

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт
з лабораторної роботи № 6
з дисципліни: «Кросплатформенні засоби програмування»
на тему: «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»

Виконав:

студент групи КІ-306

Довганюк О.С.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Мета роботи: оволодіти навиками використання засобів мови Java для роботи з потоками і файлами.

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 екземплярі розробленого класу-контейнеру.
2. Програма має розміщуватися в пакеті Група.Прізвище.Lab6 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету. Автоматично згенерувати документацію до розробленого пакету.
3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату їївиконання та фрагменту згенерованої документації та завантажити його у ВНС.
5. Дати відповідь на контрольні запитання.

Варіант 6:

6. Шафа

Вихідний код програми

Wardrobe.java

```
package KI306.Dovganiuk.Lab6;

import java.util.ArrayList;
import java.util.Collections;

/**
 * Class that implements the domain area of a Wardrobe.
 */
```

```

public class Wardrobe<T extends Comparable<T>> {
    private ArrayList<T> items;

    public Wardrobe() {
        items = new ArrayList<>();
    }

    /**
     * Method to add an item.
     * @param item The item to be added.
     */
    public void addItem(T item) {
        items.add(item);
    }

    /**
     * Method to remove an item.
     * @param item The item to be removed.
     */
    public void removeItem(T item) {
        items.remove(item);
    }

    /**
     * Method to find the minimum item.
     * @return The minimum item, or null if the wardrobe is empty.
     */
    public T findMin() {
        if (items.isEmpty()) {
            return null;
        }
        return Collections.min(items);
    }

    /**
     * Method to get all items.
     * @return An ArrayList of all items.
     */
    public ArrayList<T> getItems() {
        return items;
    }
}

```

Class1.java

```

package KI306.Dovganiuk.Lab6;

/**
 * Class that uses Wardrobe.
 */
public class Class1 {
    private Wardrobe<Integer> wardrobe;

```

```

public Class1() {
    wardrobe = new Wardrobe<>();
    wardrobe.addItem(10);
    wardrobe.addItem(20);
    wardrobe.addItem(30);
}

/**
 * Method to add an item to the wardrobe.
 * @param n The item to be added.
 */
public void AddItem(int n){
    wardrobe.addItem(n);
}

/**
 * Method to remove an item from the wardrobe.
 * @param n The item to be removed.
 */
public void RemoveItem(int n){
    wardrobe.removeItem(n);
}

/**
 * Method to get the wardrobe.
 * @return The wardrobe.
 */
public Wardrobe<Integer> getWardrobe() {
    return wardrobe;
}
}

```

Class2.java

```

package KI306.Dovganiuk.Lab6;

import java.util.ArrayList;

/**
 * Class that uses Wardrobe.
 */
public class Class2 {
    private Wardrobe<String> wardrobe;
    static int count = 0;
    static int maxLength = 0;

    public Class2() {
        wardrobe = new Wardrobe<>();
        wardrobe.addItem("Apple");
        wardrobe.addItem("Banana");
        wardrobe.addItem("Cherry");
    }

    /**
     * Method to add an item to the wardrobe.
     * @param str The item to be added.

```

```

    */
    public void AddItem(String str){
        wardrobe.addItem(str);
    }

    /**
     * Method to remove an item from the wardrobe.
     * @param str The item to be removed.
     */
    public void RemoveItem(String str){
        wardrobe.removeItem(str);
    }

    /**
     * Method to count the number of items in the wardrobe with the maximum length.
     * @return The count of items with the maximum length.
     */
    public int Count(){
        for(String s : wardrobe.getItems()){
            if(s.length() > maxLength){
                maxLength = s.length();
                count = 1;
            }
            else if(s.length() == maxLength){
                count++;
            }
        }
        return count;
    }

    /**
     * Method to get the wardrobe.
     * @return The wardrobe.
     */
    public Wardrobe<String> getWardrobe() {
        return wardrobe;
    }
}

```

Main.java

```

package KI306.Dovganiuk.Lab6;

/**
 * Driver class.
 */
public class Main {
    public static void main(String[] args) {
        Class1 class1 = new Class1();
        Class2 class2 = new Class2();
        class1.AddItem(1);
        class2.AddItem("AALemon");
        class2.RemoveItem("AALemon");

        System.out.println(class2.Count());
        System.out.println("Minimum in Class1: " + class1.getWardrobe().findMin());
        System.out.println("Minimum in Class2: " + class2.getWardrobe().findMin());
    }
}

```

```
}  
}
```

Результат виконання програми

```
D:\jdk-17.0.8.1+1\bin\java.exe "-javaagent:D:\I  
2  
Minimum in Class1: 1  
Minimum in Class2: Apple  
  
Process finished with exit code 0
```

Висновок

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