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

Факультет Программной инженерии и компьютерной техники

Отчёт по лабораторной работе №5

Программирование

Вариант № 8137

Выполнил: студент группы P3131

Серов А. А.

Проверил: Обляшевский С.А.

Санкт-Петербург 2025г.

1. **Текст задания**

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Ticket, 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.ArrayList
* 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.FileReader
* Запись данных в файл необходимо реализовать с помощью класса java.io.OutputStreamWriter
* 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\_min {element}: добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
* remove\_lower {element}: удалить из коллекции все элементы, меньшие, чем заданный
* reorder: отсортировать коллекцию в порядке, обратном нынешнему
* remove\_any\_by\_type type: удалить из коллекции один элемент, значение поля type которого эквивалентно заданному
* count\_by\_person person: вывести количество элементов, значение поля person которых равно заданному
* print\_field\_ascending\_person: вывести значения поля person всех элементов 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 Ticket {

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.time.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически

private Integer price; //Поле может быть null, Значение поля должно быть больше 0

private Integer discount; //Поле может быть null, Значение поля должно быть больше 0, Максимальное значение поля: 100

private TicketType type; //Поле может быть null

private Person person; //Поле не может быть null

}

public class Coordinates {

private long x;

private long y; //Максимальное значение поля: 166

}

public class Person {

private String passportID; //Длина строки должна быть не меньше 4, Длина строки не должна быть больше 49, Поле может быть null

private Color eyeColor; //Поле может быть null

private Color hairColor; //Поле может быть null

}

public enum TicketType {

VIP,

BUDGETARY,

CHEAP;

}

public enum Color {

GREEN,

RED,

BLUE,

YELLOW;

}

public enum Color {

GREEN,

RED,

YELLOW,

ORANGE;

}

1. **Диаграмма классов исходной программы**

<https://github.com/H4SKEY/Lab5_Prog/blob/main/UML_lab5_Prog.svg>

1. **Исходный код программы**

[**https://github.com/H4SKEY/Lab5\_Prog/tree/main/src/main/java/org/example**](https://github.com/H4SKEY/Lab5_Prog/tree/main/src/main/java/org/example)

1. **Выводы по работе**

В ходе работы было разработано консольное приложение для управления коллекцией объектов с полной реализацией требуемого функционала, включая загрузку/сохранение данных в JSON-формате, обработку множества видов команд, валидацию ввода и автоматическую генерацию полей. Приложение демонстрирует корректную работу с коллекцией, эффективную организацию кода с использованием принципов ООП и шаблона Command, а также устойчивость к ошибкам пользовательского ввода. Все требования технического задания выполнены в полном объеме, что подтверждает готовность системы к практическому использованию.