**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи № 7

з дисципліни: «Кросплатформні засоби програмування»

«ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»

Варіант №9

Виконав:

ст.гр. КІ-34

Корнєєв А.В.

Прийняв:

Іванов Ю.С.

**Львів 2022**

**Мета роботи:** 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.

Варіант 9 :

Споруда

**Код програми :**

package KI34.Kornieiev.Lab7;  
import java.util.\*;  
import java.io.\*;  
  
public class HDDDriver {  
 public static void main(String[] args)  
 {  
 Building <? super Data> building1 = new Building <Data>();  
 building1.AddData(new Flat(200.5, 4, 2));  
 building1.AddData(new UtilityRoom(40, 2));  
 building1.AddData(new Flat(150, 2, 4));  
 building1.AddData(new UtilityRoom(200, 6));  
  
 Data res = building1.findMax();  
 System.*out*.print("The biggest apartment: \n");  
 res.print();  
 }  
  
}

package KI34.Kornieiev.Lab7;  
import java.util.ArrayList;  
  
class Building <T extends Data>  
{  
 private ArrayList<T> arr;  
  
 public Building()  
 {  
 arr = new ArrayList<T>();  
 }  
  
 public T findMax()  
 {  
 if (!arr.isEmpty())  
 {  
 T max = arr.get(0);  
 for (int i=1; i< arr.size(); i++)  
 {  
 if ( arr.get(i).compareTo(max) > 0)  
 max = arr.get(i);  
 }  
 return max;  
 }  
 return null;  
 }  
  
 public void AddData(T data)  
 {  
 arr.add(data);  
 System.*out*.print("Element added: ");  
 data.print();  
 }  
 public void DeleteData(int i)  
 {  
 arr.remove(i);  
 }  
}

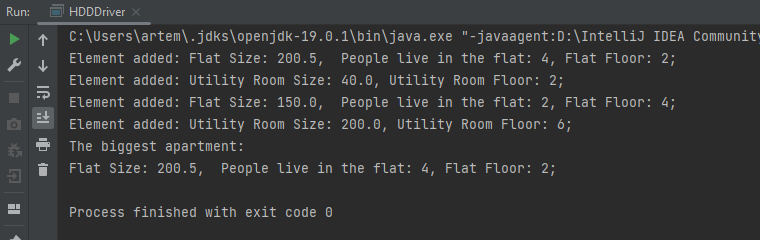
package KI34.Kornieiev.Lab7;  
  
class Flat implements Data  
{  
 private double flatSize;  
 private int flatPeople;  
  
 private int flatFloor;  
  
 public Flat(double flSize, int flPeople, int flFloor)  
 {  
 flatSize = flSize;  
 flatPeople = flPeople;  
 flatFloor = flFloor;  
 }  
  
  
 public int getPeople()  
 {  
 return flatPeople;  
 }  
  
 public void getFileName(int n)  
 {  
 flatPeople = n;  
 }  
  
 public int getFloor()  
 {  
 return flatFloor;  
 }  
  
 public void setFloor(int f)  
 {  
 flatFloor = f;  
 }  
  
 public void setSize(double m)  
 {  
 flatSize = m;  
 }  
  
 public double getSize()  
 {  
 return flatSize;  
 }  
  
 public int compareTo(Data p)  
 {  
 Double s = flatSize;  
 return s.compareTo(p.getSize());  
 }  
  
  
 public void print()  
 {  
 System.*out*.print("Flat Size: " + flatSize +  
 ", People live in the flat: " + flatPeople + ", Flat Floor: " + flatFloor + ";\n");  
 }  
}

package KI34.Kornieiev.Lab7;  
  
class UtilityRoom implements Data  
{  
  
 private double URoomSize;  
 private int URoomFloor;  
  
  
 public UtilityRoom(double URSize, int URFloor)  
 {  
 URoomSize = URSize;  
 URoomFloor = URFloor;  
 }  
  
 public int getFloor()  
 {  
 return URoomFloor;  
 }  
  
 public void setFloor(int q)  
 {  
 URoomFloor = q;  
 }  
  
 public double getSize()  
 {  
 return URoomSize;  
 }  
  
 public int compareTo(Data p)  
 {  
 Double s = URoomSize;  
 return s.compareTo(p.getSize());  
 }  
  
 public void print()  
 {  
 System.*out*.print("Utility Room Size: " + URoomSize + ", Utility Room Floor: "+ URoomFloor + ";\n");  
 }  
  
}

package KI34.Kornieiev.Lab7;  
  
interface Data extends Comparable<Data>  
{  
 public double getSize();  
 public void print();  
}

**Результат виконання програми:**

Вивід в консоль :



**Висновок :** Яоволодів навиками параметризованого програмування мовою Java.