Санкт-Петербургский Национальный
Исследовательский университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Программирование

Лабораторная работа - 5

Вариант -556874

Преподаватель - Сорокин Р.Б.
Выполнил - Дробыш Д.А.
Группа - P3110

Санкт-Петербург
2021

Задание:

Внимание! У разных вариантов разный текст задания!

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.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 `xml`
- Чтение данных из файла необходимо реализовать с помощью класса `java.io.BufferedInputStream`
- 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, отсутсвие 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)
- `head` : вывести первый элемент коллекции
- `remove_greater {element}` : удалить из коллекции все элементы, превышающие заданный
- `history` : вывести последние 10 команд (без их аргументов)
- `filter_starts_with_name name` : вывести элементы, значение поля name которых начинается с заданной подстроки
- `filter_greater_than_genre genre` : вывести элементы, значение поля genre которых больше заданного
- `print_unique_number_of_participants` : вывести уникальные значения поля numberOfParticipants всех элементов в коллекции

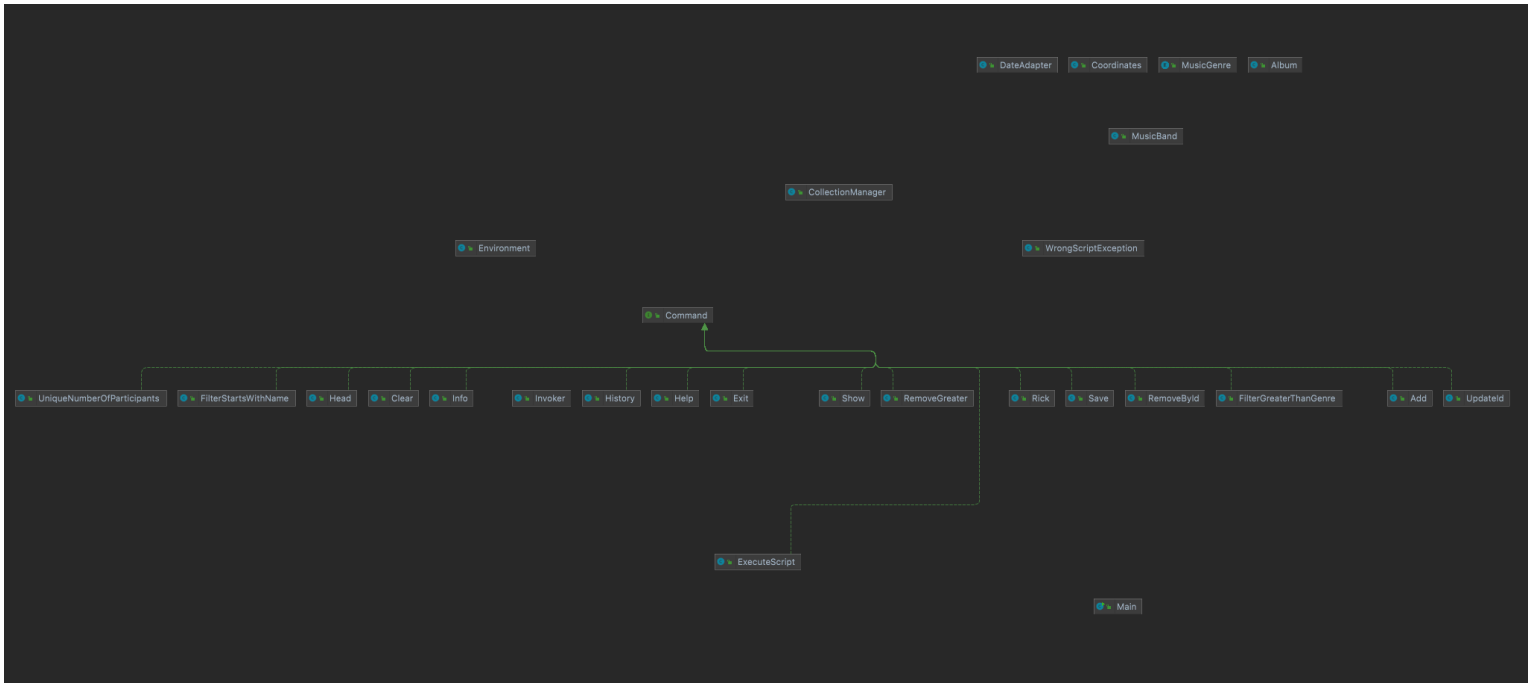
Формат ввода команд:

- 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; введённое 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 int id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генер
    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 int numberOfParticipants; //Значение поля должно быть больше 0
    private Integer albumsCount; //Поле не может быть null, Значение поля должно быть больше 0
    private MusicGenre genre; //Поле не может быть null
    private Album bestAlbum; //Поле может быть null
}
public class Coordinates {
    private Float x; //Максимальное значение поля: 955, Поле не может быть null
    private Long y; //Значение поля должно быть больше -293, Поле не может быть null
}
public class Album {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private long tracks; //Значение поля должно быть больше 0
    private Integer length; //Поле не может быть null, Значение поля должно быть больше 0
    private long sales; //Значение поля должно быть больше 0
}
public enum MusicGenre {
    ROCK,
    PSYCHEDELIC_CLOUD_RAP,
    JAZZ,
    POP,
    POST_PUNK;
}
```

UML:



Вывод: За время выполнения этой лабораторной работы, я освоил паттерн “Command”, поработал с парсерами xml, старался придерживаться принципов Solid, а также освоил Hash map и LinkedList.