

Міністерство освіти та науки України  
Національний університет «Львівська політехніка»  
Кафедра «Електронних обчислювальних машин»



## **ЗВІТ**

**з лабораторної роботи № 7**

**з дисципліни:** «Кросплатформенні засоби програмування»

**на тему:** «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»

Виконав: ст. гр. КІ-35

Дністрян Я. В.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю.С

Львів-2022

**Мета роботи:** оволодіти навиками параметризованого програмування мовою Java.

### Завдання(4варіант)

1. Створити параметризований клас, що реалізує предметну область задану варіантом.

Клас має містити мінімум 4 методи опрацювання даних включаючи розміщення та виймання елементів. Парні варіанти реалізують пошук мінімального елементу, непарні – максимального. Написати на мові Java та налагодити програму-драйвер для розробленого класу, яка мстить мінімум 2 різні класи екземпляри яких розмішуються у екземплярі розробленого класу-контейнеру. Програма має розмішуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

## 4. Конвеєр

Рис.1. Предметна область

2. Автоматично згенерувати документацію до розробленого пакету.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
4. Дати відповідь на контрольні запитання.

### Текст програми:

#### Лістинг ConveyorApp.java

```
package KI35.Dnistrian.Lab7;

import java.util.*;
import java.io.*;

public class ConveyorApp {
    public static void main(String[] args)
    {
        Conveyor<Product> conveyor = new Conveyor<Product>();
        conveyor.AddProduct(new Baking("Cake", 1200, 25));
        conveyor.AddProduct(new Fruit("Bananas", 500, 400));
        conveyor.AddProduct(new Fruit("Apples", 200, 45 ));
        conveyor.AddProduct(new Baking("Apple pie" , 10000, 100));
        Product res = conveyor.findMin();
        System.out.print("\nThe lightest product on the conveyor is:\n");
        res.print();
    }
}
```

## Лістинг Product.java

```
package KI35.Dnistrian.Lab7;

public interface Product extends Comparable<Product>
{
    public int getSize();
    public int getPrice();
    public void print();
}
```

## Лістинг Fruit.java

```
package KI35.Dnistrian.Lab7;

/**
 * class Fruit implements different fruits
 * @author SlaweeK
 */
public class Fruit implements Product
{
    private String name;
    private int size;
    private int price;
    public Fruit(String name, int size, int price)
    {
        this.name = name;
        this.size = size;
        this.price=price;
    }

    public String getName()
    {
        return name;
    }

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

    public int getPrice()
    {
        return price;
    }

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

    public void setSize(int n)
    {

```

```

        size = n;
    }

    public int getSize()
    {
        return size;
    }

    public int compareTo(Product p)
    {
        Integer s = size;
        return s.compareTo(p.getSize());
    }

    public void print()
    {
        System.out.print("Fruit: " + name + " ----- " + size + "gram" + " - " +
price + "$;\n");
    }
}

```

## Лістинг Baking.java

```

package KI35.Dnistrian.Lab7;

/**
 * class Baking implements various pastries
 * @author Slaweeek
 *
 */
public class Baking implements Product
{
    private String name;
    private int price;
    private int size;
    public Baking(String name, int size, int price)
    {
        this.name = name;
        this.price = price;
        this.size = size;
    }

    public String getName()
    {
        return name;
    }

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

    public int getPrice()
    {
        return price;
    }

    public void setPrice(int price)

```

```

    {
        this.price = price;
    }

    public int getSize()
    {
        return size;
    }

    public int compareTo(Product p)
    {
        Integer s = size;
        return s.compareTo(p.getSize());
    }

    public void print()
    {
        System.out.print("Baking: " + name + " ----- " + size +
            "gram " + " - " + price + "$;\n");
    }
}

```

## Лістинг Conveyor.java

```

package KI35.Dnistrian.Lab7;
import java.util.ArrayList;

/**
 * class Conveyor implements conveyor in a supermarket
 * @author Slaweeek
 */
public class Conveyor <T extends Product>
{
    private ArrayList<T> arr;

    public Conveyor()
    {
        arr = new ArrayList<T>();
    }

    /**
     * The method finds the lightest item among those added
     * @return min or null
     */
    public T findMin()
    {
        if (!arr.isEmpty())
        {
            T min = arr.get(0);
            for (int i=1; i< arr.size(); i++){
                if (arr.get(i).compareTo(min) < 0 )
                    min = arr.get(i);
            }
            return min;
        }
        return null;
    }
}

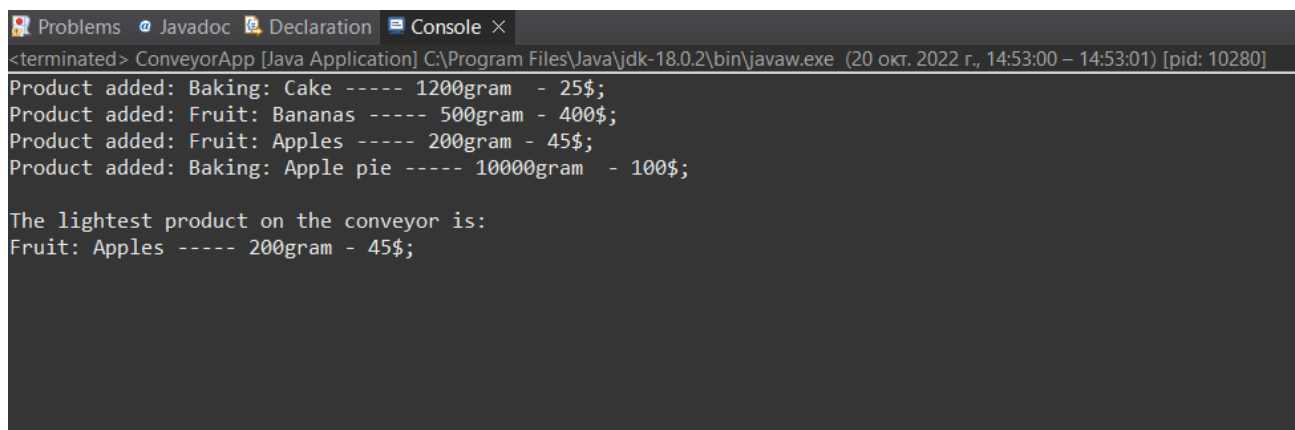
```

```

    }
    /**
     * The method adds new products
     * @param product
     */
    public void AddProduct(T product)
    {
        arr.add(product);
        System.out.print("Product added: ");
        product.print();
    }
    /**
     * The method removes some product
     * @param i number of our element
     */
    public void DeleteProduct(int i)
    {
        arr.remove(i);
    }
}

```

## Результат виконання програми



```

<terminated> ConveyorApp [Java Application] C:\Program Files\Java\jdk-18.0.2\bin\javaw.exe (20 окт. 2022 г., 14:53:00 – 14:53:01) [pid: 10280]
Product added: Baking: Cake ----- 1200gram - 25$;
Product added: Fruit: Bananas ----- 500gram - 400$;
Product added: Fruit: Apples ----- 200gram - 45$;
Product added: Baking: Apple pie ----- 10000gram - 100$;

The lightest product on the conveyor is:
Fruit: Apples ----- 200gram - 45$;

```

Рис.2.Результати роботи програми

## Фрагмент згенерованої документації

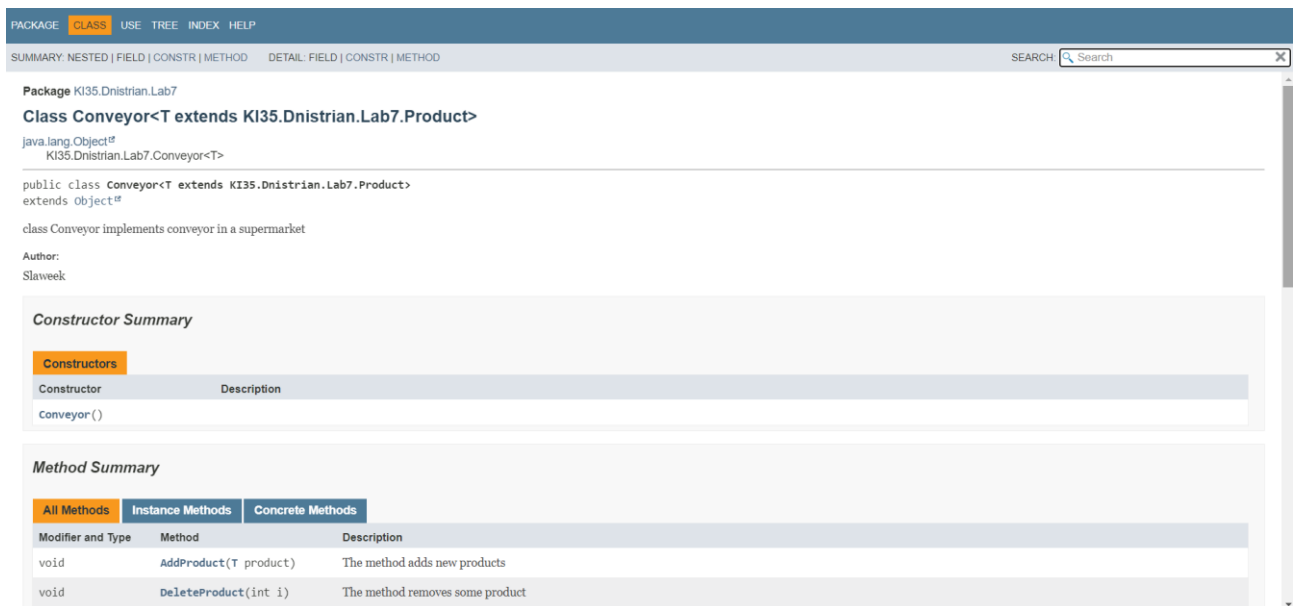


Рис .4. Результат автоматичної генерації документації

## Відповіді на контрольні запитання:

### 1. Дайте визначення терміну «параметризоване програмування».

Параметризоване програмування є аналогом шаблонів у C++. Воно полягає у написанні коду, що можна багаторазово застосовувати з об'єктами різних класів.

### 2. Розкрийте синтаксис виклику параметризованого методу.

(НазваКласу|НазваОб'єкту).[.] НазваМетоду(параметри);

У мові Java компілятор здатний самостійно визначати типи, що підставляються замість параметризованих типів, тому у трикутних дужках вказувати реальні типи не обов'язково. Проте це може призвести до помилок, якщо компілятор не зможе однозначно визначити єдиний супертип для всіх параметрів.

Наприклад:

```
String[] names = {"Ivan", "Ivanovych", "Ivanov"};
```

```
String middle = ArrayAlg.getMiddle(names);
```

**Висновок:** виконуючи лабораторну роботу №7, я оволодів навиками параметризованого програмування мовою Java.