Міністерство освіти і науки України

Національний університет „Львівська політехніка” Кафедра “Електронних обчислювальних машин”



Звіт

з лабораторної роботи №7

з дисципліни «Кросплатформні засоби програмування»

на тему: «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»

Виконав:ст.гр.КІ-34

Шкраба Р. І.

Прийняв:

Іванов Ю.С.

**Львів 2022**

**Мета:** оволодіти навичками 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.

2. Автоматично згенерувати документацію до розробленого пакету.

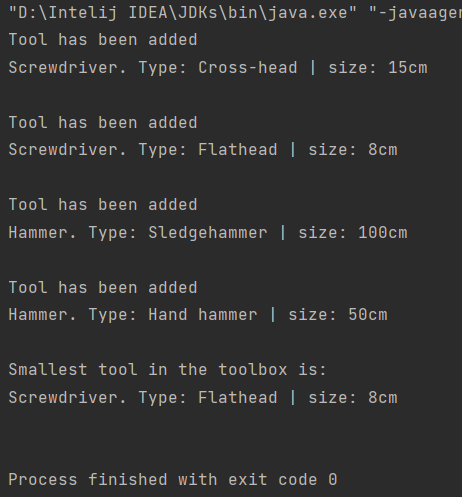
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4.Дати відповідь на контрольні запитання.

*Лістинг:*

|  |
| --- |
| import java.util.ArrayList;  public class Main {  public static void main(String[] args) {  ToolBox<? super Tool> toolBox=new ToolBox<>();  toolBox.addTool(new Screwdriver(15, "Cross-head"));  toolBox.addTool(new Screwdriver(8, "Flathead"));  toolBox.addTool(new Hammer(100,"Sledgehammer"));  toolBox.addTool(new Hammer(50,"Hand hammer"));   Tool smallest = toolBox.findMin();  System.*out*.println("Smallest tool in the toolbox is:");  smallest.print();  } }  class ToolBox<T extends Tool> {  private ArrayList<T> arr;   public ToolBox(){  arr=new ArrayList<T>();  }  public T findMin(){  if(!arr.isEmpty()){  T min = arr.get(0);  for(int i =1;i<arr.size();i++){ // can`t use 'foreach' bc arr.get() requires 'int'  if(arr.get(i).compareTo(min)<0){  min=arr.get(i);  }  }  return min;   }  return null;  }  public void addTool(T Tool){  arr.add(Tool);  System.*out*.println("Tool has been added");  Tool.print();  }   }  interface Tool extends Comparable<Tool>{  public int getSize();  public void print(); }  class Screwdriver implements Tool{  int size;  String type;   public Screwdriver(int \_size,String \_type){  size= \_size;  type=\_type;  }   public void setSize(int \_size){  size = \_size;  }  @Override  public int getSize() {  return size;  }   public void setType(String \_type){  type=\_type;  }   public String getType(){  return type;  }  @Override  public int compareTo(Tool t) {  Integer s=size; // using 'Integer' cause 'int' does not have method 'compareTo'  return s.compareTo(t.getSize()); //'compareTo' returns 0 if =; -1 if a<b; +1 if a>b; b - parameter  }   public void print(){  System.*out*.println("Screwdriver. Type: "+type+" | size: " + size+"cm\n");  } }  class Hammer implements Tool{  int size;  String type;   public Hammer(int \_size,String \_type){  size= \_size;  type=\_type;  }   public void setSize(int \_size){  size = \_size;  }  @Override  public int getSize() {  return size;  }   public void setType(String \_type){  type=\_type;  }   public String getType(){  return type;  }  @Override  public int compareTo(Tool t) {  Integer s=size; // using 'Integer' cause 'int' does not have method 'compareTo'  return s.compareTo(t.getSize()); //'compareTo' returns 0 if =; -1 if a<b; +1 if a>b; b - parameter  }   public void print(){  System.*out*.println("Hammer. Type: "+type+" | size: " + size+"cm\n");  } } |

**Результат:**

**\**

**Висновок:** виконавши цю лабораторну роботу, я частково оволодів навичками параметризованого програмування мовою Java.