# 设计模式之禅

# 迪米特法则

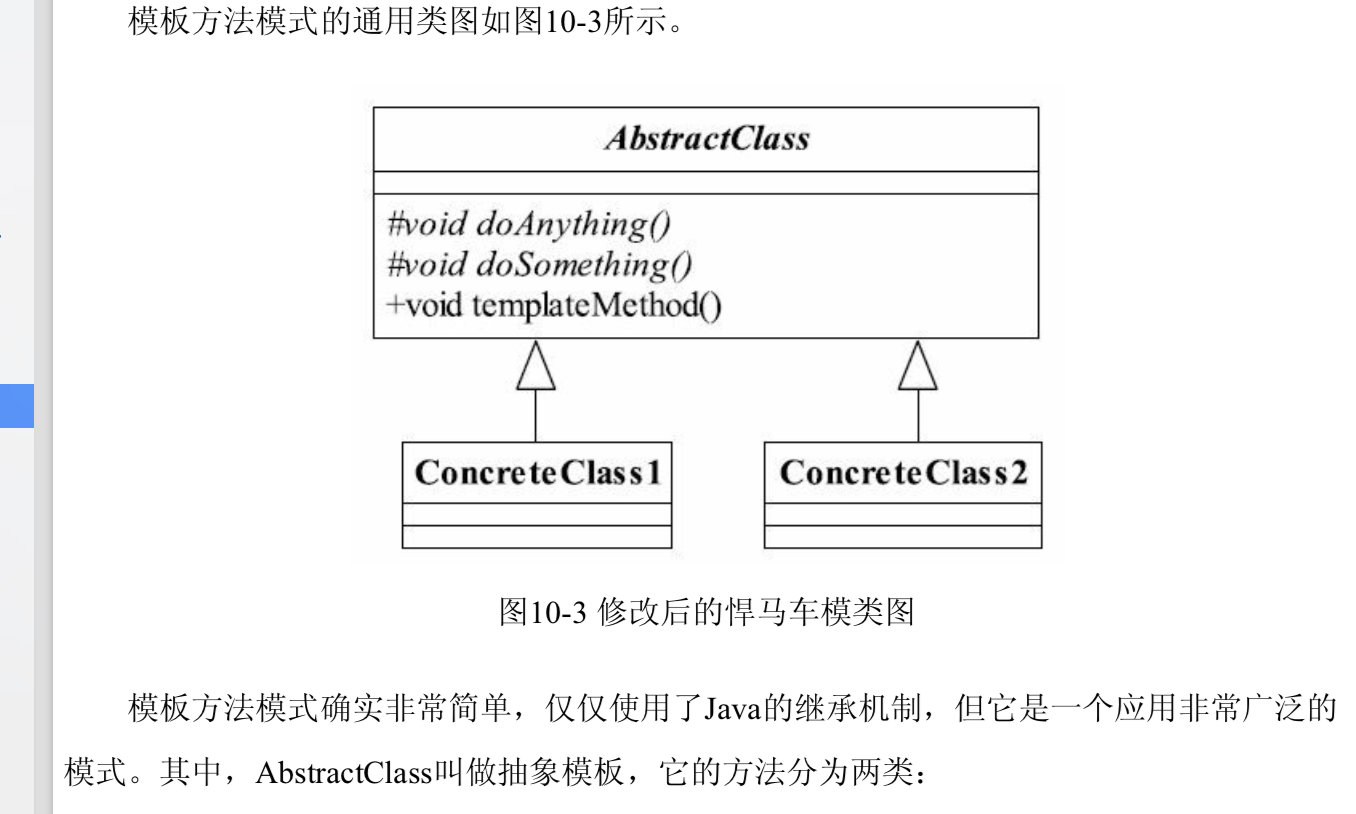
迪米特法则(Law of Demeter,LoD) 也称为最少知识原则(Least Knowledge Principle,LKP),

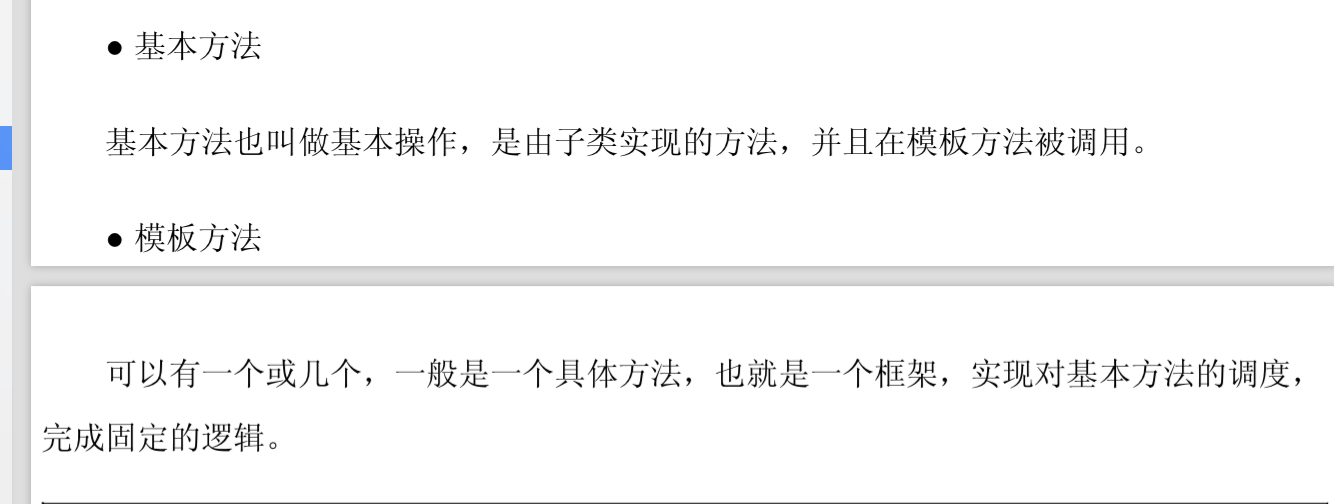
一个对象应该对其他对象有最少的了解。一个类应该对自己需要耦合的或调用的类知道的最少。

# 模板方法

## 10.2 模板方法的定义

定义一个操作中的算法的框架，而将一些步骤延迟到子类中。使得子类可以不改变一个算法的结构即可重定义该算法的某些特定步骤





注意：

1. 为了防止恶意的操作，一般模板方法都加上final关键字，不允许被覆盖
2. 抽象模板中的基本方法尽量设计为protected类型，复合迪米特法则，不需要暴露属性或方法精良不要设置为protected类型。实现类若非必要，尽量不要扩大父类中的访问权限。

## 10.3 模板方法模式的应用

优点：

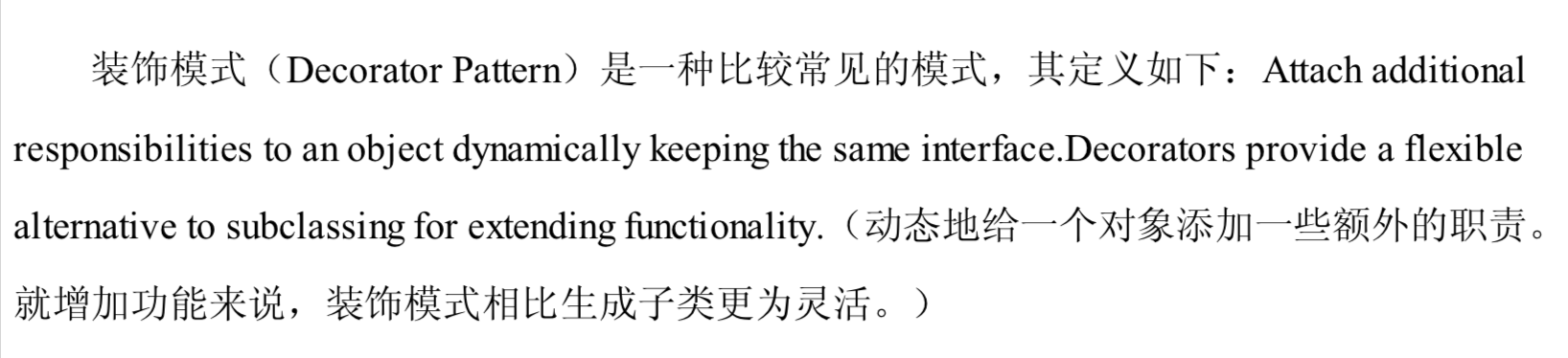
1. 封装不变不封，扩展可变部分
2. 提取公共部分代码，便于维护
3. 行为由父类控制，子类实现

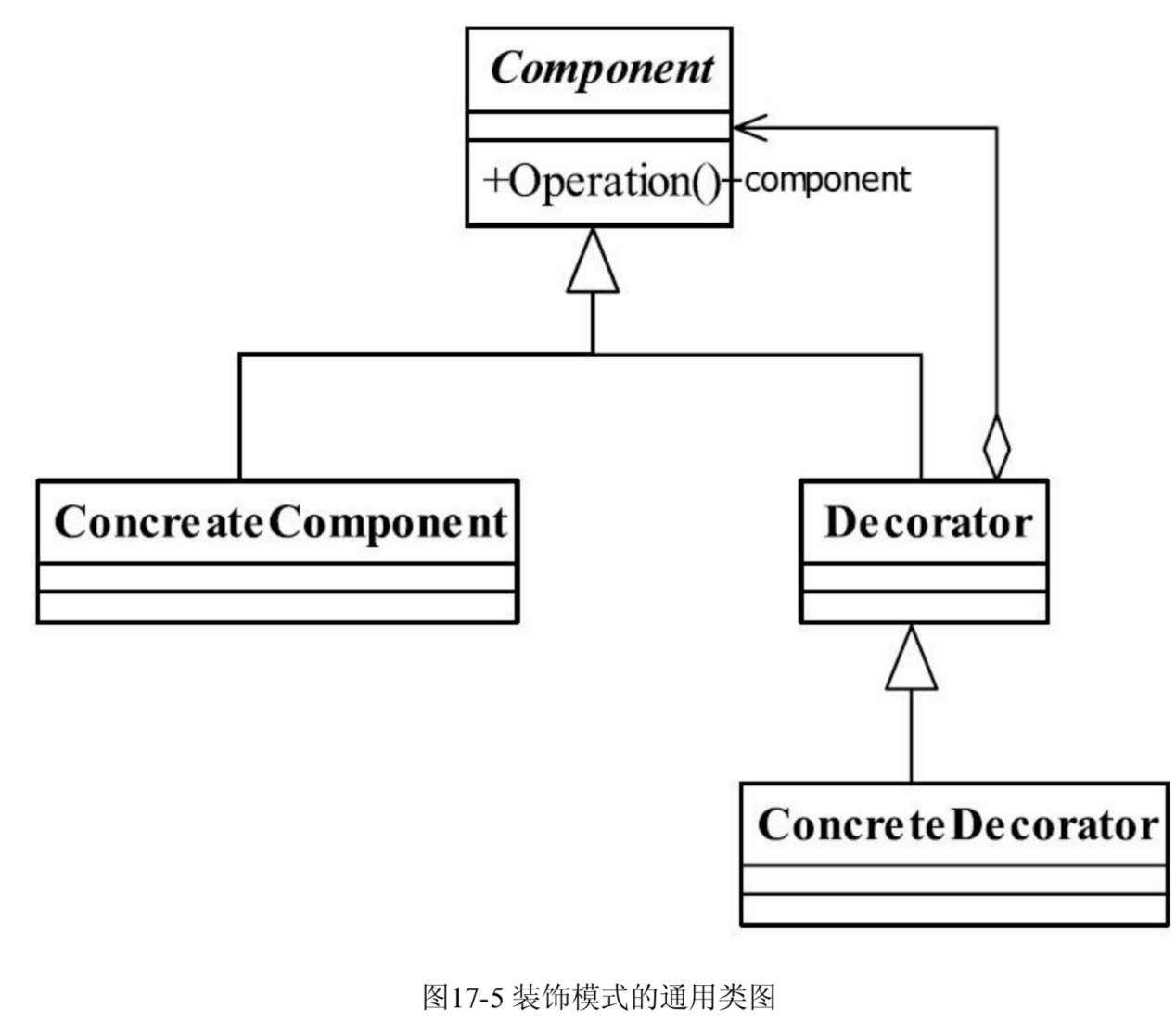
缺点：

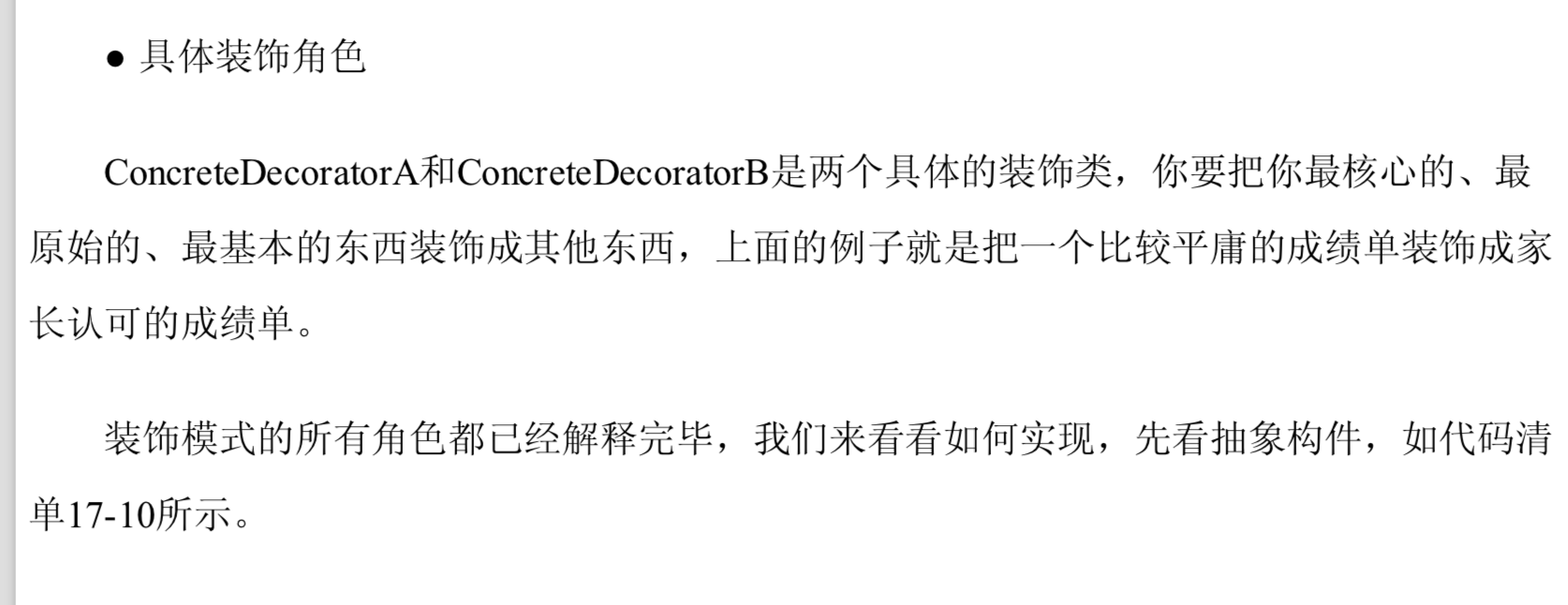
代码阅读起来有难度。

# 装饰者模式

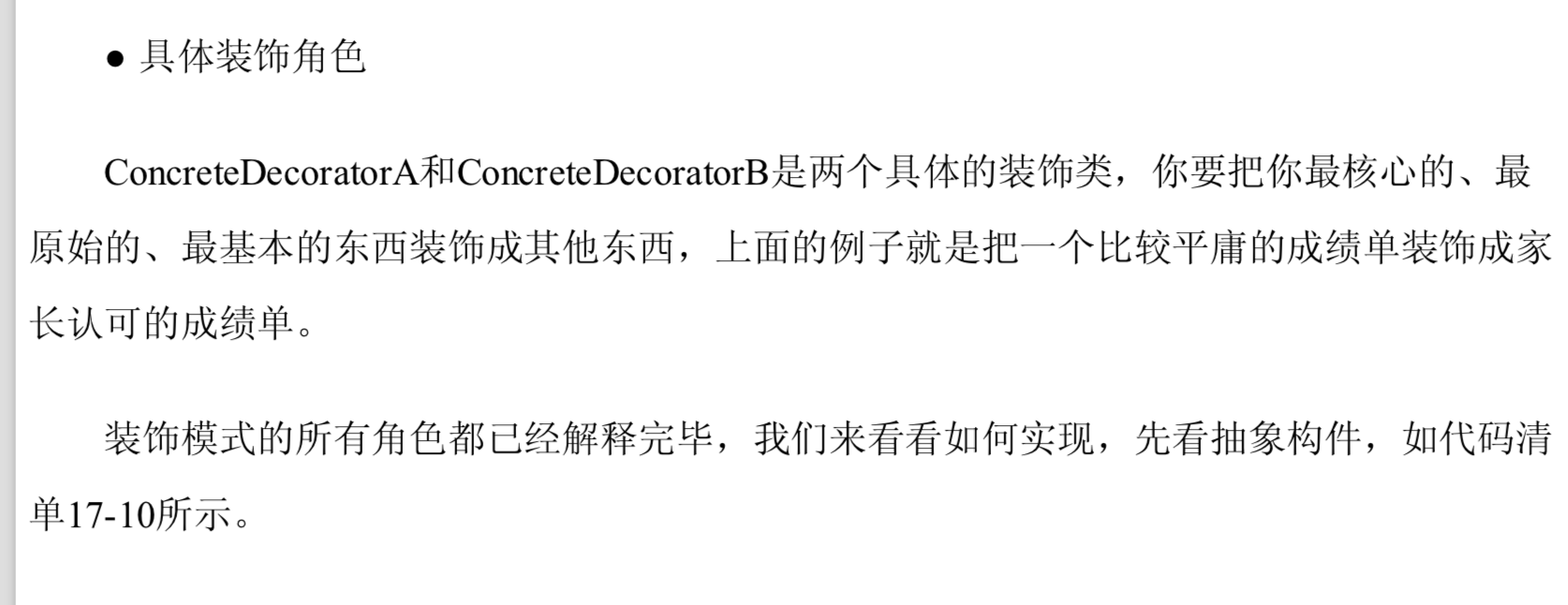
动态地给一个对象添加一些额外的职责

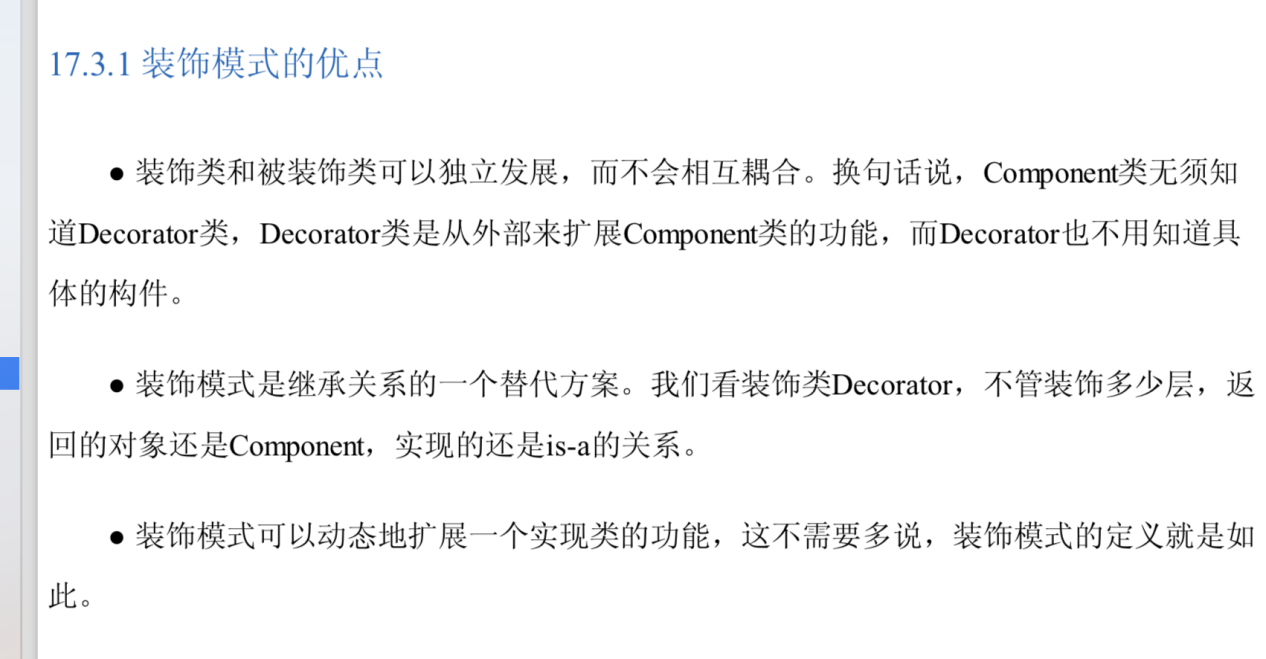


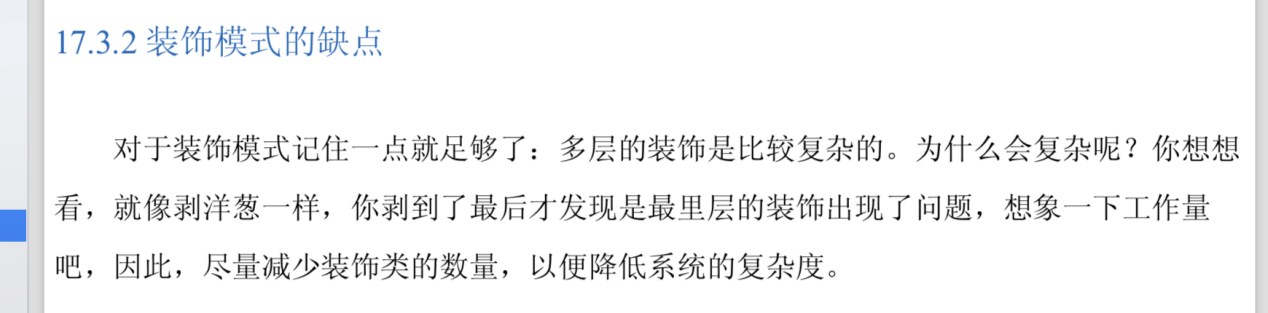


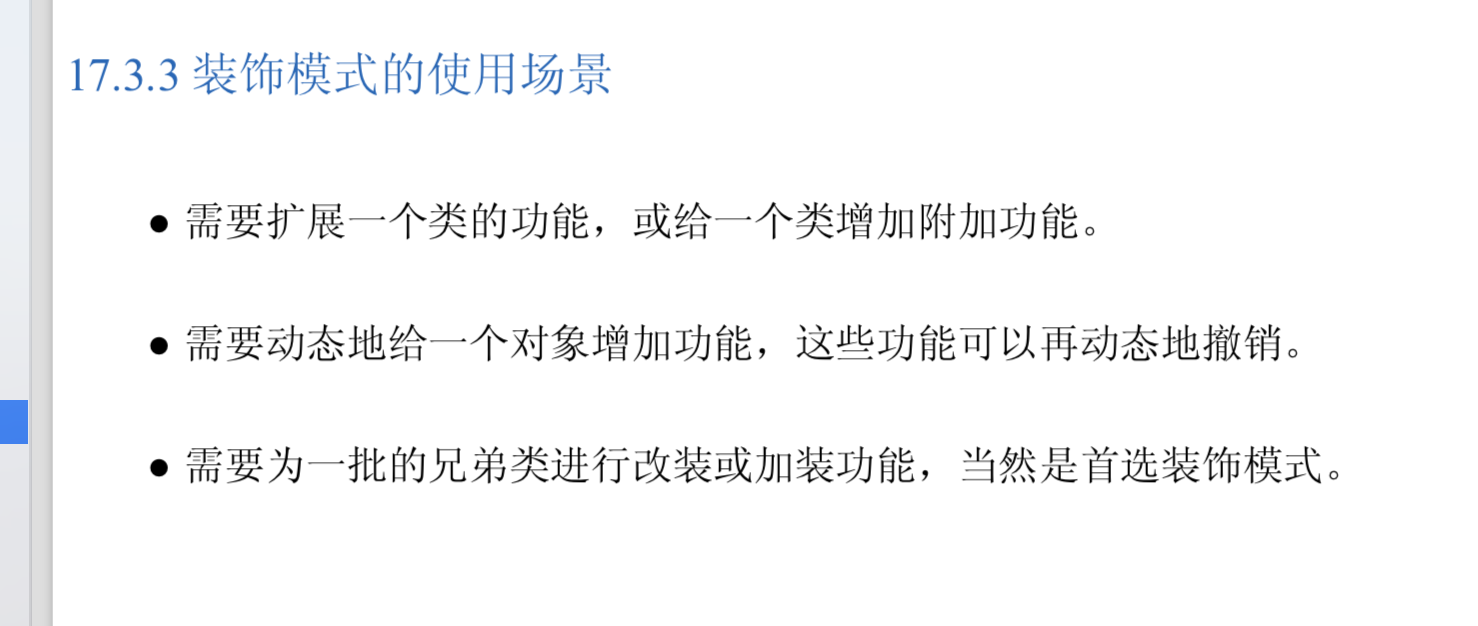




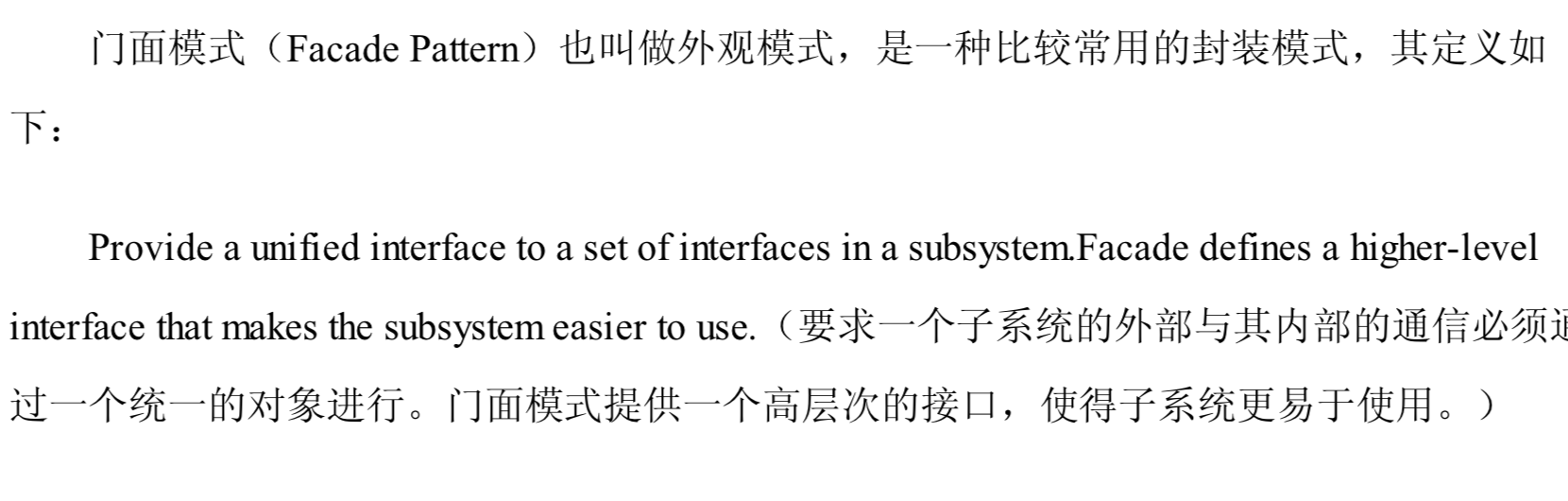


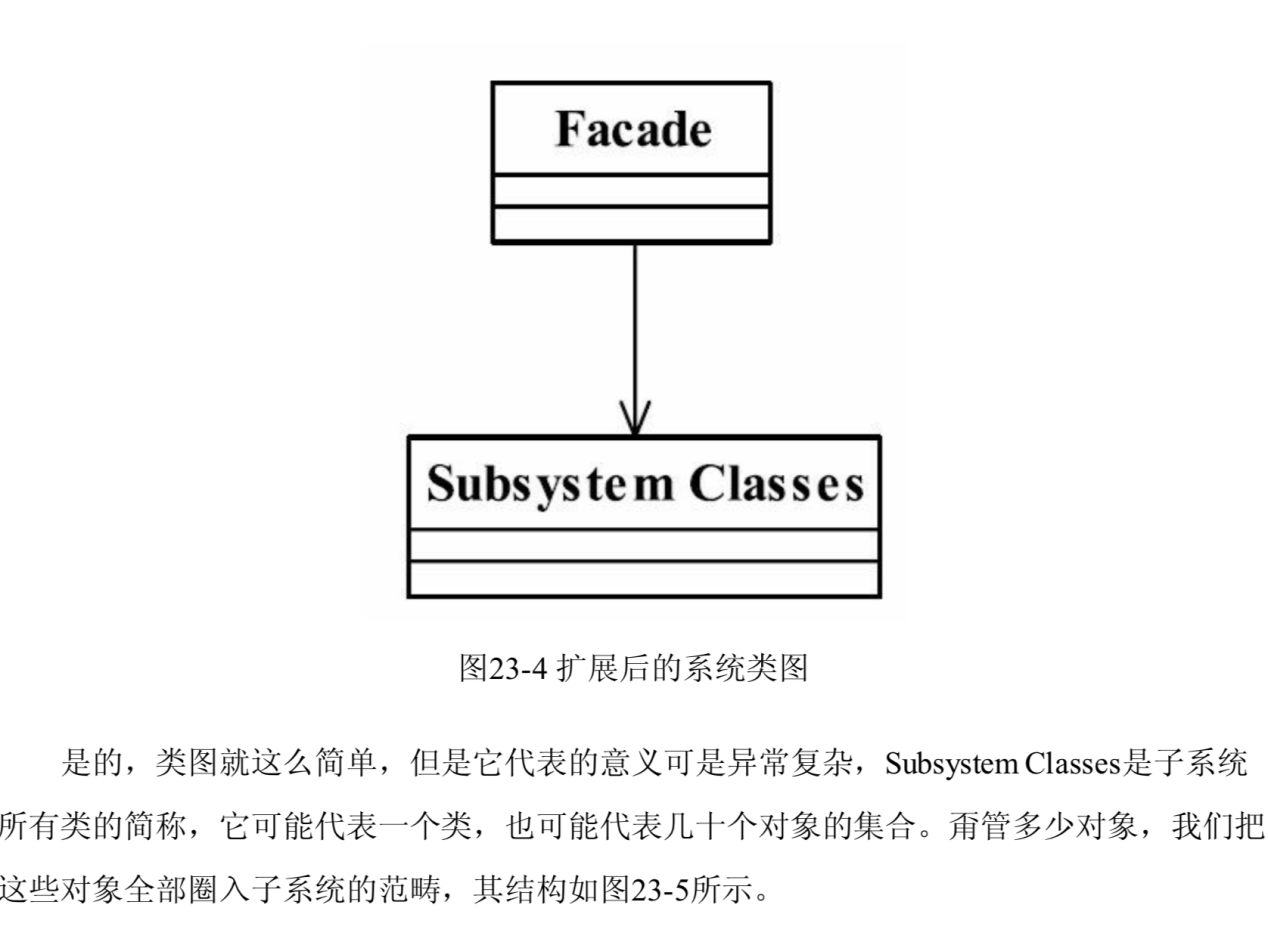






# 门面模式

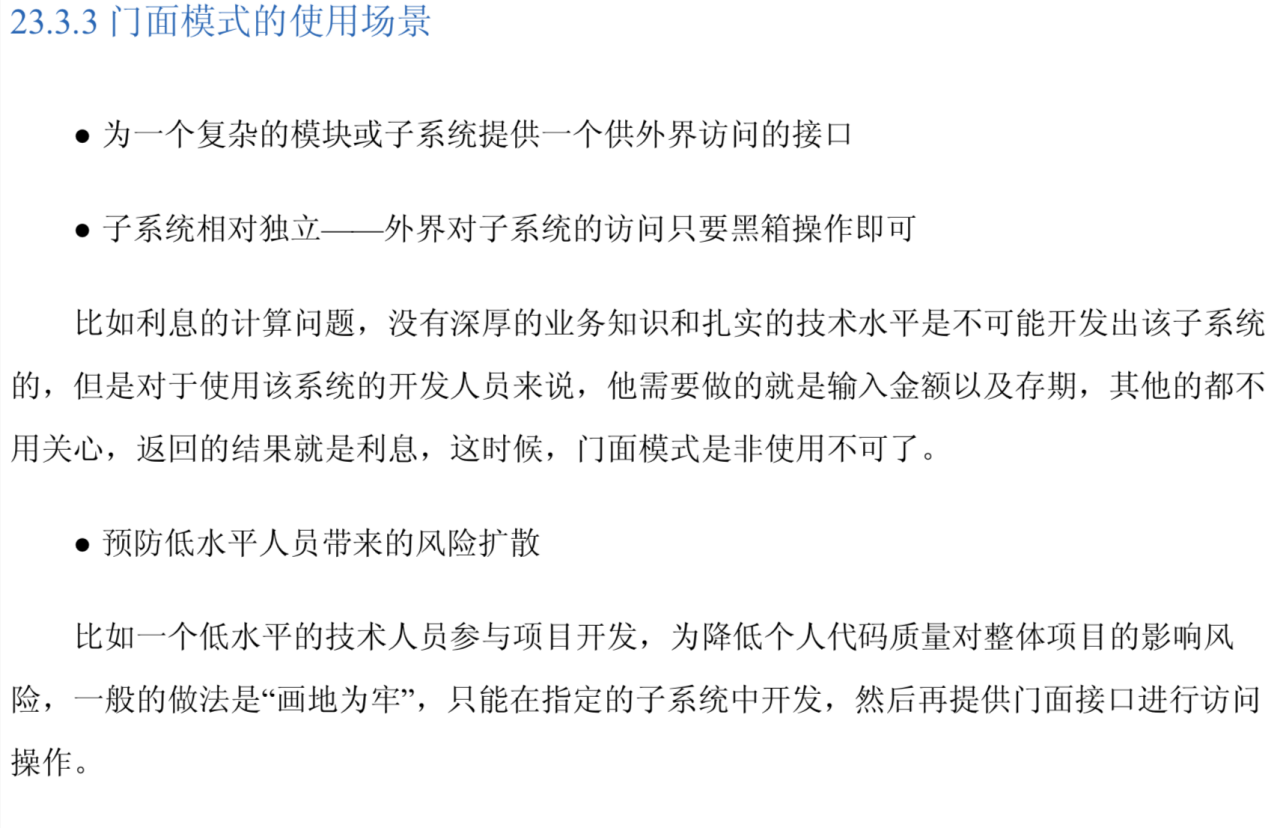




优点：

1. 减少系统的相互依赖
2. 提高了灵活性
3. 提高安全性

缺点：不符合开闭原则，对修改关闭，对扩展开放



案例：tomcat的Request，Response就使用了门面模式，netty的ByteBuf对ByteBuffer等的封装，也是门面模式。其实传统的mvc架构中的。controller，service，dao，这三层也体现了门面模式，controller是对外暴露的服务，service组合dao层处理业务逻辑。