装饰模式提供了一个非常好的解决方案，它把每个要装饰的功能放在单独的类中，并让这个类包装它所要装饰的对象，因此当需要执行特殊行为时，客户代码就可以在运行时根据需要有选择地、按顺序地使用装饰功能包装对象了。

装饰模式（Decorator）：动态地给一个对象添加一些额外的职责，就增加功能来说，装饰模式比生成子类更为灵活。

装饰模式（Decorator）结构图：



