Национальный исследовательский университет ИТМО

Факультет ПИиКТ

Лабораторная работа №5

Работу выполнил:

Асташин Сергей Сергеевич

Группа: P3130

Вариант: 289327

Преподаватель: Исаев А.С.

Санкт-Петербург 2020

Задание:

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.LinkedHashSet**
- 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 csv
- Чтение данных из файла необходимо реализовать с помощью класса **java.io.BufferedReader**
- 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_max {element}** : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- **remove_greater {element}** : удалить из коллекции все элементы, превышающие заданный
- **history** : вывести последние 7 команд (без их аргументов)
- **remove_any_by_unit_of_measure unitOfMeasure** : удалить из коллекции один элемент, значение поля unitOfMeasure которого эквивалентно заданному
- **filter_by_manufacturer manufacturer** : вывести элементы, значение поля manufacturer которых равно заданному
- **print_field_descending_price** : вывести значения поля price 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'e; 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 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 Coordinates coordinates; //Поле не может быть null

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

    private float price; //Значение поля должно быть больше 0

    private String partNumber; //Значение этого поля должно быть уникальным, Поле не может быть null

    private Float manufactureCost; //Поле не может быть null

    private UnitOfMeasure unitOfMeasure; //Поле не может быть null

    private Organization manufacturer; //Поле может быть null
```

```
}
public class Coordinates {
    private Double x; //Поле не может быть null
    private Long y; //Поле не может быть null
}
public class Organization {
    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 Long annualTurnover; //Поле может быть null, Значение поля должно быть больше 0
    private Long employeesCount; //Поле может быть null, Значение поля должно быть больше 0
    private OrganizationType type; //Поле может быть null
}
public enum UnitOfMeasure {
    KILOGRAMS,
    SQUARE_METERS,
    PCS,
    LITERS,
    MILLILITERS;
}
public enum OrganizationType {
    COMMERCIAL,
    GOVERNMENT,
    PRIVATE_LIMITED_COMPANY,
    OPEN_JOINT_STOCK_COMPANY;
}
```

Ссылка на GitHub репозиторий

https://github.com/Gramdel/lab5 (Основная ветка - **/tree/main**)

Вывод:

В ходе выполнения данной работы были изучены: Java Collection Framework, иерархия интерфейсов и их реализаций, основные методы разных типов коллекций, параметризованные типы (и то, что с ними связано), классы-оболочки, автоупаковка и автораспаковка.

Также пришлось разбираться с выводом и вводом в файл несколькими способами, с настройкой git'а, документацией в формате Javadoc.

После выполненной работы пришёл к выводу, что коллекции – тема далеко не одной лабораторной работы, ведь их много, и они все полезны для различных задач, поэтому, вероятно, буду изучать их дальше за рамками курса.