

**Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»**

Факультет программной инженерии и компьютерной техники

Дисциплина «Программирование»

**Отчёт по лабораторной работе №5
Вариант №1700**

Выполнил:

Калиев Александр Дмитриевич,

Группа: Р3111

Преподаватель:

Мой Лучший Практик

Санкт-Петербург 2024 г.

Содержание

Задание	3
Исходный код	7
Заключение	7

Задание

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.TreeSet`
- ▷ 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.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)
- ▷ `add_if_min {element}` : добавить новый элемент в коллекцию, если его значение меньше, чем у наименьшего элемента этой коллекции
- ▷ `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- ▷ `history` : вывести последние 9 команд (без их аргументов)
- ▷ `count_greater_than_group_admin groupAdmin` : вывести количество элементов, значение поля `groupAdmin` которых больше заданного
- ▷ `filter_starts_with_name name` : вывести элементы, значение поля `name` которых начинается с заданной подстроки
- ▷ `print_field_ascending_semester_enum` : вывести значения поля `semesterEnum` 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 enumом, 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 long id; //Значение поля должно быть больше 0, Значение этого поля должно б
    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 э
    private long studentsCount; //Значение поля должно быть больше 0
    private Long averageMark; //Значение поля должно быть больше 0, Поле может быть null
    private FormOfEducation formOfEducation; //Поле может быть null
    private Semester semesterEnum; //Поле может быть null
    private Person groupAdmin; //Поле не может быть null
```

```

}

public class Coordinates {
    private Long x; //Поле не может быть null
    private long y; //Максимальное значение поля: 159
}

public class Person {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private int weight; //Значение поля должно быть больше 0
    private Color eyeColor; //Поле не может быть null
    private Color hairColor; //Поле не может быть null
    private Country nationality; //Поле не может быть null
}

public enum FormOfEducation {
    DISTANCE_EDUCATION,
    FULL_TIME_EDUCATION,
    EVENING_CLASSES;
}

public enum Semester {
    FIRST,
    SECOND,
    FOURTH,
    SIXTH,
    EIGHTH;
}

public enum Color {
    GREEN,
    RED,
    YELLOW,
    ORANGE,
    WHITE;
}

public enum Color {

```

```
    GREEN,  
    BLACK,  
    YELLOW,  
    BROWN;  
}  
  
public enum Country {  
    UNITED_KINGDOM,  
    VATICAN,  
    ITALY,  
    JAPAN;  
}
```

Исходный код

Исходный код, дерево директорий и диаграмма классов в хорошем разрешении доступны на [GitHub](#). В связи с быстрой утратой актуальности дублирование кода в документ целесообразным не вижу.

Заключение

В ходе выполнения этой работы я написал реализовал консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции было необходимо хранить объекты класса StudyGroup, описание которого было дано в варианте. Программа была реализована с помощью создания классов для объектов соответственно, были созданы абстрактные классы и интерфейсы, были добавлены исключения, анонимные классы. Объектно-ориентированный стиль облегчит понимание архитектуры кода для других разработчиков и для меня.