**

Колледж космического машиностроения и технологии

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине «Прикладное программирование»**

**Тема: «Разработка абстрактного типа данных "Множество"»**

Выполнил студент:

Соколов Ярослав Алексеевич

Группа П2-17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Королев, 2020

**Задание на курсовой проект**

Краткое описание задачи:

Разработка абстрактного типа данных "Множество" на языке программирования “С”

Полное описание задачи:

Разработать программу с реализацией функционала абстрактного типа данных "Множество" посредством функций:

* Объединение множеств
* Пересечение множеств
* Проверка на равенство множеств
* Разность множеств
* Симметричная разность множеств
* Проверка является множество “B” подмножеством “A”

Входные данные:

На вход подается строка, содержащая в себе односимвольные элементы множества через пробел. Затем выбирается номер действия.

Пример входных данных:





Выходные данные:

На выходе мы получаем полученное множество, значение 0 (false) 1(true) или информацию о доступных действиях.

Примеры выходных данных:





**Оглавление:**

[**Введение**](#_lf56fqfgllfi) **4**

[**Теоретическая часть**](#_zd51r9kdqe84) **5**

[Описание предметной области](#_2et92p0) 5

[Описание существующих разработок](#_venrg4jwh3j5) 6

[**Проектная часть**](#_8l5z2qptntm9) **7**

[Диаграмма прецедентов](#_pdsevntycmyj) 7

[Выбор инструментов](#_z9zjlwwma32v) 8

[Проектирование сценария](#_90yztlcganaa) 11

[Диаграмма классов](#_xvlv6ixzcfhk) 11

[Описание главного модуля](#_goix5v72crhu) 11

[Описания спецификаций к модулям](#_v7a3grb5po9j) 12

[Описание модулей:](#_4wytcrgqztzz) 13

[Описание текстовых наборов модулей](#_5vck6tyn3jjt) 21

[Описание применения средств отладки](#_wclgffdym7wa) 21

[Анализ оптимальности использования памяти и быстродействия](#_mqaleduyck9y) 21

[**Глава 3. Эксплуатационная часть**](#_zd9qf7eev5q8) **21**

[Руководство оператора](#_3whwml4) 21

[**Заключение**](#_j20e9023kny) **22**

[**Список литературы и интернет-источников**](#_qsh70q) **23**

[**Приложения**](#_3as4poj) **23**

# **Введение**

Мир с каждым днём меняется появляются новые технологии, но всё новое основывается на более ранних открытиях и изобретениях. Информационные технологии не являются исключением и одним из важных элементов являются множества.

И мной была поставлена перед собой цель реализовать операции над множествами для того, чтобы лучше понимать данный материал и в будущем иметь возможность реализовать функционал операций над множествами в своем приложении.

И для реализации своей идеи я использовал Dev-C++, который является свободным интегрированной средой разработки приложений для языков программирования “C/C++”

В первой части будут рассмотрены:

* Предметная область
* Существующие решения.

Во второй части будут рассмотрены:

Диаграмма прецедентов, Выбор инструментов, Проектирования сценария, Описание главного модуля, Описание модулей, Описание тестовых наборов модулей, Описание применения средств отладки, Анализ оптимальности использования памяти и быстродействия.

В третьей части будет освещено взаимодействие пользователя с программой, приведены изображения графических окон программы и сообщения оператору.

В заключении будут сделаны общие выводы, о получившемся проекте и полученных знаниях при разработке проекта.

Для осуществления обозначенных целей служат следующие задачи:

* Изучение теоретического материала и готовых решений
* Разработка программы и её функции
* Написание кода программы
* Описание руководства пользователя

# **Теоретическая часть**

## **Описание предметной области**

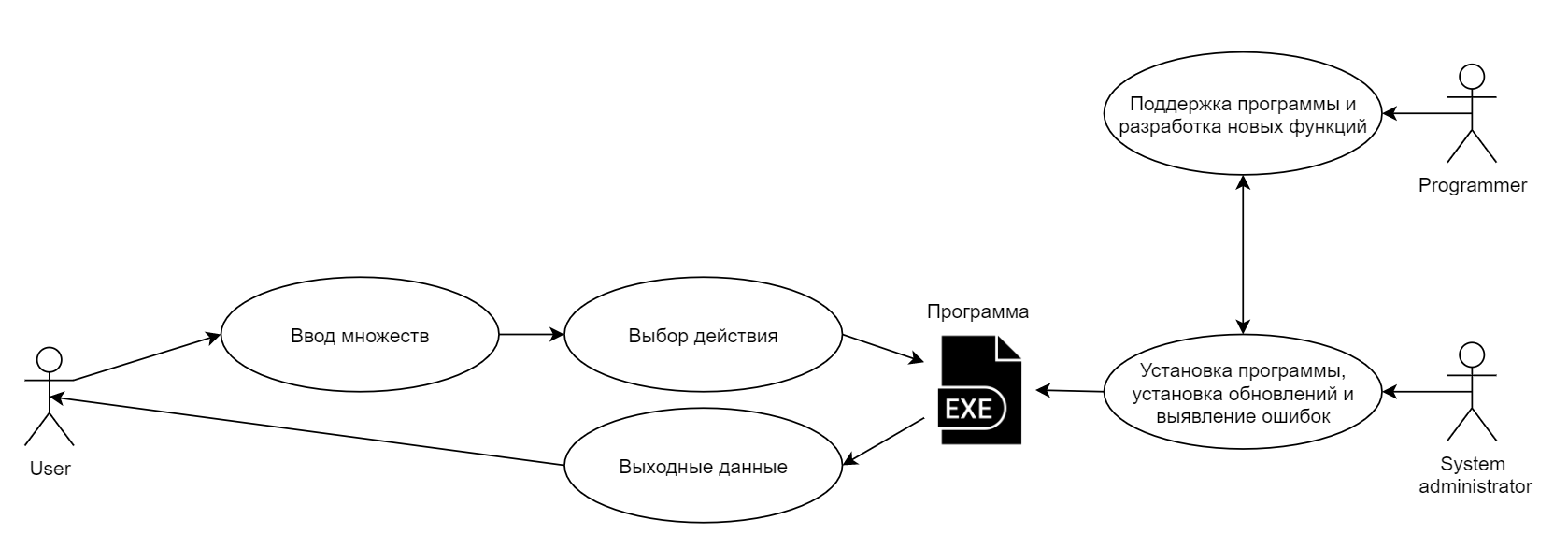
* + 1. **Множество** - является одним из ключевых понятий математики. Оно является математическим объектом, который является набором, совокупностью, собранием каких-либо объектов, которые называются **элементами** этого множества и обладают общим для всех их характеристическим свойством.
    2. **Элемент множества** - Объект, из которых состоит множество.
    3. **Пересечение множеств** - это множество, которому принадлежат те и только те элементы, которые одновременно принадлежат всем данным множествам.
    4. **Объединение множеств** - множество, содержащее в себе все элементы исходных множеств.
    5. **Разность двух множеств** - это операция, результатом которой является множество, в которое входят все элементы первого множества, не входящие во второе множество.
    6. **Симметрическая разность двух множеств** - это операция, результатом которой является новое множество, включающее все элементы исходных множеств, не принадлежащие одновременно обоим исходным множествам.

## **Описание существующих разработок**

|  |  |  |
| --- | --- | --- |
| Название языка программирования | Операции над множествами | Описание |
| Pascal | + | Объединение множеств |
| - | Пересечение множеств |
| \* | Разность множеств |
| >= | Проверка является множество “B” подмножеством “A |
| Python | union | Объединение множеств |
| intersection | Пересечение множеств |
| difference | Разность множеств |
| symmetric\_difference | Симметричная разность |
| >= | Проверка является множество “B” подмножеством “A” |
| Выводы: Реализованный тип данных множество упрощает и ускоряет работу с множествами, что делает его востребованным так множества активно используются в информационных технологиях например для моделирования объектов реального мира и компактного отображения сложных логических взаимоотношений или для 2d и 3d графики. | | |

# **Проектная часть**

## **Диаграмма прецедентов**



## 

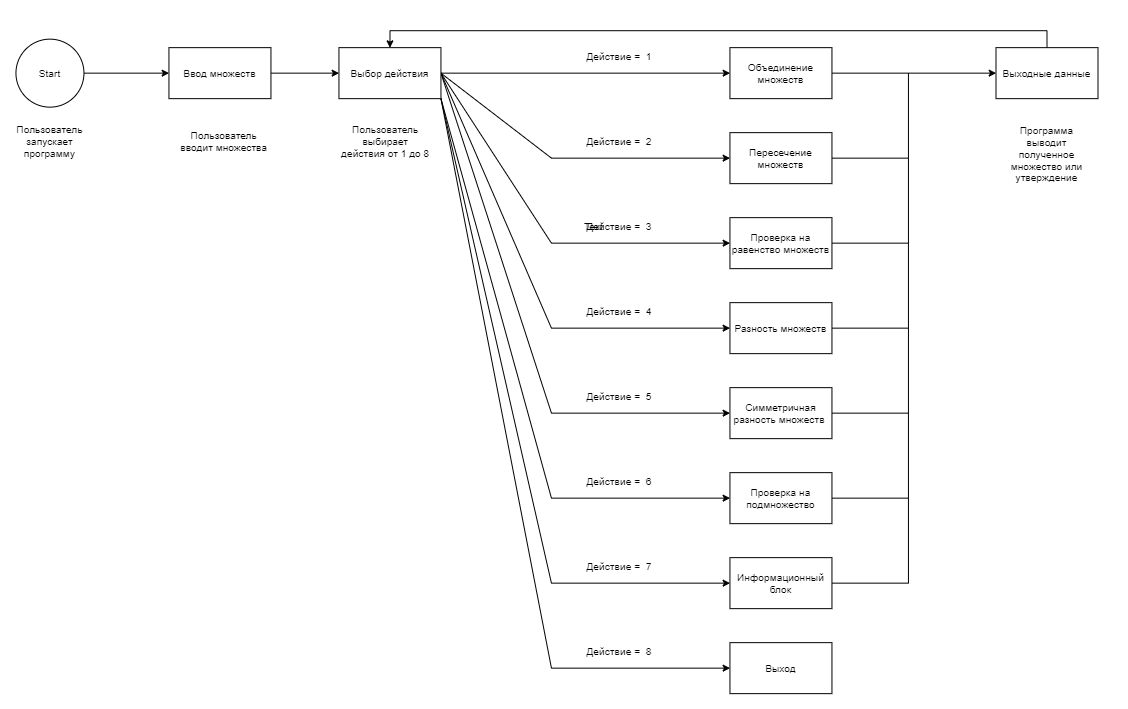
## **Выбор инструментов**

|  |  |  |  |
| --- | --- | --- | --- |
| Название среды разработки | Описание | Критерии | Оценка от 1 до 5 |
| Eclipse | Свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.  Распространяется с лицензией Eclipse Public License (EPL) | Доступность | 4 |
| Функциональность | 5 |
| Документация | 4 |
| NetBeans | NetBeans IDE — свободная интегрированная среда разработки приложений.  Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведется независимым сообществом разработчиков-энтузиастов (NetBeans Community) и компанией NetBeans Org. | Доступность | 4 |
| Функциональность | 4 |
| Документация | 5 |
| Visual Studio Code | Visual Studio Code - редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией. | Доступность | 3 |
| Функциональность | 5 |
| Документация | 4 |
| Code::Block | Code::Blocks — свободная кроссплатформенная среда разработки. Code::Blocks написана на С++ и использует библиотеку wxWidgets. Имея открытую архитектуру, может масштабироваться за счёт подключаемых модулей.  Поддерживается и разрабатывается The Code::Blocks team.  Распространяется по лицензии GNU General Public License | Доступность | 4 |
| Функциональность | 4 |
| Документация | 4 |
| Atom | Atom — бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов, написанных на Node.js, и встраиваемых под управлением Git. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом.  Поддерживается и разрабатывается GitHub Inc.  Распространяется по лицензии MIT | Доступность | 4 |
| Функциональность | 4 |
| Документация | 4 |
| Dev C++ | Dev-C++ — свободная интегрированная среда разработки приложений для языков программирования C/C++. В дистрибутив входит компилятор MinGW. Сам Dev-C++ написан на Delphi.  Разрабатывался Bloodshed Software / Orwell.  Распространяется согласно GPL. | Доступность | 4 |
| Функциональность | 4 |
| Документация | 4 |
| Итог: | Я выбрал для себя среду Dev C++, так как она наиболее простая в использовании(не имеет лишнего функционала, который захламляет среду) меньшую затратность по ресурсам системы и имеет хорошую скорость сборки проектов. В сумме даёт очень хорошие условия для создания небольших проектов. | | |

## 

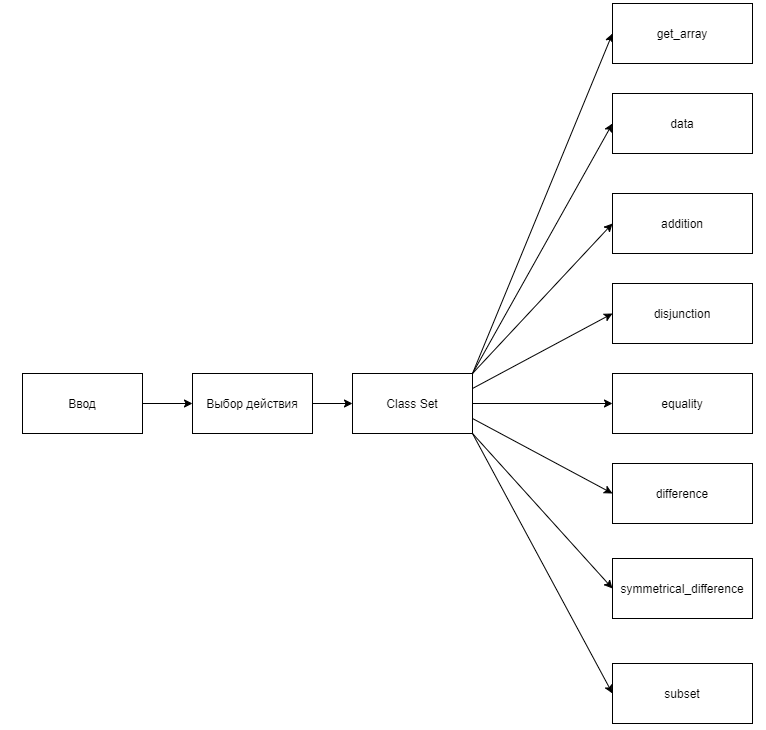
## **Проектирование сценария**

* + 1. User flow



## 

## **Диаграмма классов**



## **Описание главного модуля**

Главный модуль состоит из 3 частей:

* Запрос множеств
* Циклическая часть
* Выбор действия

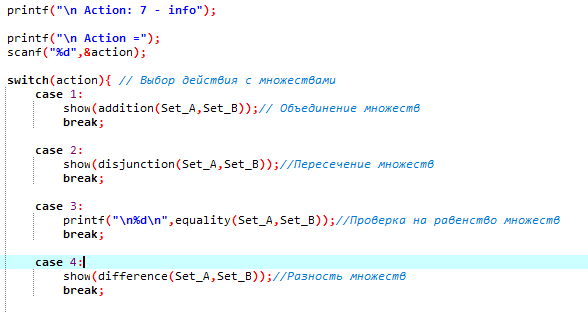
Запрос множеств представлен: Получением структуры из функционального модуля.

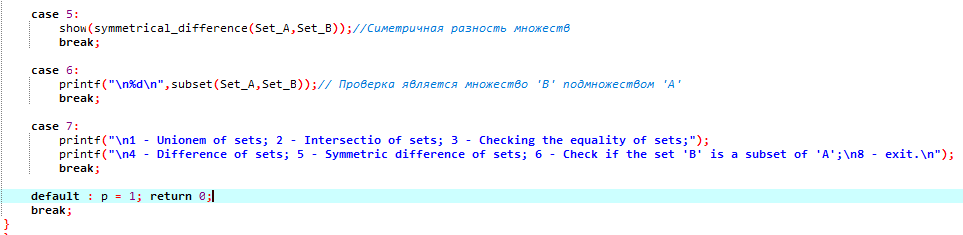


Циклическая часть представлена: циклом while который будет исполняться пока не изменится флаг.



Выбор действий представлен: Оператором swith, в который поступает выбранное действие.





Доступные действия:

* Вызов процедуры Объединения множеств
* Вызов процедуры Пересечение множеств
* Вызов процедуры Проверка на равенство множеств
* Вызов процедуры Разность множеств
* Вызов процедуры Симметричная разность множеств
* Вызов процедуры Проверка является множество 'B' подмножеством 'A'
* Вызов справочной информации о действиях
* Выход из программы

## **Описания спецификаций к модулям**

Типы модулей:

* Модуль считывания из консоли;
* Модуль возвращающий множество;
* Модуль возвращающий bool значение;
* Модуль вывода множества в консоль.

Модуль считывания из консоли:

Входные данные: Строка, состоящая из односимвольных элементов множества, которые не повторяются и разделены пробелом.

Выходные данные: Структура, состоящая из массива и его размера.

Модуль возвращающий множество:

Входные данные: Две структуры, состоящие из массива и его размера.

Выходные данные: Структура, состоящая из массива и его размера.

Модуль, возвращающий тип bool значение:

Входные данные: Две структуры состоящие из массива и его размера.

Выходные данные: bool значение.

Модуль вывода множества в консоль:

Входные данные: Структура состоящая из массива и его размера.

Выходные данные: Выводит в консоль множество.

## **Описание модулей:**

Список модулей:

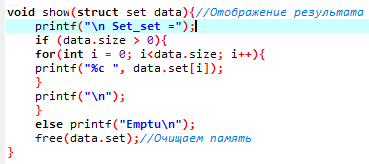
* **show**(Отображение результата)
* **get\_array**(Получение множеств от пользователя)
* **addition**(Объединение множеств)
* **disjunction**(Пересечение множеств)
* **equality**(Проверка на равенство множеств)
* **difference**(Разность множеств )
* **symmetrical\_difference**(Симметричная разность множеств)
* **subset**(Проверка является множество 'B' подмножеством 'A')

**show**(Отображение результата) описание: Функция получает множество представленное структурой. Если полученный массив из структуры пустой, то он выводит “Empty”

show состоит из:

* Проверки на пустой массив
* Вывода полученного массива

Листинг функции show:



**get\_array**(Получение множеств от пользователя) описание: Получает из консоли строку.

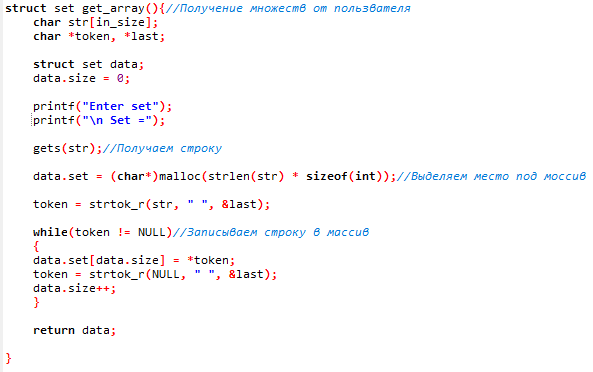
Затем выполняет следующие действия:

* Выделяет подстроку массив;
* Заполняет её элементами множества.

**get\_array** состоит из:

* Считывания строк из консоли;
* Выделение памяти под массив;
* Разделения строки и записи её в массив.

Листинг функции **get\_array:**

****

**addition**(Объединение множеств) описание: Функция получает два множества представленные структурами.

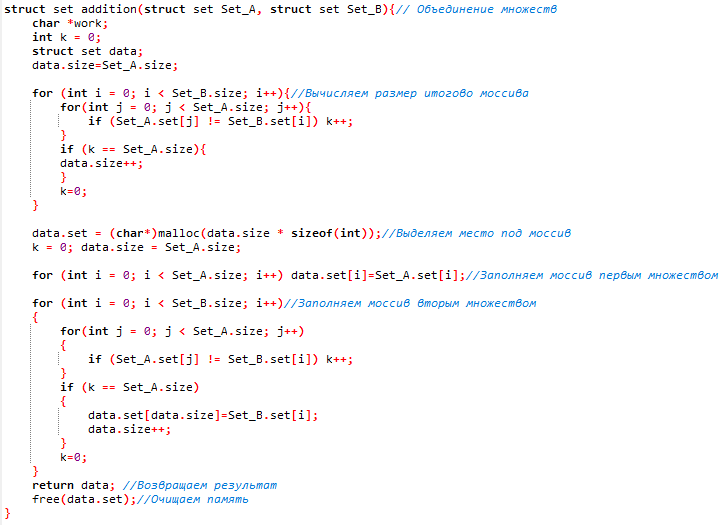
Затем выполняет следующие действия:

* Вычисляет размер итогового массива;
* Выделяет место под массив; заполняет его элементами первого множества;
* Заполняет его элементами второго множества которые подходят по условию.

**addition** состоит из:

* Вычисления размера итогового массива
* Выделения места под итоговый массив
* Заполнения массива элементами первого множества
* Заполнения массива элементами второго множества

Листинг функции **addition:**



**disjunction**(Пересечение множеств) описание: Функция получает два множества представленные структурами.

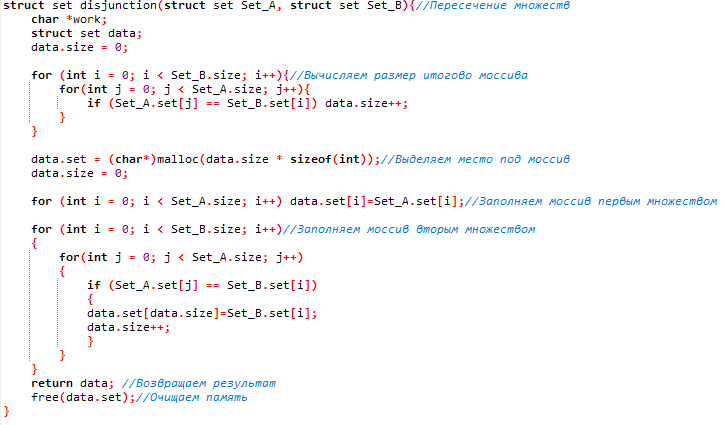
Затем выполняет следующие действия:

* Вычисляет размер итогового массива, выделяет место под массив;
* Заполняет его элементами первого множества и заполняет его элементами второго множества, которые подходят по условию.

**disjunction** состоит из:

* Вычисления размера итогового массива
* Выделения места под итоговый массив
* Заполнения массива элементами первого множества
* Заполнения массива элементами второго множества

Листинг функции **disjunction:**

****

**equality**(Проверка на равенство множеств) описание: Функция получает два множества представленные структурами.

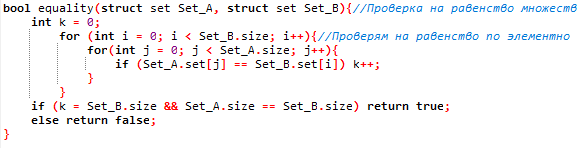
Затем выполняет следующие действия:

* Поэлементно сравнивает множества;
* Возвращает результат посредством bool значения .

**equality** состоит из:

* Поэлементного сравнения
* Вывода результата

Листинг функции **equality:**

****

**difference**(Разность множеств ) описание: Функция получает две структуры которые состоят из массива и его размера.

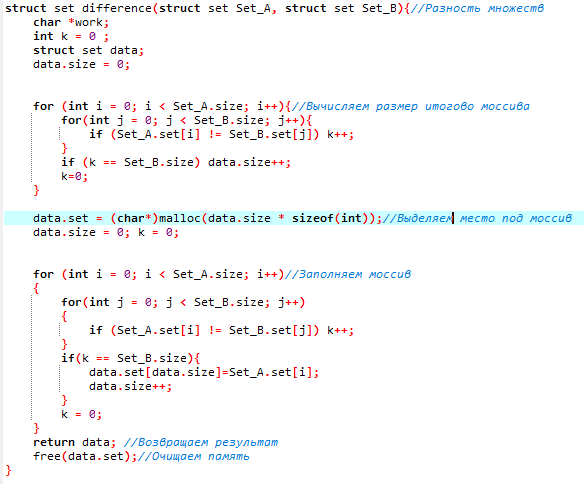
Затем выполняет следующие действия:

* Вычисляет размер итогового массива полученного в результате разности;
* Выделяет место под массив;
* Заполняет его элементами, которые не входят в первый массив.

**difference** состоит из:

* Вычисления размера итогового массива
* Выделения места под итоговый массив
* Заполнение массива

Листинг функции **difference:**

****

**symmetrical\_difference**(Симметричная разность множеств) описание: Функция получает две структуры которые состоят из массива и его размера.

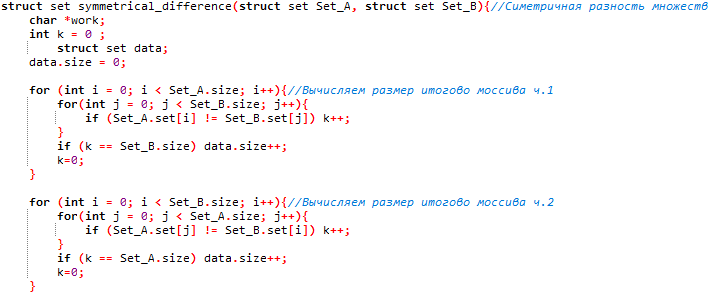
Затем выполняет следующие действия:

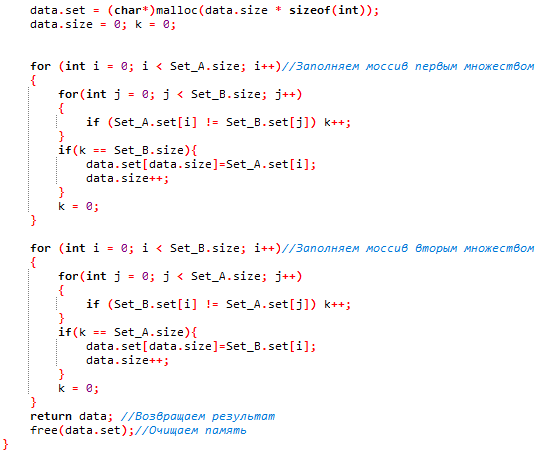
* Вычисляет размер итогового массива полученного в результате симметричной разности;
* Выделяет место под массив;
* Заполняет его элементами двух множеств кроме тех, которые присутствуют в обоих множествах.

**symmetrical\_difference** состоит из:

* Вычисления размера итогового массива ч.1
* Вычисления размера итогового массива ч.2
* Заполнения массива первым множеством
* Заполнения массива вторым множеством

Листинг функции **symmetrical\_difference:**

****

****

**subset**(Проверка является множество 'B' подмножеством 'A') описание: Функция получает две структуры которые состоят из массива и его размера.

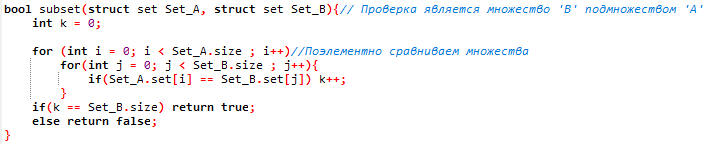
Затем выполняет следующие действия:

* Поэлементно сравнивает множества;
* Затем выводит результат.

**subset** состоит из:

* Поэлементного сравнения
* Вывода результата

Листинг функции **subset:**

****

## **Описание текстовых наборов модулей**

## **Описание применения средств отладки**

## **Анализ оптимальности использования памяти и быстродействия**

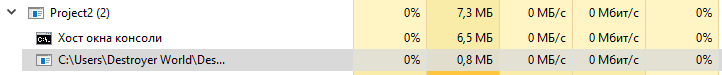
* + 1. Проект:

Информация от компилятора: Output Size: 135,3515625 KiB - Compilation Time: 0,19s

Максимальная нагрузка Загрузка ЦП 0,4%

Максимальная общая нагрузка на оперативную память 7,4 мб

Максимальная нагрузка раздельно:



* + 1. Пустая программа:

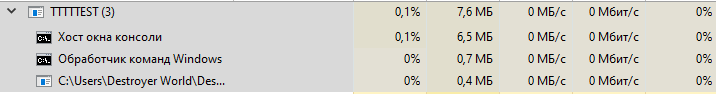
Информация от компилятора: Output Size: 127,931640625 KiB

Compilation Time: 0,25s

Максимальная нагрузка Загрузка ЦП 0,1%

Максимальная общая нагрузка на оперативную память 6,5 мб

Максимальная нагрузка раздельно:



* + 1. Итог: Программа получилась не ресурсно затратной, что делает её универсальной, так как она будет хорошо работать и на мощных системах так и на слабых.

# **Глава 3. Эксплуатационная часть**

## **Руководство по установке для Системного Администратора**

# 

# **Заключение**

# 

# **Список литературы и интернет-источников**

[https://ru.wikipedia.org/wiki/Множество](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE)

[https://ru.wikipedia.org/wiki/Пересечение\_множеств](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%81%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2)

[https://ru.wikipedia.org/wiki/Объединение\_множеств](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2)

[https://ru.wikipedia.org/wiki/Разность\_множеств](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2)

[https://ru.wikipedia.org/wiki/Симметрическая\_разность](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%80%D0%B0%D0%B7%D0%BD%D0%BE%D1%81%D1%82%D1%8C)

<https://pas1.ru/set0>

<https://python-scripts.com/sets#set-union>

Виленкин Н. Я. Рассказы о множествах. — 2005

# **Приложения**