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.

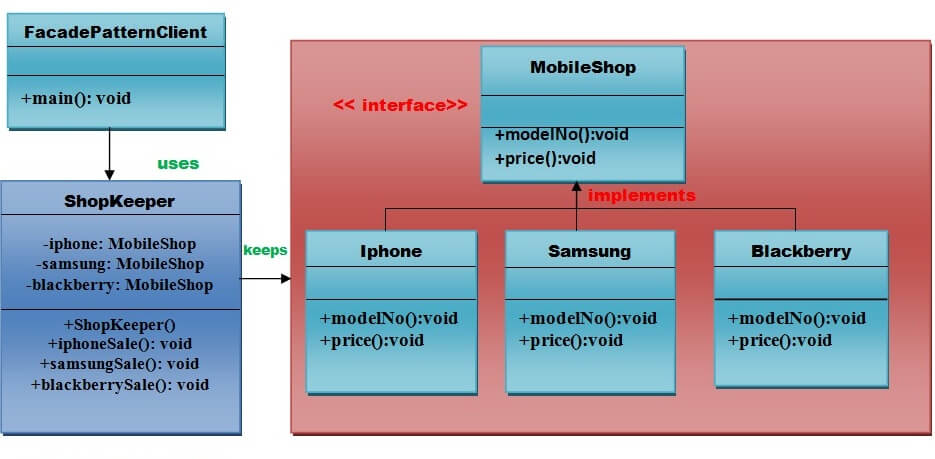
Usage of Facade Pattern:

* 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*

1. **public** **interface** MobileShop {
2. **public** **void** modelNo();
3. **public** **void** price();
4. }

Step 2

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

*File: Iphone.java*

1. **public** **class** Iphone **implements** MobileShop {
2. @Override
3. **public** **void** modelNo() {
4. System.out.println(" Iphone 6 ");
5. }
6. @Override
7. **public** **void** price() {
8. System.out.println(" Rs 65000.00 ");
9. }
10. }

Step 3

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

*File: Samsung.java*

1. **public** **class** Samsung **implements** MobileShop {
2. @Override
3. **public** **void** modelNo() {
4. System.out.println(" Samsung galaxy tab 3 ");
5. }
6. @Override
7. **public** **void** price() {
8. System.out.println(" Rs 45000.00 ");
9. }
10. }

Step 4

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

*File: Blackberry.java*

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

Step 5

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

*File: ShopKeeper.java*

1. **public** **class** ShopKeeper {
2. **private** MobileShop iphone;
3. **private** MobileShop samsung;
4. **private** MobileShop blackberry;
6. **public** ShopKeeper(){
7. iphone= **new** Iphone();
8. samsung=**new** Samsung();
9. blackberry=**new** Blackberry();
10. }
11. **public** **void** iphoneSale(){
12. iphone.modelNo();
13. iphone.price();
14. }
15. **public** **void** samsungSale(){
16. samsung.modelNo();
17. samsung.price();
18. }
19. **public** **void** blackberrySale(){
20. blackberry.modelNo();
21. blackberry.price();
22. }
23. }

Step 6

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

*File: FacadePatternClient.java*

1. **import** java.io.BufferedReader;
2. **import** java.io.IOException;
3. **import** java.io.InputStreamReader;
5. **public** **class** FacadePatternClient {
6. **private** **static** **int**  choice;
7. **public** **static** **void** main(String args[]) **throws** NumberFormatException, IOException{
8. **do**{
9. System.out.print("========= Mobile Shop ============ \n");
10. System.out.print("            1. IPHONE.              \n");
11. System.out.print("            2. SAMSUNG.              \n");
12. System.out.print("            3. BLACKBERRY.            \n");
13. System.out.print("            4. Exit.                     \n");
14. System.out.print("Enter your choice: ");
16. BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.in));
17. choice=Integer.parseInt(br.readLine());
18. ShopKeeper sk=**new** ShopKeeper();
20. **switch** (choice) {
21. **case** 1:
22. {
23. sk.iphoneSale();
24. }
25. **break**;
26. **case** 2:
27. {
28. sk.samsungSale();
29. }
30. **break**;
31. **case** 3:
32. {
33. sk.blackberrySale();
34. }
35. **break**;
36. **default**:
37. {
38. System.out.println("Nothing You purchased");
39. }
40. **return**;
41. }
43. }**while**(choice!=4);
44. }
45. }

[download this example](https://www.javatpoint.com/designpattern/designpatternexample/facadepattern.zip)

Output

1. ========= Mobile Shop ============
2. 1. IPHONE.
3. 2. SAMSUNG.
4. 3. BLACKBERRY.
5. 4. Exit.
6. Enter your choice: 1
7. Iphone 6
8. Rs 65000.00
9. ========= Mobile Shop ============
10. 1. IPHONE.
11. 2. SAMSUNG.
12. 3. BLACKBERRY.
13. 4. Exit.
14. Enter your choice: 2
15. Samsung galaxy tab 3
16. Rs 45000.00
17. ========= Mobile Shop ============
18. 1. IPHONE.
19. 2. SAMSUNG.
20. 3. BLACKBERRY.
21. 4. Exit.
22. Enter your choice: 3
23. Blackberry Z10
24. Rs 55000.00
25. ========= Mobile Shop ============
26. 1. IPHONE.
27. 2. SAMSUNG.
28. 3. BLACKBERRY.
29. 4. Exit.
30. Enter your choice: 4
31. Nothing You purchased