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

Дзьобань М. Б.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

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

**Завдання (варіант № 2)**

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 2** – Однозв’язний список

**Текст програми**

**LinkedListApp.java**

import KI\_35.Dzoban.lab7.\*;  
  
*/\*\*  
 \* LinkedListApp implements work with linked list  
 \** ***@author*** *Mykola Dzoban  
 \** ***@version*** *1.0.0  
 \*/*public class LinkedListApp {  
 */\*\*  
 \*  
 \** ***@param*** *args  
 \*/* public static void main(String[] args) {  
 LinkedList<? super IElements> list = new LinkedList<>();  
 list.addElement(new Element(3));  
 list.addElement(new Element(55));  
 list.addElement(new Element(46));  
 list.addElement(new Element(7));  
 list.addElement(new Element(20));  
 IElements res = list.findMin();  
 System.*out*.print("\nMin:");  
 res.print();  
 System.*out*.print("\n\n");  
 list.deleteElement(1);  
 list.printAll();  
  
 }  
}

**Результат виконання програми**

Added new element: Element data: 3 Next element: null

Added new element: Element data: 55 Next element: KI\_35.Dzoban.lab7.Element@34c45dca

Added new element: Element data: 46 Next element: KI\_35.Dzoban.lab7.Element@52cc8049

Added new element: Element data: 7 Next element: KI\_35.Dzoban.lab7.Element@5b6f7412

Added new element: Element data: 20 Next element: KI\_35.Dzoban.lab7.Element@27973e9b

Min:Element data: 3 Next element: null

Element data: 3 Next element: null

Element data: 46 Next element: KI\_35.Dzoban.lab7.Element@34c45dca

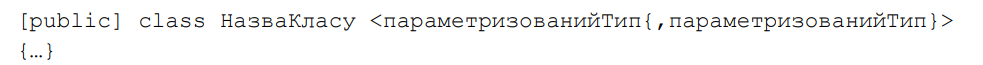
Element data: 7 Next element: KI\_35.Dzoban.lab7.Element@5b6f7412

Element data: 20 Next element: [KI\_35.Dzoban.lab7.Element@27973e9b](mailto:KI_35.Dzoban.lab7.Element@27973e9b)

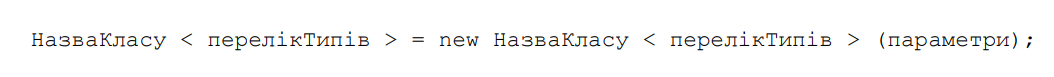
**Фрагмент згенерованої документації**

**Відповіді на контрольні запитання**

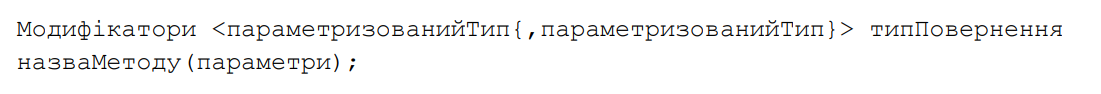
1. Розкрийте синтаксис визначення простого параметризованого класу.



1. Розкрийте синтаксис створення об’єкту параметризованого класу.



1. Розкрийте синтаксис визначення параметризованого методу.

****

**Висновок**

Під час виконання цієї лабораторної я оволодів навиками параметризованого програмування мовою Java.

**Додаток**

**LinkedList.java**

package KI\_35.Dzoban.lab7;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \*  
 \** ***@param*** <*T*>  
 *\*/*public class LinkedList<T extends IElements> {  
 private final ArrayList<T> arrayList;  
 private Element first = new Element(0);  
  
 */\*\*  
 \* Method LinkedList sets link for first element  
 \*/* public LinkedList()  
 {  
 arrayList = new ArrayList<T>();  
 first.setLink(first);  
 }  
  
 */\*\*  
 \* Method adds new element to list  
 \** ***@param*** *data  
 \*/* public void addElement(T data) {  
 if (first.getLink() != first.getLinkElement()) data.setLink(first.getLinkElement());  
 first.setLink(data.getLink());  
 arrayList.add(data);  
 System.*out*.print("Added new element: ");  
 data.print();  
 System.*out*.print("\n");  
 }  
  
 */\*\*  
 \* Method deletes element from list  
 \** ***@param*** *i  
 \*/* public void deleteElement(int i) {  
 if (i+1< arrayList.size()) arrayList.get(i+1).setLink(arrayList.get(i-1).getLink());  
 arrayList.remove(i);  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *minimum value in list  
 \*/* public T findMin()  
 {  
 if(!arrayList.isEmpty())  
 {  
 T min = arrayList.get(0);  
 for (T n : arrayList) {  
 if (n.compareTo(min) < 0) min = n;  
 }  
 return min;  
 }  
 return null;  
 }  
  
 */\*\*  
 \* Method shows all elements  
 \*/* public void printAll()  
 {  
 for (T n: arrayList  
 ) {  
 n.print();  
 System.*out*.print("\n");  
 }  
 }  
}

**Element.java**

package KI\_35.Dzoban.lab7;  
  
public class Element implements IElements{  
  
 private final int data;  
 private Element link;  
  
 */\*\*  
 \* Method adds new element and link to list  
 \** ***@param*** *data  
 \*/* public Element(int data) {  
 this.data = data;  
 this.link = null;  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *data from list  
 \*/* @Override  
 public int getData() {  
 return data;  
 }  
  
 */\*\*  
 \* Methods sets link  
 \** ***@param*** *element  
 \*/* @Override  
 public void setLink(Element element) {  
 link = element;  
 }  
  
 @Override  
 public Element getLink() {  
 return this;  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *element link  
 \*/* public Element getLinkElement() {  
 return link;  
 }  
  
 */\*\*  
 \* Print element  
 \*/* @Override  
 public void print() {  
 if (link==null) System.*out*.print("Element data: " + this.data + " Next element: " + null);  
 else System.*out*.print("Element data: " + this.data + " Next element: " + this.link);  
 }  
  
 */\*\*  
 \*  
 \** ***@param*** *o the object to be compared.  
 \** ***@return*** *compared value  
 \*/* @Override  
 public int compareTo(IElements o) {  
 Integer val = data;  
 return val.compareTo(o.getData());  
 }  
}

**IElements.java**

package KI\_35.Dzoban.lab7;  
  
  
public interface IElements extends Comparable<IElements> {  
  
 int getData();  
 void setLink(Element link);  
 Element getLink();  
 void print();  
}