

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

Факультет программной инженерии и компьютерной техники

Отчет

По лабораторной работе №5

По дисциплине «Программирование»

Вариант 2832.

Выполнил: Чураков Александр Алексеевич,
группа Р3131

Преподаватель: Абузов Ярослав Александрович.

Санкт-Петербург

~ 2024 ~

Задание

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Movie`, 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.Vector`
- 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.InputStreamReader`
- 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)
- `remove_at index`: удалить элемент, находящийся в заданной позиции коллекции (index)
- `add_if_max {element}`: добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- `sort`: отсортировать коллекцию в естественном порядке
- `print_field_ascending_total_box_office`: вывести значения поля `totalBoxOffice` всех элементов в порядке возрастания
- `print_field_descending_genre`: вывести значения поля `genre` всех элементов в порядке убывания
- `print_field_descending_usa_box_office`: вывести значения поля `usaBoxOffice` всех элементов в порядке убывания

Описание хранимых в коллекции классов:

```
public class Movie {
    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
    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 oscarsCount; //Значение поля должно быть больше 0, Поле может быть null
    private int totalBoxOffice; //Значение поля должно быть больше 0
    private Long usaBoxOffice; //Поле не может быть null, Значение поля должно быть больше 0
    private MovieGenre genre; //Поле может быть null
    private Person operator; //Поле не может быть null
}

public class Coordinates {
    private double x;
    private Long y; //Максимальное значение поля: 117, Поле не может быть null
}

public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private java.util.Date birthday; //Поле не может быть null
    private float weight; //Значение поля должно быть больше 0
    private Color eyeColor; //Поле может быть null
    private Country nationality; //Поле не может быть null
}

public enum MovieGenre {
    MUSICAL,
    THRILLER,
    FANTASY,
    SCIENCE_FICTION;
}
```

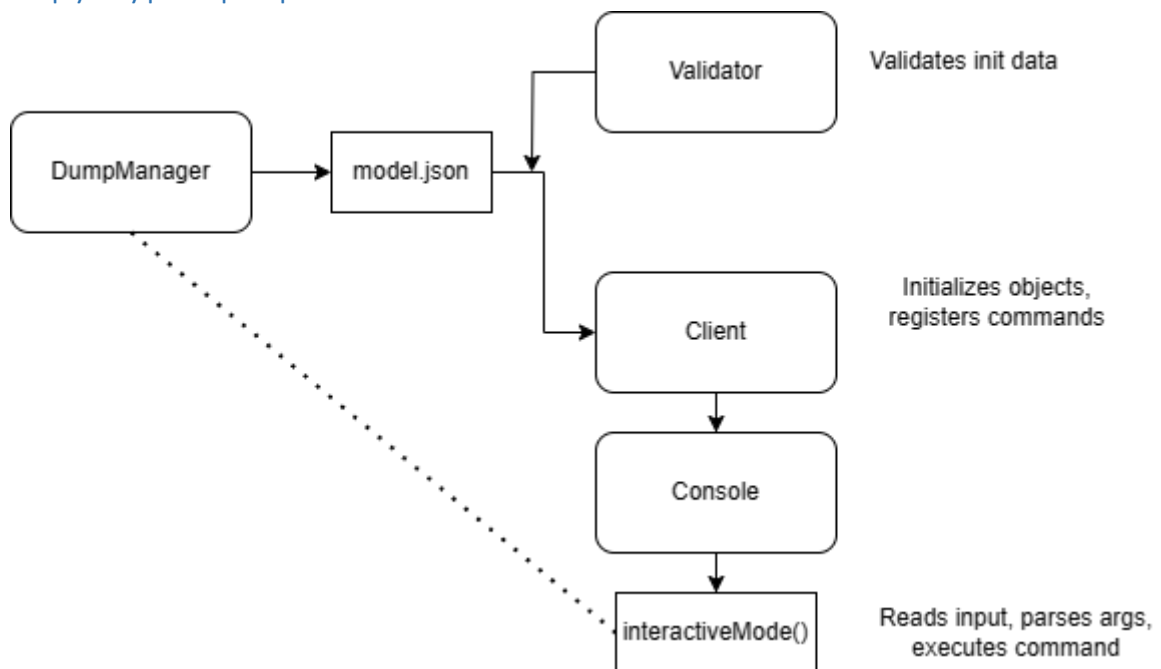
```

public enum Color {
    RED,
    BLACK,
    YELLOW,
    ORANGE,
    WHITE;
}

public enum Country {
    UNITED_KINGDOM,
    GERMANY,
    FRANCE;
}

```

Структура программы



Исходный код

[ITMO_Labs/PROG/lab5 at main · awesoma31/ITMO_Labs \(github.com\)](https://github.com/awesoma31/ITMO_Labs/tree/main/PROG/lab5)

Вывод

Во время выполнения данной лабораторной работы я научился работать с различными структурами данных в Java и файлами, а также углубил свои знания о ООП в Java, изучил параметризованные типы, wildcard-параметры и утилиту javadoc.