Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №5**

**по программированию**

**Вариант: 333049**

**Выполнила:**

Павлова Алина

Группа: P3112

**Принял:**

Сорокин Роман Борисович

Санкт-Петербург

2022 г.

1. **Текст задания**

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса **Route**, 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.TreeSet**

⦁ 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)

⦁ 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 14 команд (без их аргументов)

⦁ remove\_any\_by\_distance distance : удалить из коллекции один элемент, значение поля distance которого эквивалентно заданному

⦁ remove\_greater {element} : удалить из коллекции все элементы, превышающие заданный

⦁ 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 Route {

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

private LocationFrom from; //Поле не может быть null

private LocationTo to; //Поле не может быть null

private long distance; //Значение поля должно быть больше 1

}

public class Coordinates {

private Double x; //Поле не может быть null

private Float y; //Поле не может быть null

}

public class LocationFrom {

private Double x; //Поле не может быть null

private Double y; //Поле не может быть null

private String name; //Поле не может быть null

}

public class LocationTo {

private double x;

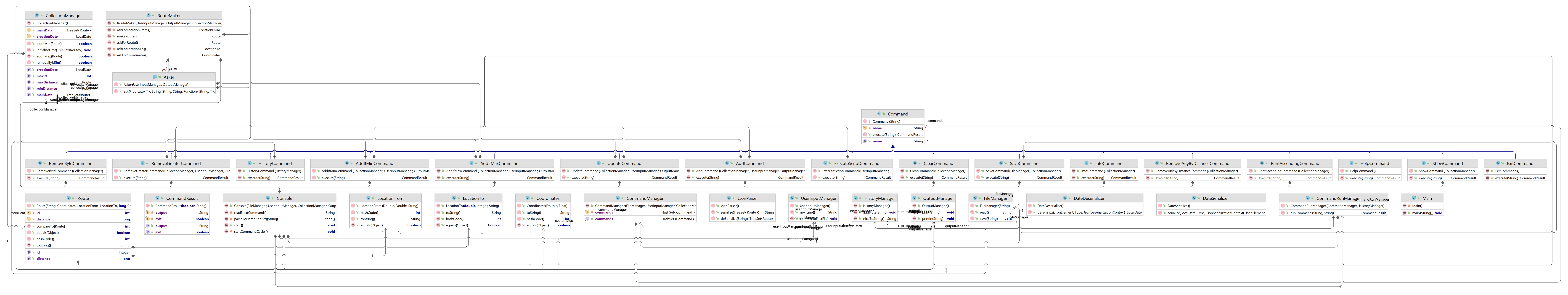
private Integer y; //Поле не может быть null

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

}

1. **Диаграмма классов разработанной программы**

https://github.com/alinautumn/itmo-lab5-alisha/blob/master/docs/lab5-diagram.png



1. **Исходный код программы**

https://github.com/alinautumn/itmo-lab5-alisha

1. **Выводы по работе**

Во время выполнения лабораторной работы я научилась работать с коллекциями в Java, с потоками ввода вывода, осуществлять чтение и запись файлов, обрабатывать исключения, строить архитектуру консольного приложения.