

## Motifs de conception - Motif de façade

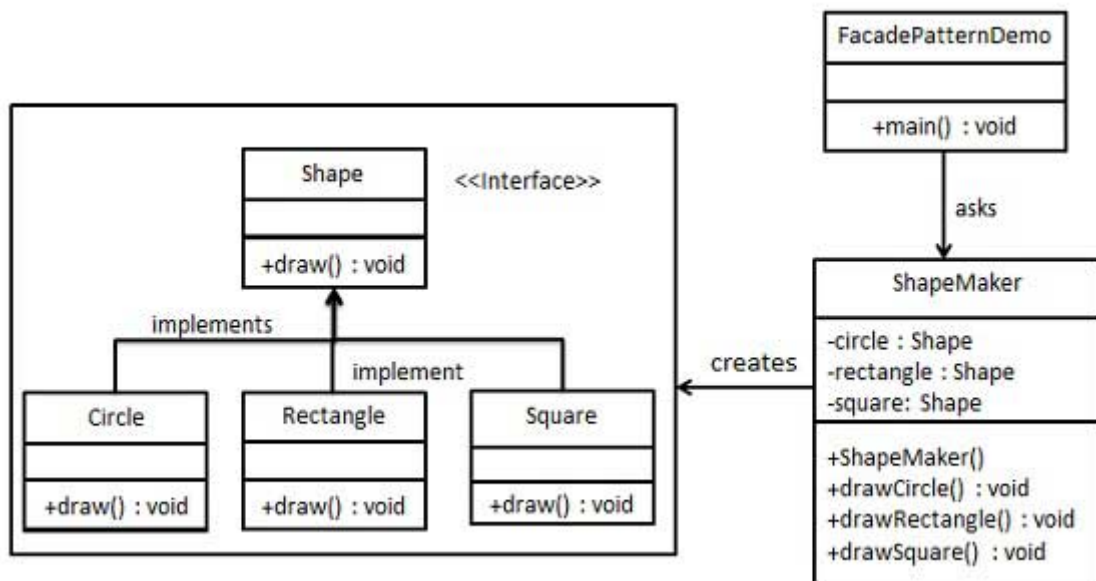
Le modèle de façade masque les complexités du système et fournit une interface au client à l'aide de laquelle le client peut accéder au système. Ce type de modèle de conception relève du modèle structurel car ce modèle ajoute une interface au système existant pour masquer ses complexités.

Ce modèle implique une seule classe qui fournit des méthodes simplifiées requises par le client et délègue les appels aux méthodes des classes système existantes.

### la mise en oeuvre

Nous allons créer une interface *Shape* et des classes concrètes implémentant l'interface *Shape*. Une classe de façade *ShapeMaker* est définie comme une étape suivante.

La classe *ShapeMaker* utilise les classes concrètes pour déléguer les appels des utilisateurs à ces classes. *FacadePatternDemo*, notre classe de démonstration, utilisera la classe *ShapeMaker* pour afficher les résultats.



### Étape 1

Créez une interface.

*Shape.java*

```

public interface Shape {
    void draw();
}
  
```

### Étape 2

Créez des classes concrètes implémentant la même interface.

### *Rectangle.java*

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

### *Square.java*

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

### *Circle.java*

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

## Étape 3

Créez une classe de façade.

### *ShapeMaker.java*

```
public class ShapeMaker {  
    private Shape circle;  
    private Shape rectangle;  
    private Shape square;  
  
    public ShapeMaker() {  
        circle = new Circle();  
        rectangle = new Rectangle();  
        square = new Square();  
    }  
  
    public void drawCircle(){  
        circle.draw();  
    }  
    public void drawRectangle(){  
        rectangle.draw();  
    }  
    public void drawSquare(){
```

```
        square.draw();  
    }  
}
```

## Étape 4

Utilisez la façade pour dessiner différents types de formes.

*FacadePatternDemo.java*

```
public class FacadePatternDemo {  
    public static void main(String[] args) {  
        ShapeMaker shapeMaker = new ShapeMaker();  
  
        shapeMaker.drawCircle();  
        shapeMaker.drawRectangle();  
        shapeMaker.drawSquare();  
    }  
}
```

## Étape 5

Vérifiez la sortie.

```
Circle::draw()  
Rectangle::draw()  
Square::draw()
```