Міністерство освіти і 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:**

студент групи КІ-35

Луцюк М.Ю.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

Мета: оволодіти навиками параметризованого програмування мовою Java.

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 9 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.

19. Збройний Кейс

Текст програми

Main.java

*/\*\*  
 \*   
 author: Maksym Lutsiuk KI-35  
 \*  
 \*\*/*public class Main {  
   
 public static void main(String[] args) {  
 WeaponSafe <? super Data> MyOwnSafe = new WeaponSafe<>();  
 MyOwnSafe.AddData(new GunClass("Carbine","78D513", "John Smith", 80));  
 MyOwnSafe.AddData(new ArmorClass("Helmet", "Metal", 45));  
 MyOwnSafe.AddData(new ArmorClass("ChestPlate", "Kevlar", 65));  
 MyOwnSafe.AddData(new GunClass("Pistol","L55546DF","Kate Smith", 25));  
 Data result = MyOwnSafe.findMax();  
 System.*out*.println("\n\nThe biggest attachment in the Safe is: ");  
 result.print();  
 MyOwnSafe.DeleteData(0);  
 result = MyOwnSafe.findMax();  
 System.*out*.println("\n\nThe biggest attachment in the Safe is: ");  
 result.print();  
  
 }  
}

WeaponSafe.java

import java.util.ArrayList;  
  
class WeaponSafe <T extends Data> {  
  
 private ArrayList<T> array;  
 public WeaponSafe(){  
 array = new ArrayList<T>();  
 }  
 public T findMax()  
 {  
 if (!array.isEmpty())  
 {  
 T max = array.get(0);  
 for (int i=1; i< array.size(); i++)  
 {  
 if ( array.get(i).compareTo(max) > 0 )  
 max = array.get(i);  
 }  
 return max;  
 }  
 return null;  
 }  
 public void AddData(T data)  
 {  
 array.add(data);  
 System.*out*.print("Element added: ");  
 data.print();  
 }  
 public void DeleteData(int i)  
 {  
  
 array.remove(i);  
 }  
  
  
}

GunClass.java

public class GunClass implements Data {  
  
 private String GunNumber;  
 private String GunType;  
 private String OwnerInitials;  
 private int size;  
  
 public GunClass(String GunType,String GunNumber, String OwnerInitials, int size){  
 this.GunType = GunType;  
 this.GunNumber = GunNumber;  
 this.OwnerInitials = OwnerInitials;  
 this.size = size;  
 }  
  
 public GunClass() {  
  
 }  
  
 public String GetGunNumber(){  
 return GetGunNumber();  
 }  
 public String getGunType(){  
 return GunType;  
 }  
 public void setGunType(String GunType){ this.GunType = GunType;}  
 public void setGunNumber( String GunNumber){  
 this.GunNumber = GunNumber;  
 }  
 public String GetOwnerIitials(){  
 return OwnerInitials;  
 }  
  
 public void setOwnerInitials( String OwnerInitials){  
 this.OwnerInitials = OwnerInitials;  
 }  
  
  
  
 public int getSize() {  
 return size;  
 }  
  
  
 public void print() {  
 System.out.print("Gun Type: " + GunType + ", Gun Number: " + GunNumber + ", Gun Owner: " + OwnerInitials +  
 ", Gun Size: " + size + ";\n");  
 }  
  
 public int compareTo(Data g) {  
 Integer s = size;  
 return s.compareTo(g.getSize());  
 }  
  
  
}

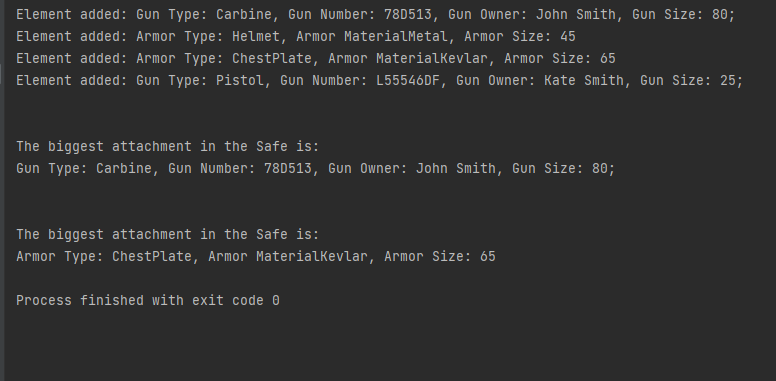
ArmorClass.java

public class ArmorClass implements Data {  
  
 private String ArmorType;  
 private String ArmorMaterial;  
 private int size;  
 public ArmorClass( String ArmorType, String ArmorMaterial, int size){  
 this.ArmorMaterial = ArmorMaterial;  
 this.ArmorType = ArmorType;  
 this.size = size;  
 }  
  
 public String getArmorType(){ return ArmorType;}  
 public String getArmorMaterial(){return ArmorMaterial;}  
 public void setArmorType( String ArmorType){this.ArmorType = ArmorType;}  
 public void setArmorMaterial( String ArmorMaterial){this.ArmorType = ArmorType;}  
  
  
  
  
 public int getSize() {  
 return size;  
 }  
  
  
 public void print() {  
 System.out.print("Armor Type: " + ArmorType + ", Armor Material" + ArmorMaterial  
 + ", Armor Size: " + size + "\n");  
 }  
  
  
 public int compareTo(Data o) {  
 Integer s = size;  
 return s.compareTo(o.getSize());  
 }  
}

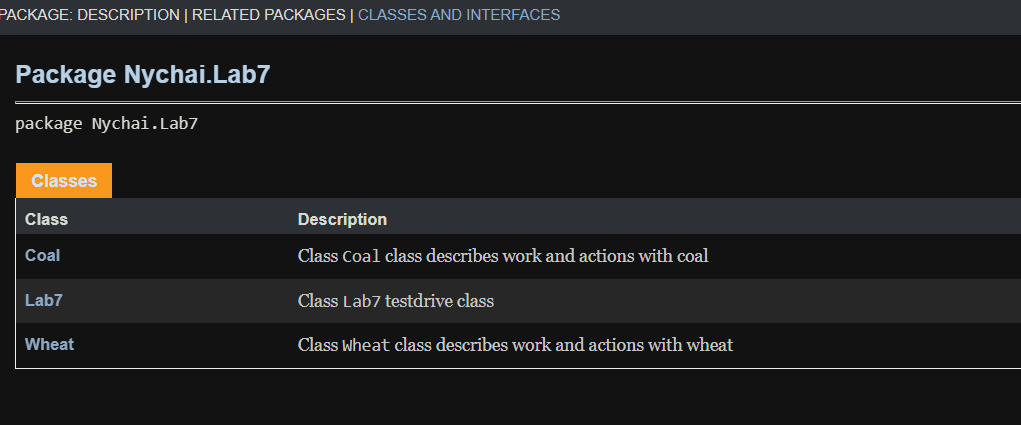
Data.java

public interface Data extends Comparable<Data> {  
 public int getSize();  
 public void print();  
}

Результат роботи програми



**Фрагмент згенерованої документації**



Контрольні питання

1. Дайте визначення терміну «параметризоване програмування».

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

1. Розкрийте синтаксис визначення простого параметризованого класу.

Параметризований клас – це клас з однією або більше змінними типу. Синтаксис оголошення параметризованого класу:

[public] class НазваКласу {…}

Висновок: на даній лабораторній роботі я ознайомився з поняттям параметризованого програмування. Дізнався його основні функції та сфери застосування. Практично оволодів навичками цїєї сфери. Написав та протестував програму з використанням параметризованого програмування.