

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 7

з дисципліни: «Кросплатформні засоби програмування»
«Параметризоване програмування»

Варіант - 17

Виконав: ст. гр. КІ-34

Панасюк М.Т.

Прийняв:

Іванов Ю

Львів 2022

ЗАВДАННЯ

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 розміщуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
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.

Варіант завдання:

17. Кошик

Код програми:

File App.java

```
package lab7;

/**
 * Class App
 * @version 1.0
 */
public class App {
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        Basket basket = new Basket();

        basket.add(new Orange(500, "Orange1"));
        basket.add(new Apple(200, "Apple1"));
        basket.add(new Orange(700, "Orange2"));
        basket.add(new Apple(250, "Apple2"));
        basket.add(new Orange(900, "Orange3"));
        basket.add(new Apple(290, "Apple3"));

        System.out.println(basket.get(0).getName());
        System.out.println(basket.get(1).getName());
        System.out.println(basket.get(2).getName());
        System.out.println(basket.get(3).getName());

        System.out.println(basket.getFruitWithMaxWeight().getName());
        System.out.println(basket.getTotalWeight());
    }
}
```

```
}  
}
```

File Basket.java

```
package lab7;  
  
import java.util.ArrayList;  
/**  
 * Class Basket  
 * @version 1.0  
 */  
public class Basket<T extends Fruit> {  
    private ArrayList<T> storage = new ArrayList();  
    /**  
     * Method returns an item by the index  
     * @param index Element index  
     * @throws Exception  
     */  
    public T get(int index)  
    {  
        try{  
            return storage.get(index);  
        }  
        catch (Exception e)  
        {  
            throw e;  
        }  
    }  
    /**  
     * Method adds an item  
     * @param item Item to be added  
     */  
    public void add(T item)  
    {  
        try {  
            storage.add(item);  
        }  
        catch (Exception e)  
        {  
            throw e;  
        }  
    }  
    /**  
     * Method shows storage content  
     */  
    public void showAll()  
    {  
        for(int i = 0; i < storage.size(); i++)  
        {  
            try{  
                storage.get(i).printData();  
            }  
            catch (Exception e)  
            {  
                System.out.println(e);  
            }  
        }  
    }  
    /**  
     * Method returns total weight  
     */  
}
```

```

public int getTotalWeight()
{
    int result = 0;
    for(int i = 0; i < storage.size(); i++)
    {
        result += storage.get(i).getWeight();
    }
    return result;
}
/**
 * Method returns fruit with max weight
 */
public T getFruitWithMaxWeight()
{
    int maxIdx = 0;

    for(int i = 0; i < storage.size(); i++)
    {
        if(i == 0)
        {
            continue;
        }
        if(storage.get(i).getWeight() < storage.get(maxIdx).getWeight())
        {
            maxIdx = i;
        }
    }
    return storage.get(maxIdx);
}
}

```

File Fruit.java

```

package lab7;
/**
 * Class Fruit
 * @version 1.0
 */
public abstract class Fruit {
    protected int weight;
    protected String name;
    /**
     * Constructor
     * @param weight Fruit weight
     * @param name Fruit name
     */
    public Fruit(int weight, String name)
    {
        this.name = name;
        this.weight = weight;
    }
    /**
     * Methods returns Fruit weight
     */
    public int getWeight() {
        return weight;
    }
    /**
     * Methods returns Fruit name
     */
    public String getName() { return name; }
    public void printData()

```

```

{
    System.out.println("Name:" + name);
    System.out.println("Weight:" + weight);
}
}

```

File Orange.java

```

package lab7;
/**
 * Class Rifle extends Gun
 * @version 1.0
 */
public class Orange extends Fruit {
    /**
     * Constructor
     * @param weight Fruit weight
     * @param name Fruit name
     */
    public Orange(int weight, String name)
    {
        super(weight, name);
    }
}

```

File Apple.java

```

package lab7;
/**
 * Class Pistol extends Gun
 * @version 1.0
 */
public class Apple extends Fruit {
    /**
     * Constructor
     * @param weight Fruit weight
     * @param name Fruit name
     */
    public Apple(int weight,String name)
    {
        super(weight, name);
    }
}

```

```

/usr/lib/jvm/java-18/bin/java -javaagent:/opt/idea/lib/idea_rt.jar
Orange1
Apple1
Orange2
Apple2
Apple1
2840

Process finished with exit code 0

```

Резуль
тат
викона
ння
роботи
:

Constructor Summary		
Constructors		
Constructor	Description	
Basket()		

Method Summary		
All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	add(T item)	Method adds an item
T	get(int index)	Method returns an item by the index
T	getFruitWithMaxWeight()	Method returns fruit with max weight
int	getTotalWeight()	Method returns total weight
void	showAll()	Method shows storage content
Methods inherited from class java.lang.Object		
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait		

Висновок: під час виконання цієї роботи я оволодів навичками параметризованого програмування мовою Java.