**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

отчет

**по лабораторной работе № 2**

**по дисциплине «Алгоритмы и структуры данных»**

Тема: Множество как объект

|  |  |
| --- | --- |
| Студенты гр. 4312 | Устинов В. А. Гущин К. Д. |
| Преподаватель | Аббас С. А. М. |

Санкт-Петербург

2025

Оглавление

[1 Цель работы 3](#_Toc210292259)

[2 Задание 35 3](#_Toc210292260)

[3 Результаты эксперимента с четырьмя структурами данных на основе классов 3](#_Toc210292261)

[4 Результат эксперимента с отслеживанием вызовов функций-членов 6](#_Toc210292262)

[5 Выводы о результатах испытания способов представления множеств в памяти 8](#_Toc210292263)

[6 Приложение 10](#_Toc210292264)

# Цель работы

Исследование эффекта от использования классов.

# Задание 35

Множество, содержащее все буквы множества A, которых нет во множествах B, C или D.

# Результаты эксперимента с четырьмя структурами данных на основе классов

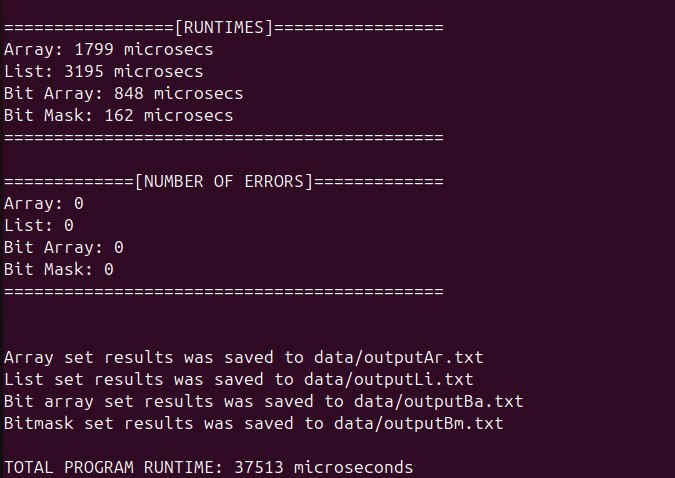
Эксперимент 1:

Рис. 1 Прогон 300 тестов

Эксперимент 2:

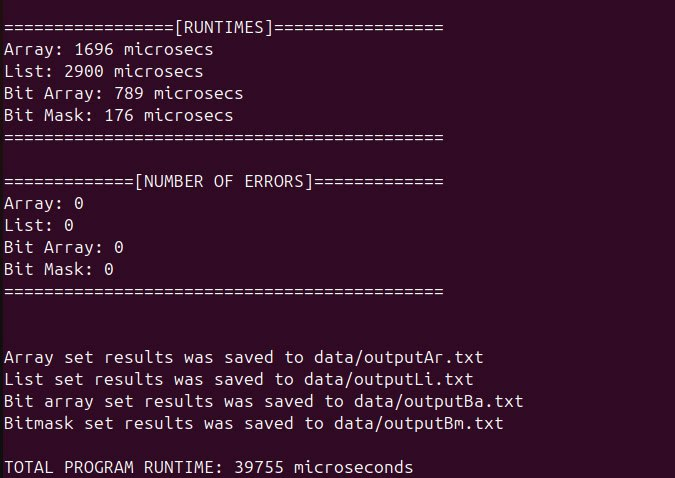


Рис. 2 Прогон 300 тестов

Эксперимент 3:

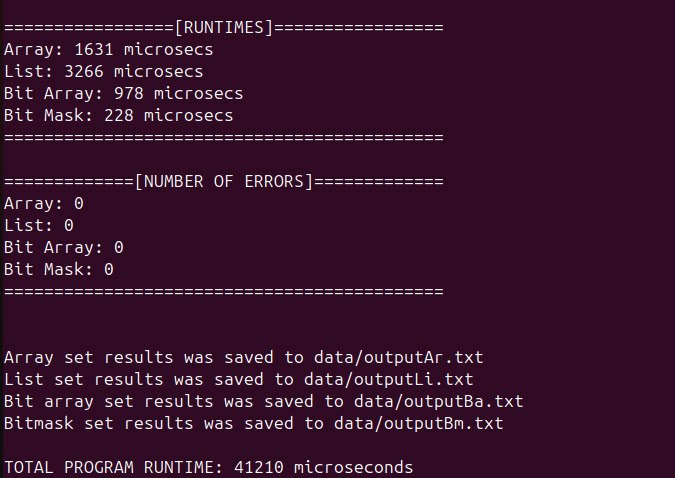


Рис. 3 Прогон 300 тестов

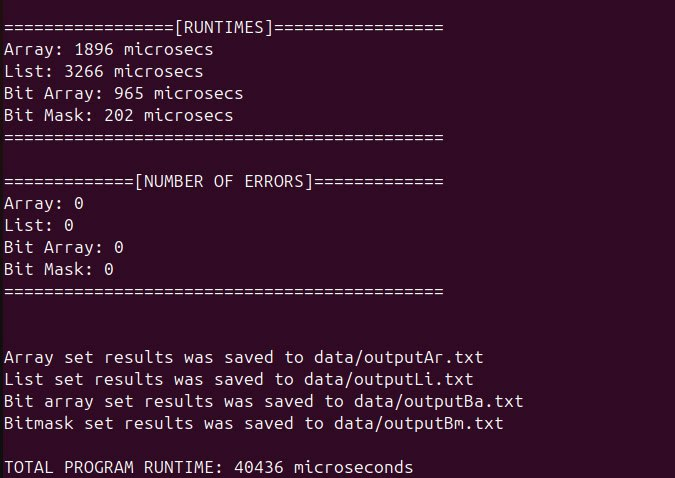
Эксперимент 4:

Рис. 4 Прогон 300 тестов

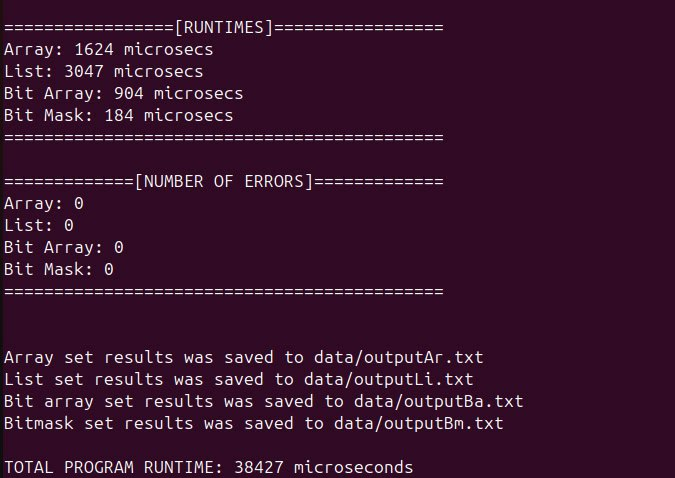
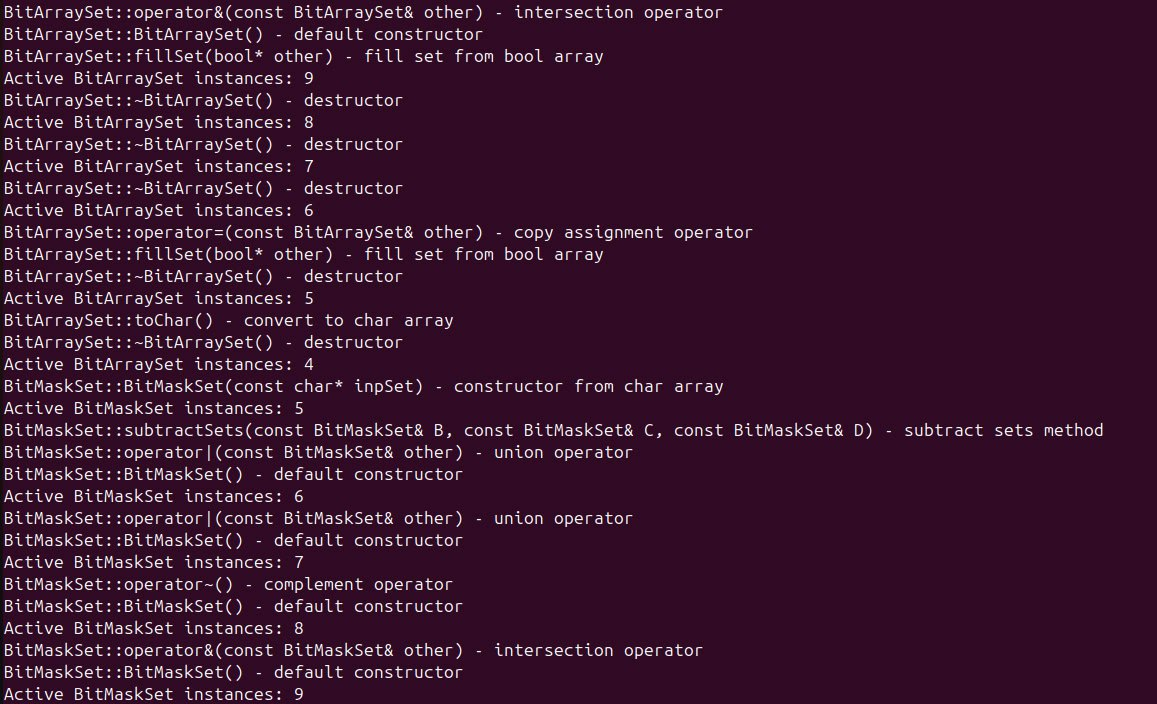
Эксперимент 5:

Рис. 5 Прогон 300 тестов

В результате экспериментов с четырьмя структурами данных на основе классов мы выяснили, что самая быстрая реализация класса — это Машинное слово.

# Результат эксперимента с отслеживанием вызовов функций-членов

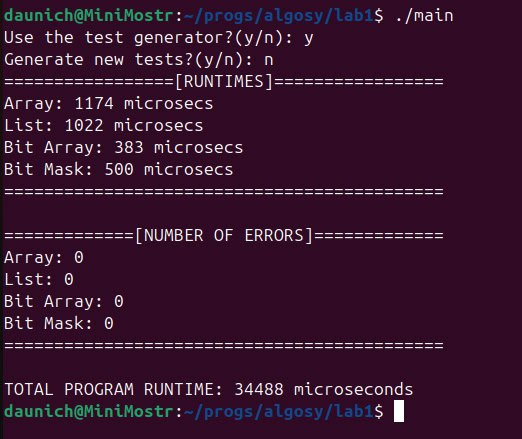
Рис. 6 Вызовы конструкторов в программе

Рис. 7 Вызовы деструкторов в программе

Исходя из данных, выводящихся в консоль, объекты классов корректно вызываются: методы, конструкторы, деструкторы отрабатывают без ошибок.

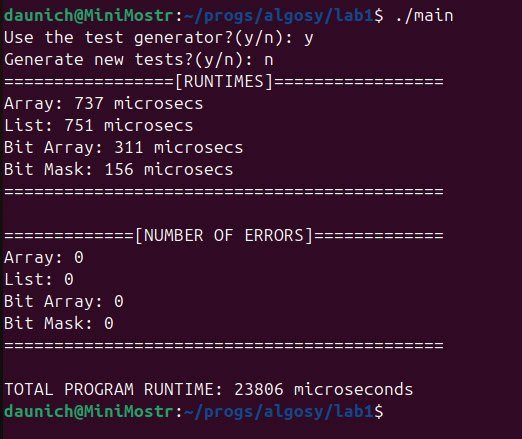
# Выводы о результатах испытания способов представления множеств в памяти

Испытание 1:



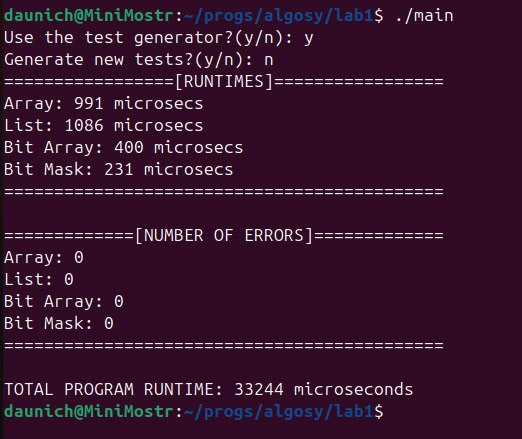
*Рис. 8 Общее время работы при 300 тестах*

Испытание 2:



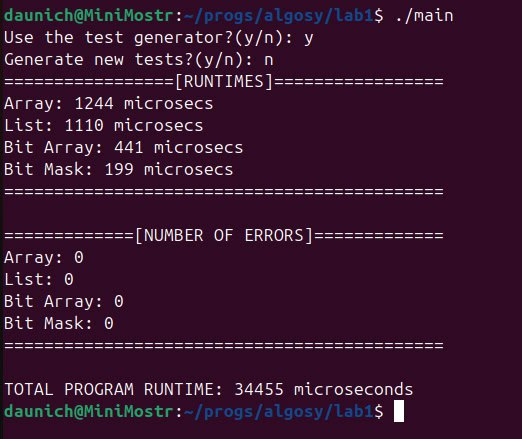
*Рис. 9 Общее время работы при 300 тестах*

Испытание 3:



*Рис. 10 Общее время работы при 300 тестах*

Испытание 3:



*Рис. 11 Общее время работы при 300 тестах*

Время работы программы, написанной с помощью использования технологий ООП, показывает относительно небольшое отставание по времени (5-6к микросекунд) по отношению к программе, написанной процедурным методом.

Данное отставание может быть обусловлено накладными расходами на объект класса, такими как обращение к методам, проверка полей приватности, вызов конструкторов и деструкторов.

Но главное преимущество ООП — это читаемость кода, простота поддержки. Сложные модели объектов очень удобно описывать с помощью классов. Можно сделать вывод из данного эксперимента, что при небольших проигрышах в скорости работы программы, мы получаем хорошо читаемый код с понятной логикой программы.

# Приложение

1) Реализация множества через массив символов ([header](array_set.hpp), [source](array_set.cpp));

2) Реализация множества через массив битов ([header](barray_set.hpp), [source](barray_set.cpp));

3) Реализация множества через двусвязный список ([header](listSet.hpp), [source](listSet.cpp));

4) Реализация множества через битовую маску ([header](bmask_set.hpp), [source](bmask_set.cpp));

5) Реализация генератора ([header](generator.hpp), [source](generator.cpp));

6) Реализация файлового менеджера ([header](fileManager.hpp), [source](fileManager.cpp));

7) Реализация консольного интерфейса ([header](consoleInterface.hpp), [source](consoleInterface.cpp));

8)Реализация тестировщика ([header](setTest.hpp), [source](setTest.cpp));

8) Реализация main ([source](file:///C:\Users\vedmix\УЧЕБА\algosy\lab1\main.cpp)).