МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

**ЛАБОРАТОРНАЯ РАБОТА № 5**

по дисциплине

«Программирование»

Вариант № 6901

**Выполнил:**

студент группы Р3117

Агей Михаил Александрович  
  
**Преподаватели:**

Алина Сагайдак

Екатерина …

Санкт-Петербург

2022

# Текст задания

**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.HashSet
* 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 xml
* Чтение данных из файла необходимо реализовать с помощью класса java.util.Scanner
* Запись данных в файл необходимо реализовать с помощью класса java.io.FileWriter
* 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
* 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 12 команд (без их аргументов)
* remove\_any\_by\_oscars\_count oscarsCount : удалить из коллекции один элемент, значение поля oscarsCount которого эквивалентно заданному
* sum\_of\_length : вывести сумму значений поля length для всех элементов коллекции
* count\_by\_oscars\_count oscarsCount : вывести количество элементов, значение поля oscarsCount которых равно заданному

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

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

private long length; //Значение поля должно быть больше 0

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 Integer x; //Значение поля должно быть больше -319, Поле не может быть null

private int y;

}

public class Person {

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

private String passportID; //Длина строки должна быть не меньше 9, Строка не может быть пустой, Поле может быть null

private Country nationality; //Поле может быть null

private Location location; //Поле может быть null

}

public class Location {

private long x;

private long y;

private double z;

}

public enum MovieGenre {

WESTERN,

ADVENTURE,

TRAGEDY;

}

public enum MpaaRating {

G,

PG\_13,

R,

NC\_17;

}

public enum Country {

FRANCE,

INDIA,

VATICAN,

THAILAND,

SOUTH\_KOREA;

}

# Диаграмма классов разработанной программы

# 

# Исходный код программы

Весь проект доступен по ссылке: [МихаилAJ/lab5\_prog (github.com)](https://github.com/MishailAJ/lab5_prog)

# Выводы по работе

Выполнив лабораторную работу №5 по программированию я научился использовать обобщения (generics) в своей программе, познакомился с библиотекой simpleXML для парсинга XML формата, а также потренировался в создании оптимальной архитектуры проекта, в обработке ошибок и написании комментариев к своим файлам (javadoc).