

10. Обробка параметризованих контейнерів

Мета: Розширення функціональності параметризованих класів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Малюга Андрій Володимирович
- НТУ “ХП” 1.KIT102.8a
- Варіант 12

1.2 Загальне завдання

- Використовуючи програму рішення завдання лабораторної роботи №9:
- Розробити параметризовані методи (Generic Methods) для обробки колекцій об'єктів згідно прикладної задачі.
- Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.
- Автоматичний режим виконання програми задається параметром командного рядка -auto. Наприклад, java ClassName -auto.
- В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.
- 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

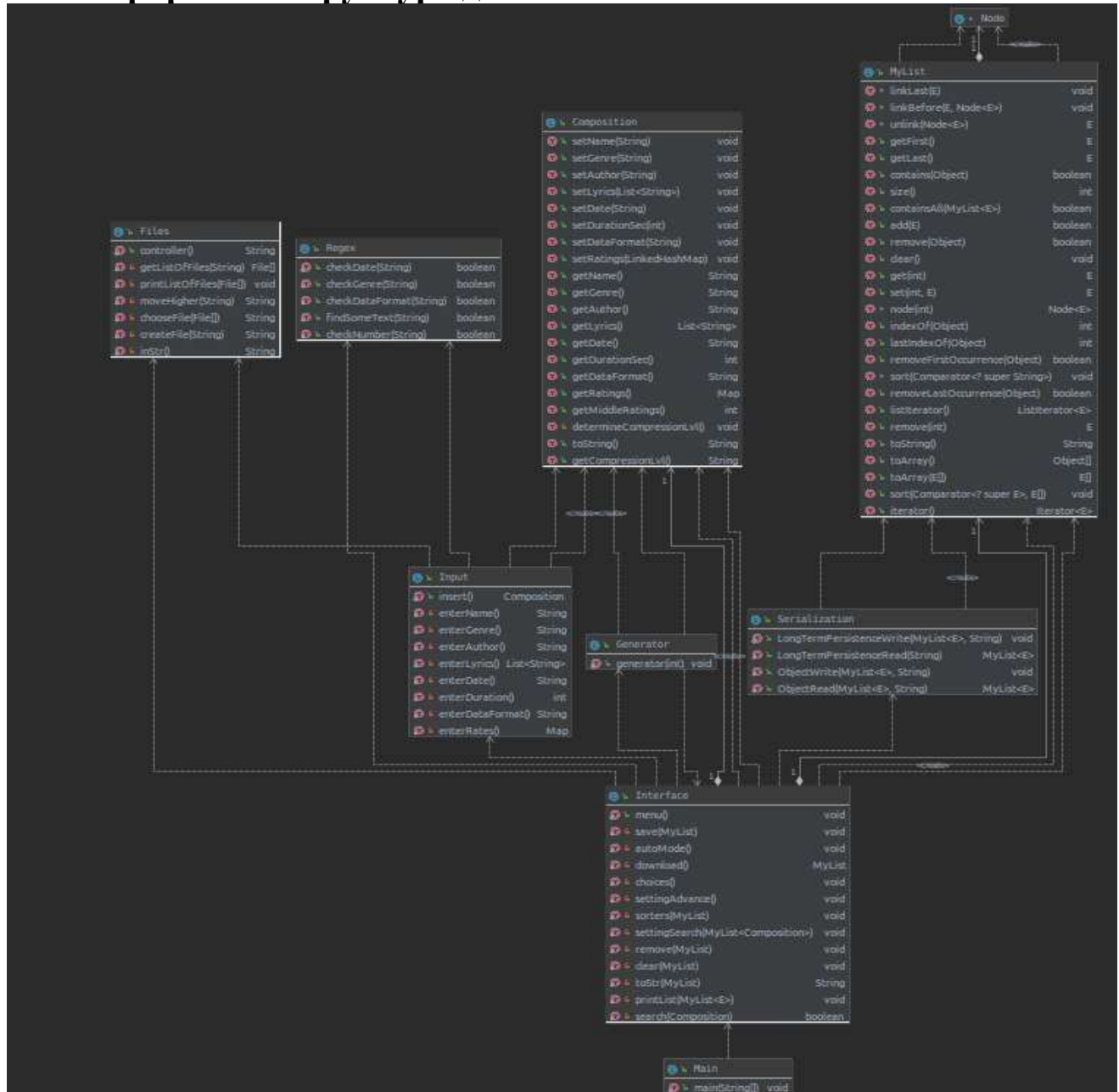


Рисунок 1 – Діаграма класів

2.3 Важливі фрагменти програми

```
public void sort(Comparator<? super E> c, E[] arr) {
    Object[] a = arr;
    Arrays.sort(a, (Comparator) c);
    ListIterator<E> i = this.listIterator();
    for (Object e : a) {
        i.next();
        i.set((E) e);
    }
}
```

Рисунок 2 — сортування в списку(саме використання знаходиться в класі Helper)

```
case 1:
    input.sort(new Comparator<Composition>() {
        @Override
        public int compare(Composition p1, Composition p2) {
            return p1.getName().compareTo(p2.getName());
        }
    }, input.toArray(new Composition[input.size()]));
    break;
case 2:
    input.sort(new Comparator<Composition>() {
        @Override
        public int compare(Composition p1, Composition p2) {
            if (p1.getMiddleRatings() != p2.getMiddleRatings()) {
                return p1.getMiddleRatings() - p2.getMiddleRatings();
            }
            return 0;
        }
    }, input.toArray(new Composition[input.size()]));
    break;
case 3:
    input.sort(new Comparator<Composition>() {
        @Override
        public int compare(Composition p1, Composition p2) {
            return p1.getAuthor().compareTo(p2.getAuthor());
        }
    }, input.toArray(new Composition[input.size()]));
    break;
```

Рисунок 3 — створення компараторів для сортування

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\а”, то перший крок програми зчитування з файлу.

```
List of settings:
0 - Exit
1 - Sort by name
2 - Sort by ratings
3 - Sort by author
4 - Show list
Select: |
```

Рисунок 4 — обираємо режим сортування

```
List of settings:
0 - Exit
1 - Sort by name
2 - Sort by ratings
3 - Sort by author
4 - Show list
Select: 2
List of settings:
0 - Exit
1 - Sort by name
2 - Sort by ratings
3 - Sort by author
4 - Show list
Select: 4|
```

Рисунок 5 — обираємо режим сортування(та виводимо на екран)

```

Lyrics:
Ratings:
FW42bqnub8i0FvFh: 7

Date release: 1965-11-27
Duration: 3
DataFormat: mp3
-----

-----

Name: Xdtke5KKJfqN40Xc
Author: Xdtke5KKJfqN40Xc
Genre: Xdtke5KKJfqN40Xc
Lyrics:
Ratings:
Xdtke5KKJfqN40Xc: 8

Date release: 1921-11-13
Duration: 4
DataFormat: mp3
-----

-----

Name: 2Mfrbl3BMvcPuDM8
Author: 2Mfrbl3BMvcPuDM8
Genre: 2Mfrbl3BMvcPuDM8
Lyrics:
Ratings:
2Mfrbl3BMvcPuDM8: 9

Date release: 2004-11-20

```

Рисунок 6 — Результат сортування

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.