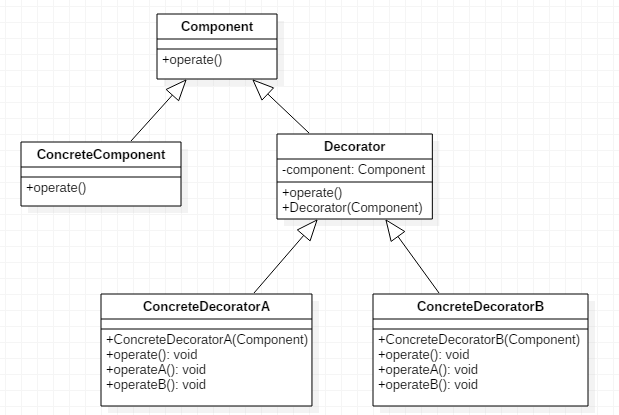
## 装饰模式

1. 定义与使用场景

定义：动态地给一个对象添加一些额外的职责。就新增加的功能来说，装饰模式相比生成子类更为灵活。

使用场景：需要透明且动态地扩展类的功能时。

1. UML类图



Component：抽象组件，可以是一个接口或者抽象类，其充当的就是被装饰的原始对象

ConcreteComponent：组件具体实现类，该类是Component类的基本实现，也是我们装饰的具体对象

Decorator：抽象装饰者，其承担的职责就是为了装饰我们的组件对象，其内部一定要有一个指向组件对象的引用。该类为抽象类，需要根据不同的装饰逻辑实现不同的具体子类。

ConcreteDecorator：装饰者具体实现类，只是对抽象装饰者作出具体的实现

1. 简单实现
2. 源码实战
3. 实战总结