Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №1257

Лабораторная работа №5

По дисциплине

Программирование

Выполнил студент группы P3113:

Бободжонов Комронджон

Преподаватель:

Письмак Алексей Евгеньевич

Санкт-Петербург 2024 г.

1. **Текст задания**

|  |
| --- |
| 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.PriorityQueue * 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.util.Scanner * 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) * remove\_first : удалить первый элемент из коллекции * head : вывести первый элемент коллекции * remove\_lower {element} : удалить из коллекции все элементы, меньшие, чем заданный * count\_by\_discount discount : вывести количество элементов, значение поля discount которых равно заданному * filter\_greater\_than\_discount discount : вывести элементы, значение поля discount которых больше заданного * print\_ascending : вывести элементы коллекции в порядке возрастания   **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 long 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.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически  private double price; //Значение поля должно быть больше 0  private int discount; //Значение поля должно быть больше 0, Максимальное значение поля: 100  private Boolean refundable; //Поле может быть null  private MovieType type; //Поле может быть null  private Venue venue; //Поле не может быть null  }  public class Coordinates {  private Long x; //Поле не может быть null  private Double y; //Поле не может быть null  }  public class Person {  private Long 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 long capacity; //Значение поля должно быть больше 0  private PersonType type; //Поле может быть null  }  public enum MovieGenre{  ACTION,  WESTERN,  HORROR,  FANTASY;  }  public enum PersonType {  PUB,  OPEN\_AREA,  STADIUM;  } |
|  |

1. **Исходный код программы.**
2. **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, wildcard-параметры.