Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет

ИТМО»

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение*

*Дисциплина «Программирование»*

**Отчет**

**По лабораторной работе №5**

**Вариант №1754**

Выполнил:

Кирячек Тимофей Алексеевич

Группа: P3109

Преподаватель:  
Остапенко Иван Вадимович

Г. Санкт-Петербург, 2024 г.

**Оглавление**

[Задание 2](#_Toc158638123)

[UML-диаграмма классов: 5](#_Toc158638124)

[Код программы: 6](#_Toc158638125)

[Заключение 7](#_Toc158638126)

[Список литературы. 8](#_Toc158638127)

# Задание

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.Stack
* 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.io.FileReader
* Запись данных в файл необходимо реализовать с помощью класса java.io.BufferedWriter
* 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)
* remove\_at index : удалить элемент, находящийся в заданной позиции коллекции (index)
* reorder : отсортировать коллекцию в порядке, обратном нынешнему
* sort : отсортировать коллекцию в естественном порядке
* count\_greater\_than\_distance distance : вывести количество элементов, значение поля distance которых больше заданного
* print\_ascending : вывести элементы коллекции в порядке возрастания
* print\_field\_descending\_distance : вывести значения поля distance всех элементов в порядке убывания

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

private Location from; //Поле может быть null

private Location to; //Поле не может быть null

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

}

public class Coordinates {

private Long x; //Поле не может быть null

private Long y; //Поле не может быть null

}

public class Location {

private float x;

private Float y; //Поле не может быть null

private double z;

private String name; //Поле может быть null

}

public class Location {

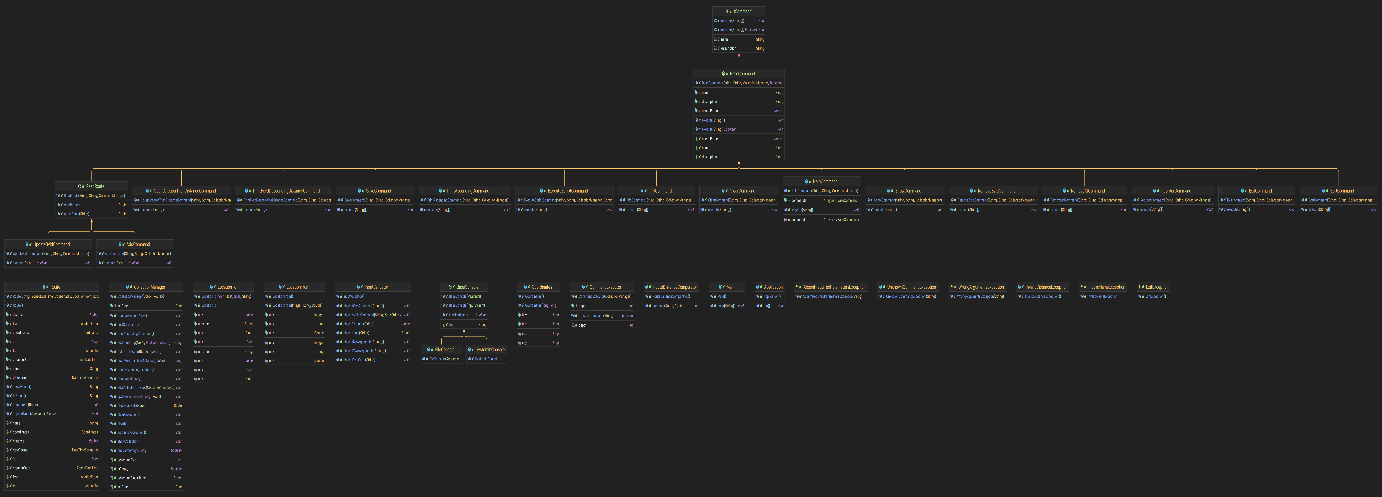
private Integer x; //Поле не может быть null

private long y;

private Double z; //Поле не может быть null

}

# UML-диаграмма классов:



# Код программы:

Исходный код может быть найден в [этом репозитории](https://github.com/TecHeReTiC3141/Lab5)

# Заключение

В процессе выполнения лабораторной работы я разработал консольное приложение, получил опыт использования встроенных коллекций, например, Stack, освоил файловый ввод-вывод, узнал о Command pattern и реализовал его, а кроме того, научился создавать JavaDoс и более-менее разумно документировать свой код, без этого такую комплексное приложение было очень тяжело разрабатывать

# Список литературы.

1. https://se.ifmo.ru/courses/programming
2. <https://javarush.ru/groups/posts/1981-dlja-chego-v-java-nuzhnih-interfeysih>
3. <https://javarush.ru/groups/posts/591-vot-tak-final>
4. <https://www.youtube.com/watch?v=3Xo6zSBgdgk>
5. <https://javarush.ru/groups/posts/1985-raznica-mezhdu-abstraktnihmi-klassami-i-interfeysami>
6. <https://skillbox.ru/media/base/klassy-i-obekty-v-java/>
7. <https://www.jetbrains.com/help/idea/class-diagram.html#analyze_graph>
8. <https://metanit.com/java/tutorial/3.3.php>
9. JAVA from EPAM, 2-е издание, исправленное. И. Н. Блинов, В. С. Романчук
10. <https://javarush.ru/groups/posts/1967-otnoshenija-mezhdu-klassami-nasledovanie-kompozicija-i-agregirovanie->
11. <https://javarush.ru/groups/posts/599-inicializacija-poley>
12. https://javarush.ru/groups/posts/2347-klass-string-v-java