкина, Т.И. Проектирование учебной деятельности студентов на основе электронных учебно-методических комплексов // Педагогическая информатика. 2008. № 1. С. 27-33.

© Миронова Л.И., 2022

УДК 371.3
https://doi.org/10.36906/AP-2022/72

Казиахмедов Т.Б.
ORCID: 0000-0002-5937-1514, канд. пед. наук
Мосягина Т.В.
ORCID: 0000-0003-0146-4176
Нижневартовский государственный университет
г. Нижневартовск, Россия

ФОРМИРОВАНИЕ WEB КОМПЕТЕНЦИЙ БАКАЛАВРОВ ИВТ В УСЛОВИЯХ НАЛИЧИЯ БОЛЬШОГО НАБОРА ИНСТРУМЕНТОВ ДЛЯ WEB ПРОЕКТИРОВАНИЯ

Аннотация. В представленной работе описаны подходы выбора инструментария web проектирования в условиях их бурного развития и цифровизации всех сфер экономической деятельности государства. В качестве инструментов web проектирования рассматриваются технологии PHP, ASP.NET, PYTHON+DJANGO, анализируется повышение производительности разработки сайтов с использованием конструкторов сайтов. Результатом работы является методическое обоснование системы формирования продвинутых компетенций у BackEnd, FrontEnd и FullStack разработчиков.

Ключевые слова: Web технологии; Web разработчик; Web администратор; технология PHP; ASP.Net; конструкторы сайтов.

Kaziakhmedov T.B.
ORCID: 0000-0002-5937-1514, Ph.D.
Mosyagina T.V.
ORCID: 0000-0003-0146-4176
Nizhnevartovsk State University
Nizhnevartovsk, Russia

FORMATION OF WEB COMPETENCIES OF IT BACHELORS IN THE PRESENCE OF A LARGE SET OF TOOLS FOR WEB DESIGN

Abstract. The presented paper describes approaches to choosing web design tools in the conditions of their rapid development and digitalization of all spheres of economic activity of the state. The technologies PHP, ASP.NET, PYTHON + DJANGO are considered as web design tools, the increase in the productivity of website development using website builders is analyzed. The result of the work is a methodological substantiation of the system for the formation of advanced competencies for BackEnd, FrontEnd and FullStack developers.

Keywords: Web technologies; Web developer; Web administrator; PHP technology; ASP.Net; site builders.

CC BY

Подготовка IT бакалавров web разработке приобретает особую значимость в эпоху цифровизации экономики государства благодаря тому, что наступает время разработки web сервисов, использование облачных ресурсов и хранилищ, вычислительных мощностей. Порой сталкиваемся с тем, что для решения проблем ИИ нужны огромные массивы данных, доступ к которым необходимо осуществлять через Web. Поэтому обучение Web технологиям нужно строить с учётом информационной инфраструктуры предприятий, региона, да и государства в целом. О переводе начал программирования на такие языки как Java Script, Python мы говорим уже не 1 год [1]. Именно эти языки сегодня становятся более актуальными и в области Web разработок. Java Script является полноценным платформенно не зависимым языком, средствами которого можно решить множество информационных или вычислительных задач.

Эти возможности более подробно приведены в [2; 3]. Рассмотрим некоторые задачи, при решении которых мы обычно используем конкретные языки как языки функционального, логического программирования.

Задача 1. Реализуйте следующие алгоритмы на массивах в виде функций:

• вычисление суммы элементов массива;

• вычисление произведения элементов массива;

• сортировка массива выборочным методом;

• нахождение минимального элемента массива.

Файл JavaScript.html:

```
<html>
<head>
</head>

<body>
 <script type="text/javascript">

var myArray = null;
  function generateArrayRandom(n, min, max) {
            var arr = [];
            for(var i = 0; i<n; i++) {
                        arr.push(Math.floor(Math.random() * (max - min + 1)) + min);
            }
            myArray = arr;
  }
  function summ(array) {
            var sum = 0;
            for(var i = 0; i<array.length; i++) {
                        sum += array[i];
            }
            return sum;
  }
  function mult(array) {
            var m = 1;
            for(var i = 0; i<array.length; i++) {
                        m *= array[i];
            }
            return m;
  }
```

CC BY

```
function sort(array) {
	for(var i = 0; i<array.length - 1; i++) {
		for(var j = i; j<array.length; j++) {
			if(array[i] > array[j]) {
				var t = array[j];
				array[j] = array[i];
				array[i] = t;
			}
		}
	}
}
function minimum(array) {
	var m = array[0];
	for(var i = 0; i<array.length; i++) {
		if(array[i] < m)
			m = array[i];
	}
	return m;
}

function run() {
	var x = parseInt(document.myForm.number.value);
	generateArrayRandom(x, 0, 100);
	var result = '[' + myArray + ']';
	var msg = `<font color="blue"> Исходный массив:<br>${result}`;
	document.getElementById("div_0").innerHTML = msg;

	var min_el = minimum(myArray);
	document.getElementById("div_4").innerHTML = `<font color="magenta">${min_el}`;

	var summ_el = summ(myArray);
	document.getElementById("div_1").innerHTML = `<font color="magenta">${summ_el}`;

	var mult_el = mult(myArray);
	document.getElementById("div_2").innerHTML = `<font color="magenta">${mult_el}`;

	sort(myArray);
	sort_el = '[' + myArray + ']';
	document.getElementById("div_3").innerHTML = `<font color="magenta">${sort_el}`;
}

</script>

<h1>Лабораторная 5</h1>
<form name ="myForm">
  Введите размер исходного массива:
   <input type="text" name="number" size=4 maxlength=4>
   <br><br>
   <input type = "button" name = "button" value = "Сгенерировать" onclick="run()">
   <br><br>
   <div id="div_0"></div>
   <br>
   Сумма элементов массива:
   <div id="div_1">Нет</div>
   <br>
   Произведение элементов массива:
```

```
  <div id="div_2">Нет</div>
  <br>
  Отсотированный массив:
  <div id="div_3">Нет</div>
  <br>
  Минимальный элемент в массиве:
  <div id="div_4">Нет</div>
 </form>
 </body>
</html>
```

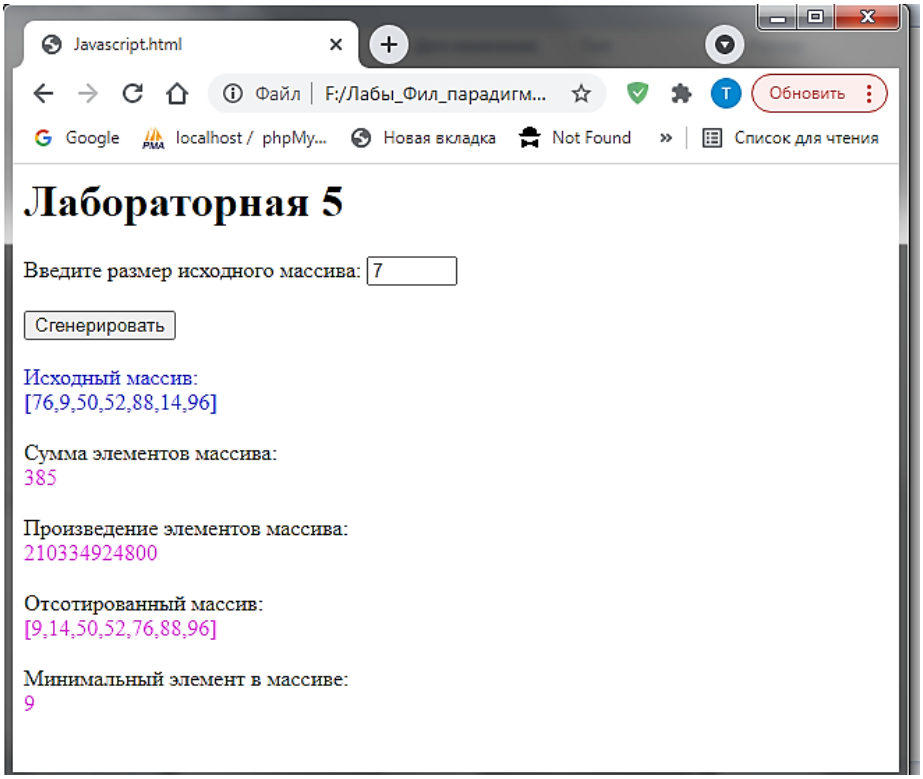


Рис. 1. Окно браузера с выполненными скриптами

Задача 2. Определите функции, вычисляющие

- неотрицательную степень целого числа;
- по натуральному числу N сумму чисел от 1 до N.

```
<html>
<head>
</head>

<body>
 <script type="text/javascript">

        function stepen(number, step) {
                if(step == 0)
                        return 1;
                else
                        return number * stepen(number, step - 1);
        }

        function summ(number) {
```

```
		if(number == 0)
			return 0;
		else
			return number + summ(number - 1);
	}

	function run() {
		var x = document.myForm.number.value;
		var y = document.myForm.number2.value;
		var result = stepen(x,y);
		var msg = `<font color="blue">Результат: ${x} в степени ${y} = ${result}</font>`;
		document.getElementById("div_0").innerHTML = msg;
		//document.write('Результат: ',x, ' в степени ',y,' = ', result);
	}

	function run2() {
		var x = parseInt(document.myForm.number3.value);
		var result = summ(x);
		var msg = `<font color="blue">Результат: Сумма от 1 до ${x} = ${result}</font>`;
		document.getElementById("div_1").innerHTML = msg;
		//document.write(msg);
	}
</script>

<form name ="myForm">
	<h1>Лабораторная 2</h1>
	<h2>Вычисление степени числа</h2>
	<br>
	Введите число:
	<input type="text" name="number" size=2 maxlength=2>
	<br><br>
	Введите степень:
	<input type="text" name="number2" size=2 maxlength=2>
	<br><br>
	<input type = "button" name = "button1" value = "Возвести в степень" onclick="run()">
	<br><br>
	<div id="div_0"></div>
	<br><br>
	<h2>Вычисление суммы от 1 до N</h2>
	Введите число N:
	<input type="text" name="number3" size=3 maxlength=3>
	<br><br>
	<input type = "button" name = "button2" value = "Посчитать сумму" onclick="run2()">
	<br><br>
	<div id="div_1"></div>
	<br><br>
</form>
</body>
</html>
```

CC BY

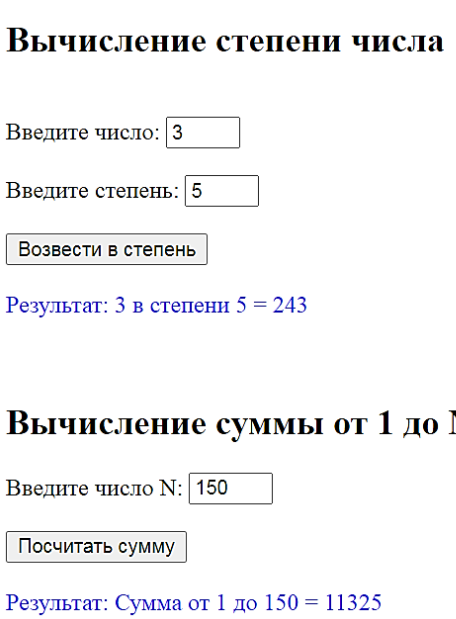


Рис. 2. Фрагмент окна браузера с выполненными скриптами

Таким образом, Java Script можно использовать как язык высокого уровня программирования и может использоваться в учебной дисциплине «Программирование» и это позволить формированию компетенций по дизайну сайтов, т. е. быть основой подготовки Frontend- разработчика.

Приведём инструментарий дисциплины, которые предназначены для подготовки Web разработчика по направлениям подготовки 09.03.01-Информатика и вычислительная техника, 09.03.02-Информационные системы и технологии (табл.).

Таблица

Уровни подготовки Web разработчика и краткое описание содержания дисциплин

Уровень подготовки / Дисциплины	FrontEnd	BackEnd	FullStack
Web программирование	Java Script,DHTML Конструктор сайтов WordPress	PHP, OpenServer(локальный Web сервер)	PHP,MYSQL Конструктор сайтов WordPress OpenServer(локальный Web сервер)
Web проектирование	Web-дизайн,CSS Конструктор сайтов Wix	Администрирование Web сервера Конструктор сайтов Wix	Web-дизайн,CSS Администрирование Web сервера, Type Script Конструктор сайтов Wix SEO-мастер – инструмент для автоматической поисковой оптимизации сайта. Wix ADI – быстрое создание ресурсов с помощь искусственного интеллекта. Средство для приема платежей. Встроенная CRM система.

Технология ASP.Net	MVC, ADO,FCL(пространство Web),C#	ADO.NET IIS,JQUERY	ADO.NET Администрирование IIS, JQUERY. Облачные Web серверы
Web разработка средствами PYTHON	PySharm,модуль Webbrowser, Java Script, HTML	PySharm, SQLite. Framework DJANGO	PySharm, модуль Webbrowser, Java Script, HTML, SQLite. Framework DJANGO

Подходы к системе подготовки FrontEnd Web разработчика более подробно описаны в [3] и практически демонстрированы в [4]. При подготовке FullStack web разработчика, обязательным является и обучение использованию нескольких конструкторов Web сайтов из следующего списка: Wix, Nethouse, Ukit, Tilda, UMI, Insales, Site123, Mozello, Fo.ru, Mottor, Joomla.

Предложенные нами подходы реализованы в системе подготовки IT специалистов в нашем вузе и востребованы работодателями региона.

Литература

1. Казиахмедов Т.Б., Яламов Г.Ю. Содержание и методы обучения программированию бакалавров по направлению «Информатика и вычислительная техника» // Педагогическая информатика. 2019. № 4. С. 47-58.

2. Казиахмедов Т.Б., Симурзина Е.А. Междисциплинарный подход в обучении программированию бакалавров ИВТ // Современное программирование: Мат-лы III Международной научно-практической конференции (г. Нижневартовск, 27-29 ноября 2020 года). Нижневартовск, 2021. С. 246-250. https://doi.org/10.36906/AP-2020/48

3. Казиахмедов Т.Б., Мосягина Т.В. Классификация видов практических работ и их соотношение с профессиональными компетенциями при изучении дисциплин по web-разработкам // Математические структуры и моделирование. 2016. № 2(38). С. 78-91.

4. Казиахмедов Т.Б., Елыкомов Р.Н. Этапы разработки автоматизированных информационных систем удалённого тестирования обучающихся // Информатизация образования-2020: Международная научно-практическая конференция (г. Орел, 29-31 октября 2020 года). Орёл, 2020. С. 329-334.

© Казиахмедов Т.Б., Мосягина Т.В., 2022

УДК 378
https://doi.org/10.36906/AP-2022/73

Натальсон А.В.
ORCID: 0000-0002-3830-1592
Казанский государственный энергетический университет
г. Казань, Россия

МЕТОДОЛОГИЯ ФОРМИРОВАНИЯ ЦИФРОВЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ

Аннотация. Цифровые компетенции являются одним из наиболее актуальных объектов изучения в рамках цифровизации современного общества. Представленная работа посвящена изучению вопроса, связанного с методологией формирования цифровых компетенций выпускников высших учебных заведений по специальностям, связанным с информационными технологиями. Автором применяются эмпирические и теоретические методы исследования. В работе используются научные материалы отечественного и зарубежного авторства.

Ключевые слова: цифровые компетенции; информационные технологии; цифровизация; общество; информация.

Natalson A.V.
ORCID: 0000-0002-3830-1592
Kazan State Power Engineering University
Kazan, Russia

METHODOLOGY OF FORMATION OF DIGITAL COMPETENCIES OF UNIVERSITY GRADUATES IN IT AREAS

Abstract. Digital competencies are one of the most relevant objects of study in the framework of digitalization of modern society. The presented work is devoted to the study of the issue related to the methodology of the formation of digital competencies of graduates of higher educational institutions in the specialties related to information technology. The author applies empirical and theoretical research methods. The work uses scientific materials of domestic and foreign authorship.

Keywords: digital competencies; information technologies; digitalization; society; information.

Повсеместная цифровизация общества обуславливается глубоким проникновением информационных технологий (ИТ), Интернета, социальных сетей и другого в повседневную жизнь человека. На основе этого на качественном уровне меняется информационно-технологическая парадигма развития экономики. Современный объем и качество ИТ, накопленные обществом, создают условия для их использования в экономике новыми

CC BY

способами, а также применения новых, так называемых сквозных технологий. Развитие ИТ и распространение создаёт инструментальную базу для развития цифровой экономики (ЦЭ), т. е. повышения эффективности хозяйственной деятельности субъектов за счёт использования достижений цифровизации общества [1].

Развитие цифровой экономики в современном мире выступает приоритетной задачей для всех стран, в том числе и России. Происходящие на рынке труда изменения, проявляющиеся в увеличении дефицита кадров требуемой квалификации, обусловлены усилением влияния на формирование рынка труда факторов, связанных с автоматизацией, цифровизацией и компьютеризацией. Развитие технологий, в свою очередь, приводит к увеличению требований работодателей к соискателям, последнее обостряет вопрос, связанный с обеспеченностью кадрами. Наряду с этим в условиях цифровизации актуализируется не только использование потенциала, но и его накопление вследствие возрастания роли научно-технической среды, влияющей на создание инноваций. Воспроизводство кадрового потенциала, предполагающее необходимые навыки и компетенции «будущего», становится одним из важных аспектов, способствующих экономическому росту [2].

Представляется целесообразным рассмотреть теоретико-методологические аспекты к определению понятий «компетенция» и «компетентность». Необходимость анализа данных терминов обусловлена тем, что существует два варианта толкования их соотношения: они либо отождествляются, либо разделяются. В связи с этим представляется целесообразным рассмотреть сущность обоих терминов для чёткого понимания и использования в дальнейшем исследовании (рис. 1) [3].

Помимо технологий движущей силой развития ЦЭ являются компетенции субъектов. Организации и граждане должны владеть компетенциями для того, чтобы использование ИТ привело к ожидаемым социальным и экономическим эффектам ЦЭ. Компетенция характеризует способность субъекта к осуществлению какой-либо деятельности. При этом компетенцией могут обладать как отдельные специалисты, так и предприятие, объединяя нескольких специалистов для ведения экономической деятельности.

Вопрос о необходимости формирования компетенций ИТ-кадров подчёркивается различными исследователями и экспертами. Так, например, отмечается, что подготовка квалифицированного ИТ-специалиста связана с накоплением необходимого набора компетенций. Обладание компетенциями, удовлетворяющими требованиям работодателей ИТ-отрасли, авторы исследований связывают с определенным образом организованным образовательным процессом (рис. 2) [4].

CC BY

Автор	Определение	
	Компетенция	Компетентность
А. В. Хуторский	Совокупность взаимосвязанных качеств личности (знаний, умений, навыков), задаваемых по отношению к определенному кругу предметов и процессов и необходимых для качественной продуктивной деятельности по отношению к ним [3; 21]	Владение, обладание человеком соответствующей компетенцией, с личностным отношением к ней и предмету деятельности [21]
Э. Ф. Зеер	Целостная и систематизированная совокупность обобщенных знаний [9]	Способность продуктивно выполнять профессиональную деятельности, то есть умение человека реализовывать на практике свои компетенции [9]
Н. Л. Гончарова	Базовое свойство, качество, которое формирует компетентного специалиста [6]	Обладание компетенцией, характеристика специалиста в его профессиональной деятельности [6]
В. Г. Зарубин	Некий круг обязанностей, которые должен выполнять человек, и именно в них он сможет проявить свою компетентность [12]	Личностно интегрированный результат, имеющий деятельностные, поведенческие признаки: практическую, опытную, мотивационно-ценностную и когнитивно-обеспеченную готовность действовать в сфере своей компетенции [12]
Ю. А. Глазков, Л. О. Денищева, К. А. Краснянская	Совокупность знаний, учебных умений и навыков, а также способы деятельности в жизни для решения практических и теоретических задач [8]	Совокупность компетенций, наличие знаний и опыта, необходимых для эффективной деятельности в заданной предметной области [8]
В. П. Симонов	Совокупность определенных знаний, умений, навыков, личностных качеств и опыта определенной сфере деятельности [17]	Широкое интегративное понятие, характеризующее обобщенную способность личности к решению жизненных и профессиональных задач благодаря имеющимся у нее знаниям, умениям и опыту [17]
Л. В. Черепанова	Совокупность специальных и общепредметных знаний, умений, навыков, способов деятельности и ценностных ориентаций, сформированных у школьников в результате изучения ими предметной образовательной области [19; 23]	Результат обучения, заключающийся в появлении у школьника особого свойства (качества), позволяющего ему осуществлять разные виды деятельности на основе полученных знаний, умений, навыков и способов познавательной деятельности, то есть опыта деятельности [19; 23]

Рис. 1. «Компетенция» и «компетентность»: определение понятий (Ильин Е., Психология помощи. Альтруизм, эгоизм, эмпатия)

CC BY

Компетенция	Описание
Коммуникация и кооперация в цифровой среде	Способность человека в цифровой среде использовать различные цифровые средства, позволяющие во взаимодействии с другими людьми достигать поставленных целей
Саморазвитие в условиях неопределенности	Способность человека ставить себе образовательные цели под возникающие жизненные задачи, подбирать способы решения и средства развития других необходимых компетенций
Креативное мышление	Способность человека генерировать новые идеи для решения задач цифровой экономики, абстрагироваться от стандартных моделей: перестраивать сложившиеся способы решения задач, выдвигать альтернативные варианты действий с целью выработки новых оптимальных алгоритмов
Управление информацией и данными	Способность человека искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач
Критическое мышление в цифровой среде	Способность человека проводить оценку информации, ее достоверности, строить логические умозаключения на основании поступающих информации и данных

Рис. 2. Перечень ключевых компетенций цифровой экономики

Цифровые компетенции подразумевают способность специалистов предприятия создавать инновации с использованием цифрового контента и технологий, повышающие эффективность экономической деятельности предприятия. При этом состав цифровых компетенций будет значительно варьироваться в зависимости от отрасли, в которой действует предприятие, и предметной области, в которой работает каждый конкретный специалист. Владение специалистом знаниями и компетенциями предметной области служит основой для формирования цифровых компетенций [5].

Для использования достижений цифровизации общества в экономической деятельности необходимы специальные компетенции по работе как самими технологиями, так и с информационными ресурсами (данными). На смену понятиям компьютерной грамотности или информационной компетентности пришло понятие «цифровые компетенции».

Применение ИТ только усиливают инновационный потенциал специалистов предприятия, обладающих соответствующими компетенциями. В современном обществе

CC BY

доступно множество ИТ и способов их применения. Цифровые технологии являются совокупностью ИТ, обеспечивающей накопление и обработку цифровых данных. Цифровые технологии включают известные и новые ИТ, для которых созданная ИТ-инфраструктура позволила найти новое или расширение применение в современной экономике за счёт удешевления стоимости приобретения, внедрения и использования ИТ [6].

Формирование цифровых компетенций должно проводиться на основе анализа тенденций развития информационных технологий и видах экономической деятельности их применения. За основу необходимо брать, к примеру, подход линейного программирования, позволяющий представить множество компетенций в виде матрицы. Информационная база исследования составляется с помощью используемых в экономике и науке методологических подходов к исследованию цифровых компетенций, а также данных отчётов ведущих консалтинговых компаний [7].

Также важно расширять контакты молодёжи с представителями профессиональной сферы и потенциальным работодателем на этапе обучения не только в профессиональных учебных заведениях, но и раньше – в процессе получения среднего общего образования. Это позволяет на раннем этапе начать процесс формирования у обучающихся необходимых компетенций для будущей профессиональной, в том числе, управленческой деятельности.

Компетенции субъектов ЦЭ подразумевают способность специалистов предприятия создавать инновации с использованием цифрового контента и технологий, повышающие эффективность экономической деятельности хозяйствующих субъектов. При этом состав этих компетенций будет значительно варьироваться в зависимости от отрасли, в которой действует субъект, и предметной области, в которой работает каждый конкретный специалист. Владение специалистом знаниями и компетенциями предметной области служит основой для формирования цифровых компетенций.

Проведённое исследование показывает, что в результате развития ИТ и многообразия сфер их применения формируется потребность общества в постоянном обновлении и получении новых компетенций. Развитие ЦЭ должно быть поддержано системой образования, где в основном происходит формирование и развитие интеллектуального капитала общества за счёт подготовки кадров и научных исследований. В тоже время система образования стоит перед вызовами цифровизации, требующих организационных, методических и технологических преобразований учебного процесса [8].

Таким образом, основной целью данной статьи являлось изучение вопроса, связанного с методологией формирования цифровых компетенций выпускников высших учебных заведений по специальностям, связанным с информационными технологиями. В результате работы были изучены такие аспекты, как: актуальность цифровизации современного общества и экономики; новые требования к специалистам в рамках цифровизации общества; подходы и инструменты, необходимые для формирования цифровых компетенций выпускников высших учебных заведений по направлениям, связанным с информационными технологиями.

CC BY

Литература

1. Авакова Э.Б., Асадов А.Н., Кузнецов А.А. Формирование цифровых компетенций при подготовке управленческих кадров // Известия Санкт-Петербургского государственного экономического университета. 2021. № 2(128). С. 116-121.

2. Бунтова Е.В. Методология создания учебного курса программы магистратуры в условиях цифровой экономики // Вестник Самарского государственного технического университета. Серия: Психолого-педагогические науки. 2019. № 1(41). С. 20-36. https://doi.org/10.17673/vsgtu-pps.2019.1.2

3. Климова Ю.О. Анализ соответствия уровня компетенций выпускников IT-специальностей требованиям работодателей // Вопросы территориального развития. 2021. Т. 9. № 1. https://doi.org/10.15838/tdi.2021.1.56.5

4. Крылова Н.П., Левашов Е.Н. Особенности гендерных показателей цифровой грамотности студентов // Science for education today. 2020. Т. 10. № 1. С. 128-148. https://doi.org/10.15293/2658-6762.2001.08

5. Маркеева А.В. Социальные последствия развития Интернета вещей (IOT) // Современные информационные технологии и ИТ-образование. 2016. Т. 12. № 2. С. 236-240.

6. Khutorskoy A.V. Definition of general subject content and key competencies as a characteristic of a new approach to the construction of educational standards // Bulletin of the Institute of Human Education. 2011. T. 1.

7. Shauhalova R.A. Organizational and pedagogical conditions for the formation of digital culture of university students // Modern Higher School: Innovative aspect. 2020.

8. Mikidenko N.L., Storozheva S.P., Kharlamov A.V. Features of the implementation of the competence model of higher engineering education in the conditions of the modern labor market // Bulletin of the NGPU. 2019.

© Натальсон А.В., 2022

УДК 519.681
https://doi.org/10.36906/AP-2022/74

Савина Е.В.
канд. физ.-мат. наук
Раджабов К.Я.
канд. экон. наук
Дагестанский государственный университет народного хозяйства
г. Махачкала, Россия

АДАПТАЦИЯ КОМПЕТЕНТНОСТНОГО ПОДХОДА К МЕТОДИКЕ ПОДГОТОВКИ IT-СПЕЦИАЛИСТОВ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ

Аннотация. На современном этапе реализации образовательных программ, направленных на формирование у выпускников профессиональных компетенций в сфере разработки программного обеспечения, возникла потребность в модификации имеющихся методик преподавания дисциплин, направленных на освоение технологий и методов программирования. При этом необходимо ориентироваться на используемые профессиональные стандарты.

Ключевые слова: программирование; компетенции; оценочные материалы; методика.

Savina E.V.
Ph.D.
Radzhabov K.J.
Ph.D.
Dagestan State University of National Economy
Makhachkala, Russia

ADAPTATION OF THE COMPETENCE APPROACH TO THE METHODOLOGY OF TRAINING IT SPECIALISTS IN THE FIELD OF PROGRAMMING

Abstract. At the present stage of the implementation of educational programs aimed at the formation of graduates' professional competencies in the field of software development, there is a need to modify the existing methods of teaching disciplines aimed at mastering programming technologies and methods. At the same time, it is necessary to focus on the professional standards used.

Keywords: programming; competencies; evaluation materials; methodology.

Современный этап развития страны характеризуется широким проникновением цифровых технологий во все сферы деятельности, включая систему образования. В условиях проводимой в стране цифровизации экономики остро стоит проблема подготовки

CC BY

высококвалифицированных IT-специалистов, владеющих умениями и навыками разработки, внедрения и сопровождения программного обеспечения. При этом многие образовательные программы, связанные с IT-сферой, содержат целый спектр учебных дисциплин, ориентированных на получение профессиональных компетенций, которые востребованы в условиях цифровой трансформации практико-ориентированных бизнес-процессов.

Существует достаточно большое количество публикаций, в которых с различной степенью глубины обсуждаются содержательные и методические аспекты обучения программированию, но при этом связь программирования с другими дисциплинами профильной подготовки практически не рассматривается, зачастую остаются без внимания и вопросы формирования профессиональных компетенций у бакалавров IT-направления в процессе обучения программированию.

Следует также отметить, что при многообразии дисциплин, обеспечивающих профессиональную подготовку бакалавров в области информационных технологий, практически отсутствуют подходы к выявлению вклада каждой из дисциплин в формирование конкретных профессиональных компетенций выпускника. Это обусловлено необходимостью формирования эффективных подходов в оценке приобретаемых компетенций, которые должны принципиально отличаться от традиционного подхода, направленного на выявление объёма и качества усвоенных знаний. Компетентностный подход предполагает в первую очередь оценку способностей к профессиональной деятельности, освоенных в процессе обучения [1].

Письмом Минобрнауки РФ руководителям образовательных организаций высшего образования от 12.07.2021 № мн-5/4611 было рекомендовано включить в образовательные программы модули «Введение в информационные технологии» и «Информационные технологии и программирование», направленные на формирование компетенций, включающих способности понимать принципы работы современных IT-технологий, использовать их для решения задач профессиональной деятельности и разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

В ходе практической реализации компетентностного подхода в подготовке бакалавров для IT-отрасли в Дагестанском государственном университете народного хозяйства для образовательной программы 09.03.03 Прикладная информатика был сформирован интегрированный бизнес-процесс, включающий в технологическом плане ряд взаимосвязанных учебных дисциплин, приведённых на схеме 1.

Бизнес-процесс предусматривает развитие у бакалавров следующих компетенций:

– ОПК1 – способность применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;

– ОПК2 – способность понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

CC BY

– ОПК7 – способность разрабатывать алгоритмы и программы, пригодные для практического применения;

– ОПК4 – способность участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;

– ПК1 – способность проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе;

– ПК2 – способность разрабатывать, адаптировать, тестировать и внедрять прикладное программное обеспечение информационных систем;

– ПК3 – способность проектировать информационные системы по видам обеспечения.

Под компетентностным подходом будем понимать альтернативу классическому подходу, в котором используются, главным образом, категории усвоения знаний. При этом оценка компетенций, в отличие от экзаменационных испытаний, нацелена на практическое применение специальных подходов и методов диагностики деятельности. Термин «компетенция» будем трактовать как совокупность взаимосвязанных качеств личности (знаний, умений, трудовых навыков), задаваемых по отношению к определенному кругу дисциплин при формировании индикаторов достижения [2].

Одним из показателей качества профессиональной подготовки программистов является профессиональная компетентность, которая определяется через умение специалиста применить полученные знания, умения и навыки для решения прикладных задач.

Схема 1

Структура интегрированного бизнес-процесса

При формировании профессиональных компетенций программиста важное место занимает профессиональная компетенция ПК2. В условиях цифровизации экономики особое внимание специалисты уделяют информационно-аналитическим компетенциям, связанным с системами искусственного интеллекта. Для комплексного решения проблемы подготовки программистов образовательные программы, реализуемые в ДГУНХ, дополнены учебными

CC BY

курсами по программированию в рамках сетевого взаимодействия с ведущими вузами Российской Федерации (НИУ ВШЭ, Санкт-Петербургский политехнический университет Петра Великого).

Для оценки усвоения компетенций, интегрированных в наш обобщённый бизнес-процесс, нами разработаны и апробированы разнообразные многофункциональные оценочные материалы, примерные варианты которых приведены ниже [3; 4]. В связи с ограничением объёма статьи здесь приведены варианты заданий по одному из возможных языков программирования.

Пример 1. Изучите программный код и выполните задания.

```
begin
  var s: ...;
  for var i := 1 to 4 do
    for var j := 1 to 6 do
      begin
       read(s[i, j]);
       if s[i, j] = '+' then s[i, j]:= '1';
      end;
  for var i := 1 to 4 do
  begin
    for var j := 1 to 6 do write(s[i, j]);
    println
  end;
end.
```

1. Фрагмент программы выполняет следующие действия:
   1) считывает числовой массив 4×6 по столбцам
   2) считывает символьный массив 4×6 по строкам
   3) считывает символьный массив 6×4 по столбцам
   4) считывает массив целых чисел 6×4 по строкам
   5) заменяет все элементы 1-й строки на единицы
   6) определяет количество знаков «плюс» и заменяет их на «1»
   7) заменяет все знаки «плюс» на единицы
   8) заменяет все единицы на знак «плюс»
   9) выводит на экран полученный массив в строку
   10) выводит на экран полученный массив в виде таблицы
2. Определите тип переменной s.
3. Замените выражение логического типа логической переменной a. Как изменится код?
4. Что изменится, если процедуру write заменить на print?

CC BY

5. Определите тип переменных i и j. Измените код так, чтобы переменные i и j получили тип byte.

Пример 2. Изучите программный код и выполните задания.

```
const n = 5;
begin
  var s: ...;
  for var i := 1 to n do readln(s[i]);
  var m := 0;
  var k := 1;
  repeat
    if s[k] in ['a'..'l'] then
    begin
     s[k] := '0';
     inc(m);
    end;
    k += 1
  until k > n;
  println('m =', m);
  foreach var a in s do print(a)
end.
```

1. Программный код выполняет следующие действия:

1) считывает строку, определяет в ней количество m букв a…l, вставляет после каждой из них пробел и выводит на экран значение m и новую строку;

2) считывает массив символов, определяет в нем количество m букв от a до l, выводит на экран значение m и элементы массива в столбец;

3) считывает строку, определяет в ней количество букв a…l, удаляет эти буквы из строки и новую строку выводит на экран;

4) считывает массив символов, определяет в нем количество m букв a…l, заменяет их нулями и выводит на экран значение m и элементы массива в строку;

5) считывает строку, определяет в ней количество m букв a…l, заменяет их пробелами и выводит на экран значение m и новую строку.

2. Определите тип переменных s, k, m и a.

3. Выпишите строку, содержащую конструктор множества, и измените её так, чтобы множество не использовалось

4. Замените оператор цикла с постусловием оператором цикла с предусловием.

5. Замените оператор foreach:

а) оператором for;

б) оператором repeat..until.

CC BY

6. Можно ли заменить оператор foreach оператором loop? Ответ обоснуйте.

Пример 3. Изучите программный код и выполните задания.

```
begin
  var s := Readln...;
  var k := 1;
  var t1 := '';
  var m := 0;
  repeat
    if (s[k] >= '0') and (s[k] <= '9') then
      begin
        inc(m);
        t1 += s[k];
      end;
    inc(k)
  until k > length(s);
  println(s);
  println('m =', m, t1)
end.
```

1. Программный код выполняет следующие действия:

1) ввод действительной переменной;

2) ввод строки;

3) вычисляет количество положительных элементов в массиве;

4) определяет, содержит ли строка знаки препинания;

5) определяет количество цифр в строке;

6) выводит цифры на экран в виде целого числа;

7) записывает цифры в новую строку;

8) выводит на экран новую строку;

9) выводит на экран количество цифр в строке;

10) выводит на экран новый массив, составленный из цифр, содержащихся в строке.

2. Определите тип переменных s, t1, k, m. Может ли переменная m иметь тип boolean? тип byte?

3. Замените оператор цикла repeat оператором while.

4. Оформите условие, которое используется в первой строке оператора цикла, в виде отдельной функции.

5. Можно ли оператор println(s) оформить с помощью оператора цикла? Ответ обоснуйте.

6. Каким образом можно изменить код, чтобы постусловие имело вид until k > N?

CC BY

Анализ методологической и научно-методической литературы, а также многолетняя практика преподавания дисциплин, связанных с программированием на языках высокого уровня, свидетельствует о повышенном интересе к этому виду деятельности, поскольку во многих образовательных программах, как правило, присутствует целый спектр дисциплин этого типа. Современные тренды рынка труда показывают, что программисты востребованы как у компаний ИТ-сектора, так и в других отраслях экономики.

Все это свидетельствует о необходимости чёткой проработки учебного материала в рамках дисциплин, предполагающих освоение технологий и методов программирования, а также использование компетентностного подхода при разработке разноплановых оценочных материалов, мотивирующих обучающихся и максимально приближенных к реальным прикладным задачам.

Литература

1. Орлов А.А. Технологии оценивания результатов образовательного процесса в вузе в контексте компетентностного подхода: учебное пособие для преподавателей и студентов. М.; Берлин: Директ-Медиа, 2017. 127 с.

2. Белкина В.Н. Методика преподавания: оценка профессиональных компетенций у студентов. М.: Юрайт, 2021. 212 с.

3. Савина Е.В. Практикум по программированию на PascalABC.NET. М.; Берлин: Директ-Медиа, 2021. 123 с.

4. Слизкова Е.В. Виды оценочных средств. Подготовка практико-ориентированного педагога. М.: Юрайт, 2021. 138 с.

© Савина Е.В., Раджабов К.Я., 2022

УДК 519.687.7
https://doi.org/10.36906/AP-2022/75

Садвакасова К.М.
Колледж экономики, бизнеса и права Карагандинского экономического университета Казпотребсоюза
г. Караганда, Казахстан

ИСПОЛЬЗОВАНИЕ МОДУЛЬНОЙ ТЕХНОЛОГИИ НА УРОКАХ

Аннотация. *Модуль* – определённый вид работы, который выполняют студентов. В педагогической литературе модуль определяется как «целевой, функциональный узел обучения, который объединяет учебное содержание и технологию овладения им». Методическая сущность модульной технологии – это предоставление студентам центрального места в системе «преподаватель–студент». При систематическом использовании данной технологии реализуются все навыки «само» студентов: самообучение, самоопределение, самоконтроль, самооценка, самоанализ, самореализация.

Ключевые слова: модуль; обучение; компетентность; учебный элемент.

Sadvakasova K.M.
College of Economics, Business and Law of Karaganda Economic University of Kazpotrebsoyuz
Karaganda, Kazakhstan

USE OF MODULAR TECHNOLOGY IN THE LESSONS

Abstract. Module is a specific type of work that students do. In pedagogical literature, a module is defined as “a target, functional learning unit that combines educational content and the technology of mastering it”. The methodological essence of modular technology is to provide students with a central place in the teacher-student system. With the systematic use of this technology, all students' self-skills are realized: self-study, self-determination, self-control, self-esteem, introspection, self-realization.

Keywords: module; learning; competence; learning element.

На современном этапе развития учебно-воспитательного процесса наблюдается постепенный отказ от приоритетного формирования ЗУН в чистом виде. Центр тяжести переносится на формирование способности личности студенты, особенно способности её к самообразованию, к самостоятельному получению знаний, умений и отработке навыков. Все эти категории входят в понятие «компетентность». Воспитание компетентного человека и должно служить главной конечной целью образовательного процесса в колледже. В связи с

CC BY

этим предъявляются новые требования к системе организации и проведения учебно-воспитательного процесса, предпринимаются попытки его «технологизации».

Модульная система организации учебно-воспитательного процесса имеет некоторые отличия принципиального характера от традиционной системы. Содержание обучения представляется в законченных, самостоятельных модулях, одновременно являющихся банком информации и методическим руководством по его применению. В основе такого обучения лежат субъект-субъектные отношения между преподавателем и студентом. Обеспечивается самостоятельное, осознанное достижение определенного уровня в учении. Наблюдается высокая степень адаптивности элементов к условиям педагогического процесса.

К ведущим принципам модульного обучения можно отнести:

~ мобильность;

~ структуризацию содержания обучения;

~ динамичность;

~ действенность и оперативность знаний;

~ гибкость;

~ осознанную перспективу;

~ разносторонность методического консультирования.

Каждый *учебный элемент (УЭ) модульного урока* – шаг к достижению интегрирующей цели урока, без овладения содержанием которого эта цель не будет достигнута. Учебных элементов не должно быть много (не более семи), но среди них обязательно должны присутствовать следующие:

– УЭ-0 – направлен на определение интегрирующей цели по достижению результатов обучения;

– УЭ-1 – включает задания по выявлению уровня знаний по теме, задания, направленные на овладение новым материалом и т. д.;

– УЭ-2 (и т. д.) – отработка учебного материала;

Завершающий УЭ – включает выходной контроль знаний, подведение итогов занятия (оценка степени достижения целей урока), выбор домашнего задания (оно должно быть дифференцированным, с учётом успешности работы учащегося на уроке), рефлексию (оценку своей работы с учётом оценки окружающих).

Модульные уроки имеют свои особенности. Одна из них заключается в том, что каждый такой урок целесообразно начинать с процедуры мотивации – может быть обсуждение эпиграфа к уроку, использование входного теста самопроверкой, небольшого математического диктанта и т. п. [1]

Модульные занятия отличаются от обычного урока тем, что они строятся в логике процесса усвоения знаний и представляют собой полный цикл познания, совпадающий по своей структуре с циклом учебной деятельности – описание, объяснение, проектирование (обычные же уроки строятся в такой логике: проверка домашнего задания, изучение нового материала, его закрепление, задание на дом).

CC BY

Начинается модульное занятие с *целеполагания.* Следующий этап в модульном занятии – *мотивация* на усвоение содержания и учебную деятельность. Это различного рода интеллектуальные разминки, математические диктанты, небольшие тесты. Далее идёт *информационный блок:* содержание в виде рассказа учителя, лекции, фильма, сообщений учащихся, чтения учебника или комбинаций этих компонентов. Далее – *отработка материала:* практические работы, решение учебных задач, проблем, ответы на вопросы, выполнение заданий, игры, конференции и др. На этом этапе используются «мягкие» формы контроля – само- и взаимоконтроль. Заканчивается модульное занятие экспертным контролем (контроль преподавателя), коррекцией знаний и умений с постоянной рефлексией относительно целей учебной деятельности. *Экспертный контроль* - это обычная проверочная работа, зачёт, устный опрос или итоговый тест. Особенность коррекции в модульном обучении заключается в том, что она проводится сразу же после контроля, на том лее уроке, а не на следующем, как при традиционном обучении.

На каждом модульном занятии как обязательный элемент проводится *рефлексия* (оценка себя, своей деятельности). В конце каждого урока студенты возвращаются к целям занятия и оценивают степень их достижения и свою работу на уроке. Обратите внимание – свою работу. В ходе модульного занятия определяется исходный уровень знаний и умений учащихся, затем они получают информацию по изучаемой теме, отрабатывают учебный материал, в конце урока проводятся контроль и коррекция знаний и умений. Поэтому, модульные занятия по времени проводятся не менее чем за пару. На модульных уроках студенты могут работать индивидуально, парами, в группах постоянного и переменного состава. Форма посадки свободная, каждый из них имеет право выбора: один он будет работать или с кем-либо из товарищей.

Преимущества и недостатки модульного обучения

Технология модульного обучения создаёт надёжную основу для индивидуальной и групповой самостоятельной работы обучающихся и приносят до 30% экономии учебного времени без ущерба для полноты и глубины изучаемого материала. Кроме того, достигается гибкость и мобильность в формировании знаний и умений обучающихся, развивается их творческое и критическое мышление.

Достоинства модульного обучения

1. Цели обучения точно соотносятся с достигнутыми результатами каждого студента.

2. Разработка модулей позволяет уплотнить учебную информацию и представить её блоками.

3. Задаётся индивидуальный темп учебной деятельности.

4. Поэтапно-модульный контроль знаний и практических умений даёт определенную гарантию эффективности обучения.

5. Достигается определенная «технологизация» обучения. Обучение в меньшей степени становится зависимым от педагогического мастерства преподавателя.

6. Обеспечение высокого уровня активизации учащихся на уроке.

7. Первоочередное формирование навыков самообразования.

CC BY

Литература

1. Яковлева О., Кондратьева Н., Семенова М. Модернизация образования: модульное обучение // Еженедельная учебно-методическая газета «Математика». 2004. № 15.

© Садвакасова К.М., 2022

УДК 004.92(076.5)
https://doi.org/10.36906/AP-2022/76

Сайидова Н.С.
ORCID: 0000-0002-8910-8987, канд. физ.-мат. наук
Бухарский государственный университет
г. Бухара, Узбекистан

СОЗДАНИЕ АНИМИРОВАННЫХ КНОПОК УПРАВЛЕНИЯ С ПОМОЩЬЮ ГРАФИЧЕСКОГО РЕДАКТОРА CorelDRAW

Аннотация. В статье рассмотрены пути создания кнопок управления для Веб-страниц с помощью инструментов программы Corel Draw.

Ключевые слова: вектор; элемент; изображение; графические объекты; эффекты; меню; векторный объект; растровый объект; параметры; панели инструментов; инструмент.

Sayidova N.S.
ORCID: 0000-0002-8910-8987, Ph.D.
Bukhara State University
Bukhara, Uzbekistan

CREATING ANIMATED CONTROL BUTTONS USING THE CorelDRAW GRAPHICS EDITOR

Abstract. In clause the ways of creation of buttons of management for Web-pages with the help of tools program Corel Draw are considered.

Keywords: vector; element; image; graphic objects; effects; vector object; parameters; panels of tools; tool.

Образ, формируемый средствами векторной графики, строится из набора простых графических объектов, соответствующих его типовому элементу (рис. 1). Основным элементом векторного изображения является линия.

В памяти компьютера эта строка не занимает слишком много места, так как в памяти отображаются только параметры строки. В нем простые объекты объединяются в сложные объекты, поэтому векторную графику также называют объектно-ориентированной графикой. Несмотря на то, что в памяти компьютера векторная графика сохраняется в виде линий, изображение выводится на экран в виде точек. Он вычисляет каждый параметр перед выводом изображения на экран. Вот почему говорят, что векторная графика является вычислимой графикой. С помощью векторной графики можно выполнять более простые виды декорирования.

CC BY

Важной характеристикой векторной графической системы служит состав ее простейших графических элементов, приведем несколько примеров простейших графических элементов.

Точка, координата которой задана, – это линия, начало которой направлено под определенным углом и имеет заданную длину.

Эллипс, центр которого расположен в указанной точке с координатами, длиной полуосей, цветом и толщиной линии, а также цветом её окраски.

Прямоугольник с координатами точки, длины сторон, толщины линии и цвета, расположенный в верхнем левом углу.

Линейные геометрические признаки простейших элементов при отображении векторного изображения вычисляют с учетом соотношения его исходного и необходимого размеров, а также допустимой погрешности в устройствах вывода (мониторы, печатающее устройство). Такой способ значительно увеличивает возможность масштабирования. Независимо от того, насколько сильно вы увеличиваете размер изображения, его качество не меняется. При слишком малых размерах изображения линии простейших элементов могут перекрываться, а за счет сжатия ошибок-изменяться (искажаться).

Программы для работы с векторной графикой CorelDRAW, AutoCAD, 3DS MAX - в них каждое изображение строится на основе определенной формы и линии. Это связано с тем, что с помощью программ можно работать с двумерными или трехмерными кадрами, при масштабировании объектов качество не меняется, а большая инструментальная база предоставляет большие возможности для пользователей. Созданные файлы хранится в таких форматах, как cdr, dwg.

Изображение, формируемое средствами растровой графики, используется в основном в электронных и полиграфических издательствах (рис. 2). Растровое изображение состоит из набора точек в виде двумерного массива (матрицы), называемых пикселями. Растровое изображение состоит из мельчайших пикселей.

Рис. 1. Векторное изображение

Рис. 2. Растровое изображение

Его атрибуты не зависят от атрибутов других пикселей. В зависимости от возможностей операционных систем, применяемых на компьютере, могут располагаться изображения с разрешением 480x640, 800x600, 1024x768 и более пикселей. В зависимости от размера изображения увеличивается и его емкость. Ёмкость экрана параметрическая и определяется количеством точек на дюйм. Изображения, выполненные (подготовленные) средствами растровой графики, лишь в очень редких случаях обрабатываются с помощью компьютерных

CC BY

программ. С этой целью в компьютер с помощью технических средств вводится рисунок или фотография, выполненная профессиональным художником.

В последнее время для ввода растровых изображений в компьютер используются цифровые фото-и видеокамеры. Поэтому основное назначение растровой графики-не создание изображения, а обработка имеющегося изображения.

Канадская фирма, носящая название Corel, считается одним из лидеров в разработке программного обеспечения.

Corel DRAW 11, который является основой программного обеспечения фирмы Corel, был выпущен в августе 2002 года. Он имеет большой потенциал в производстве рекламной продукции, подготовке публикаций и создании изображений для своих веб-страниц.

Corel DRAW – графическая программа, которая помогает вам создавать векторные изображения, графические тексты, а также воплощать в жизнь все ваши творческие идеи в вашем воображении [1].

Все изображения, с которыми работает программа Corel DRAW, делятся на два класса: точечные и векторные. В векторной графике линия рассматривается как основной элемент изображения. В растровой графике такие линии создаются с помощью точек (пикселей), тогда как в векторной графике при создании изображений используются линии, которые имеют больше общего с точкой, за счет чего изображения получаются более четкими.

Произвольное изображение векторной графики состоит из линий, а из простых линий образуются сложные. Математической основой векторной графики является изучение свойства геометрических фигур. Представление векторных изображений в памяти компьютера сложнее, чем точечных. Недостаток точечных изображений - для их хранения в памяти компьютера требуется большое пространство [2].

Часто при работе с такими большими объектами скорость работы современных компьютеров оказывается недостаточной. Работать с векторным изображением намного проще. Чтобы сделать его большим или маленьким, можно изменить только параметр изображения, которым он управляет. При этом размер файла векторного изображения не превышает даже одного байта.

Когда данные передаются через изображения, скорость их анализа и восприятия намного выше, чем при передаче информации по слуховым каналам. Вот почему даже один рекламный плакат, состоящий из изображения (которое может легко дать желаемый эффект), оказывает более сильное влияние на человека, который его смотрит, чем несколько серий объявлений.

Информация любого объема хорошо воспринимается человеком, если передается по зрительному каналу (детям нравятся книги с картинками). В другом формате такая информация недопустима (таблицы на биржах). Вот почему потребность в графических данных растет день ото дня.

Программные средства, предназначенные для работы с такими графиками и изображениями, гораздо более многочисленны и разнообразны. Одной из таких программ является программа Corel DRAW.

CC BY

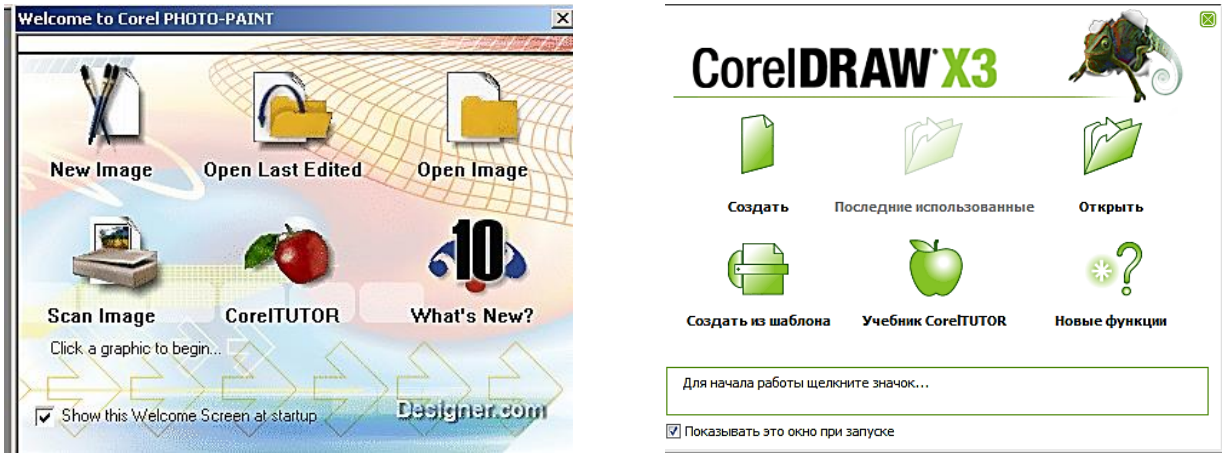
12-я версия Corel DRAW, сочетающая в себе современные возможности, была выпущена в 2004 году (13-я версия Corel DRAW была выпущена в 2005 году). Это программное обеспечение используется в промышленном дизайне, разработке рекламных продуктов, издательских системах и для создания различных изображений для веб-страниц. Corel DRAW является пакетной программой и включает в себя несколько разделов. Это обеспечивает легкий процесс обмена информацией между программами.

Для запуска графического редактора Corel Draw дается команда Pusk-Programming-Corel Draw Graphics Suite X3-Corel Draw X3 или двойной щелчок левой кнопкой мыши по его ярлыку на рабочем столе Windows XP [3].

Файлы, созданные в программе Corel DRAW принимает расширения, такие как CDR, CMX, WMF.

После загрузки программы на экране появится окно Corel DRAW, состоящее из основных элементов пользовательского интерфейса.

Как только программа CorelDRAW запустится, откроется следующее диалоговое окно.



a) b)

Рис. 3. Диалоговое окно программы CorelDRAW
(а) – CorelDRAW 10, б) – CorelDRAW X3)

Где: New – создание нового файла, Recently Used – ссылка на пять последних использованных файлов. Open – открыть файлы, присутствующие в памяти компьютера. New from template – создание нового файла на основе существующих шаблонов. Corel TUTOR – Электронная книга, в формате pdf, дающая рекомендации поработе в программе CorelDRAW. What's New – информация о том, чем отличается программа от предыдущих версий и о введённых нововведениях [4]. После выбора любого из вышеперечисленных (в нашем случае New Partition) в зависимости от цели выполняемой работы откроется рабочее окно программы.

CorelDRAW. Он состоит из следующих разделов:

1. Строка программы, в которой отображается текущая программа и имя файла;

2. Menu Bar – строка меню, в которой все команды, доступные в программе CorelDRAW, сгруппированы в эти меню;

3. Стандарт – Панель Управления по умолчанию включает в себя такие команды, как создание нового файла, открытие существующих файлов, сохранение, печать, общие для всех приложений;

4. Property Bar – панель форматирования, в которой без указания объекта отображается свойство листа, а с указанием объекта – его свойства;

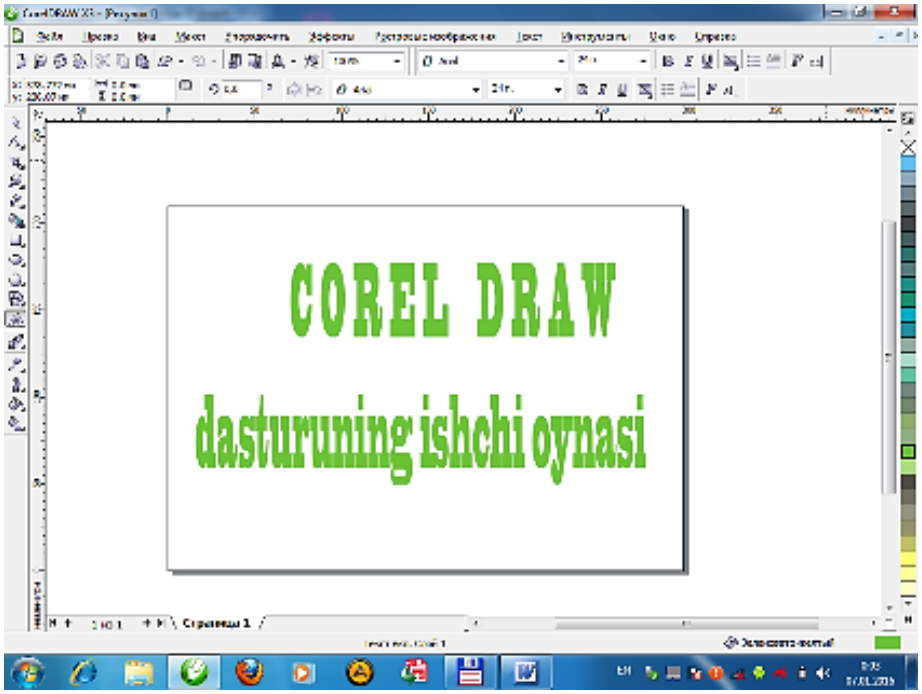


Рис. 4. Рабочее окно программы CorelDRAW X3

5. Toolbox – включает в себя панель инструментов, т. е. все инструменты, доступные в программе, и предоставляет доступ к ним.

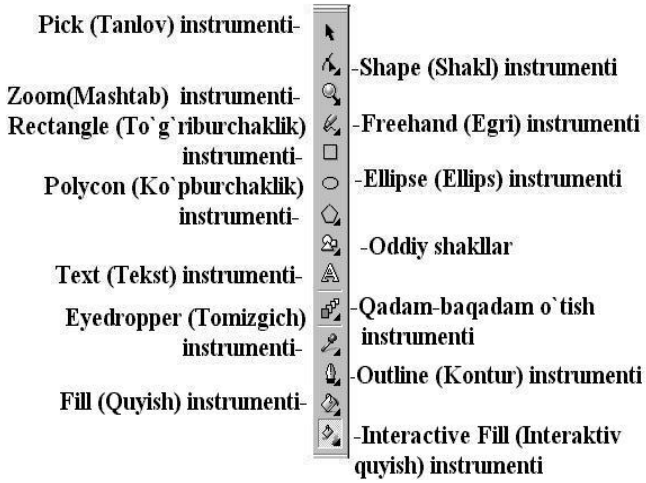


Рис. 5. Панель инструментов

6. Color Palettes – цветовые палитры, отображающие цвета в выбранной цветовой палитре, можно вывести около десяти других цветов, удерживая левую кнопку мыши на каком-либо цвете в течение 3 секунд.

CC BY

Вывод и удаление этих панелей осуществляется щелчком правой кнопки мыши в правом верхнем углу экрана и выбором нужного названия панели из открывшегося списка панелей. Их можно разместить в любой части экрана.

Главное меню. Строка главного меню находится в верхней части экрана Corel DRAW, чуть ниже строки заголовка строки. Это единственный элемент в программе, который не может быть размещён в любом месте экрана, он имеет определенное место. Главное меню состоит из 11 пунктов. В это меню пользователь также может добавить дополнительные пользовательские команды.

Стандартная строка меню: файл, расположение, вид, макет, расположение, эффекты, растровые изображения, текст, инструменты, окно, справка.

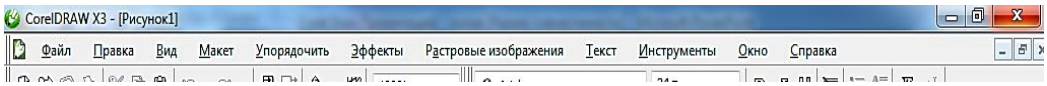


Рис. 6. Главная строка меню программы CorelDRAW X3

Стандартная панель инструментов. Панель инструментов по умолчанию находится под строкой главного меню. На этой панели расположены кнопки, обеспечивающие быстрое и удобное нахождение различных функций Corel Draw. Помимо этих кнопок, на панели есть раскрывающийся список с именем Zoom levels (Масштаб).

Рис. 7. Стандартная панель инструментов программы CorelDRAW X3

Функции кнопок стандартной панели инструментов:

1 – создать новый документ;

2 – Открыть существующий документ;

3 – сохранить документ;

4 – печать документа;

5 – соскоб;

6 – копирование;

7 – заливка;

8 – отмена последнего действия;

9 – повторение последнего действия;

10 – импорт (подключение одного документа к другому);

11 – экспорт (отправка файлов);

Модуль загрузки приложения 12;

13 – доступ к сайту Corel DRAW;

14 – уровень масштабирования.

Панель атрибутов. Панель атрибутов – панель, в которой содержимое и инструментальные средства меняются в зависимости от того, какой инструмент или объект

активен в данный момент. Панель атрибутов находится под панелью инструментов по умолчанию.

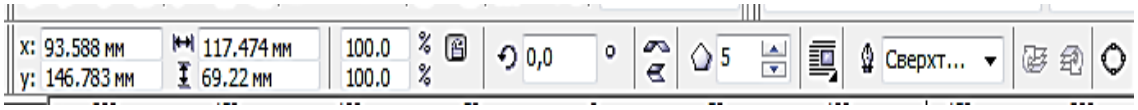


Рис. 8. Панель атрибутов программы CorelDRAW X3

Вызов меню производится с помощью щелчка правой кнопкой мыши. При нажатии правой кнопки вызывается контекстное меню.

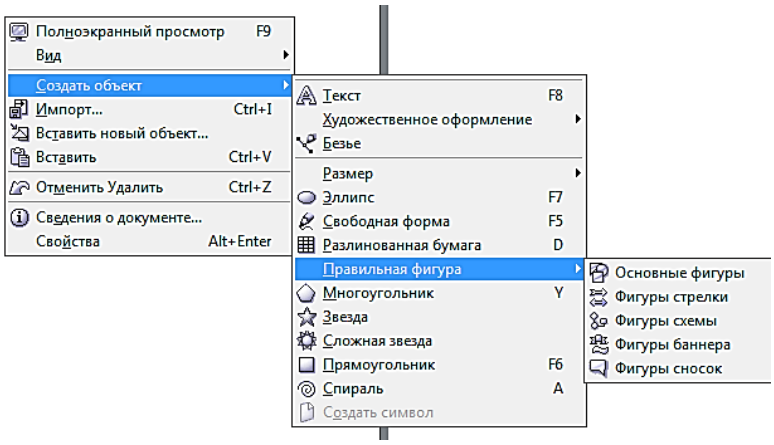


Рис. 9. Контекстное меню chican с помощью щелчка правой кнопкой мыши
в программе CorelDRAW X3

Объектное меню. Вызываем это меню, наведя на него выделенную группу объектов и щёлкнув правой кнопкой мыши. Команды в таких меню называются контекстными, так как зависят от состояния в вызываемые моменты.

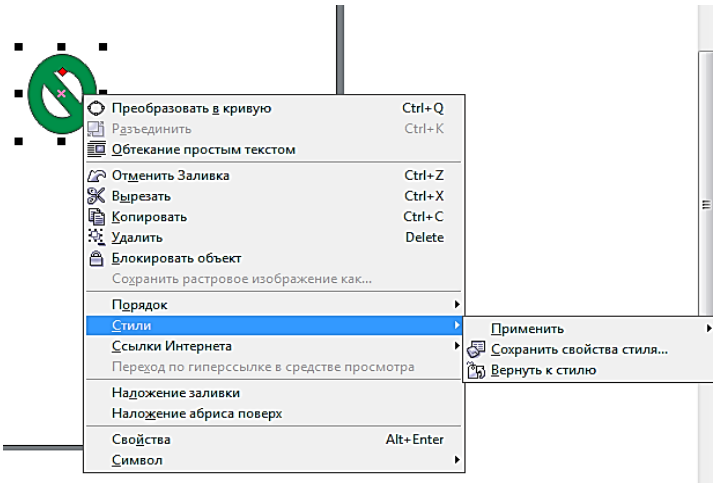


Рис. 10. Объектное меню программы CorelDRAW X3

Суть создании анимированных кнопок заключается в том, что при наведении указателя на графические ссылки на Web-страницах и при нажатии кнопки мыши происходит изменение

их внешнего вида, при этом изображение воспринимается более впечатляюще. Создание таких объектов в программе CorelDRAW автоматизировано.

В качестве примера рассмотрим создание кнопки, которая меняет свой внешний вид. Сначала необходимо создать представления, показывающие, когда кнопка нажата, а когда нет. Создайте прямоугольник с дугой вокруг него и залейте его черным цветом. Из этого прямоугольника сделайте копию вверху и слева от него, залейте ее белым цветом. Установите запись сверху, напишите текст с надписью, например, кнопка, и выделите красным внутри белым. Готов вид кнопки в не нажатом состоянии (рис.9, слева). теперь нужно создать из него объект-переменную.

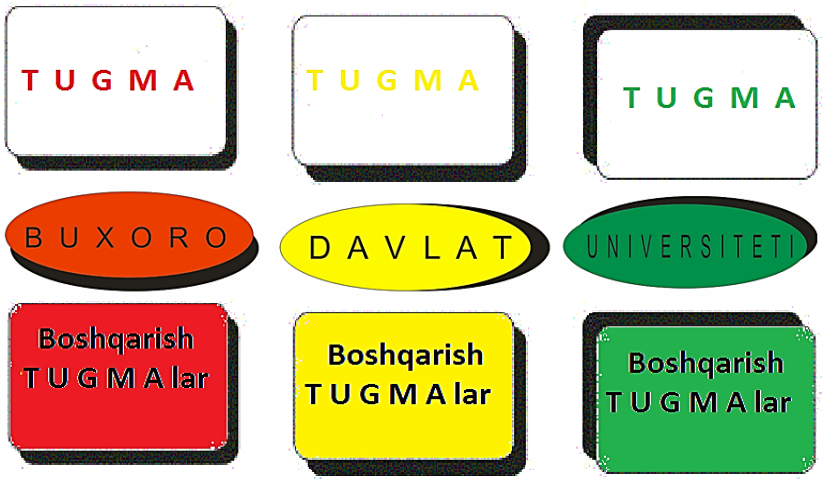


Рис. 11. Различные детали, кнопки в программе CorelDRAW X3

Выделите изображение кнопки и выберите в меню команду Effects Rollover Create Rollover (создать новую кликабельную кнопку). Внешне не видоизменяется, а вместо нескольких объектов появляется одна обычная кликабельная кнопка, о чем свидетельствует надпись Rollover [NORMAL] (кликабельная кнопка [Simple]) в строке состояния.

Чтобы изменить изображение в других режимах, щелкните правой кнопкой мыши и выберите команду Edit Rollover (изменение кликабельной кнопки) в меню утилиты. Все изображения, кроме кнопки, которая будет изменена, станут невидимыми, и на экране появится дополнительная панель инструментов интернета (Internet). Если вы часто создаёте картинку для Интернета, то стоит установить эту панель в окне программы. Если панель оборудования Internet все время остаётся видимой, то при создании кнопки не обязательно давать команду меню, а достаточно нажать кнопку «Create Rollovers» на панели. Переход в режим конвертации осуществляется нажатием кнопки «Edit Rollover».

После переключения в режим конвертации вам нужно будет создать изображение для двух оставшихся случаев кнопки. Из списка слева от панели Internet установите флажок OVER (сверху). В результате указатель мыши переключится в режим настройки положения направленной кнопки. До этого был выбран вариант NORMAL, который показывал простой вид кнопки. Сотрите строку текста и заполните её черным цветом. Кнопка (жёлтая кнопка) будет выглядеть так же, как и при наведении указателя мыши (рис. 9, по центру).

CC BY

Выберите раздел «вниз» (щелчок) из списка параметров изображения. теперь необходимо будет нарисовать вид напечатанной кнопки. Мы будем использовать первое структурированное представление кнопки. Закрасьте текст черным цветом и сотрите контур. Перетащите текст (зелёная кнопка) и белый прямоугольник вниз, а черный прямоугольник – в противоположную сторону. Сформировался вид нажатой кнопки (рис. 9, справа).

Нажмите кнопку «Готово редактирование», чтобы завершить процесс создания кнопки и перейти в начальный режим работы программы. Теперь можно увидеть, как выглядит созданный объект. Нажмите кнопку «Изменить» на панели Internet. Можно просмотреть анимацию кнопок. При установке указателя мыши на созданную кнопку меняется её изображение. Нажмите и удерживайте кнопку мыши, пока не появится третья картинка с изображением кнопки. Отодвиньте указатель в сторону, отпустив кнопку мыши. Кнопка появится в виде первого изображения. Изображения кнопок также могут меняться медленно, но при использовании в Интернете они меняются быстрее. Нажмите кнопку «Готово редактирование», чтобы перейти в начальный режим.

Если нажать кнопку «Извлечь объект» с панели Internet после того, как анимированная кнопка будет отключена, все части будут разделены в виде отдельных объектов, расположенных друг над другом. В этом можно убедиться, переместив их мышкой в другой раздел. Чтобы посмотреть, как выглядит готовое изображение в Интернете, нажмите кнопку Internet на панели Internet. В результате на экране появится диалоговое окно, в котором будет изменён формат интернет-документа. При сохранении в памяти создаются не только изображения, но и коды, с помощью которых можно изменять внешний вид объектов на Web-страницах.

Литература

1. Миронов Д.Ф. Corel DRAW 12. Учебный курс. СПб: Питер. 2004. 442 с.

2. Михайлов С.М. Кулеева Л.М. Основы дизайна. М: Союз Дизайнеров. 2002.

3. Mamarajabov M.E., Tursunov S.Q., Nabiulina L.M. Kompyuter grafikasi va web-dizayn: darslik. O`zbekiston Respublikasi Oliy va o`rta maxsus ta`lim vazirligi. Toshkent: Cho`lpon nomidagi NMIU. 2013. b. 376.

4. Миронов Д.Ф. CorelDraw X3. Учебный курс. СПб.: Питер. 2006. 397 с.

© Сайидова Н.С., 2022

УДК 004.92
https://doi.org/10.36906/AP-2022/77

Табачук Н.П.
ORCID: 0000-0002-2781-2805, канд. пед. наук
Поличка А.Е.
ORCID: 0000-0001-9581-3829, д-р пед. наук
Тихоокеанский государственный университет
г. Хабаровск, Россия

ИССЛЕДОВАТЕЛЬСКАЯ И ПРОЕКТНАЯ АКТИВНОСТИ СТУДЕНТОВ В ЦИФРОВЫХ ПРОДУКТАХ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ: ЦЕЛИ И ЦЕННОСТИ В РАЗВИТИИ ИНФОРМАЦИОННОЙ КОМПЕТЕНЦИИ

Аннотация. В статье подчёркивается актуальность формирования положительных цифровых следов студентов как результатов учебной деятельности в цифровом формате. Отмечается, что цифровой след формируется через исследовательскую и проектную активности студентов в образовательном процессе по получению цифровых продуктов учебной деятельности. Выделяются цели и ценности проявления исследовательской и проектной активности студентов в развитии информационной компетенции. Приводятся примеры цифровых продуктов учебной деятельности студентов как будущих учителей математики. Определяются перспективные направления исследований.

Ключевые слова: цифровизация образования; информационная компетенция студентов вуза; исследовательская и проектная активности студентов; цифровой продукт учебной деятельности.

Tabachuk N.P.
ORCID: 0000-0002-2781-2805, Ph.D.
Polichka A.E.
ORCID: 0000-0001-9581-3829, Ph.D.
Pacific National University
Khabarovsk, Russia

RESEARCH AND PROJECT ACTIVITY OF STUDENTS IN DIGITAL PRODUCTS OF EDUCATIONAL ACTIVITY: GOALS AND VALUES

Abstract. The article emphasizes the relevance of the formation of positive digital traces of students as the results of educational activities in a digital format. It is noted that the digital footprint is formed through the research and project activity of students in the educational process to obtain digital products of educational activities. The goals and values of the manifestation of research and project activity of students in the development of information competence are highlighted. Examples

CC BY

of digital products of educational activities of students as future teachers of mathematics are given. The promising areas of research are determined.

Keywords: digitalization of education; information competence of university students; research and project activity of students; digital product of educational activity.

В настоящее время современные и актуальные проекты в сфере высшего образования побуждают к пересмотру целей и ценностей развития образовательных систем вуза.

Отмеченные в паспорте национального проекта «Образование» положения о непрерывности и доступности образования, о развитии универсальных и профессиональных компетенций, востребованных в цифровом обществе, о выстраивании индивидуальных образовательных маршрутов, об исследовательской и проектной активности в профессиональной подготовке обучаемых актуализируются в эпоху цифровой трансформации (https://clck.ru/NBYCG). Уделяется внимание развитию информационной компетенции студентов вуза, которая является одной из универсальных компетенций.

Согласно ФГОС ВО 3++ в направлении подготовки «Педагогическое образование» выделим ряд характеристик информационной компетенции как универсальной компетенции, подчёркивающих актуальность исследования: способность осуществлять поиск, критический анализ и синтез информации на исследовательском уровне, способность выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни (https://clck.ru/akrnV).

В ФГБОУ ВО «Тихоокеанский государственный университет» (ТОГУ) уделяется внимание личностно-профессиональному становлению через «впитывание» универсальных и профессиональных компетенций, в том числе и информационной компетенции; разворачиванию системы сопровождения индивидуальных образовательных маршрутов; развитию у студентов активной позиции участия в проектах по формированию позитивного субъектного опыта по поддержке цифрового следа.

Целевым аспектом развития исследовательской и проектной активности студентов является формирование глубокого, безошибочного, позитивного цифрового образовательного следа в процессе развития их информационной компетенции.

Для развития исследовательской и проектной активности студентов в эпоху цифровой трансформации в образовательном процессе используются альтернативной технологии обучения, одной из которых является технология «хакатон».

Технология «хакатон» для педагогического сообщества предполагает совместную работу представителей заказчиков, преподавателей и студентов, посвящённую разработке конкретного цифрового образовательного продукта или развитию виртуального образовательного стартапа. Хакатон подразделяется на отдельные этапы деятельности, которые называются треками. Результатом хакатона выступает готовый прототип инновационного продукта (проект – в образовательной деятельности) (https://clck.ru/dYad3).

Технология «хакатон» позволила студентам ТОГУ выступать в роли модераторов цифровых образовательных продуктов, осуществляя выбор по критериям из многообразия

CC BY

цифрового инструментария (образовательных онлайн-платформ), определение стратегии их использования и интеграции в цифровую образовательную среду вуза.

Ценностным аспектом проявления исследовательской и проектной активности в цифровых продуктах учебной деятельности для студентов является осмысление индивидуальных образовательных маршрутов с ориентиром на будущее, поиск собственной аудитории для использования стартапов. Такие цифровые продукты есть положительный цифровой след в открытом цифровом образовательном пространстве. В проведённых ранее исследованиях нами подчёркивалось, что цифровой след есть авторский подход к самореализации в реальном и виртуальном мире [1; 2].

В рамках исследовательской и проектной деятельности студентами направления подготовки «Педагогическое образование» с двумя профилями подготовки «Математика», «Информатика» накоплен опыт педагогического и информационного сопровождения собственных идей и проектов, формирования позитивного цифрового следа как цифровых продуктов учебной деятельности студентов.

Приведём примеры цифровых продуктов учебной деятельности студентов как будущих учителей математики и информатики как показано в таблице.

Таблица

Примеры цифровых продуктов учебной деятельности
студентов как будущих учителей математики и информатики

Тематика исследований студентов	Образовательные онлайн-платформы	Адрес в сети Интернет	Сфера использования
«Развитие цифровых учебно-методических материалов по математике для школьников: на примере раздела «Степенные функции»»	OnlineTestPad	https://onlinetestpad.com/o4fl4qsi2pyd2	процесс обучения математике в школе
«Модели организации внеурочной деятельности по математике в школе: на примере раздела «Тригонометрические функции»»	OnlineTestPad, eTreniki, Банкт тестов, StoryJumper, Stepik, LearningApps, A5.ru	http://2280466.mya5.ru	организация внеурочной деятельности по математике в школе
«Образовательные платформы цифрового обучения школьников элементам теории множеств»	Stepik	https://stepik.org/course/103631/promo	онлайн-обучение математике школьников
«Цифровые образовательные ресурсы для изучения раздела «элементы теории множеств» в школьном курсе математики»	Thinglink Google формы	https://www.thinglink.com/scene/1506845171629686786 https://forms.gle/nRUndPXwZ3MV5bR29	проведение интерактивных занятий по математике

CC BY

«Интерактивный модульный курс по компьютерным сетям на основе сервиса LearningApps»	LearningApps	https://learningapps.org/display?v=puanxg8wn17	интерактивные занятия по компьютерным сетям
«Проектирование интерактивных модулей по дисциплине «Информационные системы» средствами приложений Learningapps.org»	LearningApps	https://learningapps.org/5257793 https://learningapps.org/5257798 https://learningapps.org/5257805 https://learningapps.org/5308831	интерактивные занятия по информационным системам

Проявленные студентами исследовательская и проектная активности выражаются в создаваемых цифровых продуктах учебной деятельности и апробируются в процессе педагогической практики по организации изучения математических разделов в школе, в процессе самостоятельной работы школьников и внедрения дистанционных образовательных технологий, в сфере дополнительного образования школьников.

Внедрение данных проектов-стартапов в образовательный процесс в школе, во время внеурочной деятельности, в сфере дополнительного образования позволяет осуществлять «самопроектирование образования для себя» (А. М. Кондаков) (https://clck.ru/PoShb; https://clck.ru/dYaVz).

Проявление студентами исследовательской и проектной активности в цифровых продуктах учебной деятельности способствует возникновению новых творческих идей и проектов в сфере образования, формированию положительного цифрового следа в открытом цифровом образовательном пространстве, развитию высокого уровня информационной компетенции.

На кафедре математики и информационных технологий ТОГУ работа в данном направлении ведётся ряд лет, в частности, в 2020–2021 уч. г. разработаны и реализованы более 10 студенческих проектов, которые способствовали повышению уровня развития информационной компетенции студентов и были защищены в форме выпускной квалификационной работы (ВКР как стартап). По результатам внедрения получены экспертные заключения руководителей ряда образовательных учреждений Хабаровского и Приморского краёв, где студенты проходили педагогическую практику.

Перспективными направлениями исследования являются поиски ответов на вопросы: «Как повысить мотивацию пассивных студентов к исследовательской и проектной активности?», «На сколько исследовательская и проектная активности влияют на профессиональную деятельность в будущем?», «В чем ценность цифрового образовательного следа?».

Исследования в данном направлении продолжаются членами Академии информатизации образования (Хабаровского отделения) в рамках реализации концепции развития ТОГУ как участника Федеральной программы «Приоритет-2030».

CC BY

Литература

1. Tabachuk N.P., Polichka A.E., Dvoryankina E.K., Karpova I.V. «Digital Image» in the Methodological System of Information Competency Development by University Students // International Journal of Applied Exercise Physiology. 2020. Vol. 9. No. 2. P. 81-87.

2. Табачук Н.П. Формирование «цифрового образа» личности в условиях цифровой трансформации // Воспитание в современном мире: новые контексты – новые решения: матлы V Международной научно-практической конференции (г. Хабаровск, 29-30 октября 2019 года). Хабаровск, 2019. С. 270-273.

© Табачук Н.П., Поличка А.Е., 2022

Содержание

РЕЗОЛЮЦИЯ **4-8**

СРЕДЫ, ИНСТРУМЕНТЫ, МЕТОДЫ ПРОГРАММИРОВАНИЯ

Васильев В.С., Непомнящий О.В.
ВНУТРЕННИЕ ПРЕДСТАВЛЕНИЯ В СИСТЕМЕ ОПТИМИЗАЦИИ ФУНКЦИОНАЛЬНО-ПОТОКОВЫХ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ **9-15**

Горелик А.А., Овчинников А.В., Алимбеков Д.Р., Огрызков А.В.
ИСПОЛЬЗОВАНИЕ СРЕДСТВ И ТЕХНОЛОГИЙ МОБИЛЬНОЙ РАЗРАБОТКИ ДЛЯ СОЗДАНИЯ НАВИГАЦИОННОГО ПРИЛОЖЕНИЯ **16-22**

Димитриев А.П.
ОБ ЭФФЕКТИВНОСТИ НЕКОТОРЫХ ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ КОММИВОЯЖЁРА **23-29**

Земцов А.В., Богомолов А.В.
СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ . **30-33**

Ибраев Т.Б., Касымова А.Х.
АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ НАГРУЗКИ ПРЕПОДАВАТЕЛЯ **34-40**

Никонова Е.З., Королев Р.И.
ОБЗОР ПРОТОКОЛОВ ИНТЕРНЕТА ВЕЩЕЙ **41-46**

Ужаринский А.Ю., Стычук А.А., Короткий А.В., Аршинова А.А.
ТЕХНОЛОГИЯ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ В ПОСТРОЕНИИ СОВРЕМЕННЫХ ПРИЛОЖЕНИЙ **47-51**

Слива М.В., Левкин Л.В.
КОНТЕЙНЕРИЗАЦИЯ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ПЛАТФОРМЫ DOCKER **52-57**

Мамедли Р.Э.
ВОССТАНОВЛЕНИЕ БАЗЫ ДАННЫХ ИЗ НЕСОВМЕСТИМОГО СОСТОЯНИЯ В РАНЕЕ СОГЛАСОВАННОЕ СОСТОЯНИЕ **58-61**

Позов Р.А., Рассадкин Е.Ю.
СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПЛАТФОРМ ДЛЯ СОЗДАНИЯ САЙТОВ **62-68**

Волков В.Н., Новиков С.В., Герасимова Н.М., Полонская Т.С.
ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ФРЕЙМВОРКА VUE.JS ДЛЯ СОЗДАНИЯ ГИБКИХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ **69-74**

Рачев А.А., Харсеев Н.А.
ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ БИБЛИОТЕЧНЫХ КАТАЛОГОВ **75-80**

Русина Л.Г.
О МЕТОДАХ РЕШЕНИЯ «ЖЁСТКИХ» СИСТЕМ УРАВНЕНИЙ **81-87**

CC BY

Смирнова Т.Н., Румянцев А.А., Антонов Б.В., Наумов К.Д.
РАЗРАБОТКА ПРОГРАММНОГО ПРИЛОЖЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ НАД МНОЖЕСТВАМИ 88-94

Ташатов Н.Н., Абылхан С.Б.
ХРАНЕНИЕ ДАННЫХ В ОБЛАЧНЫХ ХРАНИЛИЩАХ 95-101

Каледин В.О., Решетникова Е.В., Ульянов А.Д., Паульзен А.Е.
СРЕДСТВА РЕАЛИЗАЦИИ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ В СРЕДЕ ФУНКЦИОНАЛЬНО-ОБЪЕКТНОГО ПРОГРАММИРОВАНИЯ 102-107

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В СЛОЖНЫХ СИСТЕМАХ

Баскаков С.А., Ногтев С.С., Антонов И.М., Чемоданов В.Б.
МАТЕМАТИЧЕСКАЯ МОДЕЛЬ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА ДЛЯ МОБИЛЬНОГО ПИЛОТАЖНОГО КОМПЛЕКСА 108-113

Сулейменова С.Т., Баяхметова К.Е.
АВТОМАТИЗИРОВАННАЯ СИСТЕМА ЧАСТОТНОГО РЕГУЛИРОВАНИЯ ПРИВОДА КАСКАДНОЙ НАСОСНОЙ СТАНЦИИ ПРОИЗВОЛЬНОЙ КОНФИГУРАЦИИ 114-120

Гладких Т.Д.
К ВОПРОСУ АНАЛИЗА НАДЁЖНОСТИ СИСТЕМ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ 121-126

Доржиева Э.С., Романенков А.М.
МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ЧЕЛОВЕКА ПРИ ПРЫЖКЕ 127-132

Казымов Б.З., Зейналов Р.М.
ОПРЕДЕЛЕНИЕ ДИНАМИКИ ПОТРЕБНОГО КОЛИЧЕСТВА СКВАЖИН ПРИ РАЗРАБОТКЕ ГЛУБОКОЗАЛЕГАЮЩИХ НЕФТЯНЫХ ЗАЛЕЖЕЙ В РЕЖИМЕ РАСТВОРЕННОГО ГАЗА 133-137

Кузнецов В.Н.
МАТЕМАТИЧЕСКИЙ АНАЛИЗ ФУНКЦИЙ БУЛЕВОЙ АЛГЕБРЫ: ТАБЛИЧНО-АНАЛИТИЧЕСКОЕ ИСЧИСЛЕНИЕ, СТРОГИЙ ПОРЯДОК КОНЪЮНКТИВНЫХ НОРМАЛЬНЫХ ФОРМ И ПРОИЗВОДНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ ФУНКЦИЙ 138-152

Маликов З.М., Наврузов Д.П.
ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ОСЕСИММЕТРИЧНОЙ СТРУИ НА ОСНОВЕ НОВЫХ МОДЕЛЕЙ ТУРБУЛЕНТНОСТИ 153-159

Наврузов Д.П.
ВИХРЕВОЕ ТЕЧЕНИЕ ПОСЛЕ ВНЕЗАПНОГО РАСШИРЕНИЯ 160-165

Полкова Е.В., Кузяков О.Н.
ВИРТУАЛЬНЫЕ АНАЛИЗАТОРЫ КАК ИНСТРУМЕНТ ДЛЯ ОЦЕНКИ ТЕКУЩЕГО СОСТОЯНИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА ТРУБОПРОВОДА 166-170

Садыгов А.Б., Зейналов Р.М., Мамедли Р.Э.
ИССЛЕДОВАНИЕ РЕШЕНИЙ ОДНОЙ ГРАНИЧНОЙ ЗАДАЧИ 171-174

Сергеев А.М.
ПОИСК БИНАРНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ФОРМИРОВАНИЯ ЦИКЛИЧЕСКИХ БЛОКОВ СИММЕТРИЧНЫХ МАТРИЦ АДАМАРА ТРЁХБЛОЧНОЙ КОНСТРУКЦИИ БАЛОНИНА-СЕБЕРРИ 175-179

CC BY

Чертовской В.Д.
АДАПТИВНЫЙ ПЕРЕХОД НА ВЫПУСК ПРОИЗВОДСТВОМ НОВОЙ ПРОДУКЦИИ 180-185

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ПРИКЛАДНЫХ НАУКАХ (ФИЗИКА, ХИМИЯ, БИОЛОГИЯ И ДР.)

Кабилов М.М., Халимов И.И., Холов О.А., Баротов А.С.
ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ СРАВНЕНИЯ ОСНОВНЫХ ПАРАМЕТРОВ ФИЛЬТРАЦИОННОГО ГОРЕНИЯ ПРОПАНО-ВОЗДУШНОЙ СМЕСИ 186-191

Истратова Е.Е., Кожевников А.Н., Штрайх А.Е., Николаева Е.Д.
ОПРЕДЕЛЕНИЕ ДИНАМИЧЕСКИХ ХАРАКТЕРИСТИК ОПОР КОНТАКТНОЙ СЕТИ ГОРОДСКОГО ЭЛЕКТРИФИЦИРОВАННОГО ТРАНСПОРТА ПУТЕМ МОДЕЛИРОВАНИЯ В СРЕДЕ SOLID WORKS 192-197

Лаврёнов А.Н., Хитрушко В.В.
КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ПРИКЛАДНЫХ НАУКАХ 198-203

Kharitonov L.S., Evart T.E.
CARDIAC TISSUE MAPPING BY MOTION CAPTURE 204-211

УПРАВЛЕНИЕ РАЗРАБОТКОЙ ПРОГРАММНЫХ КОМПЛЕКСОВ И ВЫЧИСЛИТЕЛЬНЫМИ ПРОЦЕССАМИ. ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Антонов И.М., Мелюков С.А., Баскаков С.А., Чемоданов В.Б.
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МОБИЛЬНОГО ПИЛОТАЖНОГО СТЕНДА И ОСОБЕННОСТИ ЕГО РАЗРАБОТКИ 212-215

Брагинский М.Я., Тараканов Д.В.
РАСПОЗНАВАНИЕ ТРАНСПОРТНЫХ СРЕДСТВ НА ИЗОБРАЖЕНИИ ИСКУССТВЕННЫМИ НЕЙРОННЫМИ СЕТЯМИ 216-222

Есенбаева С.К.
КОРПОРАТИВНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА УЧЁТА ГОРЮЧЕ-СМАЗОЧНЫХ МАТЕРИАЛОВ 223-225

Мокряков А.В., Зяблицева Д.К.
ВОПРОСЫ АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ ОРГАНИЗАЦИИ НА БАЗЕ ПЛАТФОРМЫ 1С 226-229

Имомова Ш.М.
ЦЕЛИ, ФУНКЦИИ И МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ ... 230-234

Казаченко Т.А., Мелехин М.И., Литовка А.Д., Батыркаев А.Г., Аитов А.И.
АРХИТЕКТУРНЫЙ ТЕХНИЧЕСКИЙ ДОЛГ В AGILE-ПРОЕКТАХ 235-239

Карифуллаева И.С.
РАЗРАБОТКА CRM-СИСТЕМЫ IT-ПРЕДПРИЯТИЯ МАЛОГО БИЗНЕСА 240-242

Катермина Т.С., Туманов В.И., Зинченко С.С.
ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОПРЕДЕЛЕНИЯ COVID-19 243-247

CC BY

Катермина Т.С., Лазоренко Е.В.
СИСТЕМА АВТОМАТИЧЕСКОГО КОНТРОЛЯ АВТОМОЙКИ ПРИ ПОМОЩИ НЕЙРОННОЙ СЕТИ 248-254

Мамедли Р.Э.
УПРАВЛЕНИЯ С МЕТОДАМИ БЛОКИРОВКИ В ТРАНЗАКЦИЯХ 255-260

Никонова Е.З.
СОВРЕМЕННЫЕ КОНЦЕПЦИИ РАБОТЫ С ДАННЫМИ 261-266

Ражков А.Ф., Тимощенко Е.В.
ОПТИМИЗАЦИЯ ГИПЕРПАРАМЕТРОВ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ ДАННЫХ 267-274

Туренко С.К., Рейхерт В.С.
РАЗРАБОТКА АЛГОРИТМА РАСЧЁТА ЗОН ДРЕНИРОВАНИЯ ДОЖДЕВЫХ КОЛОДЦЕВ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ РЕКОНСТРУКЦИИ СИСТЕМ ГОРОДСКОГО ВОДООТВЕДЕНИЯ 275-279

Слива М.В.
ДИНАМИЧЕСКОЕ КОНФИГУРИРОВАНИЕ СОВМЕСТНОЙ РАБОТЫ RASPBERRY PI И ARDUINO В РАМКАХ УПРАВЛЕНИЯ УСТРОЙСТВАМИ 280-283

Раскатова М.В., Челышев Э.А.
ВЕКТОРИЗАЦИЯ ТЕКСТОВ В ЗАДАЧАХ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА: ИСТОРИЯ И РАЗВИТИЕ 284-288

Раскатова М.В., Челышев Э.А.
ИНФОРМАЦИОННАЯ СИСТЕМА АВТОМАТИЧЕСКОЙ РУБРИКАЦИИ НОВОСТНЫХ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ МАШИННОГО ОБУЧЕНИЯ 289-294

РОБОТОТЕХНИКА И ПРОГРАММИРОВАНИЕ

Аксенов Е.Н., Фурдуй О.М.
ПРИМЕНЕНИЕ ПЛАТФОРМЫ *MICRO:BIT* ДЛЯ ПОВЫШЕНИЯ УРОВНЯ ЗАИНТЕРЕСОВАННОСТИ ШКОЛЬНИКОВ И СТУДЕНТОВ В ИЗУЧЕНИИ ЦИФРОВЫХ ТЕХНОЛОГИЙ 295-301

Makhmadiev Kh.I., Ataeva G.I.
HISTORY OF THE DEVELOPMENT OF ROBOTICS 302-306

Изибаиров Ф.Ф.
ПРИМЕНЕНИЕ РОБОТОТЕХНИКИ В СОЦИАЛЬНОЙ СФЕРЕ 307-311

Исмоилова М.Н.
ЭТАПЫ РАЗВИТИЯ РОБОТОТЕХНИКИ 312-318

Никонова Е.З., Лемиш Д.В.
СЕМАНТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ БЕСКОНТАКТНОГО УПРАВЛЕНИЯ ЭВМ 319-323

Виштак О.В., Мамедов Р.Р.
РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПРЕДПРИЯТИЯ МАЛОГО БИЗНЕСА 324-329

Пащенко О.И., Рысаева Т.И.
ОСОБЕННОСТИ ФОРМИРОВАНИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ ПРИ ИЗУЧЕНИИ ПРОГРАММИРОВАНИЯ В УСЛОВИЯХ СМЕШАННОГО ОБУЧЕНИЯ ... 330-338

CC BY

Смирнов А.Р., Виштак О.В.
ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ В ПРОЦЕССЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ 339-342

Шакиров Д.Р.
СОВРЕМЕННАЯ РОБОТОТЕХНИКА И ЕЁ ПЕРСПЕКТИВЫ 343-347

Яблоков Е.Н.
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ БЛОКА КОДИРОВАНИЯ СООБЩЕНИЙ ДЛЯ ГРАЖДАНСКОЙ АВИАЦИИ 348-354

ВОПРОСЫ МЕТОДОЛОГИИ ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ И ВЗАИМОДЕЙСТВИЯ С ОБЩЕОБРАЗОВАТЕЛЬНЫМИ ШКОЛАМИ

Абрамян Г.В.
СОВРЕМЕННЫЕ ТЕНДЕНЦИИ И РИСКИ ПЕРСОНАЛИЗАЦИИ ОБУЧЕНИЯ В УСЛОВИЯХ ЦИФРОВИЗАЦИИ ОБРАЗОВАНИЯ 355-362

Алексеева В.Н., Комарова О.В., Орлова Л.А.
ФОРМИРОВАНИЕ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ СТУДЕНТОВ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ СПО НА ОСНОВЕ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ И ПРОИЗВОДСТВЕННЫХ ПРАКТИК 363-367

Басов В.А.
О СОДЕРЖАТЕЛЬНОМ И ТЕХНОЛОГИЧЕСКОМ АСПЕКТЕ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ ВЫСШАЯ МАТЕМАТИКА В УСЛОВИЯХ ЦИФРОВОЙ ТРАНСФОРМАЦИИ 368-372

Лозикова И.О., Валеева Д.Р.
СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ ДЛЯ ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ 373-378

Гибадуллин А.А.
КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ИГР В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ 379-383

Казиахмедов Т.Б., Симурзина Е.А.
МЕЖДИСЦИПЛИНАРНЫЙ ПОДХОД ПРИ ИЗУЧЕНИИ ТИПОВ И СТРУКТУР ДАННЫХ В ЯЗЫКАХ ПРОГРАММИРОВАНИЯ 384-392

Абильдинова Г.М., Каратаева Ж.Р.
ОСОБЕННОСТИ ДИСТАНЦИОННОГО ОБУЧЕНИЯ В ПЕРИОД САМОИЗОЛЯЦИИ .. 393-395

Барышева И.В., Козлов О.А., Малкина Е.В.
УРОВЕНЬ ПОДГОТОВКИ АБИТУРИЕНТОВ ПО ПРОГРАММИРОВАНИЮ И ЕГО ВЛИЯНИЕ НА ВЫБОР ПРОФЕССИИ 396-404

Коледин В.В., Зверева Е.А.
МЕТОДЫ РАЗВИТИЯ МАТЕМАТИЧЕСКОЙ КОМПЕТЕНТНОСТИ БАКАЛАВРОВ НАПРАВЛЕНИЯ «ПРОГРАММНАЯ ИНЖЕНЕРИЯ» 405-409

Махмутова М.В., Шевцова М.А.
РАЗРАБОТКА КУРСА ГРАФИЧЕСКОГО ДИЗАЙНА В РАМКАХ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ 410-415

CC BY

Махмутова М.В., Давлеткиреева Л.З., Алчинова Д.С.
ПРОЦЕСС СОЗДАНИЯ ФИЛЬТРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ НА ПРИМЕРЕ «SPARK AR STUDIO» ДЛЯ ПРОДВИЖЕНИЯ КОМПАНИИ В СОЦИАЛЬНЫХ СЕТЯХ 416-422

Меметова Ф.С.
ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ 423-429

Миронова Л.И.
ВАРИАНТ ВЗАИМОДЕЙСТВИЯ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ СО ШКОЛАМИ В ПРОЦЕССЕ ЦИФРОВОЙ ТРАНСФОРМАЦИИ ОБРАЗОВАНИЯ 430-438

Казиахмедов Т.Б., Мосягина Т.В.
ФОРМИРОВАНИЕ WEB КОМПЕТЕНЦИЙ БАКАЛАВРОВ ИВТ В УСЛОВИЯХ НАЛИЧИЯ БОЛЬШОГО НАБОРА ИНСТРУМЕНТОВ ДЛЯ WEB ПРОЕКТИРОВАНИЯ 439-445

Натальсон А.В.
МЕТОДОЛОГИЯ ФОРМИРОВАНИЯ ЦИФРОВЫХ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ ПО IT-НАПРАВЛЕНИЯМ 446-451

Савина Е.В., Раджабов К.Я.
АДАПТАЦИЯ КОМПЕТЕНТНОСТНОГО ПОДХОДА К МЕТОДИКЕ ПОДГОТОВКИ IT-СПЕЦИАЛИСТОВ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ 452-458

Садвакасова К.М.
ИСПОЛЬЗОВАНИЕ МОДУЛЬНОЙ ТЕХНОЛОГИИ НА УРОКАХ 459-462

Сайидова Н.С.
СОЗДАНИЕ АНИМИРОВАННЫХ КНОПОК УПРАВЛЕНИЯС ПОМОЩЬЮ ГРАФИЧЕСКОГО РЕДАКТОРА CorelDRAW 463-471

Табачук Н.П., Поличка А.Е.
ИССЛЕДОВАТЕЛЬСКАЯ И ПРОЕКТНАЯ АКТИВНОСТИ СТУДЕНТОВ В ЦИФРОВЫХ ПРОДУКТАХ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ: ЦЕЛИ И ЦЕННОСТИ В РАЗВИТИИ ИНФОРМАЦИОННОЙ КОМПЕТЕНЦИИ 472-476

CC BY

Научное издание

СОВРЕМЕННОЕ ПРОГРАММИРОВАНИЕ

Материалы IV Международной научно-практической конференции

Нижневартовск,
08 декабря 2021 года

Под общей редакцией Т.Б. Казиахмедова

ISBN 978-5-00047-631-4

Редактор:
Технический редактор:
Обложка: Д.В. Вилявин

Подписано в печать 30.03.2022
Формат 60×84/8
Гарнитура Times New Roman. Усл. печ. листов 23,60
Электронное издание. Объем 19,7 МБ. Заказ 2229

Издательство НВГУ
628615, Тюменская область, г.Нижневартовск, ул. Маршала Жукова, 4
Тел./факс: (3466) 24-50-51, E-mail: red@nvsu.ru, izdatelstvo@nggu.ru