



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:** «Множина TreeSet»

Виконали:  
Студенти групи ІА-24  
(бригада №1)  
Чайка А.П.  
Коханчук.М.М.  
Котлярчук М.С.

Перевірив:  
Колеснік Валерій  
Миколайович

## Хід роботи:

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.

## Завдання:

Варіант	Завдання	Тип ітератора		
		Задача 1	Задача 2	Задача 3
1	1	a	b	c

Номер	Завдання
1	<p>Класи: Інститут (назва, список факультетів) Факультет (назва, список студентів) Студент (ім'я, прізвище, номер залікової книжки, середній бал)</p> <p>Задача: 1) Знайти загальну кількість студентів, що навчається в інституті 2) Знайти факультет, на якому навчається найбільша кількість студентів 3) Скласти список студентів, у яких середній бал в діапазоні 95..100</p>

## Результати + код:

```
Number of students in institute:
6
The most popular faculty:
FICT
Students with average mark in range of 95-100
Daria Sidenko
Anton Chayka
Maxim Kotlyarchuk
Danya Yablonskiy
```

```

Main.java x Institute.java x Faculty.java x Student.java x
1  import java.util.*;
2
3  public class Main {
4  public static void main(String[] args) {
5      DataBase();
6      task1();
7      task2();
8      task3();
9  }
10
11  1 usage
12  public static void task1(){
13      System.out.println("Number of students in institute:");
14      System.out.println(Faculty.getAllStudents().size());
15  }
16
17  1 usage
18  public static void task2(){
19      System.out.println("The most popular faculty:");
20      System.out.println(Institute.getAllFaculties().last());
21  }
22
23  1 usage
24  public static void task3(){
25      Student studentMin = new Student( name: "minMark", surname: "forA", number: 10000, averageMark: 95);
26      Student studentMax = new Student( name: "maxMark", surname: "forA", number: 10007, averageMark: 100);
27      System.out.println("Students with average mark in range of 95-100");
28      NavigableSet<Student> stud = Faculty.getAllStudents().subSet(studentMin, fromInclusive: true, studentMax, toInclusive: true);
29      for (Student i : stud) {
30          System.out.println(i.toString());
31      }
32  }
33
34  1 usage
35  public static void DataBase(){
36      Student student1 = new Student( name: "Anton", surname: "Chayka", number: 10001, averageMark: 96);
37      Student student2 = new Student( name: "Daria", surname: "Sidenko", number: 10002, averageMark: 95);
38      Student student3 = new Student( name: "Maxim", surname: "Kotlyarchuk", number: 10003, averageMark: 98);
39      Student student4 = new Student( name: "Anna", surname: "Orlovska", number: 10004, averageMark: 90);
40      Student student5 = new Student( name: "Yulia", surname: "Meleshko", number: 10005, averageMark: 89);
41      Student student6 = new Student( name: "Danya", surname: "Yablonskiy", number: 10006, averageMark: 99.5);
42
43      TreeSet<Student> fictSpace = new TreeSet<>(new Student.markComparator());
44      fictSpace.add(student1);
45      fictSpace.add(student2);
46      fictSpace.add(student3);
47      fictSpace.add(student4);
48      Faculty fict = new Faculty( name: "FICT", fictSpace.size(), fictSpace);
49
50      TreeSet<Student> fpmSpace = new TreeSet<>(new Student.markComparator());
51      fpmSpace.add(student5);
52      fpmSpace.add(student6);
53      Faculty fpm = new Faculty( name: "FPM", fpmSpace.size(), fpmSpace);
54
55      Faculty.getAllStudents().addAll(fictSpace);
56      Faculty.getAllStudents().addAll(fpmSpace);
57
58      Institute inst = new Institute( name: "KPI");
59
60      inst.addFaculties(fict);
61      inst.addFaculties(fpm);
62  }
63
64  }

```

```
Main.java × Institute.java × Faculty.java × Student.java ×
1  import java.util.*;
   3 usages
2  public class Institute {
   2 usages
3      String name;
   2 usages
4      public static TreeSet<Faculty> allFaculties = new TreeSet<>();
   1 usage
5      public Institute(String name) { this.name = name; }
8
9      public String getName() { return name; }
   1 usage
12     public static TreeSet<Faculty> getAllFaculties() { return allFaculties; }
   2 usages
15     public void addFaculties(Faculty faculties) { allFaculties.add(faculties); }
18 }
19
```

```
Main.java × Institute.java × Faculty.java × Student.java ×
2  public class Faculty implements Comparable<Faculty>{
   2 usages
3      String name;
   5 usages
4      Integer quantity;
   1 usage
5      TreeSet<Student> space;
   2 usages
6      public static TreeSet<Student> allStudents = new TreeSet<>(new Student.markComparator());
   2 usages
7      public Faculty(String name, Integer quantity, TreeSet<Student> space){
8          this.name = name;
9          this.quantity = quantity;
10         this.space = space;
11     }
12
13     @ public int compareTo(Faculty other) {
14         int result = this.quantity.compareTo(other.quantity);
15         if (result != 0){
16             return result;
17         }
18         return this.quantity.compareTo(other.quantity);
19     }
   4 usages
20     public static TreeSet<Student> getAllStudents() { return allStudents; }
23
24     public void addStudents(TreeSet<Student> students) { allStudents.addAll(students); }
27
28     public String toString() { return name; }
31 }
32
```

```
Main.java x Institute.java x Faculty.java x Student.java x
2 public class Student {
3     2 usages
4     private String name;
5     2 usages
6     private String surname;
7     1 usage
8     private Integer number;
9     2 usages
10    private double averageMark;
11
12    8 usages
13    public Student(String name, String surname, int number, double averageMark) {
14        this.name = name;
15        this.surname = surname;
16        this.number = number;
17        this.averageMark = averageMark;
18    }
19    3 usages
20    public static class markComparator implements Comparator<Student>{
21        public int compare(Student student1, Student student2) {
22            if (student1.getAverageMark() > student2.getAverageMark()) {
23                return 1;
24            }
25            if (student1.getAverageMark() < student2.getAverageMark()) {
26                return -1;
27            }
28            return 0;
29        }
30    }
31    @Override
32    public String toString() { return name + " " + surname; }
33    4 usages
34    public double getAverageMark() { return averageMark; }
35 }
```

## Висновок:

У цій роботі ми ознайомились з класом TreeSet та його методами, методами інтерфейсів Comparator, Comparable, NavigableSet. Узагальнили знання із створення об'єктів, конструкторів та методів, реалізації інтерфейсів у Java.