

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

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.

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

Основной выполняемой операцией 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#.

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 А и В. 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 А, нажать кнопку со знаком «+», ввести второй элемент множества А, нажать кнопку со знаком «+». Продолжать, пока не введёте все элементы вашего множества А.



Рис. 2. Окно ввода элементов

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

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



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.

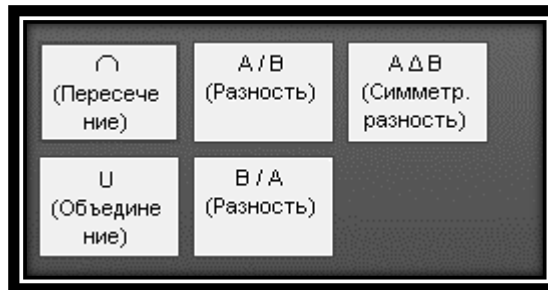


Рис. 4. Окно с операциями над множествами

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 А 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 В 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 «А Δ В (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 А и В.

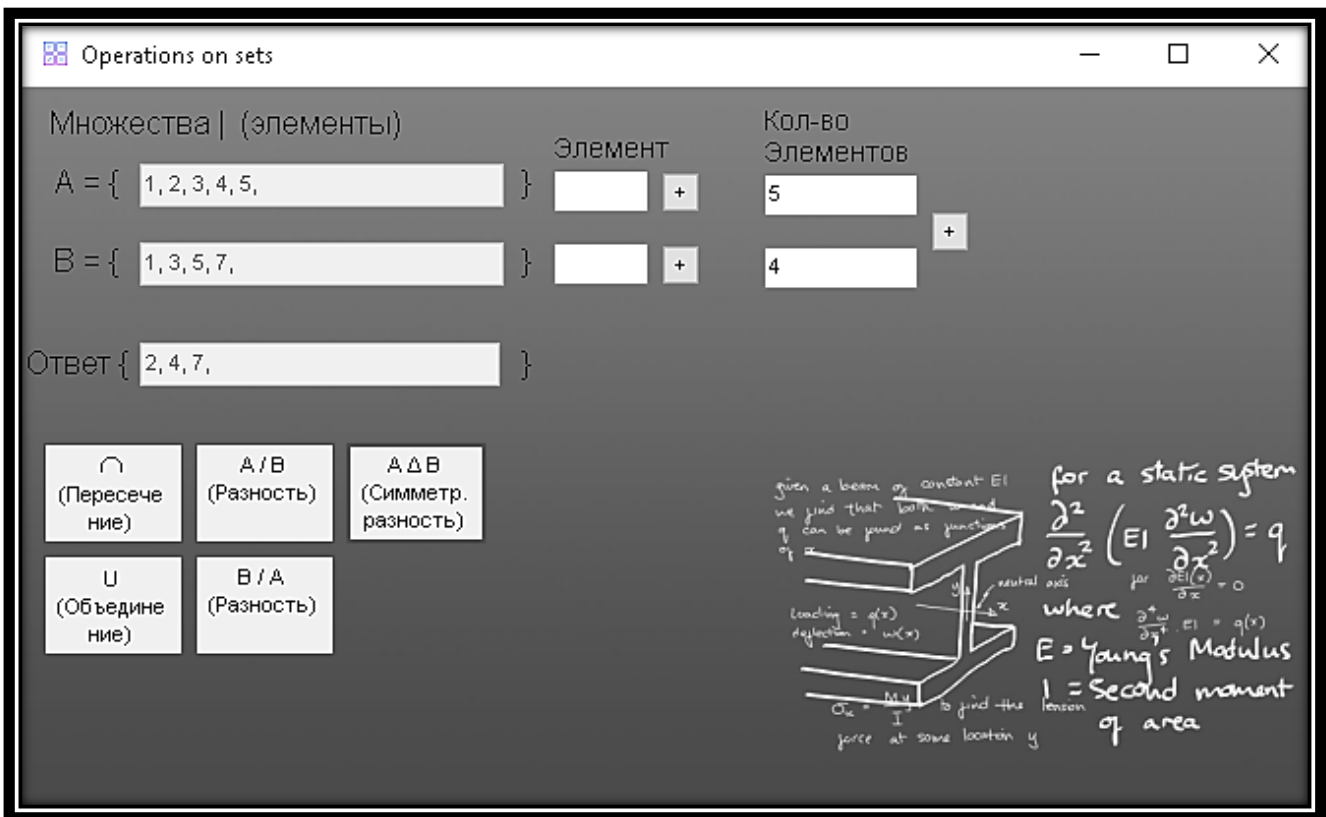


Рис. 6. Результат вычисления симметрической разности

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 статей II Всероссийской научно-практической 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. научных 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-практической 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