Modèle de conception - Modèle d'usine

Page précédente Page suivante

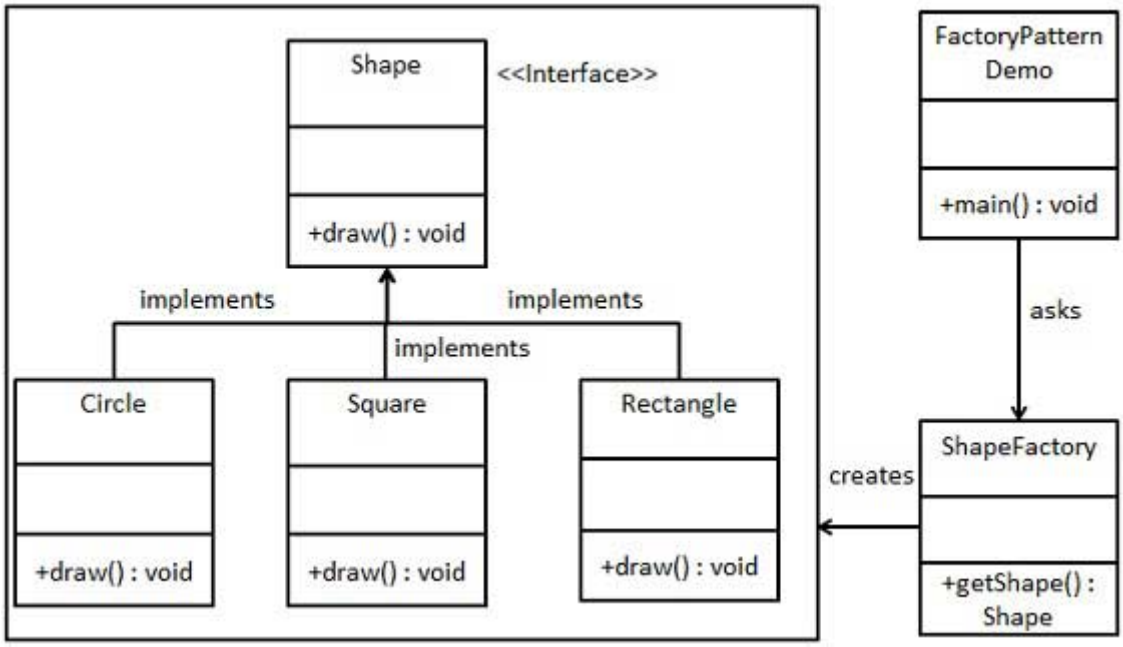
Le modèle d'usine est l'un des modèles de conception les plus utilisés en Java. Ce type de modèle de conception est inclus dans le modèle de création car il constitue l'un des meilleurs moyens de créer un objet.

Dans le modèle Factory, nous créons un objet sans exposer la logique de création au client et faisons référence au nouvel objet créé à l'aide d'une interface commune.

la mise en oeuvre

Nous allons créer une interface *Shape* et des classes concrètes implémentant l' interface *Shape* . Une classe d'usine *ShapeFactory* est définie à l'étape suivante.

FactoryPatternDemo , notre classe de démonstration utilisera *ShapeFactory* pour obtenir un objet *Shape* . Il transmettra les informations (*CIRCLE / RECTANGLE / SQUARE*) à *ShapeFactory* pour obtenir le type d'objet dont il a besoin.



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("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.");
   }
}
```

Étape 3

Créez une fabrique pour générer un objet de classe concrète en fonction d'informations données.

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;
   }
}
```

Étape 4

Utilisez Factory pour obtenir un objet de classe concrète en transmettant des informations telles que le type.

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 square
      shape3.draw();
   }
}
```

Étape 5

Vérifiez la sortie.

```
Inside Circle::draw() method.
Inside Rectangle::draw() method.
Inside Square::draw() method.
```

Page précédente Page suivante