

Міністерство освіти і науки України
Національний університет „Львівська політехніка”



Звіт
З Лабораторних робіт №6
З навчальної дисципліни: “ Кросплатформені засоби програмування”

Виконав:
ст. гр. КІ-307
Іваньчук О.Ю.
Перевірив:
Іванов Ю.С.

Тема

Оволодіти навиками параметризованого програмування мовою Java.

Код програми:

```
package KI307.Ivanchuk.Lab6;

import java.util.*;
import java.io.*;

/**
 *
 * Це клас який містить статичний метод main який являється точкою входу в програму
 * @author Ivanchyk
 */
public class CCase {

    /**
     * Метод main є точкою входу в програму
     * @param args
     */
    public static void main(String[] args){
        Case <? super Items> MyConveyor = new Case <Items>();

        MyConveyor.AddData(new PPCase("Andry", 6 , 2));
        MyConveyor.AddData(new PencilCase("Taras" , 12));
        MyConveyor.AddData(new PencilCase("Anton" , 4));
        MyConveyor.AddData(new PPCase("Roman", 7 , 22));

        Items res = MyConveyor.findMax();
        System.out.print("The minimum number of handles is: \n");
        res.print();
    }
}
```

```

/**
 * Клас реалізує пинал
 * @author Ivanchyk
 * @param <T>
 */
class Case <T extends Items>{
    private ArrayList<T> arr;

    public Case(){
        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);
        }
    }

/**
 * Це інтерфейс який описує 2 методи
 * @author Ivanchyk
 *
 */
interface Items extends Comparable<Items>{
    public int getNumPencils();
    public void print();
}

/**
 * Цей клас моделює пенал для олівців і ручки
 * @author Ivanchyk
 *
 */
class PPCase implements Items
{
    private String Owner;
    private int NumPen;
    private int NumPencils;

    public PPCase(String Owner, int NumPen, int NumPencils){
        this.Owner = Owner;
        this.NumPen = NumPen;
        this.NumPencils = NumPencils;
    }
}

```

```

/**
 * Метод повертає імя власника
 * @return Owner
 */
public String get_Collecting(){
    return Owner;
}

/**
 * Метод встановлює значення поля Owner
 * @param Owner
 */

public void set_Owner(String name){
    Owner = name;
}

public int getNumPen(){
    return NumPen;
}

public void setNumPen(int n){
    NumPen = n;
}

public void getNumPencils(int positions) {
    this.NumPencils = positions;
}

```

```
public int getNumPencils(){  
    return NumPencils;  
}
```

```
public int compareTo(Items p){  
    Integer s = NumPencils;  
    return s.compareTo(p.getNumPencils());  
}
```

```
/**  
 * Вивід інформації про пинал  
 *  
 */  
public void print(){  
    System.out.print("Owner: " + Owner + ", Number pen: " + NumPen + ",  
Number pencils: " + NumPencils + ";\n");  
}  
}
```

```
/**  
 *Клас моделює пенал для олівців  
 * @author Ivanchyk  
 *  
 */
```

```
class PencilCase implements Items  
{  
    private String Owner;  
    private int NumPencils;  
  
    public PencilCase(String Owner, int NumPencils){
```

```
        this.Owner = Owner;
        this.NumPencils = NumPencils;
    }

    public String getOwner(){
        return Owner;
    }

    public void set_Owner(String name){
        Owner = name;
    }

    public void Set_NumPencils(int n){
        NumPencils = n;
    }

    public int getNumPencils(){
        return NumPencils;
    }

    public int compareTo(Items p){
        Integer s = NumPencils;
        return s.compareTo(p.getNumPencils());
    }

    public void print(){
        System.out.print("Owner: " + Owner + ", Number pencils: " + NumPencils + ";\n");
    }
}
```

Результат виконання програми:

```
Element added: Owner: Andry, Number pen: 6, Number pencils: 2;  
Element added: Owner: Taras, Number pencils: 12;  
Element added: Owner: Anton, Number pencils: 4;  
Element added: Owner: Roman, Number pen: 7, Number pencils: 22;  
The minimum number of handles is:  
Owner: Andry, Number pen: 6, Number pencils: 2;
```

Висновок

У цій лабораторній роботі я оволодів навиками параметризованого програмування мовою Java.