

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
“Национальный исследовательский университет ИТМО”

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине
‘ПРОГРАММИРОВАНИЕ’

Вариант: 308554

выполнил:

Студент группы Р3133

Нгуен Нгок Дык

Преподаватель

Исаев Александр Сергеевич



Санкт-Петербург, 2021

1 Задание

Лабораторная работа #5

Введите вариант: 308554

Внимание! У разных вариантов разный текст задания!

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса `Person`, 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.HashSet`.
- 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)
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `remove_lower {element}` : удалить из коллекции все элементы, меньшие, чем заданный
- `history` : вывести последние 14 команд (без их аргументов)
- `group_counting_by_id` : сгруппировать элементы коллекции по значению поля id, вывести количество элементов в каждой группе
- `count_less_than_birthday` : вывести количество элементов, значение поля birthday которых меньше заданного
- `print_field_ascending_height` : вывести значения поля height всех элементов в порядке возрастания

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 Person {
    private long id; //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 long height; //Поле может быть null, Значение поля должно быть больше 0
    private java.util.Date birthday; //Поле не может быть null
    private int weight; //Значение поля должно быть больше 0
    private Country nationality; //Поле не может быть null
    private Location location; //Поле не может быть null
}

public class Coordinates {
    private int x; //Значение поля должно быть больше -801
    private Double y; //Максимальное значение поля: 687, Поле не может быть null
}

public class Location {
    private Integer x; //Поле не может быть null
    private long y;
    private String name; //Поле может быть null
}

public enum Country {
    GERMANY,
    FRANCE,
    SPAIN,
    CHINA,
    ITALY;
}
```

2 Код и Диаграмм

Исходный код доступен по ссылке или QR-коду:

<https://github.com/ndwannafly/Programming-Lab-2nd-Semester/tree/main/LAB5>



<https://github.com/ndwannafly/Programming-Lab-2nd-Semester/blob/main/LAB5//diagram.png>



3 Вывод

В ходе этой лабораторной работы мы знаем о Collection Java Framework, а также как использовать Javados, работать с потоками, файлами, интерфейсами Comparable и Comparator. Кроме того, мы знаем, как применять шаблоны проектирования для соблюдения принципов SOLID.