Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ 1**

­

|  |  |  |
| --- | --- | --- |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Д. С. Романова |
|  | подпись, дата | инициалы, фамилия |
| Студент КИ22-07Б, 032215583 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А. М. Коробков |
| номер группы, зачетной книжки | подпись, дата | инициалы, фамилия |

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc209650335)

[Ход работы 4](#_Toc209650336)

[Заключение 6](#_Toc209650337)

[Список использованных источников 7](#_Toc209650338)

# **ВВЕДЕНИЕ**

**Описание полученного задания:** необходимо разработать программу, которая должна обрабатывать файл и выполнять описанные в нем команды: ADD <object data>

REM <условие>

PRINT

При обработке команды ADD в "контейнер" добавляется "объект" с заданными в команде параметрами. Тип контейнера и объекта определяется выданным вариантом задания. Аргументом команды является описание объекта, формат описания студент выбирает самостоятельно, однако формат должен быть удобен как для обработки ЭВМ, так и для чтения пользователем. При обработке команды REM из контейнера удаляются объекты, соответствующие заданному условию. Формат описания условия студент выбирает сам. Например: "x > 10". При обработке команды PRINT содержимое контейнера печатается на экране монитора. Программа разрабатывается на объектно-ориентированном языке программирования. При выполнении работы рекомендуется использовать стандартные контейнеры (на их основе реализовать нужный вариант). Это существенно снижает сложность работы.

Вариант задания представлен на рисунке 1.

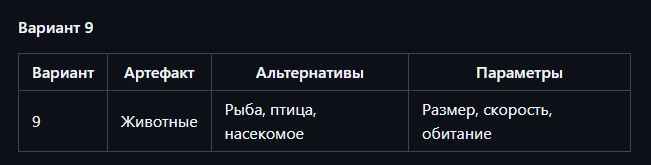


Рисунок 1 – Вариант задания

# **Ход работы**

Была разработана программа, которая принимает файл с командами в формате .txt и выполняет действия, описанные этими командами. Данная программа написана на ООП языке программирования С++. Соответствующая этой программе UML диаграмма классов, показанная на рисунке 2.

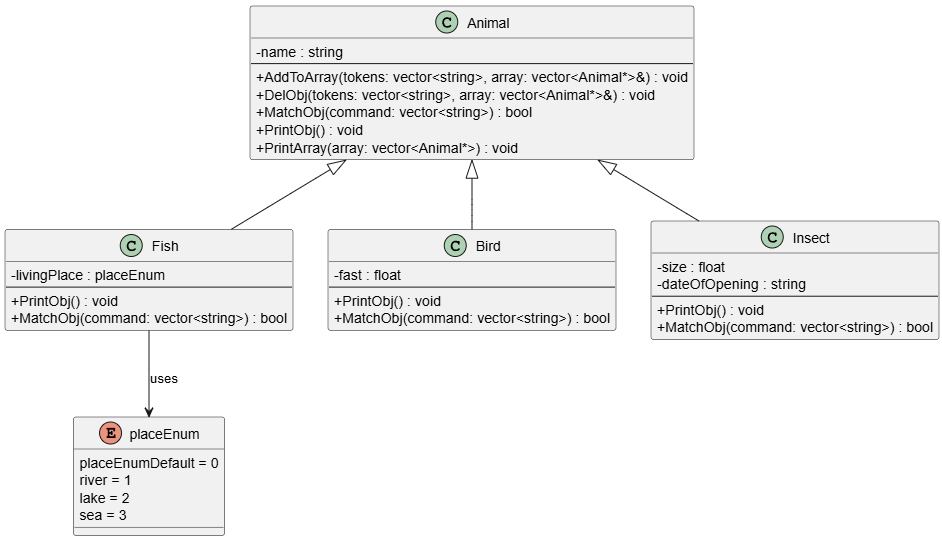


Рисунок 2 – UML–диаграмма классов

Для проверки работоспособности был составлен .txt файл, который наполнялся различными значениями. На рисунке 3 представлено содержание, а также результат выполнения файла с данными №1.

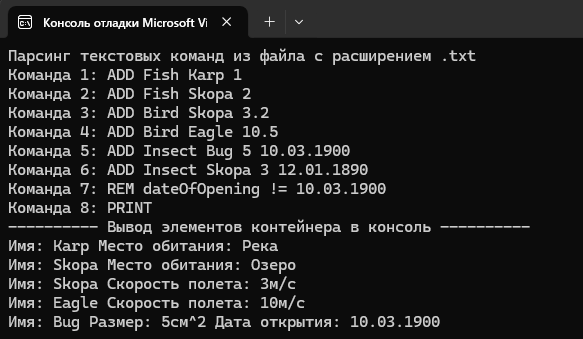


Рисунок 3 – Результат выполнения программы №1

Данные №2 представляют собой строки, содержащие ошибки. На рисунке 4 представлен результат выполнения программы с данными №2.

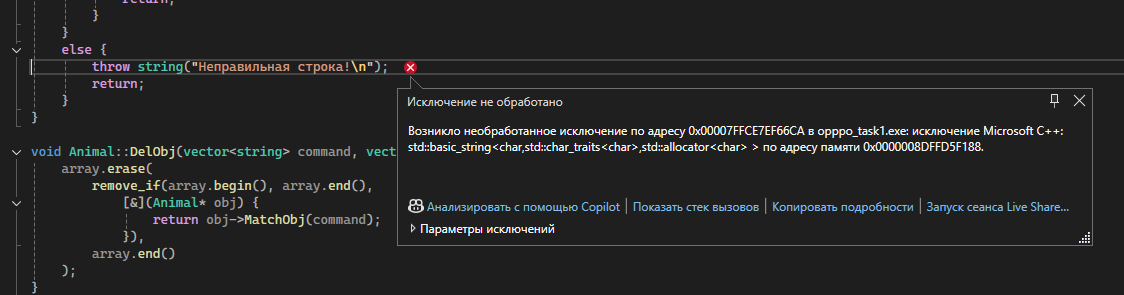


Рисунок 4 – Результат выполнения программы №2

Как можно заметить, в программе проверяется валидность строк. В случае, если какая-то из строк не валидна, выбрасывается исключение. Исходный код программы выложен на GitHub[1], ссылка приложена в списке использованных источников.

# **ЗАКЛЮЧЕНИЕ**

В результате данной практической работы удалось закрепить основные принципы ООП, а также разработать программу, которая полностью соответствует требованиям.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. GitHub. Система контроля версий: [сайт]. – URL: https://github.com/Korobkov124/OPPPO