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

Лабораторна робота №2

з дисципліни «Основи Програмування»

Тема: Колекції. Множина HashSet. Асоціативні масиви Map.

Виконали:
Студенти групи ІА-23
Мозоль Владислав,
Курач Владислав,
Лядський Дмитро
Дата здачі:07.03.2023

Перевірив:
Колеснік В.М

Київ 2023

Лабораторна робота №12

Тема:Колекції. Множина HashSet. Асоціативні масиви Map.

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів,класів та методів:
   - Set
   - HashSet
   - Object.equals(), Object.hashCode()
   - Map
   - HashMap

2. Виконати завдання лабораторної роботи №10, замінивши списки List (ArrayList та LinkedList) на множини Set (HashSet). Проаналізувати предметну область та на власний розсуд додати функціональність, для реалізації якої використати Map (TreeMap або HashMap).

3. Відповісти на контрольні питання.

5	Класи: Сервіс пошуку товарів (список інтернет-магазинів) Інтернет-магазин (назва магазина, список товарів та цін на них в цьому магазині) Товар (назва товару, рекомендована ціна виробника) 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

Код:

```java
import java.util.*;

public class Main {
    public static void main(String[] args) {
        task1( needsGood: "Iphone 45X");
        task2( needsGood: "Witcher 3");
        task3();
    }

    1 usage
    static Set<Good> listOfGoods() {
        Set<Good> listOfGoods = new HashSet<>();
        listOfGoods.add(new Good( nameOfGood: "PlayStation 5",  price: 35));
        listOfGoods.add(new Good( nameOfGood: "Iphone 45X",  price: 99));
        listOfGoods.add(new Good( nameOfGood: "Witcher 3",  price: 19));
        listOfGoods.add(new Good( nameOfGood: "Red Dead Redemption 3",  price: 14));
        return listOfGoods;
    }

    2 usages
    static Set<Shop> listOfGoodsInRzk() {
        Set<Shop> listOfGoodsInRzk = new HashSet<>();
        listOfGoodsInRzk.add(new Shop( nameOfGood: "PlayStation 5",  price: 37));
        listOfGoodsInRzk.add(new Shop( nameOfGood: "Iphone 45X",  price: 100));
        listOfGoodsInRzk.add(new Shop( nameOfGood: "Witcher 3",  price: 23));
        listOfGoodsInRzk.add(new Shop( nameOfGood: "Red Dead Redemption 3",  price: 19));
        return listOfGoodsInRzk;
    }

    2 usages
    static Set<Shop> listOfGoodsInFoxtrot() {
```

```java
2 usages
static Set<Shop> listOfGoodsInFoxtrot() {
    Set<Shop> listOfGoodsInFoxtrot = new HashSet<>();
    listOfGoodsInFoxtrot.add(new Shop( nameOfGood: "PlayStation 5",  price: 34));
    listOfGoodsInFoxtrot.add(new Shop( nameOfGood: "Iphone 45X",  price: 98));
    listOfGoodsInFoxtrot.add(new Shop( nameOfGood: "Witcher 3",  price: 18));
    listOfGoodsInFoxtrot.add(new Shop( nameOfGood: "Red Dead Redemption 3",  price: 13));
    return listOfGoodsInFoxtrot;
}

2 usages
static Set<Shop> listOfGoodsInAllo() {
    Set<Shop> listOfGoodsInAllo = new HashSet<>();
    listOfGoodsInAllo.add(new Shop( nameOfGood: "PlayStation 5",  price: 39));
    listOfGoodsInAllo.add(new Shop( nameOfGood: "Iphone 45X",  price: 115));
    listOfGoodsInAllo.add(new Shop( nameOfGood: "Witcher 3",  price: 22));
    listOfGoodsInAllo.add(new Shop( nameOfGood: "Red Dead Redemption 3",  price: 18));
    return listOfGoodsInAllo;
}

2 usages
static Set<SearchService> listOfGoodsInShops() {
    Set<SearchService> listOfGoodsInShops = new HashSet<>();
    listOfGoodsInShops.add(new SearchService( nameOfShop: "Rozetka", listOfGoodsInRzk()));
    listOfGoodsInShops.add(new SearchService( nameOfShop: "Foxtrot", listOfGoodsInFoxtrot()));
    listOfGoodsInShops.add(new SearchService( nameOfShop: "Allo", listOfGoodsInAllo()));
    return listOfGoodsInShops;
}
```

```java
1 usage
static void task1(String needsGood) {
    Set<SearchService> listOfShopsAndGoods = listOfGoodsInShops();
    Set<Shop> sortedList = new HashSet<>();
    for (SearchService goodsInShop : listOfShopsAndGoods) {
        for (Shop goodInShop : goodsInShop.listOfGoodsInShop) {
            if (goodInShop.nameOfGood.equals(needsGood)) {
                sortedList.add(new Shop(goodsInShop.nameOfShop, goodInShop.nameOfGood, goodInShop.price));
            }
        }
    }
    System.out.println("Мінімальна ціна на товар \"" + needsGood + "\" в магазині - " + sortedList.iterator().next().nameOfShop);
}


1 usage
static void task2(String needsGood) {
    Map<Set<Shop>, String> mapOfShops = new HashMap<>();
    mapOfShops.put(listOfGoodsInRzk(), "Rozetka");
    mapOfShops.put(listOfGoodsInFoxtrot(), "Foxtrot");
    mapOfShops.put(listOfGoodsInAllo(), "Allo");
    Set<Shop> sortedSet = new HashSet<>();
    for (Iterator<Set<Shop>> goodsInShops = mapOfShops.keySet().iterator(); goodsInShops.hasNext(); ) {
        Set<Shop> goodsInShop = goodsInShops.next();
        for (Iterator goods = goodsInShop.iterator(); goods.hasNext(); ) {
            Object good = goods.next();
            if (good.toString().equals(needsGood)) {
                for (Shop goodInShop : goodsInShop) {
                    if (goodInShop.nameOfGood.equals(good.toString())) {
                        sortedSet.add(new Shop(mapOfShops.get(goodsInShop), needsGood, goodInShop.price));
```

```
1 usage
68      static void task2(String needsGood) {
69          Map<Set<Shop>, String> mapOfShops = new HashMap<>();
70          mapOfShops.put(listOfGoodsInRzk(), "Rozetka");
71          mapOfShops.put(listOfGoodsInFoxtrot(), "Foxtrot");
72          mapOfShops.put(listOfGoodsInAllo(), "Allo");
73          Set<Shop> sortedSet = new HashSet<>();
74          for (Iterator<Set<Shop>> goodsInShops = mapOfShops.keySet().iterator(); goodsInShops.hasNext(); ) {
75              Set<Shop> goodsInShop = goodsInShops.next();
76              for (Iterator goods = goodsInShop.iterator(); goods.hasNext(); ) {
77                  Object good = goods.next();
78                  if (good.toString().equals(needsGood)) {
79                      for (Shop goodInShop : goodsInShop) {
80                          if (goodInShop.nameOfGood.equals(good.toString())) {
81                              sortedSet.add(new Shop(mapOfShops.get(goodsInShop), needsGood, goodInShop.price));
82                          }
83                      }
84                  }
85              }
86          }
87          System.out.print("Список магазинів розташованих за зростанням ціни на товар \"" + needsGood + "\": ");
88          for (Iterator<Shop> goods = sortedSet.iterator(); goods.hasNext(); ) {
89              System.out.print(goods.next().nameOfShop + "  ");
90          }
91          System.out.println();
92      }
93
        1 usage
94      static void task3() {
95          Set<Good> listOfGoods = listOfGoods();
```

```
93
        1 usage
94      static void task3() {
95          Set<Good> listOfGoods = listOfGoods();
96          Set<SearchService> listOfShopsAndGoods = listOfGoodsInShops();
97          for (Iterator<SearchService> shopsWithGoods = listOfShopsAndGoods.iterator(); shopsWithGoods.hasNext(); ) {
98              SearchService shopWhithGoods = shopsWithGoods.next();
99              int counter = 0;
100             for (Iterator<Shop> goodsInShop = shopWhithGoods.listOfGoodsInShop.iterator(); goodsInShop.hasNext(); ) {
101                 Shop goodInShop = goodsInShop.next();
102                 for (Iterator<Good> goods = listOfGoods.iterator(); goods.hasNext(); ) {
103                     Good good = goods.next();
104                     if (goodInShop.nameOfGood.equals(good.nameOfGood) && goodInShop.price < good.recPrice) {
105                         counter++;
106                     }
107                 }
108                 if (counter == listOfGoods.size()) {
109                     System.out.println("У магазині \"" + shopWhithGoods.nameOfShop + "\" можна придбати товари по цінам, дешевшим ніж рекомендована ці
110                 }
111             }
112         }
113     }
114 }
115
116
        12 usages
```

```java
12 usages
class SearchService {
    3 usages
    Set<Shop> listOfGoodsInShop = new HashSet<>();
    6 usages
    String nameOfShop;

    3 usages
    public SearchService(String nameOfShop, Set<Shop> listOfGoodsInShop) {
        this.listOfGoodsInShop = listOfGoodsInShop;
        this.nameOfShop = nameOfShop;
    }


    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        SearchService that = (SearchService) o;
        return nameOfShop.equals(that.nameOfShop);
    }

    @Override
    public int hashCode() { return Objects.hash(nameOfShop); }
}

34 usages
class Shop {
    5 usages
    String nameOfShop;
    9 usages
    String nameOfGood;
    8 usages
    int price;

    12 usages
    public Shop(String nameOfGood, int price) {
        this.nameOfGood = nameOfGood;
        this.price = price;
    }


    2 usages
    public Shop(String nameOfShop, String nameOfGood, int price) {
        this.nameOfShop = nameOfShop;
        this.nameOfGood = nameOfGood;
        this.price = price;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Shop shop = (Shop) o;
        return price == shop.price && nameOfShop.equals(shop.nameOfShop) && nameOfGood.equals(shop.nameOfGood);
    }

```

```java
    @Override
    public int hashCode() { return price; }

    @Override
    public String toString() { return nameOfGood; }
}

11 usages
class Good {
    5 usages
    String nameOfGood;
    5 usages
    int recPrice;

    4 usages
    public Good(String nameOfGood, int price) {
        this.nameOfGood = nameOfGood;
        this.recPrice = price;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Good good = (Good) o;
        return recPrice == good.recPrice && nameOfGood.equals(good.nameOfGood);
    }

    @Override
    public int hashCode() { return Objects.hash(nameOfGood, recPrice); }
}
```

```
    }

    @Override
    public int hashCode() { return price; }

    @Override
    public String toString() { return nameOfGood; }
}

11 usages
class Good {
    5 usages
    String nameOfGood;
    5 usages
    int recPrice;

    4 usages
    public Good(String nameOfGood, int price) {
        this.nameOfGood = nameOfGood;
        this.recPrice = price;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Good good = (Good) o;
        return recPrice == good.recPrice && nameOfGood.equals(good.nameOfGood);
    }
```

Результат виконання програми:

```
Мінімальна ціна на товар "Iphone 45X" в магазині - Foxtrot
Список магазинів розташованих за зростанням ціни на товар "Witcher 3": Foxtrot   Allo   Rozetka
У магазині "Foxtrot" можна придбати товари по цінам, дешевшим ніж рекомендована ціна виробника
```

Висновок: виконуючи лабораторну роботу ми навчилися використовувати класи HashMap та HashSet, реалізовувати методи Object.equals(), Object.hashCode() і ознайомилися із інтерфейсами Set та Map