

**Национальный исследовательский университет информационных  
технологий, механики и оптики**

**Факультет Программной Инженерии и Компьютерной Техники**

---

**Вариант № 312986**  
**Лабораторная работа №5**  
**По дисциплине:**  
**«Программирование»**

Работу выполнила:

Студентка группы Р3112

Сенина Мария Михайловна

Преподаватель:

Яркеев Александр Сергеевич

Санкт-Петербург

2020

## Текст задания

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `LabWork`, 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.LinkedList`
- При запуске приложения коллекция должна автоматически заполняться значениями из файла.
- Имя файла должно передаваться программе с помощью: **переменная окружения**.
- Данные должны храниться в файле в формате `json`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.InputStreamReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.FileOutputStream`
- 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)
- `remove_at index` : удалить элемент, находящийся в заданной позиции коллекции (index)
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `sort` : отсортировать коллекцию в естественном порядке
- `min_by_difficulty` : вывести любой объект из коллекции, значение поля difficulty которого является минимальным
- `filter_by_description description` : вывести элементы, значение поля description которых равно заданному
- `print_descending` : вывести элементы коллекции 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; введенное 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 LabWork {
    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.LocalDateTime creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private float minimalPoint; //Значение поля должно быть больше 0
    private String description; //Поле не может быть null
    private Integer averagePoint; //Поле не может быть null, Значение поля должно быть больше 0
    private Difficulty difficulty; //Поле может быть null
    private Discipline discipline; //Поле не может быть null
}

public class Coordinates {
    private int x; //Максимальное значение поля: 74
    private long y; //Значение поля должно быть больше -47
}

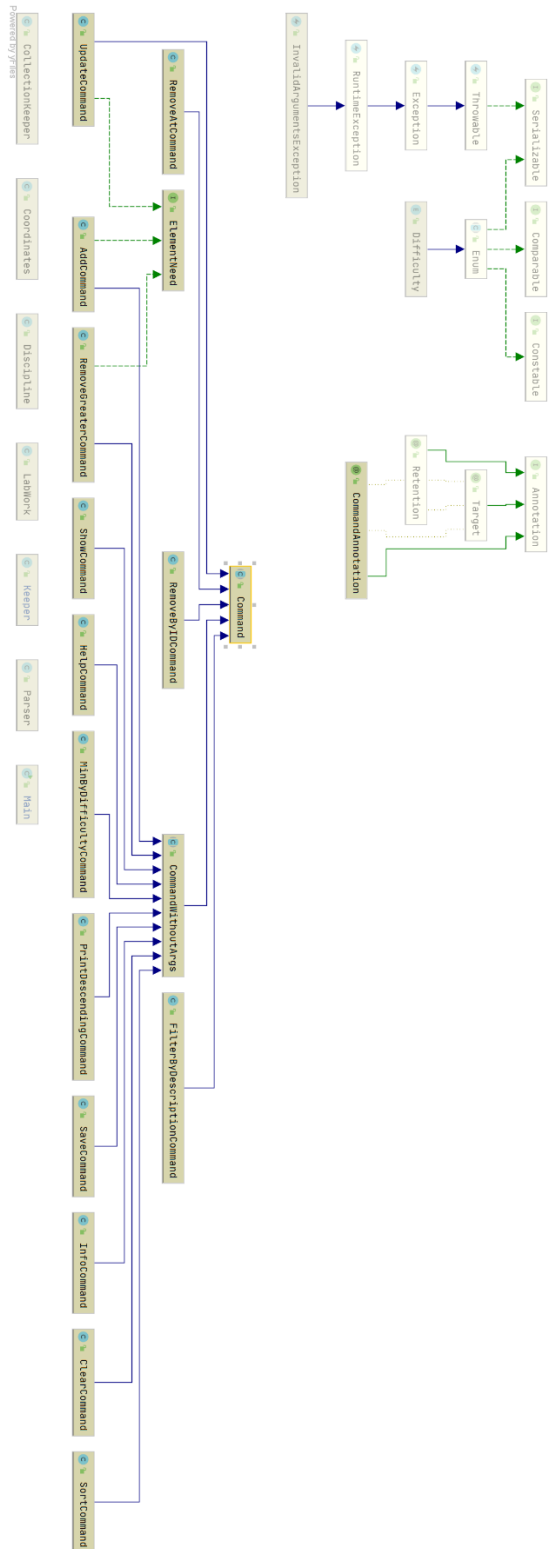
public class Discipline {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private long lectureHours;
```

```
private Integer practiceHours; //Поле может быть null
private int selfStudyHours;
}
public enum Difficulty {
    VERY_EASY,
    NORMAL,
    VERY_HARD,
    INSANE,
    HOPELESS;
}
```

## **Исходный код программы**

[https://github.com/senina-m/Programming\\_5](https://github.com/senina-m/Programming_5)

## **Диаграмма классов**



## Выводы по работе

Я познакомилась с коллекциями в java, с пакетами java.io и java.nio. Научилась читать значения из переменной среды. Попробовала использовать некоторые паттерны проектирования, читать и писать в файл и стандартный поток вывода.