Façade Design Pattern

Façade Design Pattern is one of the **Structural** design patterns. Façade design pattern is used to give client an easily interactive systems so that client applications can easily interact with the system. Façade patterns hide the complexities of the system and provide an interface to the client through which client can access the system. This pattern involves a single class which provides methods for the client and calls to the method of existing system classes.

According to GoF, Façade design pattern provide a unified interface to a set of interfaces in a system. Façade design pattern defines a higher-level interface that makes the system easier to use.

Why use Façade design pattern?

When the complexity increases or the interface behavior names become confusing, client application will find it difficult to navigate it. So we apply Façade design pattern and provide a interface on the top of the interface to help client application

Implementation

We are going to create a *Shape* interface and concrete classes implementing the *Shape* interface. A facade class *ShapeMaker* is defined as a next step.

ShapeMaker class uses the concrete classes to delegate user calls to these classes. *FacadePatternDemo*, our demo class, will use *ShapeMaker* class to show the results.

Shape.java(<<interface>>)

```
public interface Shape {

   void draw();

}
```

[Square.java] and [Circle.java] to implement same interface

```
public class Square implements Shape {
   @Override
   public void draw() {
      System.out.println("Square::draw()");
   }
}


public class Circle implements Shape {
   @Override
   public void draw() {
      System.out.println("Circle::draw()");
   }
}
```

ShapeMaker.java[Façade class]

```
public class ShapeMaker {
   private Shape circle;
   private Shape square;
   public ShapeMaker() {
      circle = new Circle();
      square = new Square();
   }
   public void drawCircle(){
```

```
circle.draw();
   }
   public void drawSquare(){
      square.draw();
   }
}
```

FacadePatternDemo.java[to draw different types of shape]

```
public class FacadePatternDemo {
   public static void main(String[] args) {
      ShapeMaker shapeMaker = new ShapeMaker();

      shapeMaker.drawCircle();
      shapeMaker.drawSquare();
   }
}
```