Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №5988

Лабораторная работа №5

По дисциплине

Программирование

Выполнил студент группы P3131:

Мироненко Артем Дмитриевич

Преподаватель:

Абузов Ярослав Александрович

Санкт-Петербург

~ 2023 ~

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: **аргумент командной строки**.
* Данные должны храниться в файле в формате xml
* Чтение данных из файла необходимо реализовать с помощью класса java.io.BufferedInputStream
* Запись данных в файл необходимо реализовать с помощью класса java.io.BufferedOutputStream
* 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\_greater {element} : удалить из коллекции все элементы, превышающие заданный
* /history : вывести последние 6 команд (без их аргументов)
* /group\_counting\_by\_distance : сгруппировать элементы коллекции по значению поля distance, вывести количество элементов в каждой группе
* /print\_descending : вывести элементы коллекции в порядке убывания
* /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 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 Location from; //Поле может быть null

private Location to; //Поле может быть null

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

}

public class Coordinates {

private int x;

private Float y; //Поле не может быть null

}

public class Location {

private int x;

private Long y; //Поле не может быть null

private String name; //Поле может быть null

}

public class Location {

private double x;

private double y;

private String name; //Поле может быть null

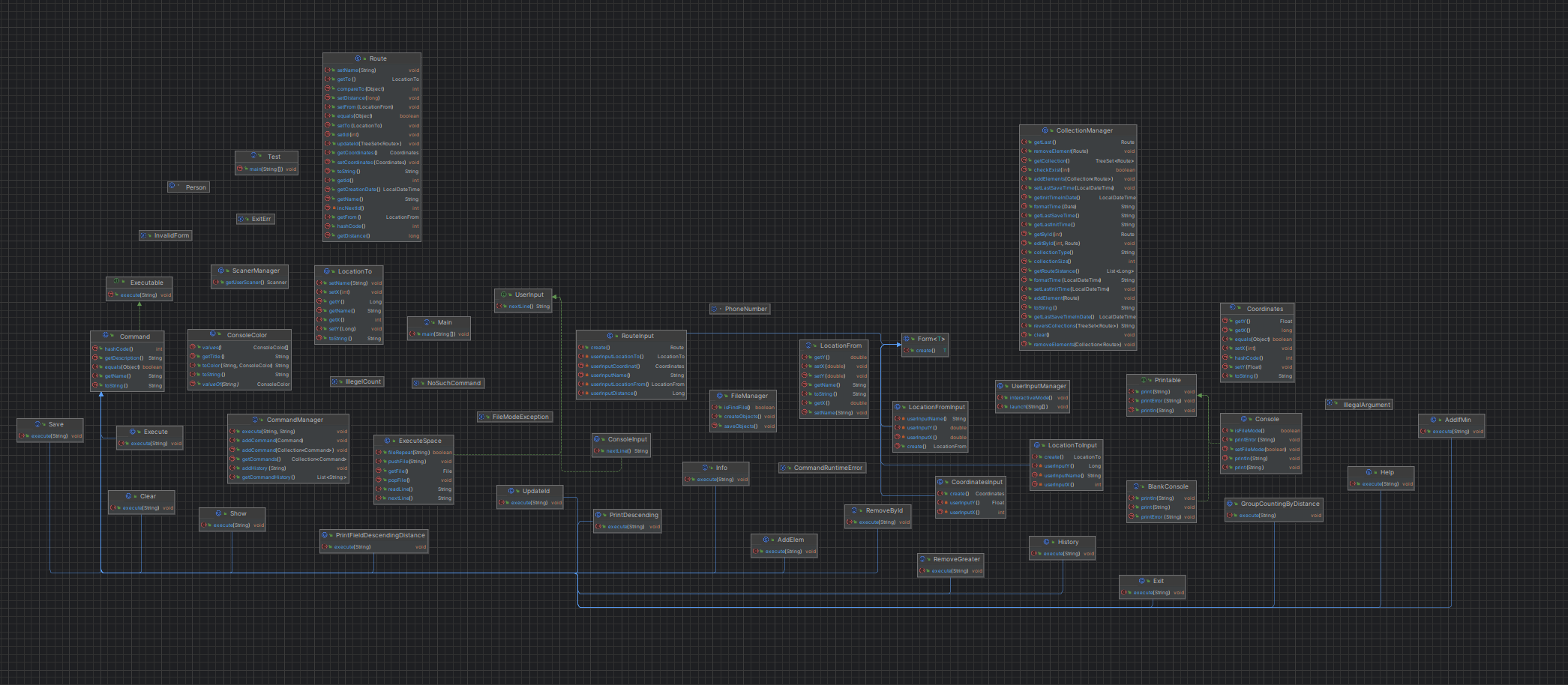
}

1. Исходный код программы

**https://github.com/tteemma/ITMO/tree/master/Sem2/Прога/labwork5**

<https://github.com/tteemma/labwork_5/tree/master/Прога/labwork5>

1. Диаграмма классов в реализованной модели



1. 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, изучил утилиту Javadoc. Научился работать с системой сборки Maven.