Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 7

з дисципліни: «Кросплатформні засоби програмування»

«Параметризоване програмування»

Варіант - 19

Виконав:

студент групи КІ - 35

Мороз М.І.

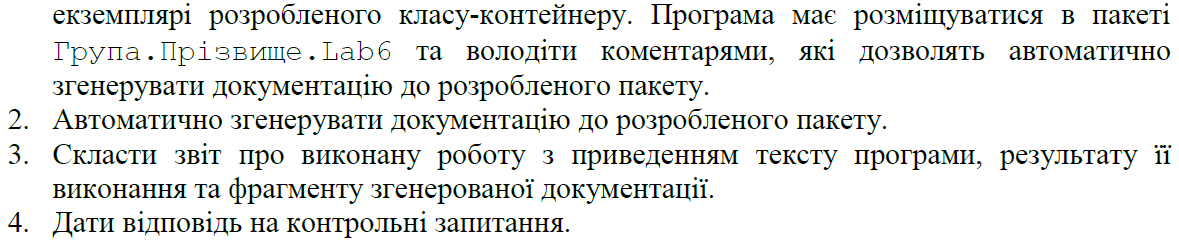
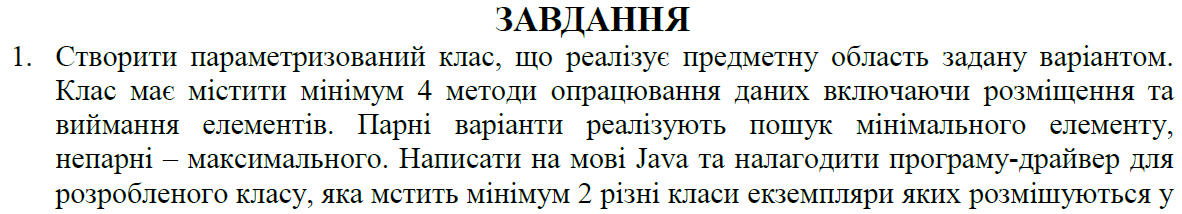
Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів 2022**

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



**Варіант завдання:**



**Код програми:**

**File App.java**

package KI35.Moroz.lab7;  
  
*/\*\*  
 \* Class App  
 \** ***@version*** *1.0  
 \*/*public class App {  
 */\*\*  
 \** ***@param*** *args  
 \*/* public static void main(String[] args)  
 {  
 Safe<? super Jewelry> safe = new Safe();  
  
 safe.add(new Ring(30, "Gold", "Ring1", "Gucci"));  
 safe.add(new Chain(250, "Gold", "Chain1", "Acer"));  
  
 safe.showAll();  
  
 System.*out*.println(safe.get(0).getName());  
  
 System.*out*.println(safe.getTotalMass());  
  
 System.*out*.println(safe.getLargestStore().getName());  
 }  
}

**File Chain.java**

package KI35.Moroz.lab7;  
*/\*\*  
 \* Class Chain  
 \** ***@version*** *1.0  
 \*/*public class Chain extends Jewelry{  
 */\*\*  
 \* Constructor  
 \** ***@param*** *weight  
 \** ***@param*** *material  
 \** ***@param*** *name  
 \** ***@param*** *brand  
 \*/* public Chain(int weight, String material, String name, String brand)  
 {  
 super(weight, material, name, brand);  
 }  
}

**File Jewelry.java**

package KI35.Moroz.lab7;  
  
*/\*\*  
 \* Class Jewelry  
 \** ***@version*** *1.0  
 \*/*public class Jewelry {  
 private int weight;  
 private String material;  
 private String name;  
 private String brand;  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *weight  
 \** ***@param*** *material  
 \** ***@param*** *name  
 \** ***@param*** *brand  
 \*/* public Jewelry(int weight, String material, String name, String brand)  
 {  
 this.name = name;  
 this.material = material;  
 this.brand = brand;  
 this.weight = weight;  
 }  
 */\*\*  
 \* Method prints data  
 \*/* public void printData()  
 {  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Material: " + material);  
 System.*out*.println("Brand: " + brand);  
 System.*out*.println("Weight: " + weight);  
 }  
 */\*\*  
 \* Method returns weight  
 \*/* public int getWeight() {return weight;}  
 */\*\*  
 \* Method returns name  
 \*/* public String getName() { return name; }  
 */\*\*  
 \* Method returns brand  
 \*/* public String getBrand() {return brand; }  
 */\*\*  
 \* Method returns material  
 \*/* public String getMaterial() {return material;}  
}

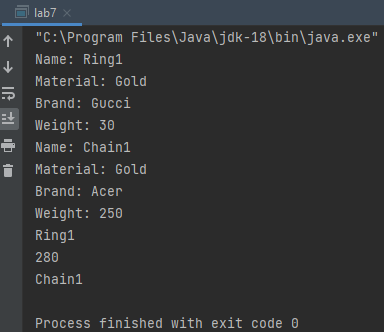
**File Ring.java**

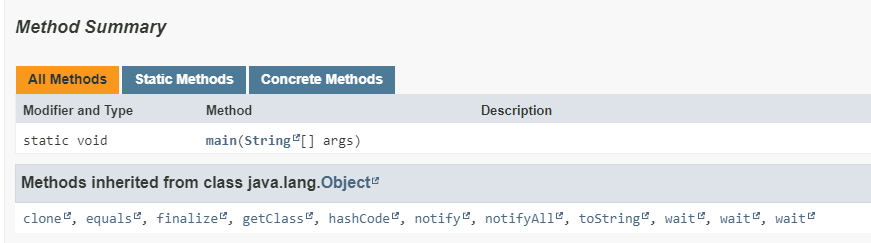
package KI35.Moroz.lab7;  
*/\*\*  
 \* Class Ring  
 \** ***@version*** *1.0  
 \*/*public class Ring extends Jewelry{  
 */\*\*  
 \* Constructor  
 \** ***@param*** *weight  
 \** ***@param*** *material  
 \** ***@param*** *name  
 \** ***@param*** *brand  
 \*/* public Ring(int weight, String material, String name, String brand)  
 {  
 super(weight, material, name, brand);  
 }  
}

**File Safe.java**

package KI35.Moroz.lab7;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* Class Safe  
 \** ***@version*** *1.0  
 \*/*public class Safe<T extends Jewelry> {  
  
 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 safe 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 content mass  
 \*/* public int getTotalMass()  
 {  
 int result = 0;  
 for(int i = 0; i < storage.size(); i++)  
 {  
 result += storage.get(i).getWeight();  
 }  
 return result;  
 }  
 */\*\*  
 \* Method returns the heaviest item  
 \*/* public T getLargestStore()  
 {  
 int largestIdx = 0;  
  
 for(int i = 0; i < storage.size(); i++)  
 {  
 if(i == 0)  
 {  
 continue;  
 }  
 if(storage.get(i).getWeight() > storage.get(largestIdx).getWeight())  
 {  
 largestIdx = i;  
 }  
 }  
 return storage.get(largestIdx);  
 }  
}

**Результат виконання роботи:**





**Відповіді на контрольні запитання**

* Параметризоване програмування є аналогом шаблонів у С++. Воно полягає у написанні коду, що можна багаторазово застосовувати з об’єктами різних класів.
* Підстановочні типи були введені у мову Java для збільшення гнучкості жорсткої існуючої системи параметризованих типів. На відміну від неї підстановочні типи дозволяють враховувати залежності між типами, що виступають параметрами для параметризованих типів. Завдяки цьому підвищується надійність параметризованого коду, полегшується робота з ним та розділяється використання безпечних методів доступу і небезпечних модифікуючих методів.
* Синтаксис оголошення параметризованого класу:

*[public] class НазваКласу <параметризованийТип{,параметризованийТип}> {…}*

**Висновок**:

На цій лабораторній роботі я ознайомився зі особливостями параметризованого програмування мови Java . В результаті виконання роботи, я розробив параметризований клас, що реалізує предметну область задану варіантом.