

# Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5 по программированию.  
Вариант 314929

**Выполнил:**

Конаныхина Антонина Александровна

**Учебная группа:** Р3115

**Преподаватель:**

Пашнин Александр Денисович

## Текст задания:

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `StudyGroup`, 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.Stack`
- 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.BufferedReader`
- Запись данных в файл необходимо реализовать с помощью класса `java.io.FileWriter`
- 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)
- `add_if_max {element}` : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- `history` : вывести последние 7 команд (без их аргументов)
- `remove_all_by_students_count studentsCount` : удалить из коллекции все элементы, значение поля `studentsCount` которого эквивалентно заданному
- `filter_starts_with_name name` : вывести элементы, значение поля `name` которых начинается с заданной подстроки
- `filter_less_than_expelled_students expelledStudents` : вывести элементы, значение поля `expelledStudents` которых меньше заданного

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; 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 StudyGroup {
    private Integer id; //Поле не может быть null, Значение поля должно быть больше 0
    , Значение этого поля должно быть уникальным, Значение этого поля должно генерировать
    ся автоматически

    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null

    private java.util.Date creationDate; //Поле не может быть null, Значение этого по
    ля должно генерироваться автоматически

    private Integer studentsCount; //Значение поля должно быть больше 0, Поле может б
    ыть null

    private Integer expelledStudents; //Значение поля должно быть больше 0, Поле може
    т быть null

    private Long averageMark; //Значение поля должно быть больше 0, Поле может быть n
    ull

    private Semester semesterEnum; //Поле не может быть null
    private Person groupAdmin; //Поле не может быть null
}

public class Coordinates {
    private long x; //Максимальное значение поля: 752
    private int y;
}

public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private java.time.LocalDateTime birthday; //Поле может быть null
    private Long weight; //Поле не может быть null, Значение поля должно быть больше
    0

    private String passportID; //Поле может быть null
}

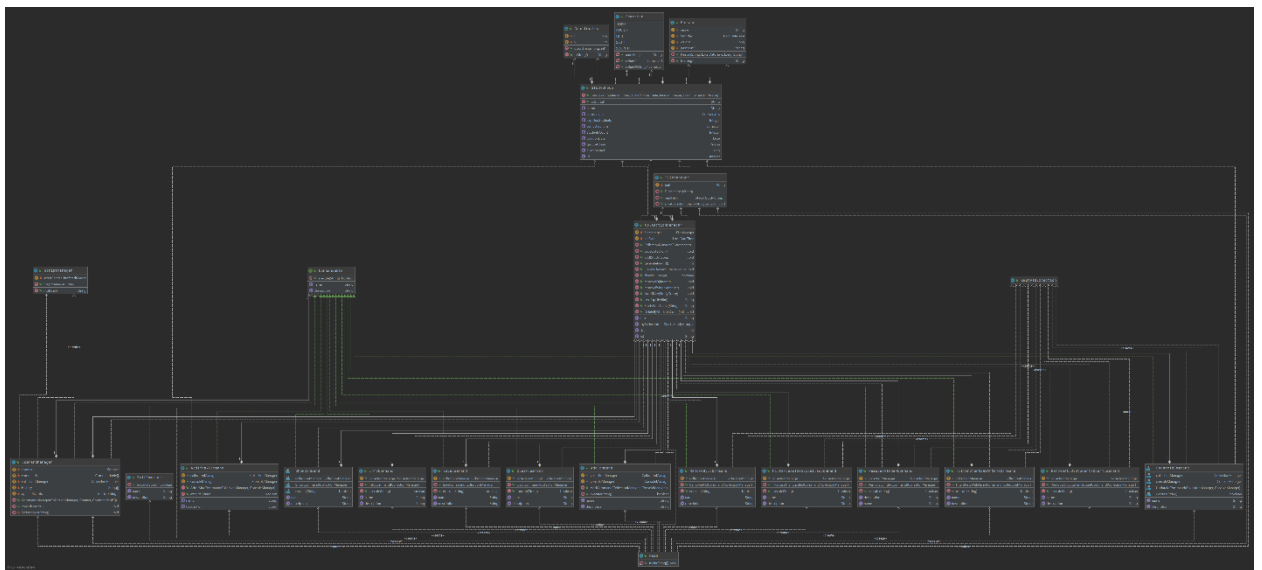
public enum Semester {
    THIRD,
```

```
    FOURTH,  
    FIFTH,  
    SIXTH,  
    SEVENTH;  
}
```

**Код:**

<https://github.com/tchn11/lab5>

**UML схема:**



**Вывод:**

В ходе лабораторной работы были разобраны использование и работа с коллекциями, работа с потоками ввода-вывода и консольными приложениями.