

**Федеральное государственное автономное образовательное  
учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»  
Факультет Программной Инженерии и Компьютерной Техники**



**Вариант №74273  
Лабораторная работа №5  
по дисциплине  
Программирование**

Выполнил Студент группы Р3115  
**Владимир Мацюк**  
Преподаватель:  
**Кустарев Иван Павлович**

Санкт-Петербург  
2023г.

## 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:

1. Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
2. Все требования к полям класса (указанные в виде комментариев) должны быть выполнены.
3. Для хранения необходимо использовать коллекцию типа `java.util.PriorityQueue`
4. При запуске приложения коллекция должна автоматически заполняться значениями из файла.
5. Имя файла должно передаваться программе с помощью: аргумент командной строки.
6. Данные должны храниться в файле в формате `csv`
7. Чтение данных из файла необходимо реализовать с помощью класса `java.util.Scanner`
8. Запись данных в файл необходимо реализовать с помощью класса `java.io.FileWriter`
9. Все классы в программе должны быть задокументированы в формате `javadoc`.
10. 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:

1. `help` : вывести справку по доступным командам
2. `info` : вывести в стандартный поток вывода информацию о коллекции (тип, дата инициализации, количество элементов и т.д.)
3. `show` : вывести в стандартный поток вывода все элементы коллекции в строковом представлении
4. `add {element}` : добавить новый элемент в коллекцию
5. `update id {element}` : обновить значение элемента коллекции, `id` которого равен заданному
6. `remove_by_id id` : удалить элемент из коллекции по его `id`
7. `clear` : очистить коллекцию
8. `save` : сохранить коллекцию в файл
9. `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.
10. `exit` : завершить программу (без сохранения в файл)
11. `remove_first` : удалить первый элемент из коллекции
12. `add_if_max {element}` : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
13. `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный

14. `min_by_manufacture_cost` : вывести любой объект из коллекции, значение поля `manufactureCost` которого является минимальным
15. `count_less_than_owner owner` : вывести количество элементов, значение поля `owner` которых меньше заданного
16. `filter_contains_name name` : вывести элементы, значение поля `name` которых содержит заданную подстроку

## Формат ввода команд:

1. 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.
2. Все составные типы данных (объекты классов, хранящиеся в коллекции) должны вводиться по одному полю в строку.
3. При вводе составных типов данных пользователю должно показываться приглашение к вводу, содержащее имя поля (например, "Введите дату рождения:")
4. Если поле является `enum`-ом, то вводится имя одной из его констант (при этом список констант должен быть предварительно выведен).
5. 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.
6. Для ввода значений `null` использовать пустую строку.
7. Поля с комментарием "Значение этого поля должно генерироваться автоматически" не должны вводиться пользователем вручную при добавлении.

## Описание хранимых в коллекции классов:

---

```
1 public class Product {
2     private Long id; //Поле не может быть null, Значение поля должно быть больше 0,
3         ↳ Значение этого поля должно быть уникальным,
4         ↳ Значение этого поля должно генерироваться автоматически
5     private String name; //Поле не может быть null, Строка не может быть пустой
6     private Coordinates coordinates; //Поле не может быть null
7     private java.time.ZonedDateTime creationDate; //Поле не может быть null,
8         ↳ Значение этого поля должно генерироваться автоматически
9     private double price; //Значение поля должно быть больше 0
10    private float manufactureCost;
11    private UnitOfMeasure unitOfMeasure; //Поле может быть null
12    private Person owner; //Поле не может быть null
13 }
14 public class Coordinates {
15     private float x;
16     private Long y; //Поле не может быть null
17 }
18 public class Person {
19     private String name; //Поле не может быть null, Строка не может быть пустой
20     private java.time.LocalDate birthday; //Поле может быть null
21     private Country nationality; //Поле не может быть null
22 }
23 public enum UnitOfMeasure {
```

## Диаграмма классов



## Вывод

3