



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:
Ковальський К.В.

Перевірів:
Колеснік В.М.

Київ 2022

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:

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

2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1. Для цього:

- проаналізувати завдання;
- створити зазначенні класи;
- для створення списків слід використовувати класи та інтерфейси з Collection Framework (заборонено використовувати масиви);
- усі списки мають бути типізованими (наприклад, ArrayList<Student>, а не просто ArrayList);
- при реалізації задач «1)», «2)», «3)» слід застосувати наступні методи перегляду колекцій у відповідності до свого варіанту (табл. 1):
 - a) нетипізований ітератор;
 - b) типізований ітератор;
 - c) типізований цикл «for-each».

3. Відповісти на контрольні питання

3	<p>Класи: База даних про кіно (Список усіх фільмів, список усіх акторів) Фільм (Назва, Список акторів які знялись у фільмі) Актор (Ім'я, Список фільмів в яких зіграв актор)</p> <p>Задача: 1) Визначити, чи є актор, який не зіграв в жодному фільмі 2) Скласти список акторів, з якими коли-небудь в одному фільмі грав заданий актор 3) Знайти фільм з найбільшою кількістю акторів</p>
---	--

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        printResults();
    }
    1 usage
    static void printResults(){
        loadDatabase();
        task1();
        task2( actor: "Chris Evans");
        task3();
    }
}
```

```

static void task1(){
    List<Actor> allActors = Data.getAllActors();
    if (allActors.size()==0){
        throw new NullPointerException("Неможливо перевірити порожній список");
    }
    for (Iterator i = allActors.iterator(); i.hasNext();){
        Actor actor = (Actor) i.next();
        if (actor.getCareer().isEmpty()){
            System.out.println(actor.getFullName() + " не знімався в жодному фільмі");
        }
    }
}

1 usage
static void task2(String actor){
    System.out.println("Конери з "+actor+":");
    Data_Proc task = new Data_Proc();
    for(Actor i:task.listActors(actor)){
        System.out.println(i);
    }
}

1 usage

```

```

static void task3(){
    int movieSize = 0;
    List<String> result = new ArrayList<>();
    List<Movie> allFilms = Data.getAllFilms();
    for (Iterator<Movie> it = allFilms.iterator(); it.hasNext(); ) {
        Movie movie = it.next();
        if (movie.getCast().size() > movieSize){
            movieSize = movie.getCast().size();
            result.add(movie.getName());
        } else if (movie.getCast().size()==movieSize) {
            result.add(movie.getName());
        }
    }
    System.out.println("Найбільша кількість акторів: "+ result);
}

```

```

public static void loadDatabase(){
    Movie film1 = new Movie( name: "Spider man");
    Movie film2 = new Movie( name: "Uncharted");
    Movie film3 = new Movie( name: "Captain America");

    Actor actor1 = new Actor( name: "Robert", surname: "Downey Jr.");
    Actor actor2 = new Actor( name: "Chris", surname: "Evans");
    Actor actor3 = new Actor( name: "Mark", surname: "Ruffalo");
    Actor actor4 = new Actor( name: "Scarlett", surname: "Johansson");
    Actor actor5 = new Actor( name: "Gwyneth", surname: "Paltrow");
    Actor actor6 = new Actor( name: "Jeff", surname: "Bridges");
    Actor actor7 = new Actor( name: "Florence", surname: "Pugh");
    Actor actor8 = new Actor( name: "Rachel", surname: "Weisz");
    Actor actor9 = new Actor( name: "Angelina", surname: "Jolie");

    film1.addCast(Arrays.asList(actor1, actor2, actor3, actor4));
    film2.addCast(Arrays.asList(actor1, actor5, actor6));
    film3.addCast(Arrays.asList(actor4, actor7, actor8));
}

```

```

actor1.addAllMovies(Arrays.asList(film1, film2));
actor2.addAllMovies(Arrays.asList(film1));
actor3.addAllMovies(Arrays.asList(film1));
actor4.addAllMovies(Arrays.asList(film1, film3));
actor5.addAllMovies(Arrays.asList(film2));
actor6.addAllMovies(Arrays.asList(film2));
actor7.addAllMovies(Arrays.asList(film3));
actor8.addAllMovies(Arrays.asList(film3));

Data.addActors(Arrays.asList(actor1, actor2, actor3, actor4, actor5, actor6, actor7, actor8, actor9));
Data.addFilms(Arrays.asList(film1, film2, film3));

```

```

}
}

```

```

1  import java.util.*;
   37 usages
2  public class Actor {
   4 usages
3      private String name;
   4 usages
4      private String surname;
   2 usages
5      private List<Movie> career = new ArrayList<>();
6
   1 usage
7      public String getFullName() {
8          return name+" "+surname;
9      }
10
   1 usage
11     public String getName() {
12         return name;
13     }
14
   1 usage
15     public String getSurname() {
16         return surname;
17     }
18

```

```

9 usages
public Actor(String name, String surname) {
    this.name = name;
    this.surname=surname;
}

8 usages
public void addAllMovies(Collection<Movie> movie){
    this.career.addAll(movie);
}

1 usage
public List<Movie> getCareer() {
    return career;
}

@Override
public String toString() {
    return name + " " + surname ;
}
}

```

```

1  import java.util.ArrayList;
2  import java.util.Collection;
3  import java.util.List;
4
17 usages
5  public class Movie {
6      3 usages
        private String name;
7      2 usages
        private List<Actor> cast = new ArrayList<>();
8
9      3 usages
        public Movie(String name) {
10         this.name = name;
11     }
12
13     2 usages
        public String getName() {
14         return name;
15     }
16
17     4 usages
        public <Actor> List<Actor> getCast() {
18         return (List<Actor>) cast;
19     }

```

```

3 usages
20     public void addCast(Collection<Actor> actors){
21         this.cast.addAll(actors);
22     }
23
24     @Override
25     public String toString() {
26         return name;
27     }
28
29
30 }

```

```

1  import java.util.*;
2
6 usages
3  public class Data {
4      2 usages
      private static List<Actor> allActors = new ArrayList<>();
5      2 usages
      private static List<Movie> allFilms = new ArrayList<>();
6
7      1 usage
      public static void addActors(Collection<Actor> actors){
8          allActors.addAll(actors);
9      }
10     1 usage
      public static void addFilms(Collection<Movie> movies){
11         allFilms.addAll(movies);
12     }
13
14     2 usages
      public static List<Actor> getAllActors() {
15         return allActors;
16     }
17
18     2 usages
      public static List<Movie> getAllFilms() {
19         return allFilms;
20     }
21 }

```

```

1  import java.util.ArrayList;
2  import java.util.Collection;
3  import java.util.List;
4
5  2 usages
   public class Data_Proc {
6      1 usage
       @ public Collection<Actor> listActors(String actorFullName) {
7          String[] names = actorFullName.split( regex: "[., ]");
8          Actor tempActor = null;
9
10         List<Actor> actors = Data.getAllActors();
11         List<Movie> movies = Data.getAllFilms();
12         ArrayList<Actor> actorsList = new ArrayList<>();
13
14         for (Actor actor : actors) {
15             if (actor.getName().equals(names[0]) &
16                 actor.getSurname().equals(names[names.length - 1])) {
17                 tempActor = actor;
18                 break;
19             }
20         }
21

```

```

22         for (Movie movie : movies) {
23             List<Actor> cast = movie.getCast();
24
25             if (cast.contains(tempActor))
26                 actorsList.addAll(cast);
27
28             actorsList.remove(tempActor);
29         }
30
31         actorsList = removeDuplicates(actorsList);
32
33         return actorsList;
34     }
35
36     1 usage
       @ public ArrayList<Actor> removeDuplicates(Collection<Actor> actors) {
37         ArrayList<Actor> newActorsList = new ArrayList<>();
38
39         for (Actor actor : actors)
40             if (!newActorsList.contains(actor))
41                 newActorsList.add(actor);
42
43         return newActorsList;
44     }
45 }

```

```
C:\Users\kovalskii\.jdk\openjdk-19.0.1\bin\java.exe "-  
Angelina Jolie не знімався в жодному фільмі  
Колеги з Chris Evans:  
Robert Downey Jr.  
Mark Ruffalo  
Scarlett Johansson  
Найбільша кількість акторів: [Spider man]
```

Висновок: Виконуючу лабораторну ми навчилися використовувати базові інтерфейси та класи, такі як, Collection, List, ArrayList, Iterator.