

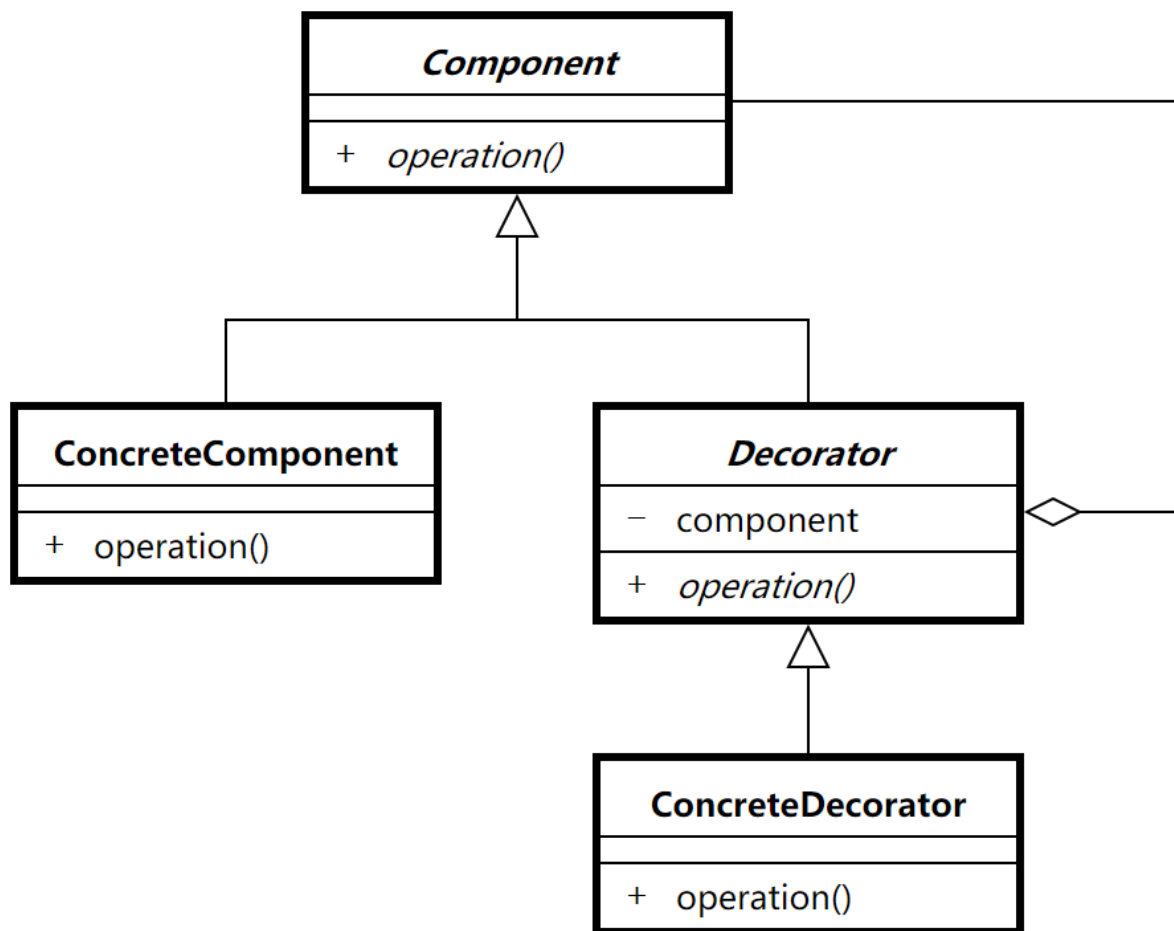
# 装饰者模式

Decorator

郭嘉

模式定义：

在不改变原有对象的基础上，将功能附加到对象上



## 案例

```
1 package com.tuling.designpattern.decorator;
2
3 /**
4  * @author 腾讯课堂-图灵学院 郭嘉
5  * @Slogan 致敬大师，致敬未来的你
6  */
7 public class DecoratorTest {
8     public static void main(String[] args) {
9         Component component= new ConcreteDecorator1( new ConcreteDecorator2(new ConcreteComponent() ) );
10        component.operation();
11    }
12 }
13
```

```
14 interface Component{
15     void operation();
16 }
17 class ConcreteComponent implements Component{
18
19     @Override
20     public void operation() {
21         System.out.println("拍照.");
22     }
23 }
24 abstract class Decorator implements Component{
25     Component component;
26
27     public Decorator(Component component) {
28         this.component=component;
29     }
30 }
31
32 class ConreteDecorator1 extends Decorator{
33
34     public ConreteDecorator1(Component component) {
35         super( component );
36     }
37
38     @Override
39     public void operation() {
40         System.out.println("添加美颜.");
41         component.operation();
42     }
43 }
44
45 class ConreteDecorator2 extends Decorator{
46
47
48     public ConreteDecorator2(Component component) {
49         super( component );
50     }
51
52     @Override
53     public void operation() {
54         System.out.println("添加滤镜.");
```

```
55  component.operation();  
56  }  
57  }  
58
```

应用场景：

扩展一个类的功能或给一个类添加附加职责

优点：

1. 不改变原有对象的情况下给一个对象扩展功能
2. 使用不同的组合可以实现不同的效果
3. 符合开闭原则

经典案例

```
1  Servlet Api:  
2  javax.servlet.http.HttpServletRequestWrapper  
3  javax.servlet.http.HttpServletResponseWrapper
```

