Facade Pattern

A Facade Pattern says that just **"just provide a unified and simplified interface to a set of interfaces in a subsystem, therefore it hides the complexities of the subsystem from the client".**

In other words, Facade Pattern describes a higher-level interface that makes the sub-system easier to use.

Practically, **every Abstract Factory** is a type of **Facade.**

Advantage of Facade Pattern

- It shields the clients from the complexities of the sub-system components.
- It promotes loose coupling between subsystems and its clients.

Usage of Facade Pattern:

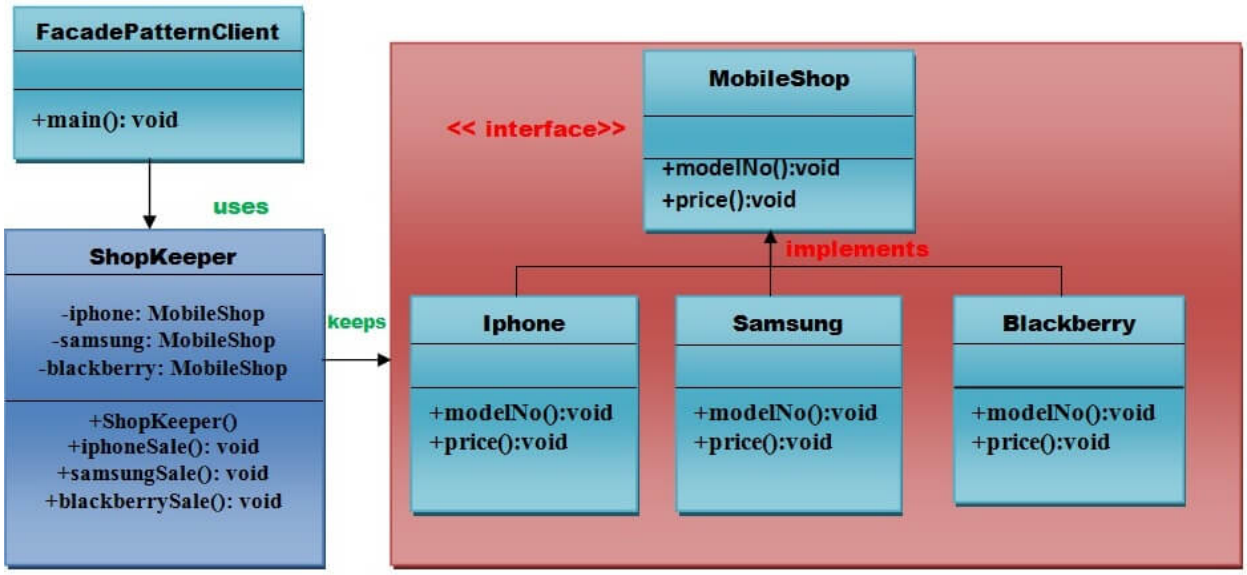
It is used:

- When you want to provide simple interface to a complex sub-system.
- When several dependencies exist between clients and the implementation classes of an abstraction.

Example of Facade Pattern

Let's understand the example of facade design pattern by the above UML diagram.

UML for Facade Pattern:



Implementation of above UML:

Step 1

Create a **MobileShop** interface.

File: MobileShop.java

```
public interface MobileShop {
   public void modelNo();
   public void price();
}
```

Step 2

Create a **Iphone** implementation class that will implement **Mobileshop** interface.

File: Iphone.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 ");
   }
```

```
}
```

Step 3

Create a **Samsung** implementation class that will implement **Mobileshop** interface.

File: Samsung.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 ");
   }
}
```

Step 4

Create a **Blackberry** implementation class that will implement **Mobileshop** interface .

File: Blackberry.java

```
public class Blackberry implements MobileShop {
   @Override
   public void modelNo() {
   System.out.println(" Blackberry Z10 ");
   }
   @Override
   public void price() {
      System.out.println(" Rs 55000.00 ");
   }
}
```

Step 5

Create a **ShopKeeper** concrete class that will use **MobileShop** interface.

File: ShopKeeper.java

```
```

```
public class ShopKeeper {
    private MobileShop iphone;
    private MobileShop samsung;
    private MobileShop blackberry;

    public ShopKeeper(){
        iphone= new Iphone();
        samsung=new Samsung();
        blackberry=new Blackberry();
    }
    public void iphoneSale(){
        iphone.modelNo();
        iphone.price();
    }
        public void samsungSale(){
        samsung.modelNo();
        samsung.price();
    }
   public void blackberrySale(){
    blackberry.modelNo();
    blackberry.price();
        }
}
```

Step 6

Now, Creating a **client** that can purchase the mobiles from **MobileShop** through **ShopKeeper.**

File: FacadePatternClient.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

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;
      case 3:
                  {
                  sk.blackberrySale();
                  }
            break;
        default:
        {
           System.out.println("Nothing You purchased");
        }
           return;
        }

     }while(choice!=4);
   }
}
```

download this example

Output

```
========= 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
```

<<prev next>>