

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №6
з дисципліни «Інженерія програмного забезпечення»
на тему: «Шаблони поведінки. Шаблони Strategy, Chain of Responsibility, Visitor»

Виконав:
студент 2-го курсу ФІОТ
групи ІВ-71
Мазан Я.В.
Номер залікової книжки: 7109
Варіант: 9

Перевірив:
доцент кафедри ОТ
Антонюк А.І.

Завдання

Варіант – 9

Визначити специфікації класів, що реалізують елементи мережевої структури (кабель, сервер, робоча станція). Реалізувати механізм додаткових операцій над мережевою структурою без зміни її елементів. В якості ілюстрації такого механізму розробити операцію визначення кошторису такої структури.

Код програми

Main.java:

```
package com.lab111.labwork6;
public class Main {
    public static void main(String[] args) {
        Cable c = new Cable(3,1500);
        Cable c2 = new Cable(1,480);
        Server s = new Server(1000,1000);
        ProcessingStation st = new ProcessingStation(200,300);
        ProcessingStation st2 = new ProcessingStation(400,450);
        NetworkElement[] elements = new NetworkElement[]{c,c2,s,st,st2};
        Network network = new Network(elements);
        System.out.println("Total network value: " + network.total_value());
    }
}
```

Network.java:

```
package com.lab111.labwork6;
public class Network {
    private NetworkElement[] elements;
    private PriceCalculator operation;
    public Network(NetworkElement el){
        operation = new PriceCalculator();
        elements = new NetworkElement[1];
        elements[0] = el;
    }
    public Network(NetworkElement[] elems) {
        operation = new PriceCalculator();
        elements = elems;
    }
    public int total_value(){
        for(NetworkElement el : elements){
            el.accept(operation);
        }
        return operation.getSum_value();
    }
}
```

PriceCalculator.java:

```
package com.lab111.labwork6;
public class PriceCalculator implements Visitor {
    private int sum_value=0;
    public int visitCable(Cable el) {
        sum_value+=el.getPrice();
        return el.getPrice();
    }
    public int visitServer(Server el) {
        sum_value+=el.getPrice();
        return el.getPrice();
    }
    public int visitProcessingStation(ProcessingStation el) {
```

```

        sum_value+=el.getPrice();
        return el.getPrice();
    }
    public int getSum_value(){
        return sum_value;
    }
}

```

Visitor.java:

```

package com.lab111.labwork6;
public interface Visitor {
    int visitCable(Cable el);
    int visitServer(Server el);
    int visitProcessingStation(ProcessingStation el);
}

```

NetworkElement.java:

```

package com.lab111.labwork6;
public interface NetworkElement {
    void accept(Visitor v);
}

```

Cable.java:

```

package com.lab111.labwork6;
public class Cable implements NetworkElement {
    private int price;
    private int length;
    public Cable(int len, int price) {
        length = len;
        this.price = price;
    }
    public void accept(Visitor v){
        v.visitCable(this);
    }
    public int getPrice(){
        return price;
    }
}

```

Server.java:

```

package com.lab111.labwork6;
public class Server implements NetworkElement {
    private int price;
    private int capacity;
    public Server(int cap, int price) {
        capacity = cap;
        this.price = price;
    }
    public void accept(Visitor v){
        v.visitServer(this);
    }
    public int getPrice(){
        return price;
    }
}

```

ProcessingStation.java:

```

package com.lab111.labwork6;
public class ProcessingStation implements NetworkElement {
    private int price;
    private int capacity;
    public ProcessingStation(int cap, int price) {

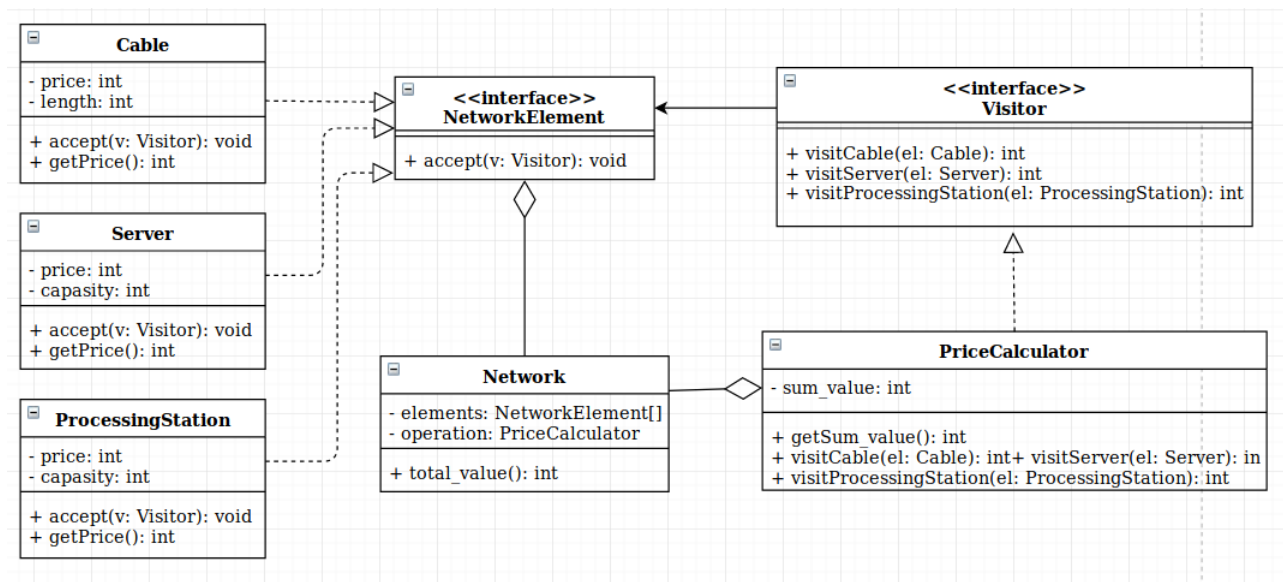
```

```

        capacity = cap;
        this.price = price;
    }
    public void accept(Visitor v){
        v.visitProcessingStation(this);
    }
    public int getPrice(){
        return price;
    }
}

```

Діаграма класів



Тестування програми

```

Run: Main x
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin
Total network value: 3730

Process finished with exit code 0

```

Висновок

Під час виконання лабораторної роботи мною було вивчено шаблони проектування Chain of Responsibility, Strategy та Visitor: для виконання даного мені варіанту було використано шаблон Visitor, в якому додаткові операції над об'єктами можна додавати без зміни їх класів.