

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ
НАУК

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине
«ПРОГРАММИРОВАНИЕ»

Вариант № 31194

Выполнил:

Студент группы Р3118

Наумов Павел

Викторович

Преподаватель:

Кулинич Ярослав

Владимирович

Санкт-Петербург, 2023

Содержание

Задание.....	3
Исходный код программы	6
Вывод программы	7
Вывод.....	8

Задание

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `LabWork`, описание которого приведено ниже.

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.PriorityQueue`
- При запуске приложения коллекция должна автоматически заполняться значениями из файла.
- Имя файла должно передаваться программе с помощью: **переменная окружения**.
- Данные должны храниться в файле в формате `json`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.BufferedReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.PrintWriter`
- Все классы в программе должны быть задокументированы в формате `javadoc`.
- Программа должна корректно работать с неправильными данными (ошибки пользовательского ввода, отсутствие 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
- `add {element}` : добавить новый элемент в коллекцию
- `update id {element}` : обновить значение элемента коллекции, id которого равен заданному
- `remove_by_id id` : удалить элемент из коллекции по его id
- `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)
- `head` : вывести первый элемент коллекции
- `remove_head` : вывести первый элемент коллекции и удалить его
- `remove_lower {element}` : удалить из коллекции все элементы, меньшие, чем заданный
- `max_by_name` : вывести любой объект из коллекции, значение поля name которого является максимальным
- `count_less_than_author author` : вывести количество элементов, значение поля author которых меньше заданного
- `print_field_descending_tuned_in_works` : вывести значения поля tunedInWorks 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'e; 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 может быть null, Значение поля должно быть больше 0
```

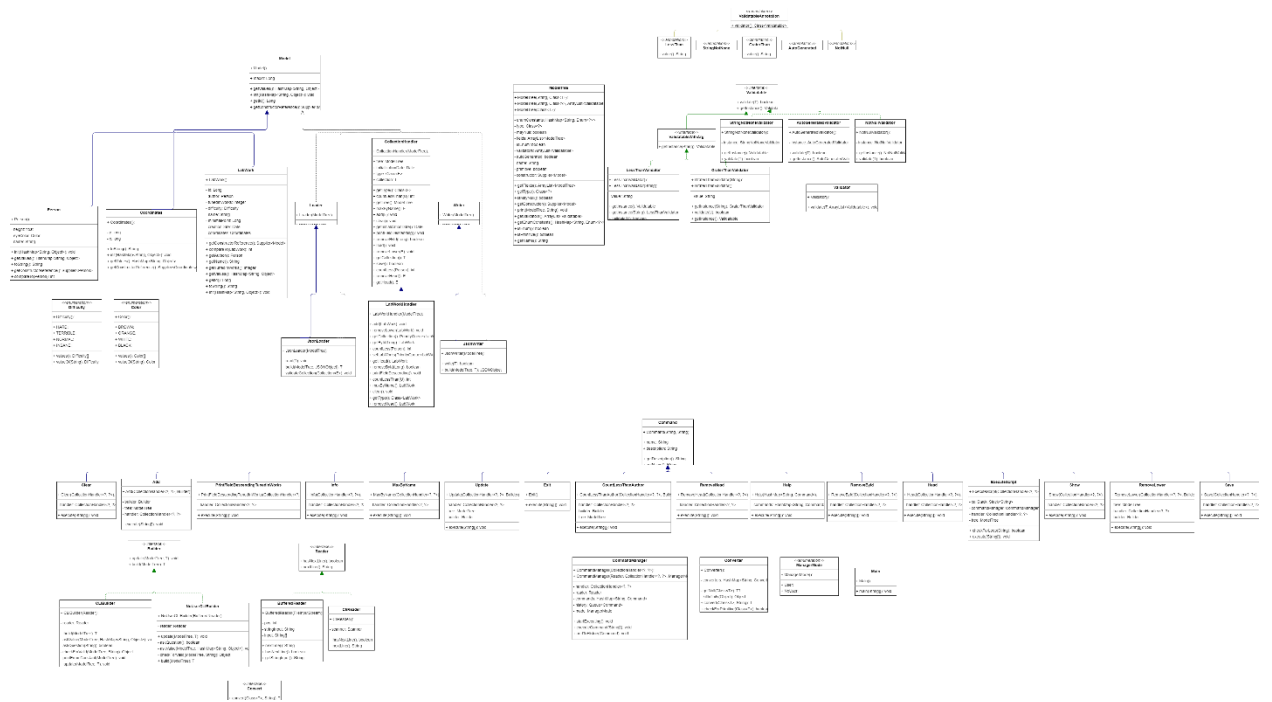
```

        private Integer tunedInWorks; //Поле может быть null
        private Difficulty difficulty; //Поле не может быть null
        private Person author; //Поле не может быть null
    }
    public class Coordinates {
        private long x; //Значение поля должно быть больше -73
        private long y; //Максимальное значение поля: 865
    }
    public class Person {
        private String name; //Поле не может быть null, Строка не может быть пустой
        private float height; //Значение поля должно быть больше 0
        private Color eyeColor; //Поле не может быть null
    }
    public enum Difficulty {
        EASY,
        NORMAL,
        INSANE,
        TERRIBLE;
    }
    public enum Color {
        RED,
        BLACK,
        YELLOW,
        BROWN;
    }
}

```

Исходный код программы

<https://github.com/naumovpavel/ITMO/tree/main/Prog/Lab5>



Вывод программы

```
show
{
  id : 1,
  name : abc,
  coordinates : {
    x : 45,
    y : 0
  },
  creationDate : Sat Mar 11 12:20:52 MSK 2023,
  minimalPoint : 54,
  tunedInWorks : 523,
  difficulty : TERRIBLE,
  author : {
    name : lack,
    height : 3.0,
    eyeColor : WHITE
  }
}
```

Вывод

Во время выполнения лабораторной работы были изучены коллекции, обобщённые типы, потоки ввода и вывода в языке Java, а также паттерн проектирования Command.