МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальная научно-образовательная корпорация ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №5

По дисциплине

«Программирование»

Вариант №31156

**Выполнил:**

Студент группы P3114

Щербинин Эдуард Павлович

**Преподаватель:**

Инячина Диана Александровна

**Содержание**

[Задание 3](#_Toc128331359)

[Исходный код программы 5](#_Toc128331360)

[Документация Javadoc 5](#_Toc128331361)

[Вывод 6](#_Toc128331362)

Задание

До сброса вариантов, был следующий:

* 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.LinkedList
* При запуске приложения коллекция должна автоматически заполняться значениями из файла.
* Имя файла должно передаваться программе с помощью: **аргумент командной строки**.
* 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.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.).

Команды:

- 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

- remove\_greater {id} : удалить из коллекции все элементы, превышающие заданный

- remove\_lower {id} : удалить из коллекции все элементы, меньшие, чем заданный

- remove\_all\_by\_event {event} : удалить из коллекции все элементы, значение поля event которого эквивалентно заданному

- filter\_contains\_comment [comment] : вывести элементы, значение поля comment которых содержит заданную подстроку

- print\_unique\_price : вывести уникальные значения поля price всех элементов в коллекции  
- sort : отсортировать коллекцию

Описание хранимых классов:

public class Ticket {

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 ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

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

private String comment; //Поле может быть null

private *TicketType* type; //Поле не может быть null

private Event event; //Поле не может быть null

}

public class Coordinates {

private float x; //Максимальное значение поля: 542

private Integer y; //Максимальное значение поля: 203, Поле не может быть null

}

public class Event {

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 LocalDate date; //Поле может быть null

private long minAge;

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

private *EventType* eventType; //Поле не может быть null

}

public enum *TicketType* {

***VIP***,

***USUAL***,

***BUDGETARY***,

***CHEAP***;

}

public enum *EventType* {

***CONCERT***,

***E\_SPORTS***,

***FOOTBALL***,

***BASKETBALL***,

***THEATRE\_PERFORMANCE***;

}

Исходный код программы

<https://github.com/edshPC/CollectionWork>

Документация Javadoc

<https://htmlpreview.github.io/?https://github.com/edshPC/CollectionWork/blob/main/doc/index.html>

Название диаграммы – diagram.png

Вывод

Во время выполнения работы я ознакомился с синтаксисом языка Java, ещё лучше узнал преимущества ООП, познакомился с ним на практике, разобрался что такое Command pattern, разобрался с коллекциями Java и их обработкой, написал интерактивную программу, создал объектную модель по тексту. Полученные знания понадобятся в процессе дальнейшего обучения.