



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
Пожар Д. Ю.
Хохол М. В.
Тюска А. Ю.

Перевів:
Колеснік В. М.

Київ 2023

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.

Class Main

```
import java.util.*;
import java.util.Arrays;

public class Main implements Comparable<Faculty> {

    public static final String ANSI_RESET = "\u001B[0m";
    public static final String ANSI_RED = "\u001B[31m";

    public static void main(String[] args) {
        Institute inst = new Institute();

        Set<Student> fiot = new TreeSet<>();
        Set<Student> fsp = new TreeSet<>();
        Set<Student> fbmi = new TreeSet<>();

        Student student1 = new Student("Andrii", "Tiuska", 26, 80.3);
        Student student2 = new Student("Maxim", "Khokhol", 5, 98.7);
        Student student3 = new Student("David", "Pozhar", 25, 96.5);
        Student student4 = new Student("Yevhenii", "Voroniuk", 12, 95);
        Student student5 = new Student("Vadim", "Voloshyn", 34, 100);
        Student student6 = new Student("Serhii", "Kalyna", 19, 87);

        fiot.add(student1);
        fiot.add(student2);
        fiot.add(student3);
        fsp.add(student4);
        fsp.add(student5);
        fbmi.add(student6);

        Faculty Fiot = new Faculty("FIOT");
        Faculty Fsp = new Faculty("FSP");
        Faculty Fbmi = new Faculty("FBMI");
        Fiot.countOfStudents = fiot.size();
        Fsp.countOfStudents = fsp.size();
        Fbmi.countOfStudents = fbmi.size();

        TreeSet<Faculty> faculties = new TreeSet<>();

        faculties.add(Fiot);
        faculties.add(Fsp);
```

```

        faculties.add(Fbmi);

        inst.addStudents(Arrays.asList(student1, student2, student3, student4,
student5, student6));

        System.out.print("Students with an average mark of 95-100 ");
        System.out.println(ANSI_RED + "(Sorting from 100 to 95) *Interface
Comparable*" + ANSI_RESET);
        task3(inst);

        System.out.println("Total amount of all students in institute");
        System.out.println(task1(inst));

        System.out.print("the largest count of students in ");
        System.out.println(ANSI_RED + faculties.last() + " *Interface Comparator*"
+ ANSI_RESET);

    }

    public static void task3(Institute institute) {
        for (Student x : institute.students) {
            if (x.averageMark >= 95) {
                System.out.print(x.name + " " + x.surname);
                System.out.println(" ");
            }
        }
    }

    public static int task1(Institute institute) {
        int count = 0;
        Iterator<Student> iterator = institute.students.iterator();

        while (iterator.hasNext()) {
            count++;
            iterator.next();
        }

        return count;
    }

    @Override
    public int compareTo(Faculty o) {
        return 0;
    }
}

```

Class Student

```
public class Student implements Comparable<Student> {
    String name;
    String surname;
    int number;
    double averageMark;
    public Student(String name,String surname,int number,double averageMark){
        this.name = name;
        this.surname = surname;
        this.number = number;
        this.averageMark = averageMark;
    }

    @Override
    public int compareTo(Student anotherStudent) {
        if(this.averageMark == anotherStudent.averageMark){
            return 0;
        } else if(this.averageMark > anotherStudent.averageMark){
            return -1;
        } else{
            return 1;
        }
    }
}
```

Class Faculty

```
import java.util.*;

public class Faculty extends Institute implements Comparator<Faculty>{
    int countOfStudents;
    String FacultyName;

    public Faculty(String facultyName) {
        FacultyName = facultyName;
    }

    @Override
    public int compare(Faculty o1, Faculty o2) {
        if(o1.countOfStudents == o2.countOfStudents){
            return 0;
        } else if(o1.countOfStudents > o2.countOfStudents){
            return 1;
        } else {
            return -1;
        }
    }

    Set<Student> students = new TreeSet<>();

    public void addStudents(Collection<Student> studentCollection){
        this.students.addAll(studentCollection);
    }

    @Override
    public String toString() {
        return "Faculty{" +
            "countOfStudents=" + countOfStudents +
            ", FacultyName='" + FacultyName + '\'' +
            '}';
    }
}
```

Class Institute

```
import java.util.*;

public class Institute implements Comparable<Institute> {

    Set<Student> students = new TreeSet<>();

    public void addStudents(Collection<Student> studentCollection) {

        this.students.addAll(studentCollection);
    }

    @Override
    public int compareTo(Institute o) {
        return 0;
    }
}
```

Результат роботи програми:

```
Students with an average mark of 95-100 (Sorting from 100 to 95) *Interface Comparable*
Vadim Voloshyn
Maxim Khokhol
David Pozhar
Yevhenii Voroniuk
Total amount of all students in institute
6
the largest count of students in Faculty{countOfStudents=3, FacultyName='FIOT'} *Interface Comparator*
```

Висновок

Виконавши лабораторну роботу ми ознайомились із інтерфейсом Set та його нащадками. Провели рефакторинг нашого попереднього коду та замінили списки List на множини Set, використовуючи Comparable та Comparator.