



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 №10**  
із дисципліни «Програмування. Частина 1. Основи програмування»  
**Тема:** «Колекції. Списки»

Виконали:  
Студентки групи ІА-24  
Ганжа Х. М.  
Кійко А. О.  
Мелешко Ю. С

## Варіант5

Код:

```
import java.lang.reflect.Array;
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Product product1 = new Product("bread", 14);
        Product product2 = new Product("butter", 40);
        Product product3 = new Product("juice", 55);
        Product product4 = new Product("milk", 30);
        Product product5 = new Product("cheese", 70);

        List<Product> productsFromManufacture = new ArrayList<Product>();

        productsFromManufacture.addAll(List.of(product1, product2, product3, product4,
product5));

        Shop shop = new Shop("Ціни від виробника: ", List.of(product1, product2,
product3, product4, product5));
        System.out.println(shop);

        Product product11=new Product(product1.getProduct(),12);
        Product product12=new Product(product2.getProduct(),12);

        Shop shop1 = new Shop("jj", List.of(product5, product4, new Product("bread",
16)));
        Shop shop2 = new Shop("ABC", List.of(product11, product12));
        Shop shop3 = new Shop("Cool", List.of(product3, product4, new Product("butter",
77)));

        AllShop allShop = new AllShop();

        allShop.addShop(Arrays.asList(shop1, shop2, shop3));
        System.out.println(allShop);

        String nameProduct = "bread";
        int minpriceProduct = task1(nameProduct, allShop);
        System.out.println("minpriceProduct: " + nameProduct + ": " + minpriceProduct);
        List<Shop> shopListMinPriceOfProduct = task2(minpriceProduct, allShop,
nameProduct);
        System.out.println("Можна купити по мінімальній ціні у: " +
shopListMinPriceOfProduct);
        task3(productsFromManufacture, allShop);

    }

    public static int task1(String product, AllShop allShop) {
        int minprice = Integer.MAX_VALUE;
        for (Shop shop : allShop.getAllshops()) {
            for (Product p : shop.getList()) {
                if (p.getPrice() < minprice) {
                    minprice = p.getPrice();
                }
            }
        }
        return minprice;
    }

    public static List<Shop> task2(int minpriceProduct, AllShop allShop, String
productMin) {
        List<Shop> shopMinPrice = new ArrayList<Shop>();
```

```

boolean isProductWithMinPrice;
for (Iterator i = allShop.getAllshops().iterator(); i.hasNext(); ) {

    Object o = i.next();
    if (o instanceof Shop) {
        Shop shop = (Shop) o;
        isProductWithMinPrice = false;
        for (Iterator p = shop.getList().iterator(); p.hasNext(); ) {

            Object object = p.next();
            if (object instanceof Product) {
                Product product = (Product) object;
                if (product.getProduct().equals(productMin)) {
                    if (product.getPrice() == minpriceProduct) {
                        isProductWithMinPrice = true;
                    }
                }
            }
        }
        if (isProductWithMinPrice) {
            shopMinPrice.add(shop);
        }
    }
}
return shopMinPrice;
}

public static void task3(List<Product> productsFromManufacture, AllShop allShop) {

    List<Shop> shopMinPrice = new ArrayList<Shop>();
    boolean isProductWithLessPriceManuf;
    for (Iterator<Shop> i = allShop.getAllshops().iterator(); i.hasNext(); ) {
        Shop shop = i.next();
        isProductWithLessPriceManuf = false;
        int countProduct = 0;
        for (Iterator<Product> p = shop.getList().iterator(); p.hasNext(); ) {
            Product product = p.next();
            for (Iterator<Product> pm = productsFromManufacture.iterator();
pm.hasNext(); ) {
                Product productM = pm.next();

                if (product.getProduct().equals(productM.getProduct())) {
                    if (product.getPrice() < productM.getPrice()) {
                        isProductWithLessPriceManuf = true;
                        countProduct++;
                    }
                }
            }
        }

        if (countProduct == shop.getList().size()) {
            shopMinPrice.add(shop);
            System.out.println("У " + shop.getName() + " можна купити товари по
цінам, дешевшим ніж рекомендована ціна виробника");
            System.out.println(shop);
        }
        else{
            System.out.println("У " + shop.getName() +" не можна купити товари по
цінам, дешевшим ніж рекомендована ціна виробника");
        }
    }
}
}

```

```

public class Shop {
    private String name;
    private List<Product> list = new ArrayList<Product>();

    public Shop(String name, List<Product> list) {
        this.name = name;
        this.list = list;
    }

    public Shop() {
    }

    public void addProduct(Collection<Product> productCollection) {
        this.list.addAll(productCollection);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public List<Product> getList() {
        return list;
    }

    public void setList(List<Product> list) {
        this.list = list;
    }

    @Override
    public String toString() {
        return name + " " + list.toString();
    }
}

```

```

public class AllShop {

    private List<Shop> Allshops = new ArrayList<Shop>();

    public AllShop(List<Shop> allshops) {
        Allshops = allshops;
    }

    public AllShop() {
    }

    public List<Shop> getAllshops() {
        return Allshops;
    }

    public void setAllshops(List<Shop> allshops) {
        Allshops = allshops;
    }

    public void addShop(Collection<Shop> shopsCollection) {
        this.Allshops.addAll(shopsCollection);
    }

    @Override
    public String toString() {
        return "Allshops: " + Allshops;
    }
}

```

```

public class Product {
    private int price;
    private String product;

    public Product(String product, int price) {
        this.product = product;
        this.price = price;
    }

    public Product() {
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public String getProduct() {
        return product;
    }

    public void setProduct(String product) {
        this.product = product;
    }

    @Override
    public String toString() {
        return product + " price: " + price;
    }
}

```

## Результат:

```

Ціни від виробника: [bread price: 14, butter price: 40, juice price: 55, milk price: 30, cheese price: 70]
Allshops: [jj [cheese price: 70, milk price: 30, bread price: 16], ABC [bread price: 12, butter price: 12], Cool [juice price: 55, milk price: 30, butter price: 77]]
minpriceProduct: bread: 12
Можна купити по мінімальній ціні у: [ABC [bread price: 12, butter price: 12]]
У jj не можна купити товари по цінам, дешевшим ніж рекомендована ціна виробника
У ABC можна купити товари по цінам, дешевшим ніж рекомендована ціна виробника
ABC [bread price: 12, butter price: 12]
У Cool не можна купити товари по цінам, дешевшим ніж рекомендована ціна виробника

Process finished with exit code 0

```

Висновок: На цій лабораторній роботі ознайомитись з javadoc для наступних інтерфейсів та класів: Collection, List, ArrayList, LinkedList, Iterator, RandomAccess, створили 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), знайшли мінімальну ціну на заданий товар, склали список магазинів, в яких заданий товар можна купити по мінімальній ціні, визначили, чи є магазин, усі товари якого можна купити по цінам, дешевшим ніж рекомендована ціна виробника.