

Міністерство освіти і науки України
Національний університет „Львівська політехніка”



Звіт

З Лабораторних робіт №7

З навчальної дисципліни: “Кросплатформені засоби програмування”

Варіант 11

Виконав:
ст. гр. КІ-36
Басько С.І.
Перевірив:
Іванов Ю.С.

Львів-2022

Тема: оволодіти навиками параметризованого програмування мовою Java.

Код програми:

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

8

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.

11. Торговий тентр

```
import java.util.*;
```

```
import java.io.*
```

```
/**
```

```
*
```

```
* Це клас який містить статичний метод main який являється точкою вхлду в програму
```

```
* @author Matviets
```

```
*/
```

```
public class Center {
```

```
/**
```

```
* Метод main є точкою входу в програму
```

```
* @param args
```

```
*/
```

```
public static void main(String[] args){  
    ShopCenter <? super Floors> MyConveyor = new ShopCenter <Floors>();  
  
    MyConveyor.AddData(new PPCase("Наш", 1 , 5));  
    MyConveyor.AddData(new PencilCase("Фортуна"      ,2));  
    MyConveyor.AddData(new PencilCase("Орнава" , 3));  
}
```

```
MyConveyor.AddData(new PPCase("Гаріна", 4 , 15));
```

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

```
/**
```

```
 * Клас реалізує торговий центр
```

```
 * @author Matviets
```

```
 * @param <T>
```

```
 */
```

```
class ShopCenter <T extends Floors>{  
    private ArrayList<T> arr;  
  
    public ShopCenter(){  
        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 Matviets
```

```
 *
```

```
 */
```

```
interface Floors extends Comparable<Floors>{  
    public int getNumInputs();  
    public void print();  
}
```

```
/**
```

```
 * Цей клас моделює пенал для олівців і ручки
```

```
 * @author Matviets
```

```
*
```

```
*/
```

```
class PPCase implements Floors
```

```
{
```

```
    private String Name;
```

```
    private int NumFloors;
```

```
    private int NumInputs;
```

```
    public PPCase(String Name, int NumFloors, int NumInputs){
```

```
        this.Name = Name;
```

```
        this.NumFloors = NumFloors;
```

```
        this.NumInputs = NumInputs;
```

```
    }
```

```
    /**
```

```
     * Метод повертає значення поля Name
```

```
     * @return Name
```

```
    */
```

```
    public String get_Name(){
```

```
        return Name;
```

```
    }
```

```
    /**
```

```
     * Метод встановлює значення поля Name
```

```
     * @param Name
```

```
    */
```

```
    public void set_Name(String name){
```

```
        Name = name;
```

```
    }
```

```
public int getNumFloors(){  
    return NumFloors;  
}
```

```
public void setNumFloors(int n){  
    NumFloors = n;  
}
```

```
public void getNumInputs(int positions) {  
    this.NumInputs = positions;  
}
```

```
public int getNumInputs(){  
    return NumInputs;  
}
```

```
public int compareTo(Floors p){  
    Integer s = NumInputs;  
    return s.compareTo(p.getNumInputs());  
}
```

```
/**
```

```
 * Вивід інформації торговий центр
```

```
 *
```

```
 */
```

```
public void print(){
```

```
        System.out.print("Name: " + Name + ", Number pen: " + NumFloors + ",  
Number pencils: " + NumInputs + ";\n");  
    }  
}
```

```
/**
```

```
 *Клас моделює пенал для олівців
```

```
 * @author Basko
```

```
 *
```

```
 */
```

```
class PencilCase implements Floors {  
    private String Name;  
    private int NumInputs;  
  
    public PencilCase(String Name, int NumInputs){  
        this.Name = Name;  
        this.NumInputs = NumInputs;  
    }  
  
    public String getName(){  
        return Name;  
    }  
  
    public void set_Name(String name){  
        Name = name;  
    }  
}
```



```

    public void Set_NumInputs(int n){
        NumInputs = n;
    }

    public int getNumInputs(){
        return NumInputs;
    }

    public int compareTo(Floors p){
        Integer s = NumInputs;
        return s.compareTo(p.getNumInputs());
    }

    public void print(){
        System.out.print("Name: " + Name + ", Number pencils: " + NumInputs + ";\n");
    }
}

```

}Результат виконання програми:

```

Element added: Name: Наш, Number pen: 1, Number pencils: 5;
Element added: Name: Фортуна, Number pencils: 2;
Element added: Name: Орнава, Number pencils: 3;
Element added: Name: Гаріна, Number pen: 4, Number pencils: 15;
The minimum number of floors is:
Name: Гаріна, Number pen: 4, Number pencils: 15;

```

Фрагмен документації:

PACKAGE

CLASS

USE

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

SEARCH:

Package KI36.Matviets.Lab7

Class Center

java.lang.Object[↗]
KI36.Matviets.Lab7.Center

```
public class Center
extends Object↗
```

Це клас який містить статичний метод main який являється точкою входу в програму

Author:
Matviets

Constructor Summary

Constructors

Constructor	Description
Center()	

Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type	Method	Description
static void	main(String [↗] [] args)	Метод main є точкою входу в програму

Methods inherited from class java.lang.Object[↗]
equals[↗], getClass[↗], hashCode[↗], notify[↗], notifyAll[↗], toString[↗], wait[↗], wait[↗], wait[↗]

Constructor Details

Center

```
public Center()
```

Method Details

main

```
public static void main(String↗[] args)
```

Метод main є точкою входу в програму

Parameters:
args -

Висновок: на цій лабораторній роботі я оволодів навиками параметризованого програмування мовою Java.