Министерство образования Республики Беларусь

Учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет компьютерного проектирования

Дисциплина: СиТАиРИС

**ЛАБОРАТОРНАЯ РАБОТА № 5**

**«Паттерны проектирования»**

Студент гр. 814301

Чарыев С.

Проверил:

Артем .Л

Минск 2020

**Цель:** знакомство и освоение практических навыков использования паттернов проектирования.

**Индивидуальное задание**: Магазин украшений. Используемые паттерны: Абстрактная фабрика, Singleton.

**Исходный текст XML-документов**

Shop.xml

<?xml version="1.0"?>

<shop>

<Shop>

 <name>Zara</name>

 <city>Minsk</city>

 <number>1</number>

 </Shop>

 <Shop>

 <name>7karat</name>

 <city>Minsk</city>

<number>2</number>

</Shop>

</shop>

Products.xml

<products>

 <product>

<productname>koltso</productname>

 <price>900</price>

 <material>zoloto</ material >

  <shop>1</shop>

 </product>

 <product>

 <productname>chepochka</productname>

 <price>600</price>

 < material >silver</ material >

  <shop>2</shop>

 </product>

  <product>

 <productname>ojerelye</productname>

 <price>210</price>

  <shop>1</shop>

 </product>

 <product>

 <productname>koltso</productname>

 <price>325</price>

  <shop>2</shop>

 </product>

 <product>

 <productname>sergi</productname>

 <price>70</price>

  <shop>2</shop>

 </product>

 </products>

**Реализация абстрактной фабрики**

Абстрактный класс Product

public abstract class Product {

    String name;

    String price;

    String shop;

    String material;

    public Product(String name, String price, String shop)

    {

        this.name=name;

        this.price=price;

        this.shop=shop;

        Singleton s=Singleton.getInstance();

        s.addCount();

        s.addPrice(price);}

    public String getName() {

        return name;

    }

    public String getPrice() {

        return price;

    }

    public String getShop() {

        return shop;

    }

    public abstract void isMax(String price, String name);

    public abstract void expensiveProduct();

}

Продукт Socks

public class Socks extends Product {

    String size;

public Socks (String name, String price, String size, String shop)

{

    super(name,price,shop);

    this.size=size;

}

    @Override

    public String toString() {

        return " Socks {" +

                "size='" + size + '\'' +

                ", name='" + name + '\'' +

                ", price='" + price + '\'' +

                ", material='" + material + '\'' +

                ", shop='" + shop + '\'' +

                '}';

}}

Мост:

public interface Interface {

    void printMax();

    void printProducts();

    void countProducts();

    void totalPrice();

}

Компоновщик:

public class Information implements Interface {

   private Compositor com;

    private CompositorProducts compr;

    public Information (Compositor com, CompositorProducts compr) {

        this.com = com;

        this.compr = compr;

    }

    public void printProducts()

{…}

public void printMax()

{…}

public void countProducts()

{…}

public void totalPrice()

{…}

Результат выполнения программы:

Shop{name='Zara', city='Minsk', number='1'}

    Koltso{name='Lakky', price='900', shop='1'}

    Chepochka{name='God', price='210', shop='1'}

Shop {name='7karat', city='Minsk', number='2'}

    Ojerelye{name='Golden Boy', price='370', shop='2'}

    Koltso{name='Good job', price='25', shop='2'}

Zara count = 2

7karat count = 2

Zara

     total koltso price = 900

     total chepochka price = 210

7karat

     total koltso price = 25

     total ojerelye price = 370