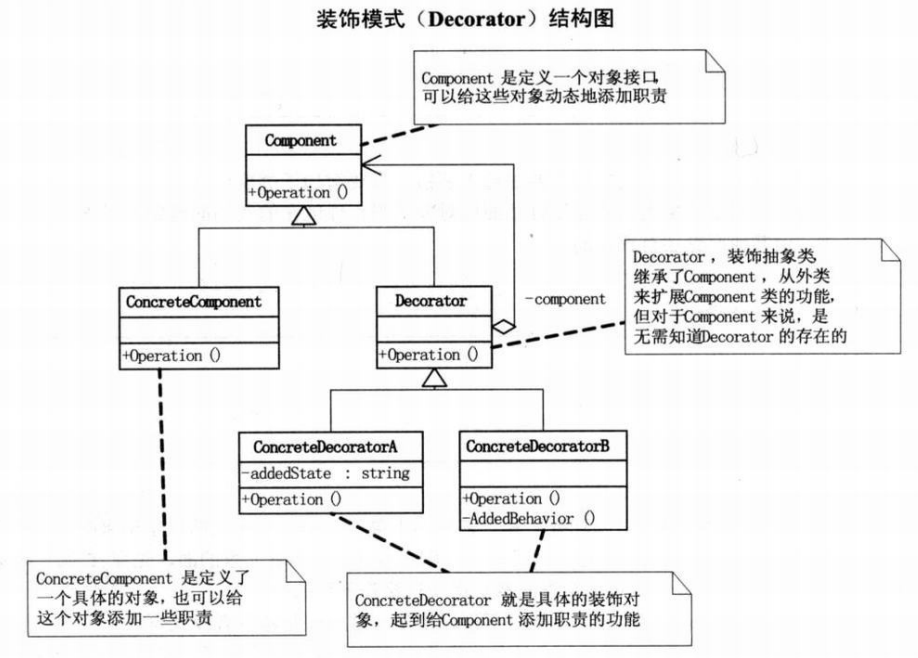
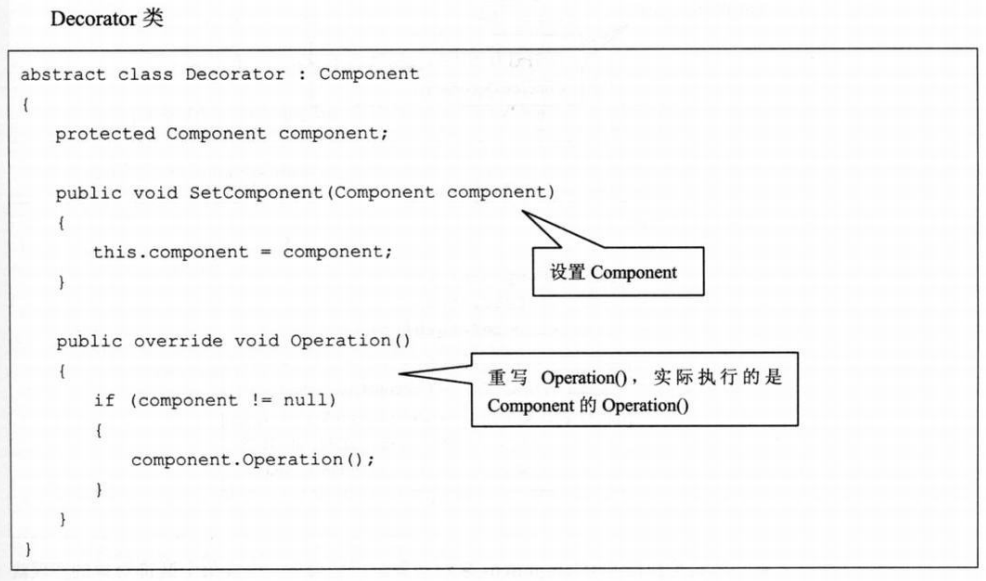
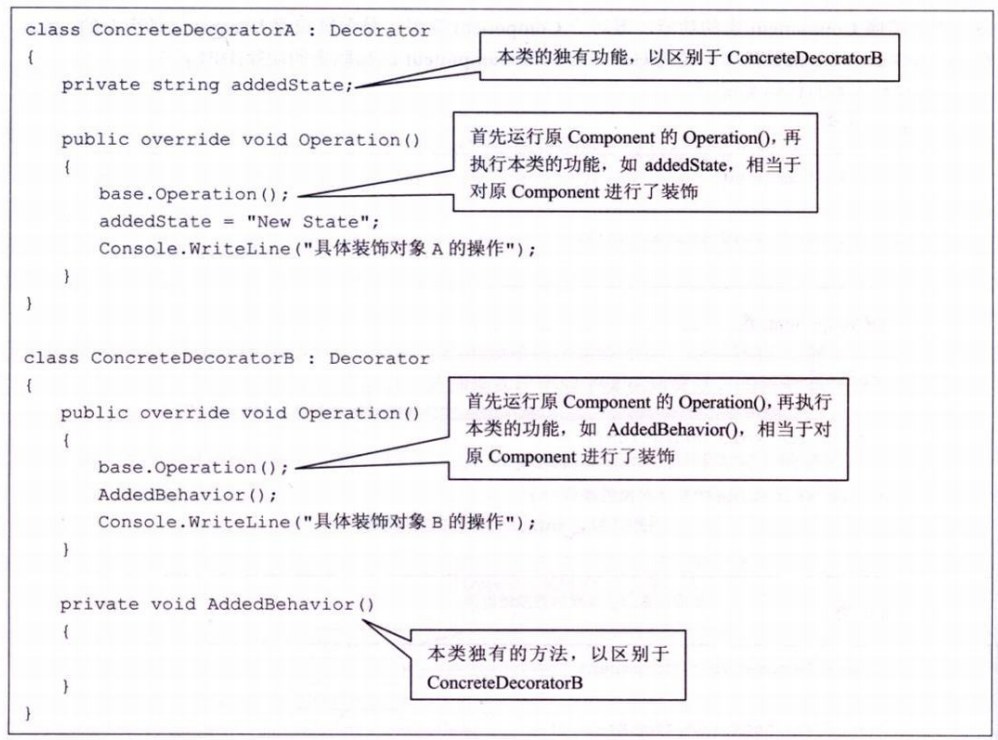
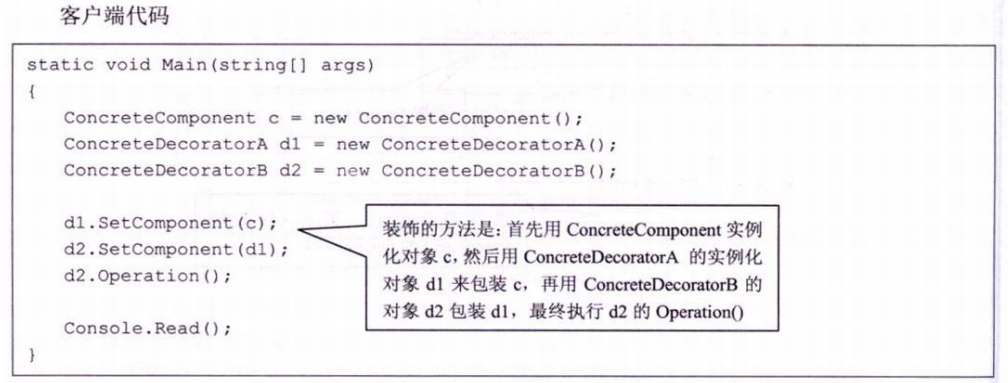
问题：为一个人搭配不同的服饰

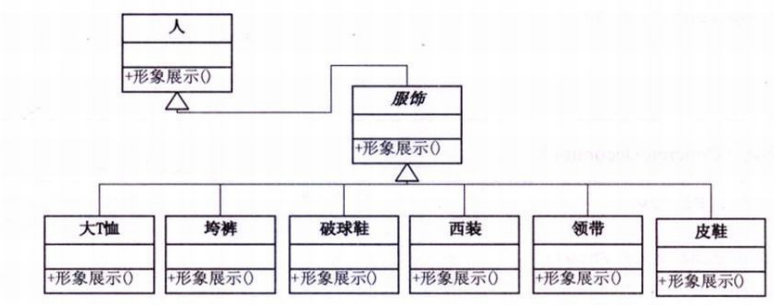
分析：









1. component是一个对象接口，ConcreteComponent是一个具体的对象，Decorator是抽象类，扩展Component的功能，但对于Component来说，是无需知道Decorator的存在。ConcreteDecorator是具体的装饰对象。
2. 如上利用setComponent来对对象进行包装。每个装饰对象只关心自己的功能，不需要关心如何被添加到对象链中。
3. 如果只有一个ConcreteComponent类，没有Component类，Decorator可以使ConcreteComponent的子类。如果只有一个ConcreteDecorator，那么可以将Decorator和ConcreteDecorator合并成一个类。
4. 

在这里是将人作为ConcreteComponent，服饰是Decorator

1. 装饰模式是为已有功能动态地添加更多功能的方式。有时候这些新添的功能仅仅为了满足一些只在某种特定情况下才会执行的特殊行为的需要。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。有效地将类的核心职责和装饰功能区分开。
2. 装饰模式的顺序很重要，例如加密数据和过滤词汇是数据持久化前的装饰功能。最好保证装饰类之间彼此独立。