

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

Лабораторна робота №10

з дисципліни «Основи Програмування»

Тема: Колекції. Списки

Виконали:
Студенти групи ІА-23
Мозоль Владислав,
Курач Владислав,
Лядський Дмитро
Дата здачі: 04.01.2022

Перевірив:
Колеснік В.М

Київ 2022

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:

- Collection
- List
- ArrayList
- LinkedList
- Iterator
- RandomAccess

2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1. Для цього:

5	5	c	a	b
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 заданій 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>			

- проаналізувати завдання;
- створити зазначенні класи;
- для створення списків слід використовувати класи та інтерфейси з Collection Framework (заборонено використовувати масиви);
- усі списки мають бути типізованими (наприклад, ArrayList<Student>, а не просто ArrayList);
- при реалізації задач «1)», «2)», «3)» слід застосувати наступні методи перегляду колекцій у відповідності до свого варіанту (табл. 1):
 - a) нетипізований ітератор;
 - b) типізований ітератор;
 - c) типізований цикл «for-each».

Хід роботи:

Пошук товарів:

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.List;

2 usages
public class SearchService {
    2 usages
    String name;
    3 usages
    List<Shops> shopsList = new ArrayList<Shops>();

    1 usage
    public SearchService(String serviceName) { this.name = serviceName; }

    1 usage
    public void addAllShops(Collection<Shops> shops) { this.shopsList.addAll(shops); }

    @Override
    public String toString() { return name + " -- " + shopsList; }
}
```

Товари:

```
public class Goods {  
    15 usages  
    String nameG;  
    10 usages  
    int recPrice;  
    5 usages  
    int shopPrice;  
  
    14 usages  
    public Goods(String nameGood, int recPrice) {  
        this.nameG = nameGood;  
        this.recPrice = recPrice;  
    }  
  
    2 usages  
    public int changePrice() {  
        shopPrice = (int) (recPrice * (Math.random() * 5));  
        if (shopPrice == 0){  
            shopPrice = recPrice;  
            return shopPrice;  
        }  
        else{  
            return shopPrice;  
        }  
    }  
  
    public String toString() {  
        return nameG + " -- " + changePrice();  
    }  
}
```

Интернет магазин:

```
import java.util.*;  
  
15 usages  
public class Shops{  
    8 usages  
    String nameShop;  
    4 usages  
    HashMap<String, Integer> prices = new HashMap<>();  
    6 usages  
    List<Goods> goodsList = new ArrayList<>();  
    6 usages  
    public Shops(String nameShop) { this.nameShop = nameShop; }  
    4 usages  
    public void addAllGoods(Collection<Goods> good) { this.goodsList.addAll(good); }  
    6 usages  
    public void addGood(String nameGood, int price){  
        int shopPrice = (int) (price * (Math.random() * 5));  
        if(shopPrice == 0)  
            shopPrice = (int) (price * 0.1);  
        prices.put(nameGood, shopPrice);  
    }  
  
    2 usages  
    public int price(String nameGood) {  
        if (prices.containsKey(nameGood)) {  
            return prices.get(nameGood);  
        } else {  
            return -99;  
        }  
    }  
}  
  
@Override
```

```
@Override
public String toString() { return nameShop + " -- " + goodsList; }

2 usages
public Set<String> products() {
    return prices.keySet();
}
}
```

Main:

```
import java.util.*;

public class Main {
    1 usage
    public static void task1(String product) {
        //Створюємо нові товари -----
        Goods first = new Goods( nameGood: "PlayStation 5", recPrice: 35);
        Goods second = new Goods( nameGood: "Iphone 45X", recPrice: 99);
        Goods third = new Goods( nameGood: "Witcher 3", recPrice: 19);
        Goods four = new Goods( nameGood: "Red Dead Redemption 3", recPrice: 14);
        Goods five = new Goods( nameGood: "GTA 7", recPrice: 11);
        //Створюємо нові магазини -----
        Shops shops1 = new Shops( nameShop: "Rozetka");
        Shops shops2 = new Shops( nameShop: "Foxtrot");
        //Створюємо список товарів в першому та другому магазині -----
        shops1.addGood(first.nameG, first.recPrice);
        shops1.addGood(second.nameG, second.recPrice);
        shops1.addGood(third.nameG, third.recPrice);
        shops2.addGood(four.nameG, four.recPrice);
        shops2.addGood(five.nameG, five.recPrice);
        shops2.addGood(third.nameG, third.recPrice);

        int firstPrice = 0;
        int secondPrice = 0;

        for (String names : shops1.products()) {
            for (String nameS : shops2.products()) {
                if (names.equals(product) & nameS.equals(product)) {
                    firstPrice = shops1.price(names);
                    secondPrice = shops2.price(nameS);
                }
            }
        }
    }
}
```



```

    }

    }

    }

    if(abc.equals(shops3)){
        for(Iterator b = shops3.goodsList.iterator(); b.hasNext();){
            Object jk = b.next();
            if(jk.equals(find)){
                shops.add(shops3.nameShop);
            }
        }
    }

    }

    }

    System.out.println("Список магазинів в яких можна купити " + find.nameG + " : " + shops);
}

}

```

1 usage

```

public static void task3(){
    Goods first = new Goods( nameGood: "PlayStation 5", recPrice: 35);
    Goods second = new Goods( nameGood: "Iphone 45X", recPrice: 99);
    Goods third = new Goods( nameGood: "Witcher 3", recPrice: 19);
    Goods four = new Goods( nameGood: "Red Dead Redemption 3", recPrice: 14);
    //Створюємо нові магазини -----
    Shops shops1 = new Shops( nameShop: "Rozetka");
    //Створюю список рекомендовних цін
    List<Goods> pricesList = new ArrayList<>();
}

```

```

List<Goods> pricesList = new ArrayList<>();
//Пошукові сервіси магазинів -----
shops1.addAllGoods(Arrays.asList(first,second,third,four));
pricesList.addAll(Arrays.asList(first,second,third,four));

int sum = 0;
for(Iterator<Goods> b = shops1.goodsList.iterator(); b.hasNext();){
    Goods abc = b.next();
    for(Iterator<Goods> a = pricesList.iterator(); a.hasNext();){
        Goods obj = a.next();
        if((abc.nameG.equals(obj.nameG)) &&(abc.changePrice() < obj.recPrice)){
            sum += 1;
        } else {
            sum += 0;
        }
    }
}

if(sum == 4)
    System.out.println(shops1.nameShop);
else {
    System.out.println("Магазину який працює собі в збиток немає");
}

}

public static void main(String[] args) {
    task1( product: "Witcher 3");
    task2( nameGood: "Witcher 3");
    task3();
}

}

```

Результат:

```
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:D:\Java\IntelliJ IDEA 2022.2.2\lib\idea_rt.jar=58497:D:\Java\IntelliJ IDEA 2022.2.2\bin" -Dfile.encoding=UTF-8 -clas
Найщи́на ці́на у магази́ні: Foxtrot -- 19 Шекелів
Список магазинів в яких можна купити Witcher 3 : [Rozetka, Steam]
Магазину який працює собі в збиток немає

Process finished with exit code 0
```

Висновок: Отже, на цій лабораторній роботі ми ознайомились з javadoc та Collection Framework інтерфейсами. Також працювали з ітераторами. Та дізнались переваги колекцій над масивами і які види колекцій існують.