Университет ИТМО

Факультет ПИиКТ

Дисциплина: программирование

Лабораторная работа №5

Выполнил: Григорьев Александр Алексеевич,

группа Р3130

Преподаватель: Харитонова Анастасия Евгеньевна

г. Санкт-Петербург, 2021 год

**Задание к лабораторной работе:**

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.Hashtable
* 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.BufferedInputStream
* 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, отсутсвие 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
* insert null {element} : добавить новый элемент с заданным ключом
* update id {element} : обновить значение элемента коллекции, id которого равен заданному
* remove\_key null : удалить элемент из коллекции по его ключу
* 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\_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный
* replace\_if\_greater null {element} : заменить значение по ключу, если новое значение больше старого
* remove\_greater\_key null : удалить из коллекции все элементы, ключ которых превышает заданный
* sum\_of\_oscars\_count : вывести сумму значений поля oscarsCount для всех элементов коллекции
* filter\_starts\_with\_name name : вывести элементы, значение поля name которых начинается с заданной подстроки
* filter\_greater\_than\_oscars\_count oscarsCount : вывести элементы, значение поля oscarsCount которых больше 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; 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 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 автоматически

private String name; //Поле не может быть null, Строка не может быть пустой

private Coordinates coordinates; //Поле не может быть null

private java.util.Date creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private int oscarsCount; //Значение поля должно быть больше 0

private MovieGenre genre; //Поле может быть null

private MpaaRating mpaaRating; //Поле не может быть null

private Person director; //Поле может быть null

}

public class Coordinates {

private double x; //Максимальное значение поля: 393

private double y;

}

public class Person {

private String name; //Поле не может быть null, Строка не может быть пустой

private String passportID; //Значение этого поля должно быть уникальным, Длина строки должна быть не меньше 10, Длина строки не должна быть больше 49, Поле может быть null

private Color eyeColor; //Поле не может быть null

private Color hairColor; //Поле может быть null

private Country nationality; //Поле может быть null

}

public enum MovieGenre {

THRILLER,

HORROR,

FANTASY;

}

public enum MpaaRating {

G,

PG,

PG\_13,

NC\_17;

}

public enum Color {

GREEN,

ORANGE,

WHITE;

}

public enum Color {

GREEN,

RED,

YELLOW,

ORANGE;

}

public enum Country {

UNITED\_KINGDOM,

SPAIN,

VATICAN,

ITALY,

JAPAN;

}

**Исходный код:**

<https://github.com/aleksgright/prog_labs/tree/master/Lab5>

**Вывод:** во время выполнения данной лабораторной работы мы закрепили принципы SOLID, собственные исключения и многое другое. Также мы научились использовать Javadoc, работать с потоками, файлами, интерфейсами Comparator и Comparable. Узнали что такое сериализация и десериализация.