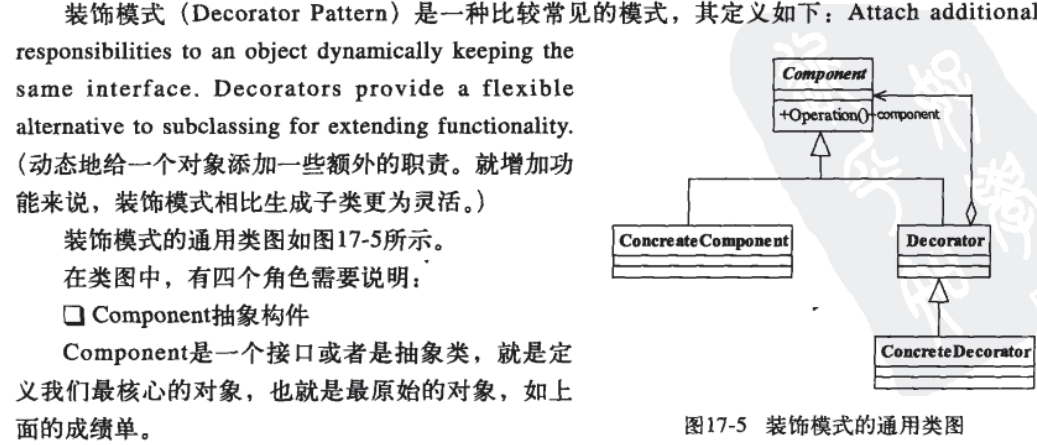
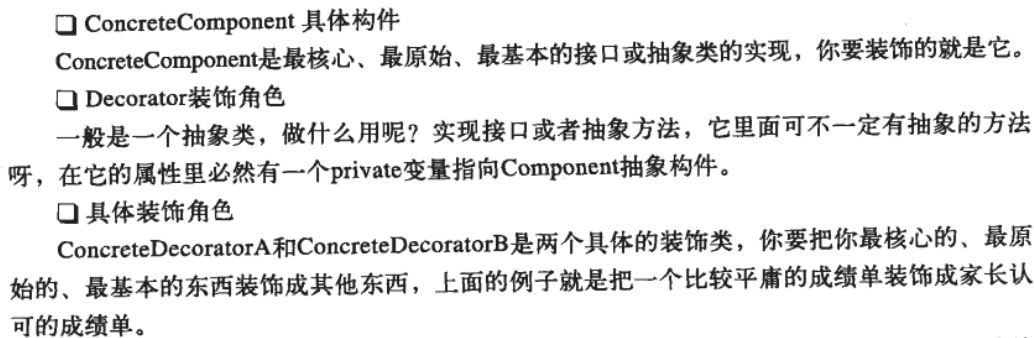
# 装饰模式

1. 定义

Attach additional responsibilities to an object keeping the same interface.Decorators provide a flexible alternative to subclass for extending functionality.( 在保持相同的接口情况下，动态的给一个对象添加一些额外的职责。就增加功能来说，装饰模式比生成子类更为灵活)

1. 类图





1. 模式的应用
2. 模式的优点
3. 装饰类和被装饰类可以独立的发展而不互相耦合。换句话说Component类无需知道Decorator类，Decorator类是从外部来扩展Component类的功能，而Decorator类也不需要知道具体的构建（ConcreteComponent）。
4. 装饰模式是继承关系的一种替代方案。我们看装饰类Decorator，不管装饰多少层，返回的对象还是Component类型，实现的还是is-a的关系。
5. 装饰模式可以动态的扩展一个实现类的功能。
6. 模式的缺点

多层装饰比较复杂。如果最底层的装饰类需要修改，对于后面的装饰类来说影响非常大。因此尽量减少装饰类的数量，以降低系统的复杂度。

1. 实际应用场景
2. 需要扩展一个类的功能或者给一个类增加附加功能；
3. 动态的给一个类增加功能，这些功能还是动态地撤销（Client类中组装时，撤销对应功能的Decorator类就可以）；
4. 需要为一批兄弟类进行改装或加装功能，首选装饰模式。