

## 11. Регулярні вирази. Перевірка даних

**Мета:** Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.

### 1 ВИМОГИ

#### 1.1 Розробник

Інформація про розробника:

- Малюга Андрій Володимирович
- НТУ “ХПІ” 1.KIT102.8a
- Варіант12

#### 1.2 Загальне завдання

- Продемонструвати ефективне (оптимальне) використання регулярних виразів для перевірки коректності (валідації) даних, що вводяться, перед записом в domain-об'єкти відповідно до призначення кожного поля для заповнення розробленого контейнера:
- при зчитуванні даних з текстового файла в автоматичному режимі;
- при введенні даних користувачем в діалоговому режимі.
- Забороняється використання алгоритмів з Java Collections Framework.

#### 1.3 Задача

**Варіант 12.** Аудіотека. Сортування за назвою композиції, за виконавцем, за середнім значенням рейтингу.

## 2 ОПИС ПРОГРАМИ

### 2.1 Засоби ООП

У даній програмі присутні об'єктно-орієнтовані методи: Інкапсуляція – захист даних від неправомірного користування та поліморфізм.

### 2.2 Ієрархія та структура даних

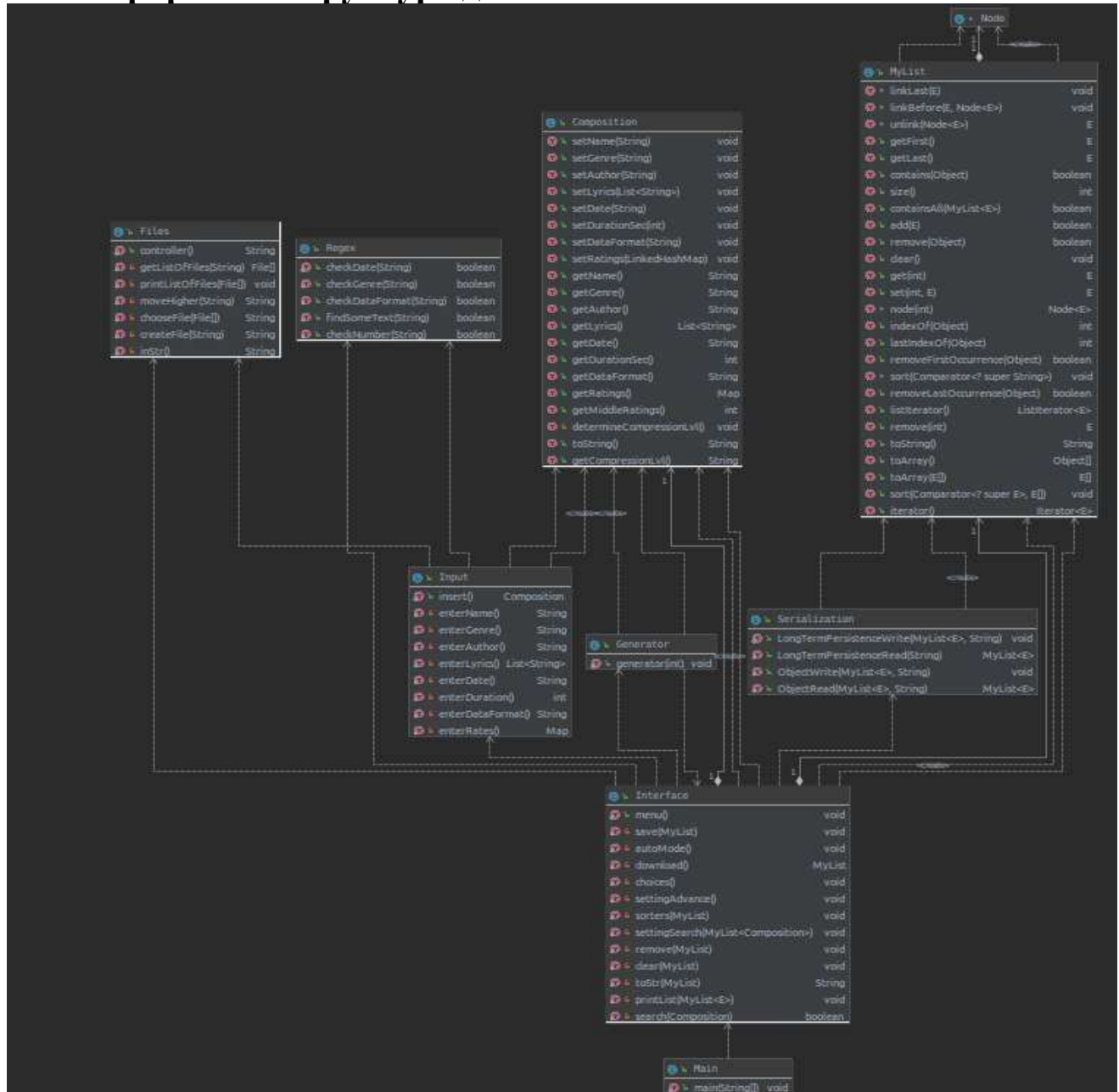


Рисунок 1 – Діаграма класів

## 2.3 Важливі фрагменти програми

```
public class Regex {

    public static boolean checkDate(String input) {
        return input.matches( regex: "^\\d{4}\\\\-(0?|[1-9]|1[012])\\\\-(0?|[1-9]|1[0-9]|3[01])$" );
    }

    public static boolean checkGenre(String input) { return input.matches( regex: "[A-Z, A-Я, a-z, a-я\\\\s\\\\-]+"); }

    public static boolean checkDateFormat(String input) {
        return input.matches( regex: "[mp3]*[ape]*[wav]*[aiff]*[flac]*[ogg]*[mp2]*");
    }

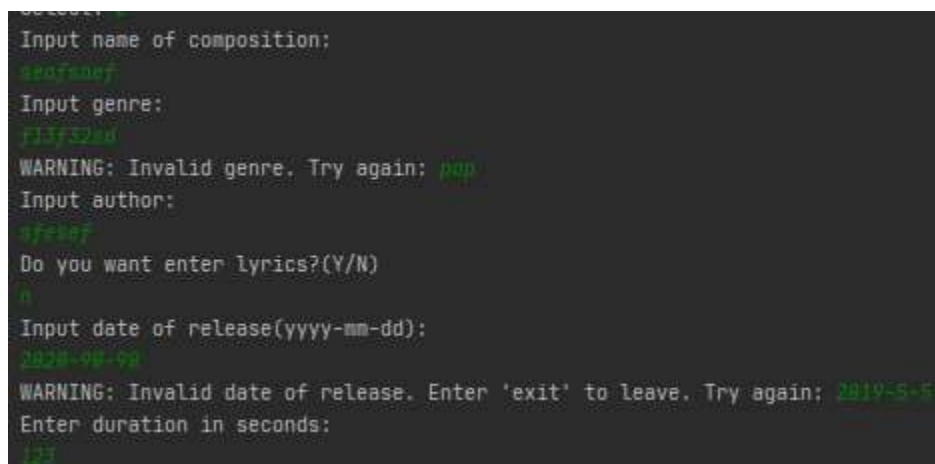
    public static boolean findSomeText(String input) {
        Pattern pattern = Pattern.compile("[H][o][p][p][v][\\\\s][N][e][=][\\\\s][Y][=][a][r]");
        Matcher matcher = pattern.matcher(input);
        if (!matcher.find()) {
            pattern = Pattern.compile("[U][e][=][\\\\s][Y][=][a][r]");
            matcher = pattern.matcher(input);
            return matcher.find();
        }
        else {
            return true;
        }
    }

    public static boolean checkNumber(String input) { return input.matches( regex: "^[1-9{1}][\\\\d{1,8}]*" ); }
}
```

Рисунок 2 — регулярні вирази

### 3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє створювати об'єкти – “композиції”, що заносяться у запис каталогу, тобто створюється масив об'єктів. Користувач може додавати об'єкти до масиву, видаляти елементи вибірково, а також очистити увесь масив одним викликом відповідної кнопки меню. Також присутня можливість серіалізувати/десеріалізувати об'єкти з файлу. Якщо програма починає свою роботу з параметром “-auto\ -a”, то перший крок програми зчитування з файлу.



```
Input name of composition:
geofseef
Input genre:
f13f32ad
WARNING: Invalid genre. Try again: pap
Input author:
sfeef
Do you want enter lyrics?(Y/N)
n
Input date of release(yyyy-mm-dd):
2020-00-00
WARNING: Invalid date of release. Enter 'exit' to leave. Try again: 2019-5-5
Enter duration in seconds:
123
```

Рисунок 3 — використання регулярних виразів

### ВИСНОВКИ

В даній лабораторній роботі було розроблено та реалізовано класи та методи відповідно прикладної галузі, реалізував управління списком domain-об'єктів, а методи сортування та обробка початкових параметрів.