

Aula Prática - 1 Padrões de Criação - Factory Method

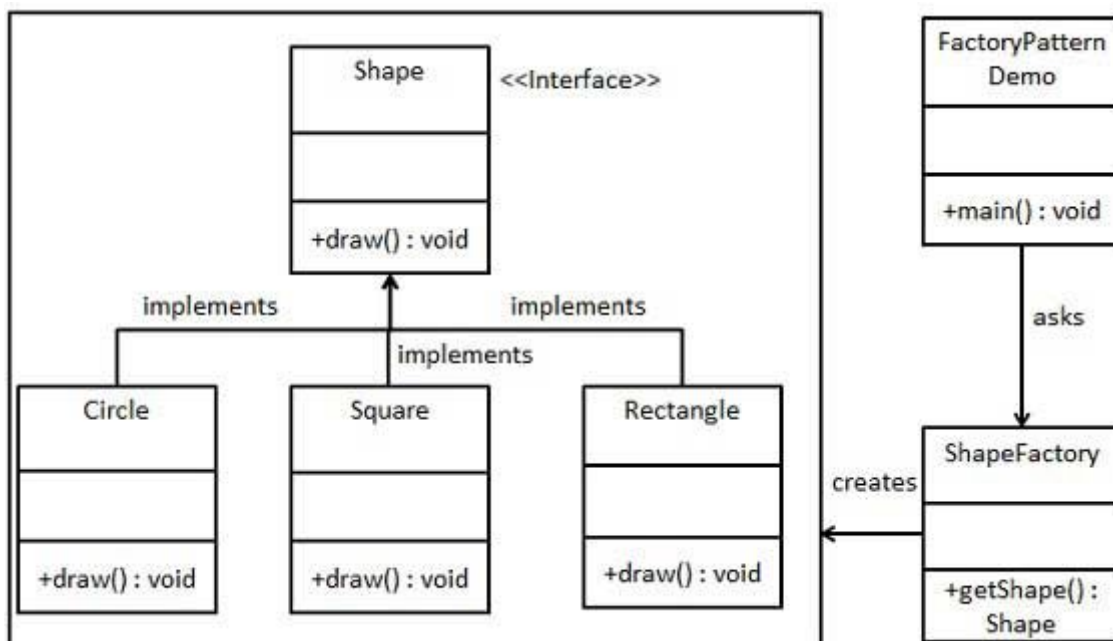
O padrão de fábrica é um dos padrões de design mais usados em Java. Esse tipo de padrão de design está em padrão de criação, pois esse padrão fornece uma das melhores maneiras de criar um objeto.

Em Padrão de fábrica, criamos objeto sem expor a lógica de criação ao cliente e nos referimos ao objeto recém-criado usando uma interface comum.

Implementação

Nós vamos criar uma interface Shape e classes concretas implementando a interface Shape. Uma classe de fábrica ShapeFactory é definida como uma próxima etapa.

FactoryPatternDemo, a classe de demonstração usará ShapeFactory para obter um objeto Shape. Ele irá passar informações (CIRCLE / RECTANGLE / SQUARE) para ShapeFactory para obter o tipo de objeto que precisa.



Passo 1

Crie uma interface.

Shape.java

```

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

Passo 2

Crie classes concretas implementando a mesma interface.

Rectangle.java

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

Square.java

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

Circle.java

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

Passo 3

Crie uma Fábrica para gerar um objeto de classe concreto com base em informações dadas.

ShapeFactory.java

```
public class ShapeFactory {  
  
    //use getShape method to get object of type shape  
    public Shape getShape(String shapeType){  
        if(shapeType == null){  
            return null;  
        }  
        if(shapeType.equalsIgnoreCase("CIRCLE")){  
            return new Circle();  
        }  
        else if(shapeType.equalsIgnoreCase("RECTANGLE")){  
            return new Rectangle();  
        }  
        else if(shapeType.equalsIgnoreCase("SQUARE")){  
            return new Square();  
        }  
        return null;  
    }  
}
```

Passo 4

Use a Fábrica para obter o objeto da classe concreta passando uma informação como o tipo.

FactoryPatternDemo.java

```
public class FactoryPatternDemo {  
  
    public static void main(String[] args) {  
        ShapeFactory shapeFactory = new ShapeFactory();  
  
        //get an object of Circle and call its draw method.  
        Shape shape1 = shapeFactory.getShape("CIRCLE");  
  
        //call draw method of Circle  
        shape1.draw();  
  
        //get an object of Rectangle and call its draw method.  
        Shape shape2 = shapeFactory.getShape("RECTANGLE");  
  
        //call draw method of Rectangle  
        shape2.draw();  
  
        //get an object of Square and call its draw method.  
        Shape shape3 = shapeFactory.getShape("SQUARE");  
  
        //call draw method of circle  
        shape3.draw();  
    }  
}
```

Passo 5

Teste sua implementação!