

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №11

з дисципліни «Основи Програмування»

Тема: Колекції. Множина TreeSet

Виконали:
Студенти групи ІА-23
Мозоль Владислав,
Курач Владислав,
Лядський Дмитро
Дата здачі: 18.02.2023

Перевірів:
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.

3. Відповісти на контрольні питання.

5	<p>Класи: Сервіс пошуку товарів (список інтернет-магазинів) Інтернет-магазин (назва магазину, список товарів та цін на них в цьому магазині) Товар (назва товару, рекомендована ціна виробника)</p> <p>Задача: 1) знайти мінімальну ціну на заданий товар 2) скласти список магазинів, в яких заданий товар можна купити по мінімальній ціні 3) Визначити, чи є магазин, усі товари якого можна купити по цінам, дешевшим ніж рекомендована ціна виробника</p>
---	--

Main

```
1 import java.util.*;
2
3 public class Main {
4     1 usage
5     public static void task1( String needsGood) {
6         Set<Shop> setRzk = new TreeSet<>();
7         setRzk.add(new Shop( nameOfGood: "PlayStation 5", price: 37));
8         setRzk.add(new Shop( nameOfGood: "Iphone 45X", price: 100));
9         setRzk.add(new Shop( nameOfGood: "Witcher 3", price: 23));
10        setRzk.add(new Shop( nameOfGood: "Red Dead Redemption 3", price: 19));
11        Shop rzk = new Shop( nameOfShop: "Rozetka", setRzk);
12
13        Set<Shop> setFoxtrot = new TreeSet<>();
14        setFoxtrot.add(new Shop( nameOfGood: "PlayStation 5", price: 34));
15        setFoxtrot.add(new Shop( nameOfGood: "Iphone 45X", price: 98));
16        setFoxtrot.add(new Shop( nameOfGood: "Witcher 3", price: 18));
17        setFoxtrot.add(new Shop( nameOfGood: "Red Dead Redemption 3", price: 13));
18        Shop foxtrot = new Shop( nameOfShop: "Foxtrot", setFoxtrot);
19
20        Set<Shop> setAllo = new TreeSet<>();
21        setAllo.add(new Shop( nameOfGood: "PlayStation 5", price: 39));
22        setAllo.add(new Shop( nameOfGood: "Iphone 45X", price: 115));
23        setAllo.add(new Shop( nameOfGood: "Witcher 3", price: 22));
24        setAllo.add(new Shop( nameOfGood: "Red Dead Redemption 3", price: 18));
25        Shop allo = new Shop( nameOfShop: "Allo", setAllo);
26
27        ComparatorByAlph comparator1 = new ComparatorByAlph();
28        Set<Shop> listOfShops = new TreeSet<>(comparator1);
29        listOfShops.add(rzk);
30        listOfShops.add(foxtrot);
```

```

30 listOfShops.add(foxtrot);
31 listOfShops.add(allo);
32 SearchService list = new SearchService(listOfShops);
33 System.out.println("Список магазинів: " + list);
34
35 for (Iterator<Shop> rzkIterator = rzk.setOfGoods.iterator(); rzkIterator.hasNext();){
36     Shop goodInRzk = rzkIterator.next();
37     for (Iterator<Shop> foxtrotIterator = foxtrot.setOfGoods.iterator(); foxtrotIterator.hasNext();){
38         Shop goodInFoxtrot = foxtrotIterator.next();
39         for (Iterator<Shop> alloIterator = allo.setOfGoods.iterator(); alloIterator.hasNext();){
40             Shop goodInAllo = alloIterator.next();
41             if (goodInRzk.nameOfGood.equals(needsGood) && goodInFoxtrot.nameOfGood.equals(needsGood) && goodInAllo.nameOfGood.equals(needsGood)) {
42                 if (goodInRzk.price < goodInFoxtrot.price && goodInRzk.price < goodInAllo.price)
43                     System.out.println("Мінімальна ціна на товар \" + needsGood + "\" + \" в магазині \" + rzk.nameOfShop);
44                 else if (goodInRzk.price > goodInFoxtrot.price && goodInFoxtrot.price < goodInAllo.price)
45                     System.out.println("Мінімальна ціна на товар \" + needsGood + "\" + \" в магазині \" + foxtrot.nameOfShop);
46                 else if (goodInRzk.price > goodInAllo.price && goodInFoxtrot.price > goodInAllo.price)
47                     System.out.println("Мінімальна ціна на товар \" + needsGood + "\" + \" в магазині \" + allo.nameOfShop);
48             }
49         }
50     }
51 }
52 }
53 }
54 }
55 }
56 }
57 1 usage
58
59 public static void task2( String needsListGoods) {
60     Shop shops2 = new Shop( nameOfShop: "Foxtrot");
61     shops2.newSetOfGoods.add(new Goods( nameOfGood: "PlayStation 5", recPrice: 34));
62     shops2.newSetOfGoods.add(new Goods( nameOfGood: "Iphone 45X", recPrice: 98));
63 }

```

```

56 public static void task2( String needsListGoods) {
57     Shop shops2 = new Shop( nameOfShop: "Foxtrot");
58     shops2.newSetOfGoods.add(new Goods( nameOfGood: "PlayStation 5", recPrice: 34));
59     shops2.newSetOfGoods.add(new Goods( nameOfGood: "Iphone 45X", recPrice: 98));
60     shops2.newSetOfGoods.add(new Goods( nameOfGood: "Witcher 3", recPrice: 18));
61     shops2.newSetOfGoods.add(new Goods( nameOfGood: "Red Dead Redemption 3", recPrice: 13));
62
63
64     Shop shops3 = new Shop( nameOfShop: "Allo");
65     shops3.newSetOfGoods.add(new Goods( nameOfGood: "PlayStation 5", recPrice: 39));
66     shops3.newSetOfGoods.add(new Goods( nameOfGood: "Iphone 45X", recPrice: 115));
67     shops3.newSetOfGoods.add(new Goods( nameOfGood: "Witcher 3", recPrice: 22));
68     shops3.newSetOfGoods.add(new Goods( nameOfGood: "Red Dead Redemption 3", recPrice: 18));
69
70     ComparatorByPrice comparator = new ComparatorByPrice();
71     Set<Goods> sortedGoods = new TreeSet<>(comparator);
72
73     Goods first = new Goods( nameOfGood: "PlayStation 5", recPrice: 35);
74     Goods second = new Goods( nameOfGood: "Iphone 45X", recPrice: 99);
75     Goods third = new Goods( nameOfGood: "Witcher 3", recPrice: 19);
76     Goods four = new Goods( nameOfGood: "Red Dead Redemption 3", recPrice: 14);
77
78     Shop shops1 = new Shop( nameOfShop: "Rozetka");
79
80     shops1.newSetOfGoods.add(first);
81     shops1.newSetOfGoods.add(second);
82     shops1.newSetOfGoods.add(third);
83     shops1.newSetOfGoods.add(four);
84
85     for (Iterator shopRzk = shops1.newSetOfGoods.iterator(); shopRzk.hasNext();) {
86         Object abc = shopRzk.next();

```

```

84
85     for (Iterator shopRzk = shops1.newSetOfGoods.iterator(); shopRzk.hasNext();) {
86         Object abc = shopRzk.next();
87
88         for (Iterator shopAllo = shops2.newSetOfGoods.iterator(); shopAllo.hasNext();) {
89             Object adc = shopAllo.next();
90
91             for (Iterator shopFxt = shops3.newSetOfGoods.iterator(); shopFxt.hasNext();) {
92                 Object acc = shopFxt.next();
93
94                 if (abc.toString().equals(needsListGoods) && adc.toString().equals(needsListGoods) && acc.toString().equals(needsListGoods)) {
95                     for (Goods names: shops1.newSetOfGoods) {
96                         if(abc.equals(names)) {
97                             sortedGoods.add(names);
98                         }
99                     }
100
101                     for (Goods names1: shops2.newSetOfGoods){
102                         if(adc.equals(names1)) {
103                             sortedGoods.add(names1);
104                         }
105                     }
106
107                     for (Goods names2: shops3.newSetOfGoods){
108                         if(acc.equals(names2)) {
109                             sortedGoods.add(names2);
110                         }
111                     }
112
113                     System.out.print("Список магазинів розташованих за зростанням ціни на товар \"\" + needsListGoods + "\": ");
114                     for (Goods nameOfShop : sortedGoods) {
115                         for (Goods goodNames : shops1.newSetOfGoods) {
116                             if (nameOfShop.recPrice == goodNames.recPrice && nameOfShop.equals(goodNames)) {

```

```

107                             sortedGoods.add(names2);
108                         }
109                     }
110
111                     System.out.print("Список магазинів розташованих за зростанням ціни на товар \"\" + needsListGoods + "\": ");
112                     for (Goods nameOfShop : sortedGoods) {
113                         for (Goods goodNames : shops1.newSetOfGoods) {
114                             if (nameOfShop.recPrice == goodNames.recPrice && nameOfShop.equals(goodNames)) {
115                                 System.out.println(shops1.nameOfShop + " -- " + nameOfShop.recPrice);
116                             }
117                         }
118                         for (Goods goodNames1 : shops2.newSetOfGoods){
119                             if (nameOfShop.recPrice == goodNames1.recPrice && nameOfShop.equals(goodNames1)) {
120                                 System.out.println(shops2 + " -- " + nameOfShop.recPrice);
121                             }
122                         }
123                         for (Goods goodNames2 : shops3.newSetOfGoods){
124                             if (nameOfShop.recPrice == goodNames2.recPrice && nameOfShop.equals(goodNames2)) {
125                                 System.out.println(shops3.nameOfShop + " -- " + nameOfShop.recPrice);
126                             }
127                         }
128                     }
129                 }
130             }
131         }
132     }
133 }
134 }
135 }

```

```

136 public static void task3() {
137     Goods good1 = new Goods( nameOfGood: "PlayStation 5", recPrice: 35);
138     Goods good2 = new Goods( nameOfGood: "Iphone 45X", recPrice: 99);
139     Goods good3 = new Goods( nameOfGood: "Witcher 3", recPrice: 19);
140     Goods good4 = new Goods( nameOfGood: "Red Dead Redemption 3", recPrice: 14);
141     Set<Goods> setOfGoods = new TreeSet<>();
142     setOfGoods.add(good1);
143     setOfGoods.add(good2);
144     setOfGoods.add(good3);
145     setOfGoods.add(good4);
146
147
148     Set<Shop> setRzk = new TreeSet<>();
149     setRzk.add(new Shop(good1.nameOfGood, price: 37));
150     setRzk.add(new Shop(good2.nameOfGood, price: 100));
151     setRzk.add(new Shop(good3.nameOfGood, price: 23));
152     setRzk.add(new Shop(good4.nameOfGood, price: 19));
153     Shop rzk = new Shop( nameOfShop: "Rozetka", setRzk);
154
155
156     Set<Shop> setFoxtrot = new TreeSet<>();
157     setFoxtrot.add(new Shop(good1.nameOfGood, price: 34));
158     setFoxtrot.add(new Shop(good2.nameOfGood, price: 98));
159     setFoxtrot.add(new Shop(good3.nameOfGood, price: 18));
160     setFoxtrot.add(new Shop(good4.nameOfGood, price: 13));
161     Shop foxtrot = new Shop( nameOfShop: "Foxtrot", setFoxtrot);
162
163     Set<Shop> setAllo = new TreeSet<>();

```

```

161     Shop foxtrot = new Shop( nameOfShop: "Foxtrot", setFoxtrot);
162
163     Set<Shop> setAllo = new TreeSet<>();
164     setAllo.add(new Shop(good1.nameOfGood, price: 39));
165     setAllo.add(new Shop(good2.nameOfGood, price: 115));
166     setAllo.add(new Shop(good3.nameOfGood, price: 22));
167     setAllo.add(new Shop(good4.nameOfGood, price: 18));
168     Shop allo = new Shop( nameOfShop: "Allo", setAllo);
169     int number = setOfGoods.size();
170     int counterRzk = 0;
171     int counterFoxtrot = 0;
172     int counterAllo = 0;
173     for (Goods good : setOfGoods){
174         for (Shop goodInShop : rzk.setOfGoods){
175             if (good.nameOfGood.equals(goodInShop.nameOfGood) && good.recPrice > goodInShop.price) {
176                 counterRzk++;
177                 if (counterRzk == number)
178                     System.out.println("У магазині \"\" + rzk.nameOfShop + "\" можна придбати товари по цінам, дешевшим ніж рекомендована ціна виробника");
179             }
180         }
181         for (Shop goodInShop : foxtrot.setOfGoods){
182             if (good.nameOfGood.equals(goodInShop.nameOfGood) && good.recPrice > goodInShop.price) {
183                 counterFoxtrot++;
184                 if (counterFoxtrot == number)
185                     System.out.println("У магазині \"\" + foxtrot.nameOfShop + "\" можна придбати товари по цінам, дешевшим ніж рекомендована ціна виробника");
186             }
187         }

```

```

186     }
187 }
188 for (Shop goodInShop : allo.setOfGoods){
189     if (good.nameOfGood.equals(goodInShop.nameOfGood) && good.recPrice > goodInShop.price) {
190         counterAllo++;
191         if (counterAllo == number)
192             System.out.println("У магазині \" " + allo.nameOfShop + "\" можна придбати товари по цінам, дешевшим ніж рекомендована ціна виробника");
193     }
194 }
195 }
196 }
197 }
198
199 public static void main(String[] args) {
200     task1( needsGood: "PlayStation 5");
201     task2( needsListGoods: "Red Dead Redemption 3");
202     task3();
203 }
204 }

```

Goods

```

40 usages
1 public class Goods implements Comparable<Goods>{
2     19 usages
3     String nameOfGood;
4     16 usages
5     int recPrice;
6
7     16 usages
8     public Goods(String nameOfGood, int recPrice) {
9         this.nameOfGood = nameOfGood;
10        this.recPrice = recPrice;
11    }
12
13    public String toString() { return nameOfGood; }
14
15    public int toShowPrice() { return recPrice; }
16
17    @Override
18    public int compareTo(Goods o) { return this.nameOfGood.compareTo(o.nameOfGood); }
19
20    }
21
22
23

```

Shop

```

1  import java.util.Set;
2  import java.util.TreeSet;
3
4  66 usages
   public class Shop implements Comparable<Shop>{
5      14 usages
       String nameOfShop;
6      10 usages
       String nameOfGood;
7      7 usages
       Set<Shop> setOfGoods = new TreeSet<>();
8      22 usages
       Set<Goods> newSetOfGoods = new TreeSet<>();
9      17 usages
       int price;
10
11     6 usages
       public Shop(String nameOfShop, Set setOfGoods) {
12         this.nameOfShop = nameOfShop;
13         this.setOfGoods = setOfGoods;
14     }
15
16     3 usages
       public Shop(String nameOfShop) { this.nameOfShop = nameOfShop; }
17
18
19     @Override
20     public String toString() { return nameOfShop ; }
21
22     24 usages
       public Shop(String nameOfGood, int price) {
23         this.nameOfGood = nameOfGood;
24         this.price = price;
25
26

```

```

16     public Shop(String nameOfShop) { this.nameOfShop = nameOfShop; }
17
18
19     @Override
20     public String toString() { return nameOfShop ; }
21
22     24 usages
       public Shop(String nameOfGood, int price) {
23         this.nameOfGood = nameOfGood;
24         this.price = price;
25     }
26
27
28
29
30     public String toStringNameGood() { return "--" + newSetOfGoods; }
31
32
33     public Shop(String nameOfShop, String nameOfGood, int price) {
34         this.nameOfShop = nameOfShop;
35         this.nameOfGood = nameOfGood;
36         this.price = price;
37     }
38
39
40     @Override
41     public int compareTo(Shop o) { return this.nameOfGood.compareTo(o.nameOfGood); }
42
43
44 }
45
46

```

SearchService

```
1
2 import java.util.Set;
3 import java.util.TreeSet;
4
5 public class SearchService{
6     String nameOfShop;
7     Set<Shop> listOfShops = new TreeSet<>();
8
9     public SearchService(Set<Shop> listOfShops) { this.listOfShops = listOfShops; }
10
11     @Override
12     public String toString() { return "" + listOfShops; }
13 }
14
15
```

ComparatorByPrice

```
1 public class ComparatorByPrice implements java.util.Comparator<Goods> {
2     @Override
3     public int compare(Goods o1, Goods o2) { return Integer.compare(o1.recPrice, o2.recPrice); }
4 }
5
6
7
```

ComparatorByAlph

```
1 import java.util.Comparator;
2
3 public class ComparatorByAlph implements Comparator<Shop> {
4     @Override
5     public int compare(Shop o1, Shop o2) { return o1.nameOfShop.compareTo(o2.nameOfShop); }
6 }
7
8
9
```

Вивід

```
Список магазинів: [Allo, Foxtrot, Rozetka]
Мінімальна ціна на товар "PlayStation 5" в мазазині Foxtrot
Список магазинів розташованих за зростанням ціни на товар "Red Dead Redemption 3": Foxtrot -- 13
Rozetka -- 14
Allo -- 18
У магазині "Foxtrot" можна придбати товари по цінам, дешевшим ніж рекомендована ціна виробника
```

Висновок: виконуючи цю лабораторну роботу ми ознайомилися з класом TreeSet та інтерфейсами Set, Comparable, Comparator, SortedSet і NavigableSet та навчилися реалізовувати їх на практиці.