



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

Лабораторна робота №10
із дисципліни «Основи програмування»
Тема: «Колекції. Списки»

Виконав:
Студенти групи ІА-24
Сіденко Д.Д.
Філімонов Є.А.
Іскандаров Е. Е. огли
Яблонський Д.Б.

Перевірив:
Колеснік Валерій Миколайович

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	<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:</p> <ol style="list-style-type: none">1) Визначити, чи є актор, який не зіграв в жодному фільмі2) Скласти список акторів, з якими коли-небудь в одному фільмі грав заданий актор3) Знайти фільм з найбільшою кількістю акторів
---	--

Angelina Jolie did not act in any film

Colleagues of Chris Evans

Robert Downey Jr.

Mark Ruffalo

Scarlett Johansson

The largest number of actors: Avengers

```

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Database();
        Task1();
        System.out.println();
        Task2("Chris Evans");
        System.out.println();
        Task3();
    }

    static void Task1(){
        List<Actor> allActors = DataBase.getAllActors();
        for (Iterator i = allActors.iterator(); i.hasNext();){
            Actor actor = (Actor) i.next();
            if (actor.getCareer().isEmpty()){
                System.out.println(actor.getFullName() + " did not act in any film");
            }
        }
    }

    static void Task2(String actor){
        System.out.println("Colleagues of "+actor);
        DataProcessor task = new DataProcessor();
        for(Actor i:task.listActors(actor)){
            System.out.println(i);
        }
    }

    static void Task3(){
        int movieSize = 0;
        List<Movie> allFilms = DataBase.getAllFilms();
        for (Iterator<Movie> it = allFilms.iterator(); it.hasNext(); ) {
            Movie movie = it.next();
            if (movie.getCast().size() > movieSize){
                movieSize = movie.getCast().size();
                System.out.println("The largest number of actors: "+movie.getName());
            }
        }
    }

    public static void Database(){ //load info to data base
        Movie film1 = new Movie("Avengers");
        Movie film2 = new Movie("Iron Man");
        Movie film3 = new Movie("Black Widow");

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

```

```

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

    }}

```

```

import java.util.*;

public class Actor {
    String name;
    String surname;
    List<Movie> career = new ArrayList<>();

    public String getFullName() {
        return name+" "+surname;
    }

    public String getName() {
        return name;
    }

    public String getSurname() {
        return surname;
    }

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

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

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

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

```

```

import java.util.*

public class Movie {
    String name;
    List<Actor> cast = new ArrayList<>();

    public Movie(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public <Actor> List<Actor> getCast() {
        return (List<Actor>) cast;
    }
}

```

```

    public void addCast(Collection<Actor> actors){
        this.cast.addAll(actors);
    }

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

```

```

import java.util.*;

public class DataBase {
    static List<Actor> allActors = new ArrayList<>();
    static List<Movie> allFilms = new ArrayList<>();

    public static void addActors(Collection<Actor> actors){
        allActors.addAll(actors);
    }
    public static void addFilms(Collection<Movie> movies){
        allFilms.addAll(movies);
    }

    public static List<Actor> getAllActors() {
        return allActors;
    }

    public static List<Movie> getAllFilms() {
        return allFilms;
    }
}

```

```

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;

public class DataProcessor {
    public Collection<Actor> listActors(String actorFullName) {
        String[] names = actorFullName.split("[., ]");
        Actor tempActor = null;

        List<Actor> actors = DataBase.getAllActors();
        List<Movie> movies = DataBase.getAllFilms();
        ArrayList<Actor> actorsList = new ArrayList<>();

        for (Actor actor : actors) {
            if (actor.getName().equals(names[0]) &
                actor.getSurname().equals(names[names.length - 1])) {
                tempActor = actor;
                break;
            }
        }

        for (Movie movie : movies) {
            List<Actor> cast = movie.cast;

            if (cast.contains(tempActor))
                actorsList.addAll(cast);
        }
    }
}

```

```

        actorsList.remove(tempActor);
    }

    actorsList = removeDuplicates(actorsList);

    return actorsList;
}

public ArrayList<Actor> removeDuplicates(Collection<Actor> actors) {
    ArrayList<Actor> newActorsList = new ArrayList<>();

    for (Actor actor : actors)
        if (!newActorsList.contains(actor))
            newActorsList.add(actor);

    return newActorsList;
}
}

```

3. Відповісти на контрольні питання

Висновок: завдяки цій лабораторній роботі ми навчилися використовувати базовий інтерфейс Collection, нащадок-інтерфейс List та його реалізацію клас ArrayList. Також ми попрацювали з інтерфейсом Iterator та циклом for-each.