

Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по программированию №5

Вариант 5179

Выполнил

Пчелкин Илья Игоревич

P3106

Проверил

Вербовой А. А.

Санкт-Петербург 2025

ОГЛАВЛЕНИЕ

<i>Задание</i>	3
<i>UML – Диаграмма</i>	5
<i>Код программы.....</i>	6
<i>Результат работы программы</i>	7

Задание

Реализовать консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. В коллекции необходимо хранить объекты класса **MusicBand**, 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: **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.OutputStreamWriter**
- 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_max {element}** : добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
- **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
- **max_by_best_album** : вывести любой объект из коллекции, значение поля bestAlbum которого является максимальным
- **filter_starts_with_name name** : вывести элементы, значение поля name которых начинается с заданной подстроки
- **print_field_ascending_establishment_date** : вывести значения поля establishmentDate всех элементов в порядке возрастания

Формат ввода команд:

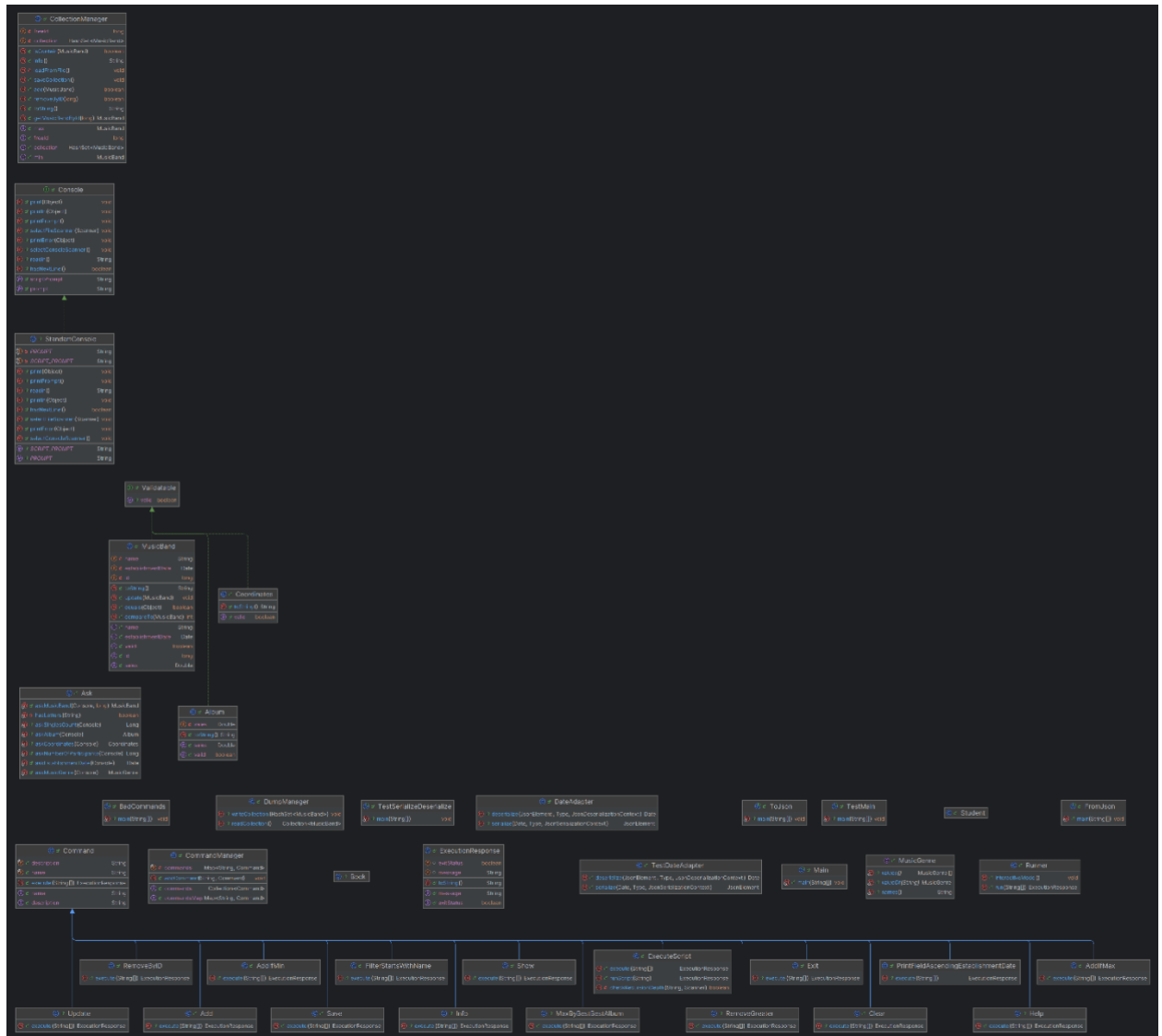
- Все аргументы команды, являющиеся стандартными типами данных (примитивные типы, классы-оболочки, String, классы для хранения дат), должны вводиться в той же строке, что и имя команды.

- 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'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 MusicBand {
    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.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно
    генерироваться автоматически
    private Long numberOfParticipants; //Поле может быть null, Значение поля должно быть больше 0
    private Long singlesCount; //Поле не может быть null, Значение поля должно быть больше 0
    private java.util.Date establishmentDate; //Поле может быть null
    private MusicGenre genre; //Поле не может быть null
    private Album bestAlbum; //Поле не может быть null
}
public class Coordinates {
    private Integer x; //Поле не может быть null
    private float y; //Максимальное значение поля: 751
}
public class Album {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Long tracks; //Поле не может быть null, Значение поля должно быть больше 0
    private long length; //Значение поля должно быть больше 0
    private Double sales; //Поле не может быть null, Значение поля должно быть больше 0
}
public enum MusicGenre {
    PROGRESSIVE_ROCK,
    HIP_HOP,
    POP;
}
```

UML – Диаграмма



Код программы

<https://github.com/ipka23/ITMO/tree/main/subj/PROG/Labs/Lab5.0/Lab5Main/lab5>