

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
Высшего образования
Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа 5 по программированию
Вариант №5276

Группа: Р3107

Выполнил:

Садовой Г.В.

Проверил:

Гаврилов А.В.

Г. Санкт-Петербург

2024

Оглавление

Текст задания	3
Диаграмма классов разработанной программы.....	6
Исходный код программы.....	7
Вывод.....	8

Текст задания

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса Product, 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: **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.PrintWriter`
- 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_min {element}` : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
- `sort` : отсортировать коллекцию в естественном порядке
- `history` : вывести последние 13 команд (без их аргументов)

- `remove_all_by_manufacturer manufacturer` : удалить из коллекции все элементы, значение поля `manufacturer` которого эквивалентно заданному
- `filter_by_part_number partNumber` : вывести элементы, значение поля `partNumber` которых равно заданному
- `filter_starts_with_part_number partNumber` : вывести элементы, значение поля `partNumber` которых начинается с заданной подстроки

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 Product {

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

    private String partNumber; //Длина строки должна быть не меньше 20, Длина строки не должна
    быть больше 69, Значение этого поля должно быть уникальным, Поле может быть null

    private Float manufactureCost; //Поле может быть null

    private UnitOfMeasure unitOfMeasure; //Поле может быть null

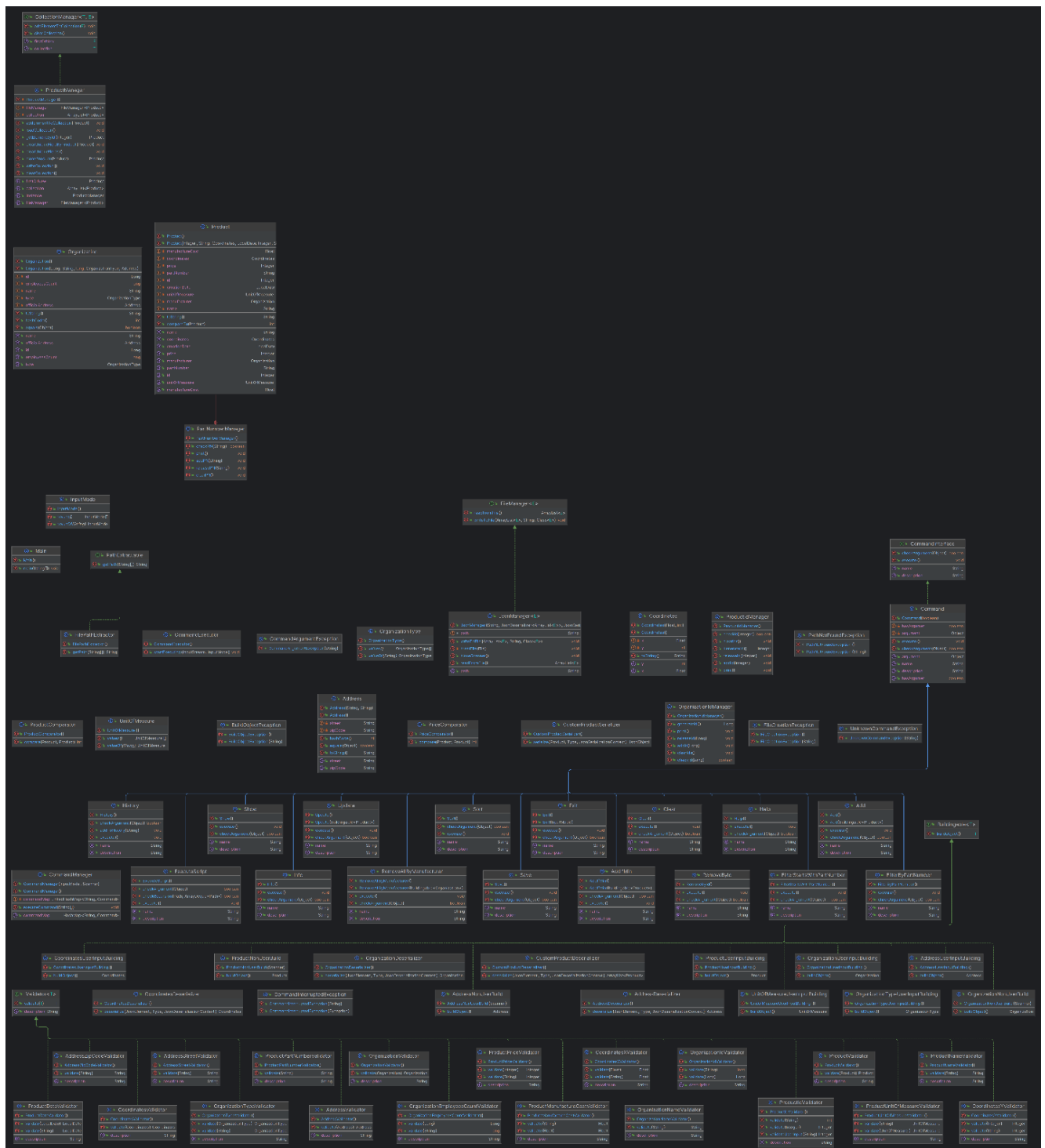
    private Organization manufacturer; //Поле не может быть null

}

public class Coordinates {
```

```
private Float x; //Поле не может быть null
private int y;
}
public class Organization {
    private Long 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 long employeesCount; //Значение поля должно быть больше 0
    private OrganizationType type; //Поле может быть null
    private Address officialAddress; //Поле может быть null
}
public class Address {
    private String street; //Длина строки не должна быть больше 148, Поле может быть null
    private String zipCode; //Длина строки не должна быть больше 14, Поле не может быть null
}
public enum UnitOfMeasure {
    METERS,
    CENTIMETERS,
    MILLIGRAMS;
}
public enum OrganizationType {
    GOVERNMENT,
    PRIVATE_LIMITED_COMPANY,
    OPEN_JOINT_STOCK_COMPANY;
}
```

Диаграмма классов разработанной программы



Исходный код программы.

<https://github.com/Umchik1222/MyITMOProjects>

Вывод

В процессе выполнения лабораторной работы я научился работать с Collection API, дженериками. Потоками ввод-вывода, классами оболочками и javadoc. Написал консольное приложение оперирующее с коллекцией элементов.