



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

Лабораторна робота №9
з дисципліни «Основи програмування»
Тема: « Основи ООП »

Виконали:

студенти групи ІА-23
Каширов Д. О.
Проценко. В. І.
Ковальський К. В.

Перевірив:

Колеснік Валерій

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:

- Set
- TreeSet
- Comparable
- Comparator
- SortedSet
- NavigableSet

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

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.

```

1  import java.util.TreeSet;
2  import java.util.Comparator;
3  import java.util.Iterator;
4
5  public class Task6_Lab11 {
6
7      public static void main(String[] args) {
8          Institut institut = new Institut( name: "KPI", Algoritm.c);
9          Fakultet fakultet = new Fakultet( name: "ABCD");
10         fakultet.addStudent(new Student( firstName: "Ivan", lastName: "Ivanov", nomZK: 1, averGrad: 80));
11         fakultet.addStudent(new Student( firstName: "Petr", lastName: "Kozlov", nomZK: 2, averGrad: 96));
12         institut.addFakultet(fakultet);
13         fakultet = new Fakultet( name: "TYRE");
14         fakultet.addStudent(new Student( firstName: "Alex", lastName: "Kovalenko", nomZK: 3, averGrad: 75));
15         fakultet.addStudent(new Student( firstName: "Olga", lastName: "Petrenko", nomZK: 4, averGrad: 88));
16         fakultet.addStudent(new Student( firstName: "Nina", lastName: "Dimitradze", nomZK: 50, averGrad: 96.5));
17         institut.addFakultet(fakultet);
18         fakultet = new Fakultet( name: "BRICK");
19         fakultet.addStudent(new Student( firstName: "Alex", lastName: "Kovalenko", nomZK: 13, averGrad: 100));
20         fakultet.addStudent(new Student( firstName: "Kostik", lastName: "Malyshev", nomZK: 44, averGrad: 95));
21         fakultet.addStudent(new Student( firstName: "Dima", lastName: "Kabanov", nomZK: 11, averGrad: 65));
22         institut.addFakultet(fakultet);

```

```

24         // використані ітератори та цикли з лаб10
25         System.out.println("Total students: " + institut.getStudQuontity());
26
27         // використаний компаратор по двом властивостям факультетів:
28         // 1) кількість студентів; 2) найменування факультету
29         System.out.println("Biggest fakultet: " + institut.getBiggestFakultet());
30
31         // використаний фіктивний студент з єдиним реквізитом - averGrad - виключно для
32         // відбору діапазону 95...100 за допомогою інтерфейсу Comparable
33         System.out.println("Excellent students: " + institut.getExcellentStudents());
34     }
35 }
36
37
38 // варіанти алгоритмів
39 // a - нетипізований ітератор
40 // b - типізований ітератор
41 // c - типізований цикл «for-each»
42 enum Algoritm {
43     a, b, c;
44 }

```

```

46 class Student implements Comparable<Student> {
47     private String firstName;
48     private String lastName;
49     private int nomZK;
50     private double averGrad;
51
52     Student(String firstName, String lastName, int nomZK, double averGrad) {
53         this.firstName = firstName;
54         this.lastName = lastName;
55         this.nomZK = nomZK;
56         this.averGrad = averGrad;
57     }
58
59     @Override
60     public String toString() {
61         return "\nstudent " + firstName + " " + lastName + ", nom zaL. " + nomZK + ", aver. grad " + averGrad;
62     }

```

```

64     @Override
65     public int compareTo(Student s) {
66         int result = Double.compare(this.averGrad, s.averGrad);
67         if (result != 0) {
68             return result;
69         }
70         return this.lastName.compareTo(s.lastName);
71     }
72 }
73
74 class Fakultet {
75     private String name;
76     private TreeSet<Student> students = new TreeSet<>();
77
78     Fakultet(String name) { this.name = name; }
79
80
81
82     public void addStudent(Student student) { students.add(student); }
83
84
85
86     public int getStudQuantity() { return students.size(); }
87
88
89
90     @Override
91     public String toString() { return name; }

```

```

95     public TreeSet<Student> getAllStudents() { return students; }
96 }
97
98
99
100 class Institut {
101     public static final ComparatorStudQuant COMPARATOR_STUD_QUANT = new ComparatorStudQuant();
102     private String name;
103     private TreeSet<Fakultet> fakultets = new TreeSet<>(Institut.COMPARATOR_STUD_QUANT);
104     private Algoritm algoritm;
105
106     Institut(String name, Algoritm algoritm) {
107         this.name = name;
108         this.algoritm = algoritm;
109     }
110
111     public void addFakultet(Fakultet fakultet) { fakultets.add(fakultet); }

```

```

111     public void addFakultet(Fakultet fakultet) { fakultets.add(fakultet); }
114
115     public int getStudQuontity() {
116         int result = 0;
117         if (algorithm == Algoritm.a) {
118             for (Iterator i = fakultets.iterator(); i.hasNext();) {
119                 Object o = i.next();
120                 Fakultet fakultet = (Fakultet) o;
121                 result += fakultet.getStudQuantity();
122             }
123         } else if (algorithm == Algoritm.b) {
124             for (Iterator<Fakultet> i = fakultets.iterator(); i.hasNext();) {
125                 Fakultet fakultet = i.next();
126                 result += fakultet.getStudQuantity();
127             }
128         } else if (algorithm == Algoritm.c) {
129             for (Fakultet fakultet : fakultets) {
130                 result += fakultet.getStudQuantity();
131             }
132         }
133         return result;
134     }

```

```

136     public Fakultet getBiggestFakultet() { return fakultets.last(); }
139
140     public static class ComparatorStudQuant implements Comparator<Fakultet> {
141         @Override
142         public int compare(Fakultet fakultet_left, Fakultet fakultet_right) {
143             int result = Integer.compare(fakultet_left.getStudQuantity(), fakultet_right.getStudQuantity());
144             if (result != 0) {
145                 return result;
146             }
147             return fakultet_left.toString().compareTo(fakultet_right.toString());
148         }
149     }

```

```

151     public TreeSet<Student> getExcellentStudents() {
152         TreeSet<Student> result = new TreeSet<>();
153         if (algorithm == Algoritm.a) {
154             for (Iterator i = fakultets.iterator(); i.hasNext();) {
155                 Object o = i.next();
156                 Fakultet fakultet = (Fakultet) o;
157                 result.addAll(fakultet.getAllStudents());
158             }
159         } else if (algorithm == Algoritm.b) {
160             for (Iterator<Fakultet> i = fakultets.iterator(); i.hasNext();) {
161                 Fakultet fakultet = i.next();
162                 result.addAll(fakultet.getAllStudents());
163             }
164         } else if (algorithm == Algoritm.c) {
165             for (Fakultet fakultet : fakultets) {
166                 result.addAll(fakultet.getAllStudents());
167             }
168         }
169         return (TreeSet) result.tailSet(new Student( firstName: "Just", lastName: "For_comparison", nomZK: 0, averGrad: 94.9));
170     }
171 }

```

Вивід програми:

```
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:\Program
Total students: 8
Biggest fakultet: TYRE
Excellent students: [
student Kostik Malyshev, nom zal. 44, aver. grad 95.0,
student Petr Kozlov, nom zal. 2, aver. grad 96.0,
student Nina Dimitradze, nom zal. 50, aver. grad 96.5,
student Alex Kovalenko, nom zal. 13, aver. grad 100.0]

Process finished with exit code 0
```

Висновок: Під час виконання цієї лабораторної роботи ми ознайомились з інтерфейсами та класами Set, TreeSet, SortedSet, NavigableSet. Визначили різницю між інтерфейсами Set та Collection. Дослідили випадки коли слід використовувати списки, а в яких краще звернутись до множин. Безпосередньо мали справу з TreeSet та розібрались які елементи можуть зберігатись у ньому. Вдосконалили свої знання.