



Factory Design Pattern

By: Kaden Ciesielski, Ivan Zivlak

What Is Factory Design?

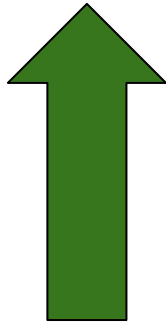
- Creation of objects
- Define an abstract class or interface
- Superclass with multiple sub-classes
- The name “factory”



Pros & Cons

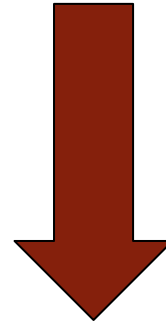
Pros:

- Reusability
- Flexibility
- Centralization
- Decoupling



Cons:

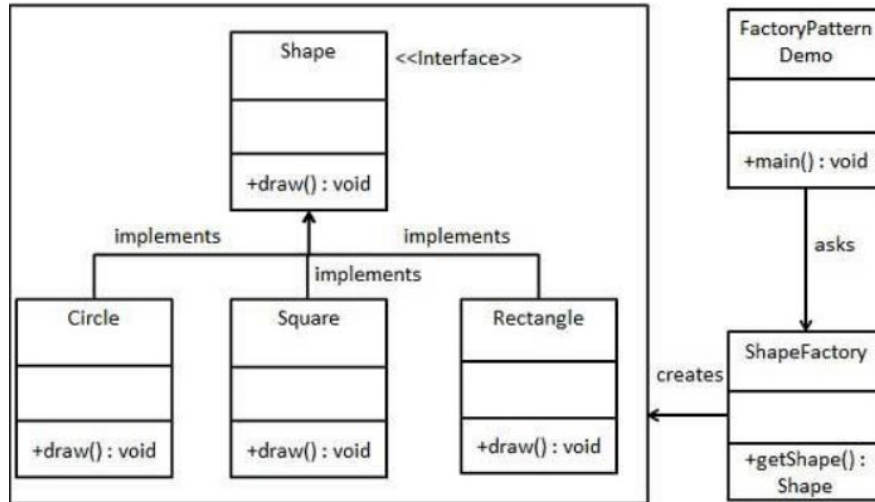
- Complexity
- Increased code
- Maintenance



What's It Used For?

- Dynamic Handling of GUI
- GUI Components
- Database Connection
- Object Creation in Video Games
- Message Handling

Factory Design In Practice



Interface

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

Subclasses

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

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

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

Factory

```
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;

    }

}
```

```
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();

    }

}
```


Works Cited

<https://www.digitalocean.com/community/tutorials/factory-design-pattern-in-java>

<https://www.geeksforgeeks.org/factory-method-design-pattern-in-java/>

<https://home.csulb.edu/~pnguyen/cecs277/lecnotes/factory.pdf>