Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5

по программированию

Вариант № 33592

Выполнила:Гафурова Фарангиз

Фуркатовна

Группа: P3120

Принял:

Санкт-Петербург. 2024

Оглавление

Текст задания

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `SpaceMarine`, 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.TreeSet`
- 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 `csv`
- Чтение данных из файла необходимо реализовать с помощью класса `java.util.Scanner`
- 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, отсутсвие прав доступа к файлу и т. п.).

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 SpaceMarine {
    private Long 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.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Integer health; //Поле не может быть null, Значение поля должно быть больше 0
    private boolean loyal;
    private Weapon weaponType; //Поле может быть null
    private MeleeWeapon meleeWeapon; //Поле может быть null
    private Chapter chapter; //Поле не может быть null
}
public class Coordinates {
    private Double x; //Значение поля должно быть больше -267, Поле не может быть null
    private long y; //Максимальное значение поля: 803
}
public class Chapter {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private String world; //Поле не может быть null
}
public enum Weapon {
    BOLTGUN,
    COMBI_PLASMA_GUN,
    FLAMER,
    INFERNO_PISTOL;
}
public enum MeleeWeapon {
    POWER_SWORD,
    MANREAPER,
    LIGHTING_CLAW,
    POWER_BLADE;
}
```

Диаграмма классов разработанной программы

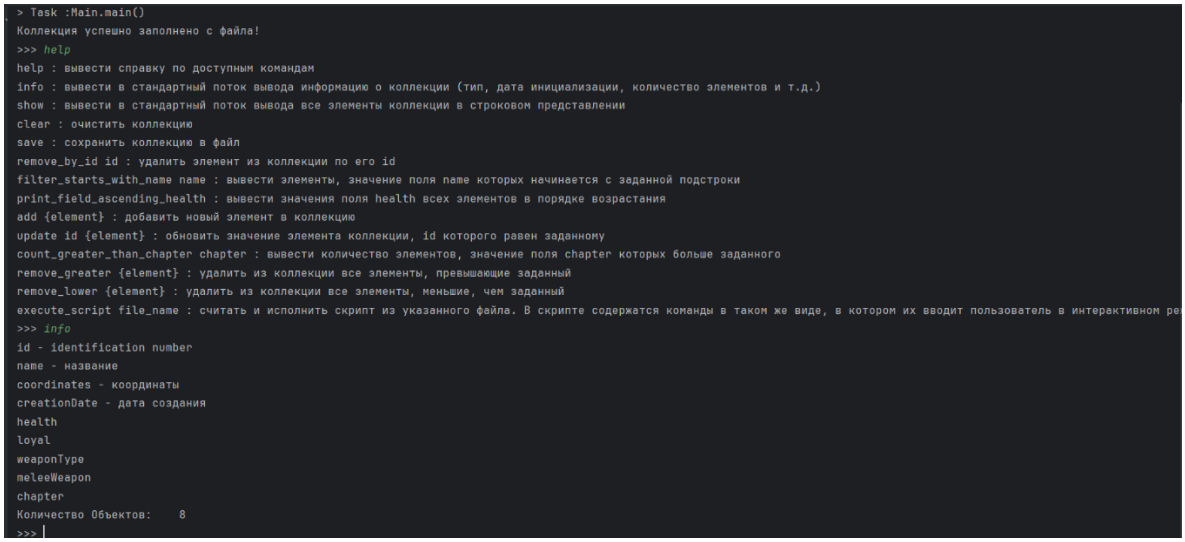
Исходный код программы (класс main)

Main.java

```java
package org.example;

import classes.*;
import collection.CollectionManager;
import command.CommandManager;
import statics.ConsoleManager;

import java.io.IOException;
import java.util.TreeSet;

public class Main {
    public static void main(String[] args) throws IOException {
        //String filename = "Datas/baza.csv";
        String filename = System.getenv( name: "BD"); //(BD=Datas/baza.csv) environment variable  if in IntelliJ*/
        CollectionManager collectionManager = new CollectionManager();
        CommandManager cmd = new CommandManager();
        ConsoleManager console = new ConsoleManager();

        TreeSet<SpaceMarine> mySet = new TreeSet<>();
        mySet = collectionManager.getAllData(filename);

        console.start(cmd, filename, mySet);
    }
}
```

Весь код доступен тут: https://github.com/karillisa/Laboratory-work-5

Результат работы программы



Вывод

Во время выполнения данной лабораторной работы я научилась использовать Javadoc, попробовала проектировать архитектуру проекта, работать с потоками, файлами, интерфейсами Comparable и Comparator, научился сереализировать и десериализировать данные в различные форматы.