**装饰者模式（Decorator Pattern）**

**定义**

动态地给一个对象添加一些额外的职责，就增加功能来说，装饰者模式比生成子类更为灵活。——《设计模式：可复用面向对象软件的基础》

装饰者模式是一种对象结构型模式。

**使用场景**

* 在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责。
* 需要动态地给一个对象增加功能，这些功能也可以动态地被撤销。
* 当不能采用继承的方式对系统进行扩充或者采用继承不利于系统扩展和维护时。不能采用继承的情况主要有两类：第一类是系统中存在大量独立的扩展，为支持每一种组合将产生大量的子类，使得子类数目呈爆炸性增长；第二类是因为类定义不能继承（如final类）.

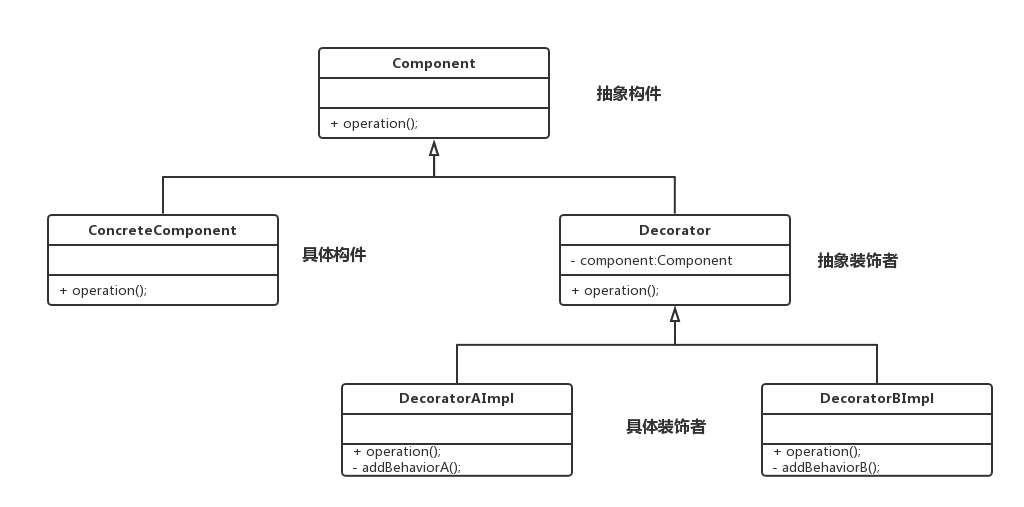
**角色**

抽象构件角色（Component）：定义一个对象接口或抽象类，可以给这些对象动态地添加职责。

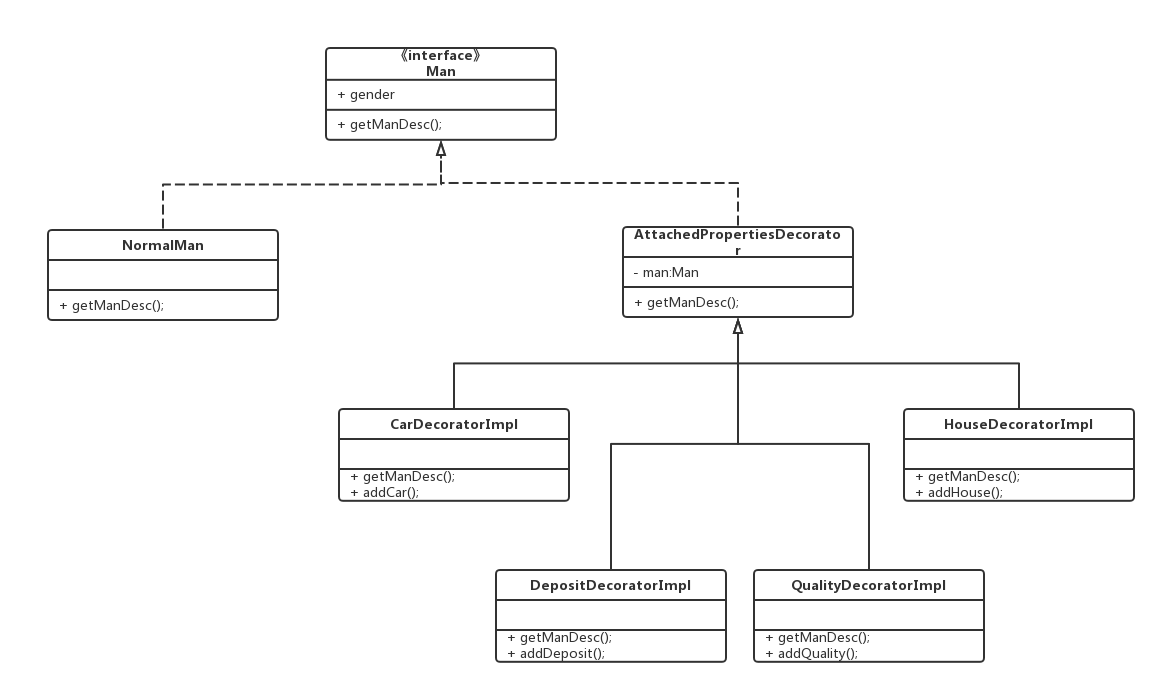
具体构件角色（ConcreteComponent）：实际被动态地添加职责的对象。

抽象装饰者角色（Decorator）：实现了Component接口，用来扩展Component类的功能，但对于Component来说，是无需知道Decorator的存在的。

具体装饰者角色（ConcreteDecorator）：动态地添加职责的对象。

装饰者模式（Decorator）结构图

装饰者模式（Decorator）代码示例类图



来着

<https://www.cnblogs.com/mingmingcome/p/9798248.html>