



**ITMO UNIVERSITY**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**Факультет Систем управления и робототехники**

**Лабораторная работа №5**

по дисциплине

«ПРОГРАММИРОВАНИЕ»

Вариант №314201

**Выполнили:**

Студенты группы R3142

Макунина Арина Артуровна

Лоскутова Ирина Вадимовна

**Преподаватель:**

Максимова Марина Михайловна

Санкт-Петербург

2022

## Содержание

Текст задания.....	3
Исходный код .....	6
Вывод .....	7

## Текст задания

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `HumanBeing`, 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.ArrayList`
- 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.util.Scanner`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.BufferedWriter`
- 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)
- `insert_at index {element}` : добавить новый элемент в заданную позицию
- `add_if_max {element}` : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `remove_all_by_weapon_type weaponType` : удалить из коллекции все элементы, значение поля `weaponType` которого эквивалентно заданному
- `average_of_minutes_of_waiting` : вывести среднее значение поля `minutesOfWaiting` для всех элементов коллекции
- `filter_starts_with_name name` : вывести элементы, значение поля `name` которых начинается с заданной подстроки

### 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; введенное 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 HumanBeing {
    private int id; //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.ZonedDateTime creationDate; //Поле не может быть null, Значение
    этого поля должно генерироваться автоматически
    private Boolean realHero; //Поле не может быть null
    private Boolean hasToothpick; //Поле не может быть null
```

```
private Integer impactSpeed; //Поле не может быть null
private String soundtrackName; //Поле не может быть null
private float minutesOfWaiting;
private WeaponType weaponType; //Поле не может быть null
private Car car; //Поле не может быть null
}
public class Coordinates {
    private Double x; //Поле не может быть null
    private Double y; //Поле не может быть null
}
public class Car {
    private String name; //Поле может быть null
    private boolean cool;
}
public enum WeaponType {
    AXE,
    PISTOL,
    SHOTGUN;
}
```

## Исходный код



## Вывод

В процессе выполнения лабораторной работы мы подробно познакомились с коллекциями и сортировкой её элементов, изучили интерфейсы `java.util.Comparable` и `java.util.Comparator`, поработали с файлами класса `java.io.File` и утилитой `javadoc`.