Facade Pattern

Facade Pattern казва, че „просто предоставяте унифициран и опростен интерфейс на набор от интерфейси в подсистема, следователно той крие сложността на подсистемата от клиента".

С други думи, Facade Pattern описва интерфейс от по-високо ниво, който улеснява използването на подсистемата.

Практически всяка Abstract Factory е вид Facade.

Предимства на Facade Pattern:

- Той предпазва клиентите от сложността на компонентите на подсистемата.
- Той насърчава свободното свързване между подсистемите и техните клиенти.

Използване на Facade Pattern:

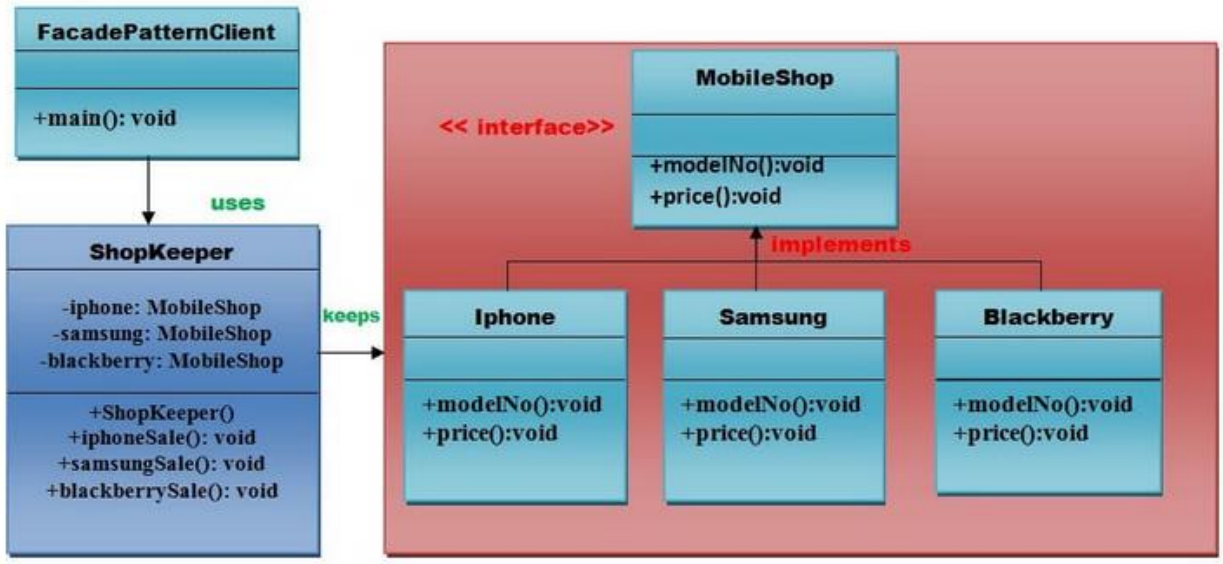
Той се използва:

- Когато искате да осигурите опростен интерфейс на сложна подсистема.
- Когато съществуват няколко зависимости между клиенти и класовете за изпълнение на абстракция.

Пример за Facade Pattern

Нека разгледаме и разберем примера за facade design pattern чрез следната UML диаграма.

UML за Facade Pattern:



Имплементиране на горния UML:

Стъпка 1

Създайте интерфейс MobileShop.

Файл: MobileShop.java

MobileShop.java

```
1
2  public interface MobileShop {
3
4
5      public void modelNo();
6
7      public void price();
8
9  }
10
```

Стъпка 2

Създайте имплементиран клас Iphone, който ще имплементира интерфейса Mobileshop.

Файл: Iphone.java

MobileShop.java | Iphone.java

```java

public class Iphone implements MobileShop {

    @Override
    public void modelNo() {
        System.out.println(" Iphone 6 ");

    }

    @Override
    public void price() {
       System.out.println(" Rs 65000.00 ");

    }

}

```

Стъпка 3

Създайте имплементиран клас Samsung, който ще имплементира интерфейса Mobileshop.

Файл: Samsung.java

Samsung.java

```java

public class Samsung implements MobileShop {

    @Override
    public void modelNo() {
        System.out.println(" Samsung galaxy tab 3 ");

    }

    @Override
    public void price() {
         System.out.println(" Rs 45000.00 ");

    }

}
```

Стъпка 4

Създайте имплементиран клас Blackberry, който ще имплементира интерфейса Mobileshop.

Файл: Blackberry.java

Blackberry.java

```
1
2  public class Blackberry implements MobileShop {
3
4      @Override
5      public void modelNo() {
6          System.out.println(" Blackberry Z10 ");
7
8      }
9
10     @Override
11     public void price() {
12         System.out.println(" Rs 55000.00 ");
13
14     }
15
16 }
```

Стъпка 5

Създайте конкретен клас ShopKeeper, който ще използва интерфейса MobileShop.

Файл: ShopKeeper.java

ShopKeeper.java

```
1
2  public class ShopKeeper {
3
4
5      private MobileShop iphone;
6      private MobileShop samsung;
7      private MobileShop blackberry;
8
9
10     public ShopKeeper(){
11
12         iphone= new Iphone();
13
14         samsung=new Samsung();
15
16         blackberry=new Blackberry();
17
18     }
19
20
21
22     public void iphoneSale(){
23
24         iphone.modelNo();
25         iphone.price();
26
27     }
28
29     public void samsungSale(){
30
31         samsung.modelNo();
32         samsung.price();
33
34     }
35
36
37    public void blackberrySale(){
38
39     blackberry.modelNo();
40     blackberry.price();
41
42 }
43
44
45 }
46
```

Стъпка 6

Сега, създайте клиент, който може да закупи мобилните телефони от MobileShop чрез ShopKeeper.

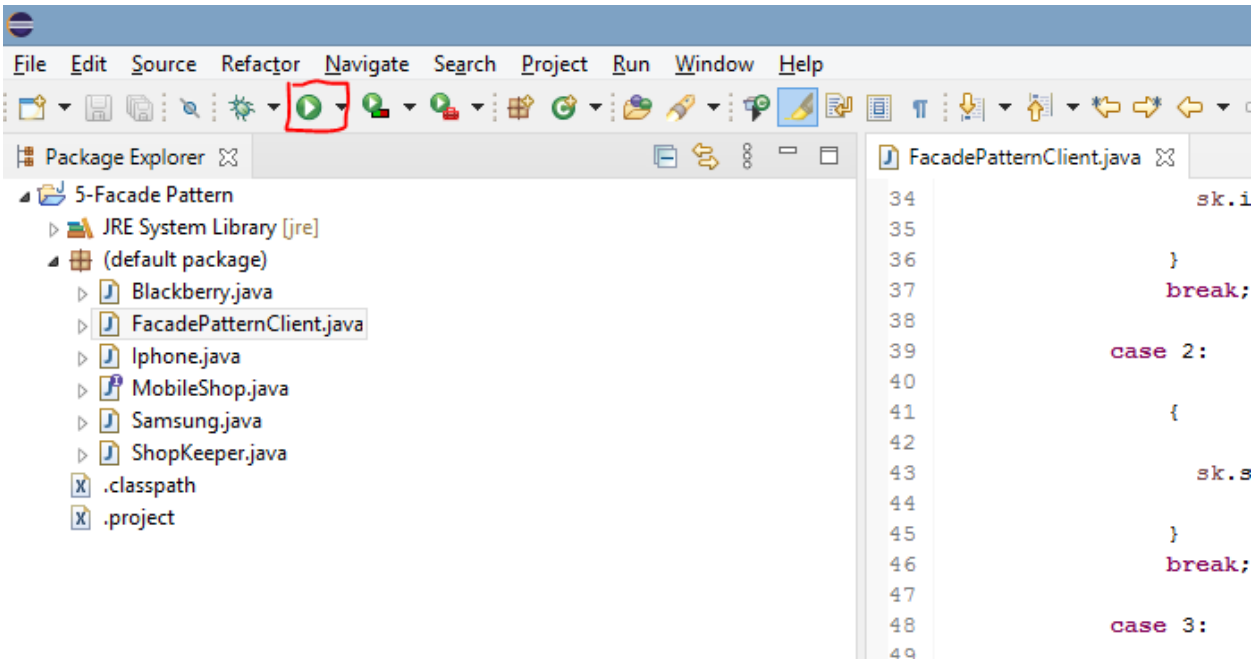
Файл: FacadePatternClient.java

FacadePatternClient.java

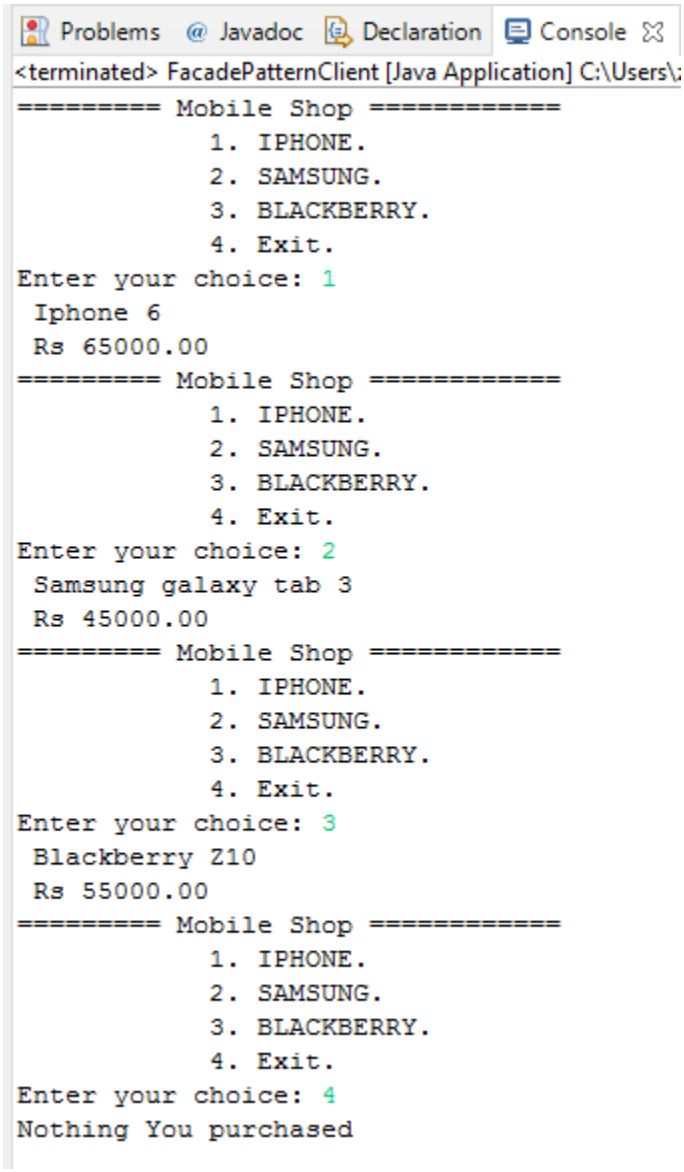
```
import java.io.BufferedReader;


public class FacadePatternClient {

    private static int  choice;



    public static void main(String args[]) throws NumberFormatException, IOException{


        do{
            System.out.print("========= Mobile Shop ============ \n");
            System.out.print("            1. IPHONE.              \n");
            System.out.print("            2. SAMSUNG.              \n");
            System.out.print("            3. BLACKBERRY.            \n");
            System.out.print("            4. Exit.                    \n");
            System.out.print("Enter your choice: ");

            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

            choice=Integer.parseInt(br.readLine());

            ShopKeeper sk=new ShopKeeper();

            switch (choice) {
            case 1:

                {

                  sk.iphoneSale();

                }
                break;

            case 2:

                {

                  sk.samsungSale();

                }
                break;

```

```
47
48              case 3:
49
50
51                  {
52
53
54                   sk.blackberrySale();
55
56                  }
57                 break;
58
59
60              default:
61              {
62                  System.out.println("Nothing You purchased");
63
64              }
65                  return;
66              }
67
68         }while(choice!=4);
69
70     }
71
72
73
74
75 }
```

ТЕСТ И РЕЗУЛТАТ



РЕЗУЛТАТ


Problems
Javadoc
Declaration
Console
<terminated> FacadePatternClient [Java Application] C:\Users\
========= Mobile Shop ============
1. IPHONE.
2. SAMSUNG.
3. BLACKBERRY.
4. Exit.
Enter your choice: 1
Iphone 6
Rs 65000.00
========= Mobile Shop ============
1. IPHONE.
2. SAMSUNG.
3. BLACKBERRY.
4. Exit.
Enter your choice: 2
Samsung galaxy tab 3
Rs 45000.00
========= Mobile Shop ============
1. IPHONE.
2. SAMSUNG.
3. BLACKBERRY.
4. Exit.
Enter your choice: 3
Blackberry Z10
Rs 55000.00
========= Mobile Shop ============
1. IPHONE.
2. SAMSUNG.
3. BLACKBERRY.
4. Exit.
Enter your choice: 4
Nothing You purchased