

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа по программированию №5

Вариант №4665

Выполнил:

Студент группы Р3106

Мельник Фёдор Александрович

Проверил:

Вербовой. А. А.,

Преподаватель-практик ФПиКТ

Санкт-Петербург, 2025

Оглавление

Задание.....	3
UML-диаграмма.....	5
Исходный код	6
Вывод	7

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.ArrayDeque`
- 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.io.BufferedReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.FileOutputStream`
- 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
- `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
- `max_by_operator`: вывести любой объект из коллекции, значение поля `operator` которого является максимальным
- `count_by_operator operator`: вывести количество элементов, значение поля `operator` которых равно заданному
- `count_less_than_genre genre`: вывести количество элементов, значение поля `genre` которых меньше заданного

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 Movie {
    private long id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться авт
    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 Long oscarsCount; //Значение поля должно быть больше 0, Поле не может быть null
    private MovieGenre genre; //Поле может быть null
    private Mpaarating mpaarating; //Поле может быть null
    private Person operator; //Поле может быть null
}

public class Coordinates {
    private Float x; //Поле не может быть null
    private int y; //Значение поля должно быть больше -486
}

public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private java.util.Date birthday; //Поле может быть null
    private long weight; //Значение поля должно быть больше 0
    private String passportID; //Длина строки не должна быть больше 25, Строка не может быть пустой, Поле не может быть null
}

public enum MovieGenre {
    DRAMA,
    MUSICAL,
    HORROR;
}

public enum Mpaarating {
    G,
    PG,
    PG_13,
    R,
    NC_17;
}
```

UML-диаграмма

UML-диаграмму можно найти по ссылке:

https://github.com/ldpst/ProgrammingLab5/blob/main/diagrams/UML_Prog_Lab5.png

Исходный код

Исходный код можно найти по ссылке:

<https://github.com/ldpst/ProgrammingLab5/tree/main/src/main/java/ru/itmo/prog/lab5>

Вывод

Во время выполнения лабораторной работы я изучил множество новых для меня методов и функций. Научился писать свои консольные приложения на языке программирования Java, работать с потоками, параметризованными типами, Generic'ами и многим другими полезными фишками Java.