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 №11

із дисципліни «*Основи програмування*»

Тема: «Колекції.Множина TreeSet»

Виконав:
Студент групи ІА-24
Красношапка Р.О.
Бакалець А.І.
Орловська А.В.

Перевірив:
Колеснік В.М.

Київ-2022

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:
   - Set
   - TreeSet
   - Comparable
   - Comparator
   - SortedSet
   - NavigableSet

2. Виконати завдання попередньої лабораторної роботи, замінивши списки List (ArrayList та LinkedList) на множини Set (TreeSet). При цьому необхідно щонайменше один раз використати Comparable та щонайменше один раз – Comparator.
3. Відповісти на контрольні питання.

Код програми:

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        printResult();
    }

    public static void printResult() {
        Film film1 = new Film("Deadpool");
        Film film2 = new Film("Focus");
        Film film3 = new Film("The Pursuit of Happyness");
        Actor actor1 = new Actor("Ryan Reynolds");
        Actor actor2 = new Actor("Will Smith");
        Actor actor3 = new Actor("Syre Smith");
        Actor actor4 = new Actor("Brad Pitt");

        Database.addFilm(film1);
        Database.addFilm(film2);
        Database.addFilm(film3);
        Database.addActor(actor1);
        Database.addActor(actor2);
        Database.addActor(actor3);
        Database.addActor(actor4);

        film1.addActor(actor1);
        film2.addActor(actor2);
        film3.addActor(actor2);
        film3.addActor(actor3);
        actor1.addFilm(film1);
        actor2.addFilm(film2);
        actor2.addFilm(film3);
        actor3.addFilm(film3);

        System.out.println(getCoActors(actor2));
        System.out.println("The largest number of actors in the film - " + mostActors());
        notFilmed();
    }

    private static String mostActors() {
        int countActor = 0;
        String mostActored = "";
        Set<Film> allfilms = Database.getFilms();
        for (Film film : allfilms) {
            if (countActor < film.getActors().size()) {
                countActor = film.getActors().size();
                mostActored = film.getName();
            } else if (countActor == film.getActors().size() && film.getActors().size() != 0) {
                mostActored = film.getName() + " and " + mostActored;
            }
        }
        return mostActored;
    }
```

```
    private static void notFilmed() {
        Set<Actor> allActors = Database.getActors();
        for (Actor actor : allActors) {
            if (actor.getFilms().isEmpty()) {
                System.out.println(actor.getName() + " - didn't act in any film");
            }
        }
    }

    public static String getCoActors(Actor actor) {
        Set<Actor> coActors = new TreeSet<>(new ActorComparator());
        Set<Film> films = Database.getFilms();

        Iterator<Film> filmIter = films.iterator();
        while (filmIter.hasNext()) {
            Film film = filmIter.next();
            if (film.getActors().contains(actor)) {
                Set<Actor> actors = film.getActors();
                Iterator<Actor> actorIter = actors.iterator();
                while (actorIter.hasNext()) {
                    Actor coActor = actorIter.next();
                    if (!coActor.equals(actor) && !coActors.contains(coActor)) {
                        coActors.add(coActor);
                    }
                }
            }
        }
        StringBuilder sb = new StringBuilder();
        sb.append("Co-actors of ").append(actor.getName()).append(": ");

        Iterator<Actor> coActorIter = coActors.iterator();
        while (coActorIter.hasNext()) {
            Actor coActor = coActorIter.next();
            sb.append(coActor.getName()).append(", ");
        }
        if (coActors.size() > 0) {
            sb.delete(sb.length() - 2, sb.length());
        }
        return sb.toString();
    }
}
import java.util.Set;
import java.util.TreeSet;

class Film implements Comparable<Film> {
    private String name;
    private Set<Actor> actors;

    public Film(String name) {
        this.name = name;
        this.actors = new TreeSet<>();
    }

    public String getName() {
        return name;
    }

    public Set<Actor> getActors() {
        return actors;
    }

    public void addActor(Actor actor) {
        actors.add(actor);
    }

    @Override
    public int compareTo(Film other) {
        return this.getName().compareTo(other.getName());

    }
}
import java.util.Set;
import java.util.TreeSet;
```

```
class Database {
    private static final Set<Film> films = new TreeSet<>();
    private static final Set<Actor> actors = new TreeSet<>();

    public static void addFilm(Film film) {
        films.add(film);
    }

    public static void addActor(Actor actor) {
        actors.add(actor);
    }

    public static Set<Film> getFilms() {
        return films;
    }

    public static Set<Actor> getActors() {
        return actors;
    }
}
import java.util.Comparator;
import java.util.Set;
import java.util.TreeSet;

class Actor implements Comparable<Actor> {
    private final String name;
    private final Set<Film> films;

    public Actor(String name) {
        this.name = name;
        this.films = new TreeSet<>();
    }

    public String getName() {
        return name;
    }

    public Set<Film> getFilms() {
        return films;
    }

    public void addFilm(Film film) {
        films.add(film);
    }

    @Override
    public int compareTo(Actor other) {
        return this.getName().compareTo(other.getName());
    }
}

class ActorComparator implements Comparator<Actor> {
    @Override
    public int compare(Actor a1, Actor a2) {
        return a1.getName().compareTo(a2.getName());
    }
}
```

Висновок: під час виконання лабораторної роботи, ознайомилися з javadoc для інтерфейсів та класів: Set, TreeSet, Comparable, Comparator, SortedSet, NavigableSet. Виконали завдання попередньої лабораторної роботи, замінивши списки List (ArrayList та LinkedList) на множини Set (TreeSet). При цьому використали Comparable та Comparator.Також відповіли на контрольні питання.