**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО”**

Факультет программной инженерии и компьютерной техники (ПИКТ)

Направление подготовки (специальность) – 09.0**3**.04 (Нейротехнологии и программная инженерия)

Программирование

Лабораторная работа № 5

Выполнил студент

Немыкин Ярослав Алексеевич

Группа № Р3122

Преподаватель: Кобик Никита

г. Санкт-Петербург

2025 г.

**Оглавление**

[Задание: 3](#__RefHeading___Toc557_470747639)

[Ход работы: 4](#__RefHeading___Toc439_741411608)

[*UML диаграмма* 5](#__RefHeading___Toc443_741411608)

[*Исходный код программ* 5](#__RefHeading___Toc445_741411608)

[Вывод: 5](#__RefHeading___Toc465_741411608)

# **Задание:**

Разработанная программа должна удовлетворять следующим требованиям:

* Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
* Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.
* Для хранения необходимо использовать коллекцию типа java.util.HashMap
* При запуске приложения коллекция должна автоматически заполняться значениями из файла.
* Имя файла должно передаваться программе с помощью: **аргумент командной строки**.
* Данные должны храниться в файле в формате csv
* Чтение данных из файла необходимо реализовать с помощью класса java.util.Scanner
* Запись данных в файл необходимо реализовать с помощью класса java.io.FileOutputStream
* Все классы в программе должны быть задокументированы в формате javadoc.
* Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутсвие прав доступа к файлу и т.п.).

**В интерактивном режиме программа должна поддерживать выполнение следующих команд:**

* help : вывести справку по доступным командам
* info : вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)
* show : вывести в стандартный поток вывода все элементы коллекции в строковом представлении
* insert null {element} : добавить новый элемент с заданным ключом
* update id {element} : обновить значение элемента коллекции, id которого равен заданному
* remove\_key null : удалить элемент из коллекции по его ключу
* clear : очистить коллекцию
* save : сохранить коллекцию в файл
* execute\_script file\_name : считать и исполнить скрипт из указанного файла. В скрипте содержатся команды в таком же виде, в котором их вводит пользователь в интерактивном режиме.
* exit : завершить программу (без сохранения в файл)
* replace\_if\_greater null {element} : заменить значение по ключу, если новое значение больше старого
* replace\_if\_lowe null {element} : заменить значение по ключу, если новое значение меньше старого
* remove\_lower\_key null : удалить из коллекции все элементы, ключ которых меньше, чем заданный
* remove\_any\_by\_personal\_qualities\_minimum personalQualitiesMinimum : удалить из коллекции один элемент, значение поля personalQualitiesMinimum которого эквивалентно заданному
* min\_by\_name : вывести любой объект из коллекции, значение поля name которого является минимальным
* print\_field\_descending\_author : вывести значения поля author всех элементов в порядке убывания

**Формат ввода команд:**

* Все аргументы команды, являющиеся стандартными типами данных (примитивные типы, классы-оболочки, String, классы для хранения дат), должны вводиться в той же строке, что и имя команды.
* Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.
* При вводе составных типов данных пользователю должно показываться приглашение к вводу, содержащее имя поля (например, "Введите дату рождения:")
* Если поле является enum'ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).
* При некорректном пользовательском вводе (введена строка, не являющаяся именем константы в enum'е; введена строка вместо числа; введённое число не входит в указанные границы и т.п.) должно быть показано сообщение об ошибке и предложено повторить ввод поля.
* Для ввода значений null использовать пустую строку.
* Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.

**Описание хранимых в коллекции классов:**

public class LabWork {

private Integer id; //Поле не может быть null, Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически

private String name; //Поле не может быть null, Строка не может быть пустой

private Coordinates coordinates; //Поле не может быть null

private java.time.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private Long minimalPoint; //Поле не может быть null, Значение поля должно быть больше 0

private Long personalQualitiesMinimum; //Поле не может быть null, Значение поля должно быть больше 0

private Difficulty difficulty; //Поле не может быть null

private Person author; //Поле может быть null

}

public class Coordinates {

private int x;

private Integer y; //Максимальное значение поля: 858, Поле не может быть null

}

public class Person {

private String name; //Поле не может быть null, Строка не может быть пустой

private java.time.ZonedDateTime birthday; //Поле не может быть null

private Double height; //Поле не может быть null, Значение поля должно быть больше 0

private double weight; //Значение поля должно быть больше 0

private Location location; //Поле может быть null

}

public class Location {

private Double x; //Поле не может быть null

private Float y; //Поле не может быть null

private double z;

}

public enum Difficulty {

EASY,

NORMAL,

VERY\_HARD,

HOPELESS;

}

# **Ход работы****:**

В ходе работы были проделаны\* следующие действия:

Создан главный файл, из которого запускается программа.

Создана коллекция, требуемая в задании.

Создан менеджер, отвевающий за передачу информации с пользователем.

Создан менеджер, определяющий команду, которую пользователь запрашивает.

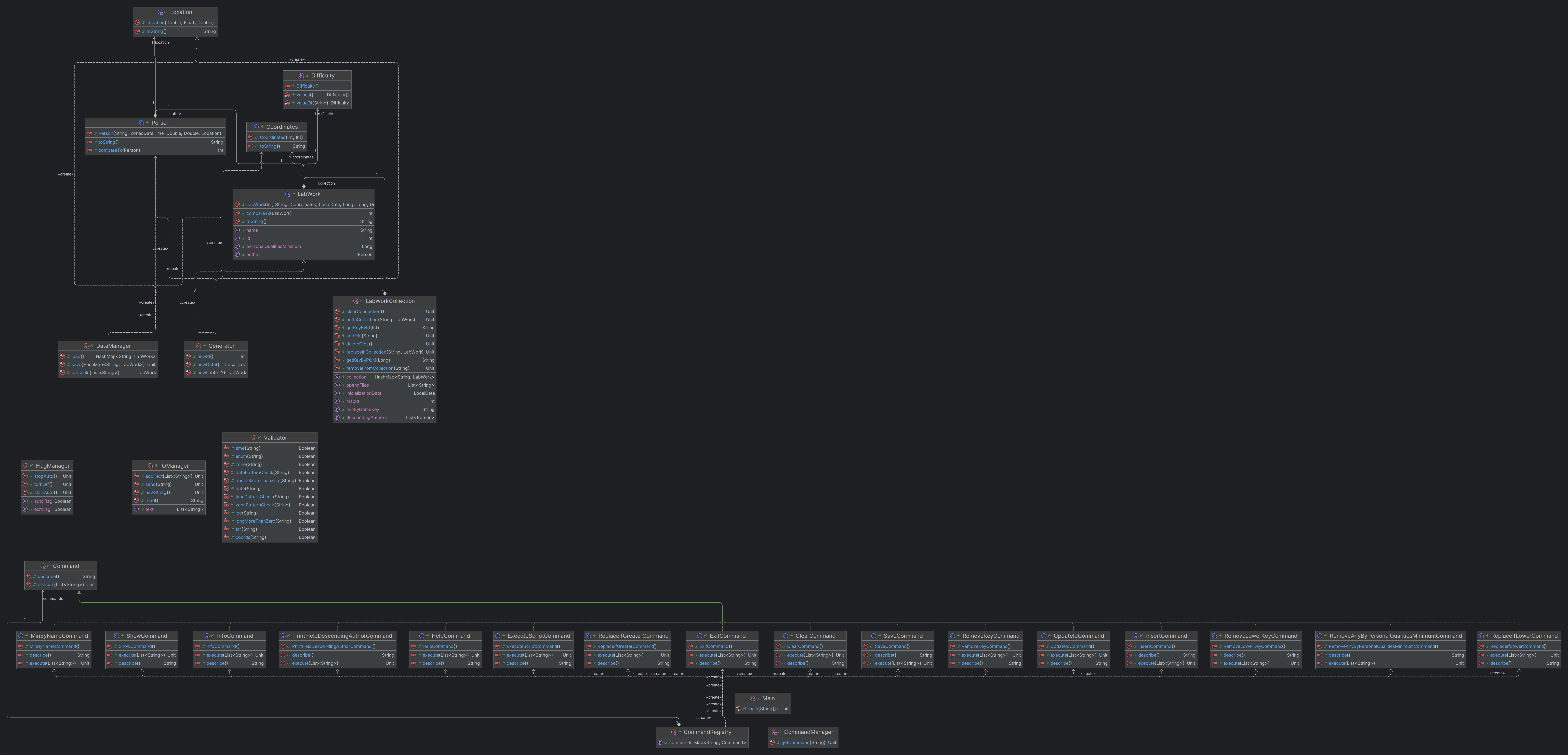
Созданы все требуемые команды.

Создан менеджер для сохранения и загрузки данных о коллекции в папку files.

Созданы вспомогательные классы, по типу generator, flag manager.

*\*В процессе работы программа многократно переделывалась, поэтому здесь представлена теоретический вариант того, как можно было все сделать, планируя следующий шаг*

## ***UML диаграмма***



*Рис. 1. UML диаграмма*

*Диаграмму также можно увидеть и в репозитории*

## ***Исходный код программ***

Ввиду большого объема программы, здесь будут размещены ссылка на репозиторий в Github, где лежат все файлы, относящиеся к данной работе.

[ссылка](https://github.com/yargames1/ITMO/tree/main/programming/lab5)

# **Вывод:**

В ходе работы я познакомился с коллекциями, поработал с файлами и сериализацией данных. А также написал первый раз программу на Kotlin и узнал новые подробности о базовых вещах в информатике.