



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №10

## **Колекції. Списки**

Виконали

студенти групи ІА-23:

Степанов Н. О.

Семашко О. В.

Ширяєв Д. Ю.

Перевірів:

Колеснік В.М

Київ 2022

## Варіант 3

Мета: Ознайомитись з javadoc для наступних інтерфейсів та класів:

- Collection
- List
- ArrayList
- LinkedList
- Iterator
- RandomAccess

Хід роботи:

### Main.java:

```
1 import java.util.Iterator;
2 import java.util.List;
3 import java.util.ArrayList;
4 import java.util.Collections;
5
6 public class Main {
7     public static void main(String[] args) {
8         // Creating actors-----
9
10        // actor bob
11        List<String> actorFilmsList = new ArrayList<>();
12        String[] actorFilms = {"Rock", "Mountain"};
13        Collections.addAll(actorFilmsList, actorFilms);
14        Actor bob = new Actor( name: "Bob", actorFilmsList);
15
16        // actor john
17        actorFilmsList = new ArrayList<>();
18        actorFilms = new String[]{"Rock", "Mountain", "Moon"};
19        Collections.addAll(actorFilmsList, actorFilms);
20        Actor john = new Actor( name: "John", actorFilmsList);
21
22        // actor david
23        actorFilmsList = new ArrayList<>();
24        actorFilms = new String[]{"David", "Hawaii"};
25        Collections.addAll(actorFilmsList, actorFilms);
26        Actor david = new Actor( name: "David", actorFilmsList);
27
28        // actor ryan
29        actorFilmsList = new ArrayList<>();
30        actorFilms = new String[]{"Drive"};
31        Collections.addAll(actorFilmsList, actorFilms);
32        Actor ryan = new Actor( name: "Ryan", actorFilmsList);
```

```

34 // actor vova
35 actorFilmsList = new ArrayList<>();
36 actorFilms = new String[]{};
37 Collections.addAll(actorFilmsList, actorFilms);
38 Actor vova = new Actor( name: "Vova", actorFilmsList);
39
40 //Creating films-----
41
42 // film rock
43 List<Actor> whoPlayedInFilm = new ArrayList<>();
44 whoPlayedInFilm.add(bob);
45 whoPlayedInFilm.add(john);
46 Film rock = new Film( title: "Rock", whoPlayedInFilm);
47
48 // film mountain
49 whoPlayedInFilm = new ArrayList<>();
50 whoPlayedInFilm.add(bob);
51 whoPlayedInFilm.add(john);
52 whoPlayedInFilm.add(david);
53 Film mountain = new Film( title: "Mountain", whoPlayedInFilm);
54
55 // film moon
56 whoPlayedInFilm = new ArrayList<>();
57 whoPlayedInFilm.add(john);
58 Film moon = new Film( title: "Moon", whoPlayedInFilm);
59
60 // film hawaii
61 whoPlayedInFilm = new ArrayList<>();
62 whoPlayedInFilm.add(david);
63 Film hawaii = new Film( title: "Hawaii", whoPlayedInFilm);
64
65 // film drive
66 whoPlayedInFilm = new ArrayList<>();
67 whoPlayedInFilm.add(ryan);
68 Film drive = new Film( title: "Drive", whoPlayedInFilm);
69
70 List<Actor> actorsList = new ArrayList<>();
71 actorsList.add(bob);
72 actorsList.add(john);
73 actorsList.add(david);
74 actorsList.add(ryan);
75 actorsList.add(vova);
76
77 List<Film> filmsList = new ArrayList<>();
78 filmsList.add(rock);
79 filmsList.add(mountain);
80 filmsList.add(moon);
81 filmsList.add(hawaii);
82 filmsList.add(drive);
83
84 // create an object of the Cinema class and add the created actorsList and filmsList to it
85 Cinema cinema = new Cinema(filmsList, actorsList);
86
87 // Task1
88 System.out.println("Task1");
89 System.out.println(Task1.getActorWhoNeverPlayed(cinema));
90
91 // Task2
92 System.out.println("Task2");
93 List<Actor> actors = Task2.getActorsListThisActorHasEverPlayedWith(bob, cinema);
94 for (Iterator i = actors.iterator(); i.hasNext();) {
95     Actor actor = (Actor) i.next();
96     System.out.print(actor.getActorName() + " ");
97 }
98 System.out.println();

```

## Task1.java:

```
1 import java.util.Iterator;
2
3 public class Task1 {
4     @ 1 usage
5     public static String getActorWhoNeverPlayed(Cinema cinema) {
6         Iterator<Actor> iterator = cinema.getCinemaActorsInfo().iterator();
7         while (iterator.hasNext()) {
8             Actor actor = iterator.next();
9             if (actor.filmsWhereActorPlayed().isEmpty()) {
10                 return actor.getActorName() + " has never played in a film";
11             }
12         }
13         return "There are not actors who never played in a film";
14     }
15 }
```

## Task2.java:

```
1 import java.util.*;
2
3 public class Task2 {
4     @ 1 usage
5     public static List<Actor> getActorsListThisActorHasEverPlayedWith(Actor actorToCheck, Cinema cinema) {
6         System.out.print("Actors who ever played with " + actorToCheck.getActorName() + ": ");
7         Set<Actor> actors = new LinkedHashSet<>();
8         for (Iterator i = cinema.getCinemaFilmsInfo().iterator(); i.hasNext();) {
9             Film film = (Film) i.next();
10             if (film.whoPlayedInFilm().contains(actorToCheck)) {
11                 actors.addAll(film.whoPlayedInFilm());
12                 actors.remove(actorToCheck);
13             }
14         }
15         List<Actor> actorsList = new ArrayList<>(actors);
16         return actorsList;
17     }
18 }
```

## Task3.java:

```
Main.java × Task1.java × Task2.java × Task3.java × Cinema.java × Actor.java × Film.java ×
1 1 usage
2 public class Task3 {
3     1 usage
4     public static String filmWithBiggestAmountOfActers(Cinema cinema) {
5         Film filmWithBiggestAmountOfActers = null;
6         int biggestNumberOfActersInFilms = 0;
7         for (Film film: cinema.getCinemaFilmsInfo()) {
8             if (film.whoPlayedInFilm().size() > biggestNumberOfActersInFilms) {
9                 biggestNumberOfActersInFilms = film.whoPlayedInFilm().size();
10                filmWithBiggestAmountOfActers = film;
11            }
12        }
13        return "The film with the biggest amount of acters: " +
14            filmWithBiggestAmountOfActers.getFilmTitle() +
15            ": " + biggestNumberOfActersInFilms;
16    }
17 }
```

## Класи Cinema, Actor, Film:

```
Main.java × Task1.java × Task2.java × Task3.java × Cinema.java × Actor.java × Film.java ×
1 import java.util.List;
2
3 5 usages
4 public class Cinema {
5     2 usages
6     private List<Film> filmsList;
7     2 usages
8     private List<Actor> actorsList;
9
10    1 usage
11    public Cinema(List<Film> filmsList, List<Actor> actorsList) {
12        this.filmsList = filmsList;
13        this.actorsList = actorsList;
14    }
15
16    2 usages
17    public List<Film> getCinemaFilmsInfo() { return this.filmsList; }
18
19    1 usage
20    public List<Actor> getCinemaActersInfo() { return this.actorsList; }
21 }
```

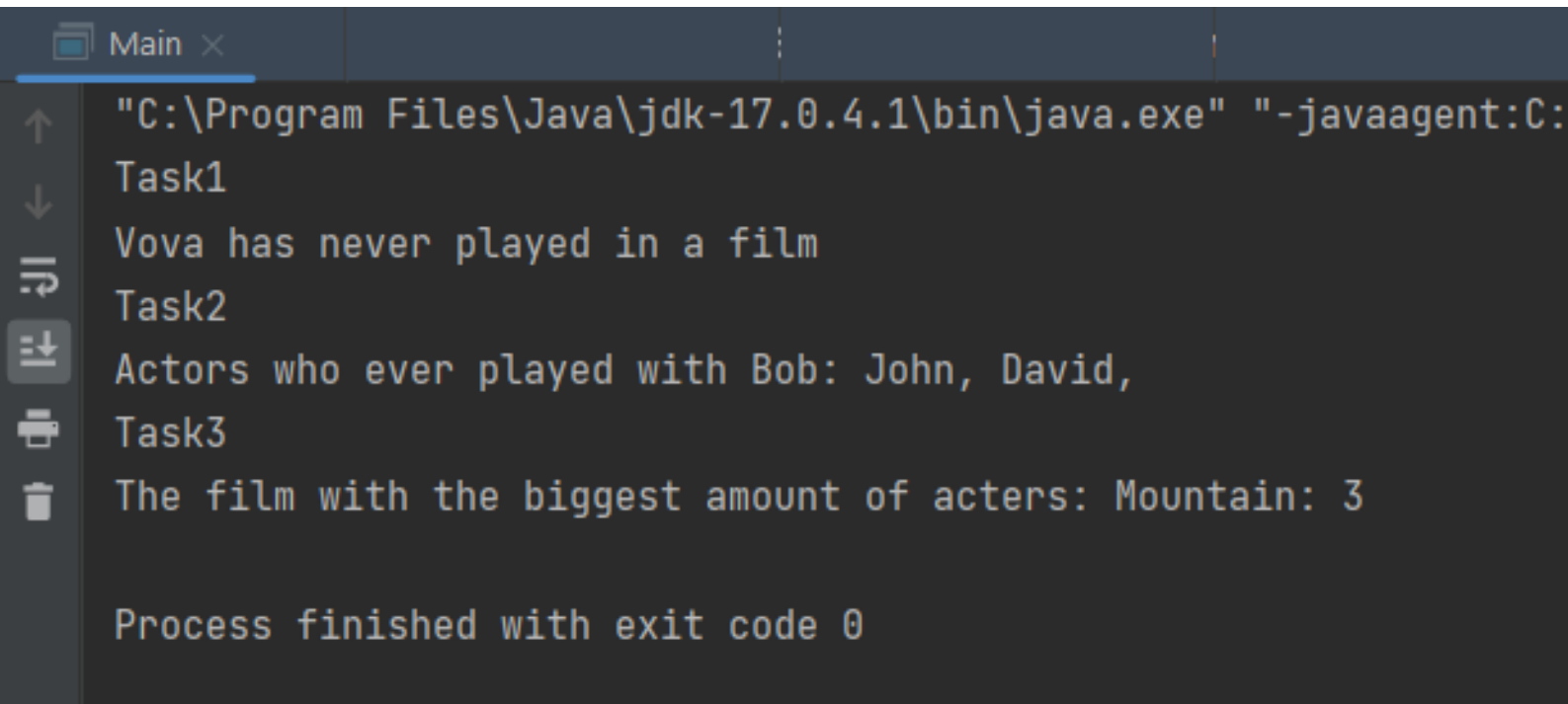
Main.java × Task1.java × Task2.java × Task3.java × Cinema.java × Actor.java × Film.java ×

```
1  import java.util.List;
2
3  27 usages
4  public class Actor {
5      2 usages
6      private String name;
7      2 usages
8      private List<String> filmList;
9      5 usages
10     public Actor(String name, List<String> filmList) {
11         this.name = name;
12         this.filmList = filmList;
13     }
14
15     3 usages
16     public String getActorName() { return this.name; }
17
18     1 usage
19     public List<String> filmsWhereActorPlayed() { return this.filmList; }
20 }
```

Main.java × Task1.java × Task2.java × Task3.java × Cinema.java × Actor.java × Film.java ×

```
1  import java.util.List;
2
3  18 usages
4  public class Film {
5      2 usages
6      private String title;
7      2 usages
8      private List<Actor> actorList;
9
10     5 usages
11     public Film(String title, List<Actor> actorList) {
12         this.title = title;
13         this.actorList = actorList;
14     }
15
16     1 usage
17     public String getFilmTitle() { return this.title; }
18
19     4 usages
20     public List<Actor> whoPlayedInFilm() { return this.actorList; }
21 }
```

## Консольний вивід:



The screenshot shows a Java IDE's console window. The title bar at the top says 'Main' with a close button. The console output is as follows:

```
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:  
Task1  
Vova has never played in a film  
Task2  
Actors who ever played with Bob: John, David,  
Task3  
The film with the biggest amount of acters: Mountain: 3  
  
Process finished with exit code 0
```

On the left side of the console, there is a vertical toolbar with icons for running, stepping through, and other debugging actions.

**Висновки:** На цій лабораторній роботі ми ознайомилися з Javadoc для таких інтерфейсів і класів, як: Collection List, ArrayList, LinkedList, Iterator, RandomAccess.