

Национальный исследовательский университет информационных технологий, механики и оптики

Факультет Программной Инженерии и Компьютерной Техники

Вариант № 3311706

Лабораторная работа №5

По дисциплине:
«Программирование»

Работу выполнила:

Студентка группы Р3112

Никонова Наталья Игоревна

Преподаватель:

Яркеев Александр Сергеевич

Санкт-Петербург

2021

Задание

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.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.BufferedWriter`
- 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)
- `insert_at index {element}` : добавить новый элемент в заданную позицию
- `sort` : отсортировать коллекцию в естественном порядке
- `history` : вывести последние 8 команд (без их аргументов)
- `remove_any_by_transport transport` : удалить из коллекции один элемент, значение поля transport которого эквивалентно заданному
- `average_of_number_of_rooms` : вывести среднее значение поля numberOfRooms для всех элементов коллекции
- `filter_less_than_view view` : вывести элементы, значение поля view которых меньше заданного

Описание хранимых в коллекции классов:

```
public class Flat {
    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.ZonedDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private long area; //Значение поля должно быть больше 0
    private Integer numberOfRooms; //Поле может быть null, Значение поля должно быть больше 0
    private boolean centralHeating;
    private View view; //Поле не может быть null
    private Transport transport; //Поле не может быть null
    private House house; //Поле не может быть null
}

public class Coordinates {
    private double x;
    private Double y; //Поле не может быть null
}

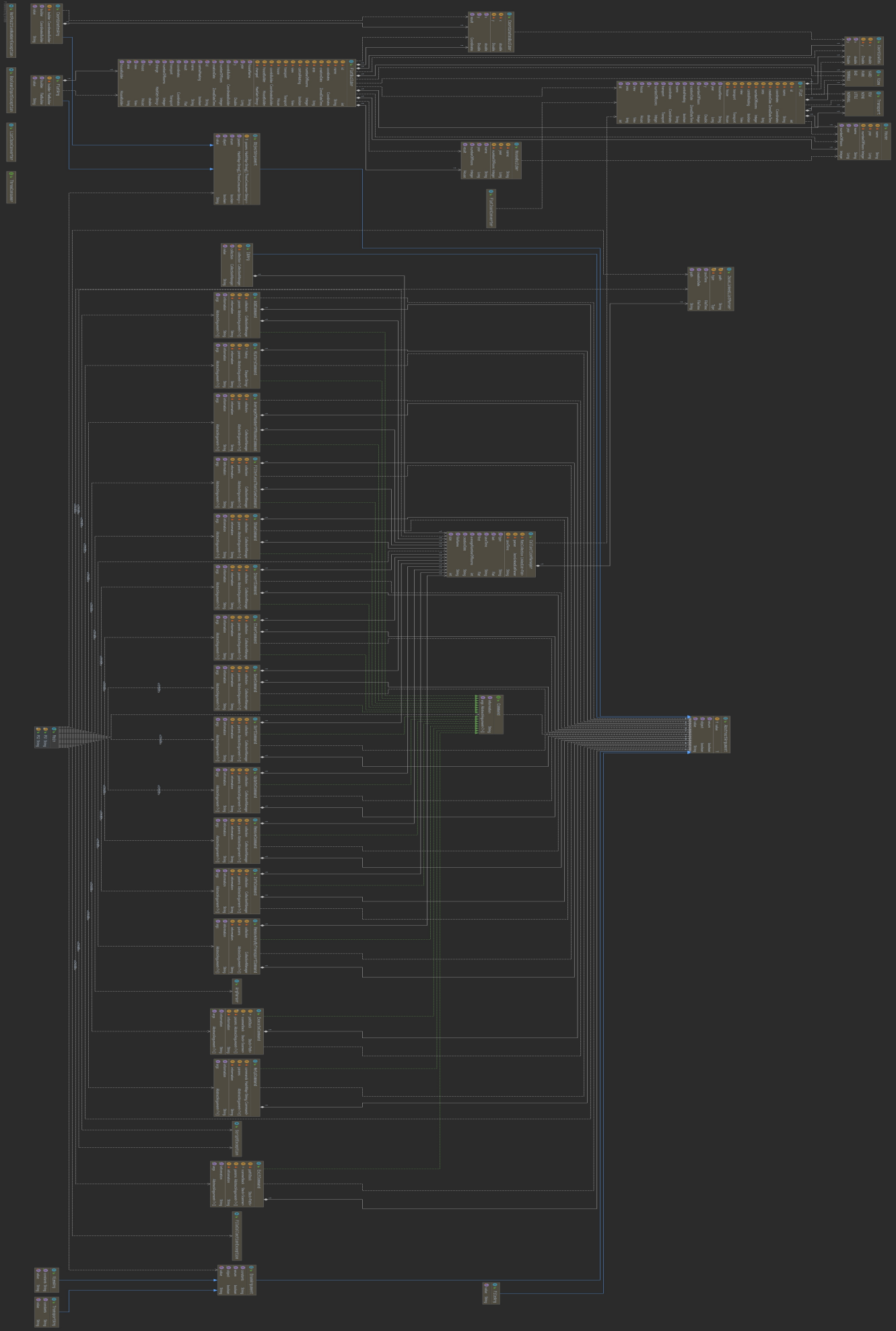
public class House {
    private String name; //Поле может быть null
    private Long year; //Поле может быть null, Значение поля должно быть больше 0
    private Integer numberOfFloors; //Поле может быть null, Значение поля должно быть больше 0
}

public enum View {
    YARD,
    PARK,
    BAD,
    TERRIBLE;
}

public enum Transport {
    FEW,
    NONE,
    LITTLE,
    NORMAL;
}
```

Исходный код: <https://github.com/nanikon/FlatCollection>

Диаграмма:



Выводы

В ходе выполнения лабораторной работы я познакомилась с некоторыми коллекциями в Java, обобщенным программированием, некоторыми потоками ввода\вывода. Опробовала такие паттерны проектирования, как Команда и Билдер. Вспомнила принципы SOLID и снова попыталась применять их при проектировании классов, поработала с собственными исключениями, функциональными интерфейсами вместе с лямбда-выражениями и ссылками на методы. Попробовала подключить к своему проекту стороннюю библиотеку.