



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 №1

Колекції. Множина TreeSet

Виконали

студенти групи ІА-23:

Семашко Олександр,

Ширяєв Даніїл,

Степанов Нікіта.

Перевірив:

Колеснік В. М.

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.

```
public class Main {
    public static void main(String[] args) {
        FilmsDataBase base = new FilmsDataBase();

        Actor bob = new Actor("Bob", new String[]{"Mountain", "Cave"});
        base.appendWithActor(bob);
        Actor dave = new Actor("Dave", new String[]{"Mountain", "Desert"});
        base.appendWithActor(dave);
        Actor nik = new Actor("Nik", new String[]{"Mountain", "Desert", "Cave",
"Rose"});
        base.appendWithActor(nik);
        Actor actorWhoNeverPlayed = new Actor("ActorWhoNeverPlayed", new
String[]{});
        base.appendWithActor(actorWhoNeverPlayed);

        Film mountain = new Film("Mountain", new String[]{"Bob", "Dave",
"Nik"});
        base.appendWithFilm(mountain);
        Film cave = new Film("Cave", new String[]{"Bob", "Nik"});
        base.appendWithFilm(cave);
        Film desert = new Film("Desert", new String[]{"Dave", "Nik"});
        base.appendWithFilm(desert);
        Film rose = new Film("Rose", new String[]{"Nik"});
        base.appendWithFilm(rose);

        System.out.println(Task1.getActorWhoNeverPlayed(base));
        System.out.println(Task2.actorsGivenActorHasEverPlayedWith(bob, base));
        System.out.println(Task3.filmWithBiggestActorsNumber(base));
    }
}

import java.util.TreeSet;
import java.util.Set;

public class Actor implements Comparable<Actor>{
    final private String name;
    final private Set<String> films;
    Actor (String name, String[] films) {
        this.name = name;
        this.films = new TreeSet<>(Set.of(films));
    }

    @Override
    public int compareTo(Actor actor) {
        return name.compareTo(actor.toString());
    }
}
```

```

@Override
public String toString() {
    return name;
}
public Set<String> filmsWithActor () {
    return films;
}
}

import java.util.Set;
import java.util.TreeSet;

public class Film{
    final private String name;
    final private Set<String> actors;

    public Film(String name, String[] actors) {
        this.name = name;
        this.actors = new TreeSet<>(Set.of(actors));
    }
    public String toString () {
        return name;
    }
    public Set<String> getActors () {
        return actors;
    }
}

import java.util.Comparator;
import java.util.Set;
import java.util.TreeSet;

public class FilmsDataBase {
    final private Set<Actor> actorsBase;
    final private Set<Film> filmsBase;
    FilmsDataBase () {
        actorsBase = new TreeSet<>();
        filmsBase = new TreeSet<>(new FilmComparator());
    }
    public void appendWithFilm (Film film) {
        filmsBase.add(film);
    }
    public void appendWithActor (Actor actor) {
        actorsBase.add(actor);
    }
    public Set<Film> getFilmsBase () {
        return filmsBase;
    }
    public Set<Actor> getActorsBase () {
        return actorsBase;
    }
}

class FilmComparator implements Comparator<Film> {
    @Override
    public int compare(Film o1, Film o2) {
        return o1.toString().compareTo(o2.toString());
    }
}

import java.util.Comparator;
import java.util.Set;
import java.util.TreeSet;

```

```

public class FilmsDataBase {
    final private Set<Actor> actorsBase;
    final private Set<Film> filmsBase;
    FilmsDataBase () {
        actorsBase = new TreeSet<>();
        filmsBase = new TreeSet<>(new FilmComparator());
    }
    public void appendWithFilm (Film film) {
        filmsBase.add(film);
    }
    public void appendWithActor (Actor actor) {
        actorsBase.add(actor);
    }
    public Set<Film> getFilmsBase () {
        return filmsBase;
    }
    public Set<Actor> getActorsBase () {
        return actorsBase;
    }
}

class FilmComparator implements Comparator<Film> {
    @Override
    public int compare(Film o1, Film o2) {
        return o1.toString().compareTo(o2.toString());
    }
}

public class Task1 {
    static String getActorWhoNeverPlayed (FilmsDataBase base) {
        for (Actor actor: base.getActorsBase()) {
            if (actor.filmsWithActor().isEmpty()) {return actor + " never played
in film";}
        }
        return "Every actor played in film";
    }
}

import java.util.Set;
import java.util.TreeSet;

public class Task2 {
    static String actorsGivenActorHasEverPlayedWith (Actor givenActor,
FilmsDataBase base) {
        Set<String> actors = new TreeSet<>();
        for (Film film: base.getFilmsBase()) {
            for (String actor: film.getActors()) {
                if (actor.equals(givenActor.toString())) {
                    actors.addAll(film.getActors());
                }
            }
        }
        actors.remove(givenActor.toString());
        StringBuilder builder = new StringBuilder(String.format("Actors who ever
played with %s: ", givenActor));
        for (String actor: actors) {
            builder.append(actor);
            builder.append(", ");
        }
        return builder.toString();
    }
}

```

```

public class Task3 {
    static String filmWithBiggestActorsNumber (FilmsDataBase base) {
        int biggestActorsNumber = 0;
        StringBuilder output = new StringBuilder();
        for (Film film: base.getFilmsBase()) {
            if (film.getActors().size() > biggestActorsNumber) {
                biggestActorsNumber = film.getActors().size();
                output.replace(0, output.capacity(), film.toString());
            }
        }
        return output.insert(0, "Film with biggest actors number: ").toString();
    }
}

```

Результат роботи програми:

```

ActorWhoNeverPlayed never played in film
Actors who ever played with Bob: Dave, Nik,
Film with biggest actors number: Mountain

```

Висновок: на цій лабораторній роботі ми вивчили та навчилися використовувати інтерфейс Set та його нащадки, застосували на практиці інтерфейси Comparable та Comparator, покращили навички кодування на java.