

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники
(ФПИиКТ)

Программирование
Лабораторная работа №5
Вариант 3611

Выполнил

Григорьев Даниил Александрович

Группа Р3116

Санкт-Петербург 2025

Текст задания

Лабораторная работа #5

Введите вариант: 3611

Внимание! У разных вариантов разный 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: **аргумент командной строки**.
- Данные должны храниться в файле в формате `json`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.InputStreamReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.BufferedOutputStream`
- 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)
- `add_if_max {element}` : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- `add_if_min {element}` : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
- `history` : вывести последние 15 команд (без их аргументов)
- `sum_of_health` : вывести сумму значений поля `health` для всех элементов коллекции
- `filter_by_achievements achievements` : вывести элементы, значение поля `achievements` которых равно заданному
- `print_ascending` : вывести элементы коллекции в порядке возрастания

Описание хранимых в коллекции классов:

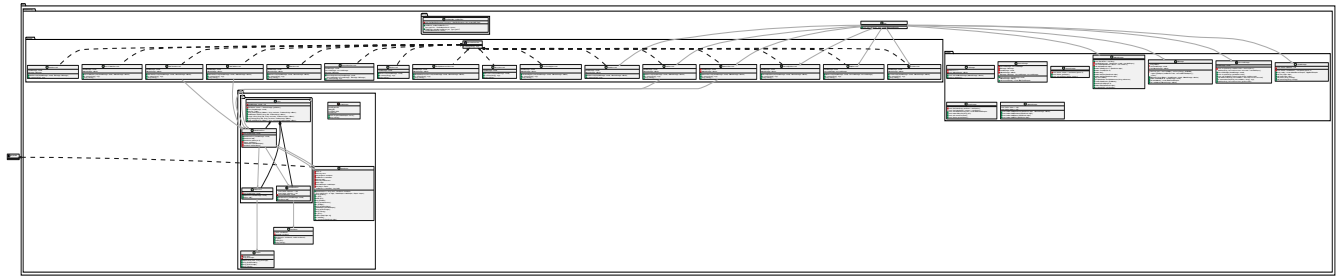
```
public class SpaceMarine {
    private int id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.time.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private Long health; //Поле не может быть null, Значение поля должно быть больше 0
    private String achievements; //Поле может быть null
    private int height;
    private MeleeWeapon meleeWeapon; //Поле не может быть null
    private Chapter chapter; //Поле не может быть null
}

public class Coordinates {
    private int x; //Значение поля должно быть больше -549
    private double y; //Значение поля должно быть больше -267
}

public class Chapter {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private String parentLegion;
}

public enum MeleeWeapon {
    POWER_SWORD,
    CHAIN_AXE,
    LIGHTING_CLAW,
    POWER_FIST;
}
```

Диаграмма классов объектной модели



Исходный код программы

Представлен по ссылке:

<https://github.com/special-meme/labs-Megadeth2006>

Вывод

Я изучил работу с потоками ввода-вывода в Java, включая буферизацию, цепочки обёрток, работу с `InputStream`, `Reader`, `BufferedReader`, `ByteArrayInputStream` и конвертацией байтов в символы. Освоил `generics` и `wildcard`-параметры, разобрался в их типах и применении. Ознакомился с основными аннотациями `Javadoc` и их использованием для документирования классов, методов и полей. Также я написал консольное приложение на Java, реализующее заданный функционал, и продемонстрировал его выполнение в `helios`.