

# Facade Pattern

Lecture 15

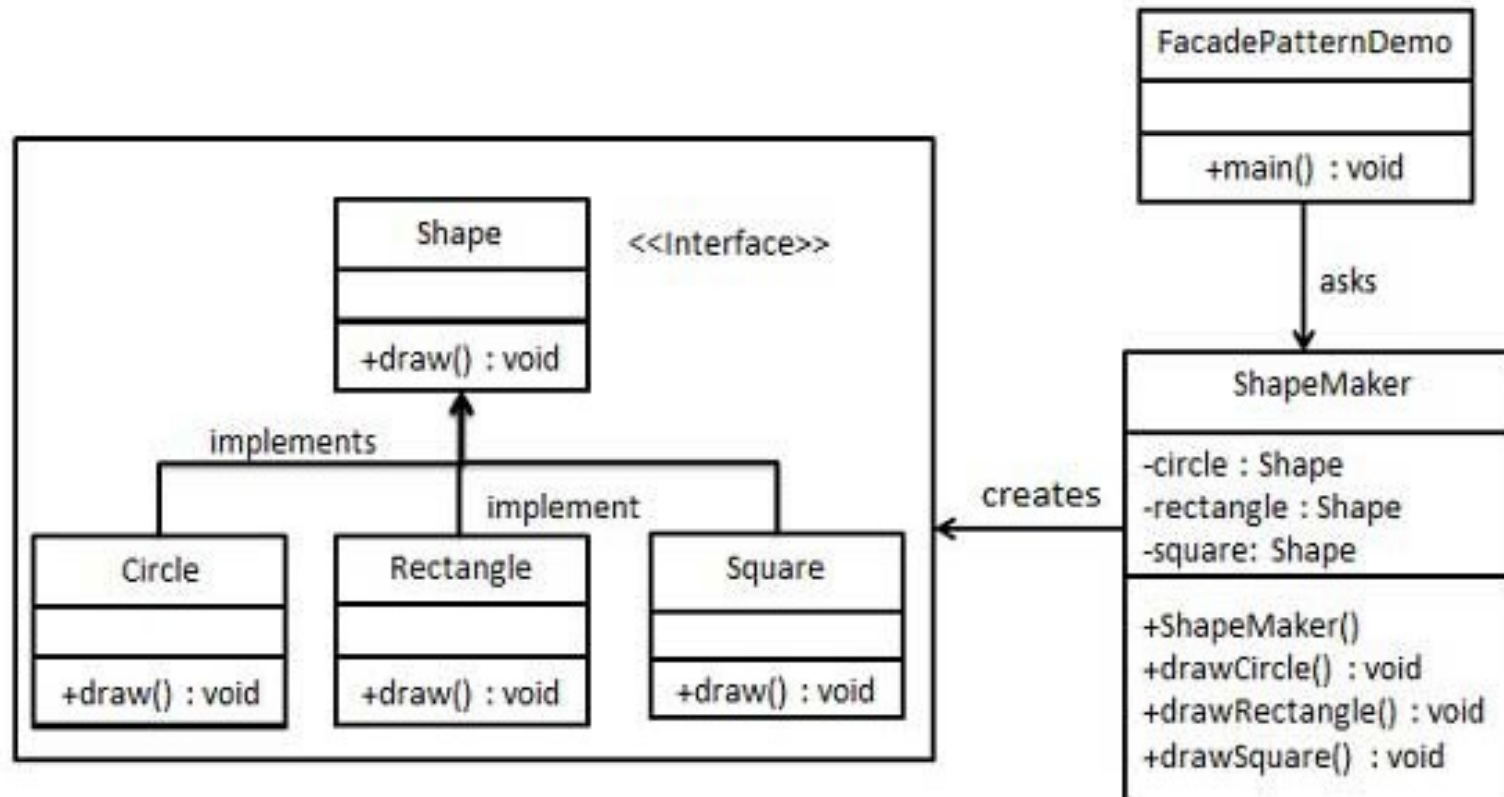
# Facade

- “Provide a unified interface to a set of interfaces in a collection of subsystems.
- Facade defines a higher-level interface that makes the subsystems easier to use.”

# Motivation

- “Common design goal is to minimize the communication and dependencies between subsystems. One way to achieve this goal is to introduce a facade object that provides a single, simplified interface to the more general facilities of a subsystem.”
- With façade clients don't have to be concerned with exactly which object in a subsystem they're dealing with. They just call methods on the facade in blissful ignorance.

# Implementation



## Step 1

Create an interface.

*Shape.java*

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

## Step 2

Create concrete classes implementing the same 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()");  
    }  
}
```

```
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();  
    }  
}
```



Use the facade to draw various types of shapes.

*FacadePatternDemo.java*

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

## Step 5

Verify the output.

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

---

# References

- [https://www.tutorialspoint.com/design\\_pattern/facade\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/facade_pattern.htm)