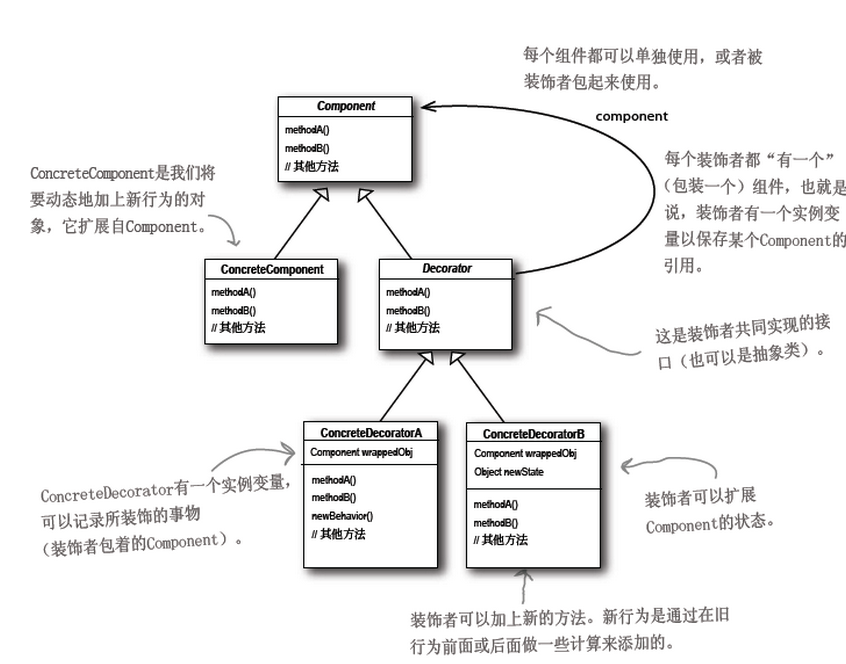
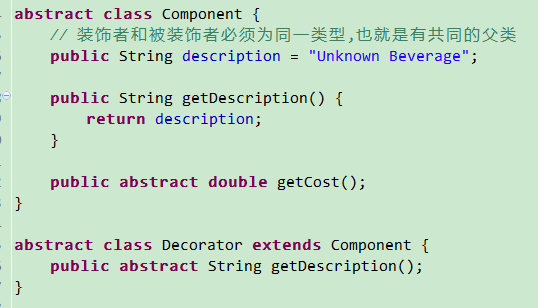
1. 装饰者(Decorator)模式：也叫包装器模式(Wrapper), **动态地将责任附加到对象上**。若要扩展功能，装饰者提供了比继承更有弹性的替代方案(动态的给一个对象添加额外的功能, 比继承更加灵活)
   1. 动态地为一个对象增加新的功能
2. 类图

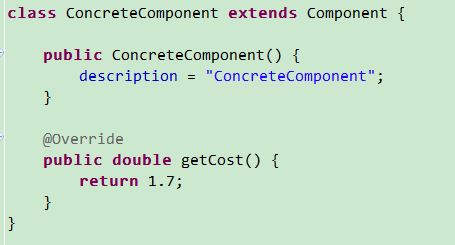


1. 例子：

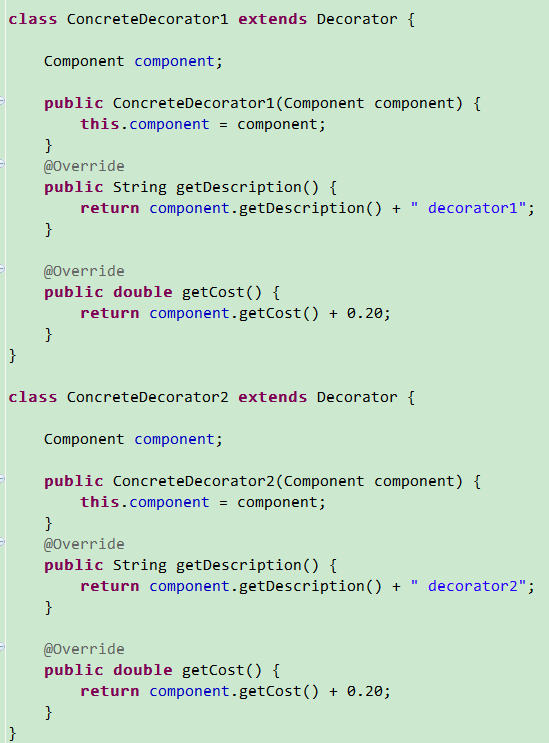
定义抽象基类以及装饰者基类



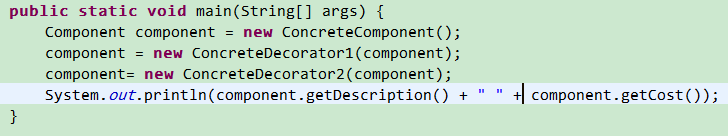
**定义被装饰者**



**定义具体装饰者**



**测试**



**输出**

1. **开发中使用场景**
   1. IO中输入流和输出流的设计
   2. Servlet API中提供了一个Request对象的装饰者设计模式的默认实现类 HttpServletWrapper, 增强了Request对象的功能
   3. Struts2中, Request、Response、Session对象的处理
2. IO流实现细节
   1. Component抽象构件角色:
      1. IO流中的InputStream/OutputStream/Reader/Writer
   2. ConcreteComponent具体构件角色
      1. FileInputStream/FileOutputStream
   3. Decorator装饰角色
      1. 持有抽象构件的引用: IO流中的FileInputStream/FileOutputStream
   4. ConcreteDecorator具体装饰角色
      1. 负责给构件对象增加新的责任. IO流中的BufferedOutputStream/BufferedInputStream等

