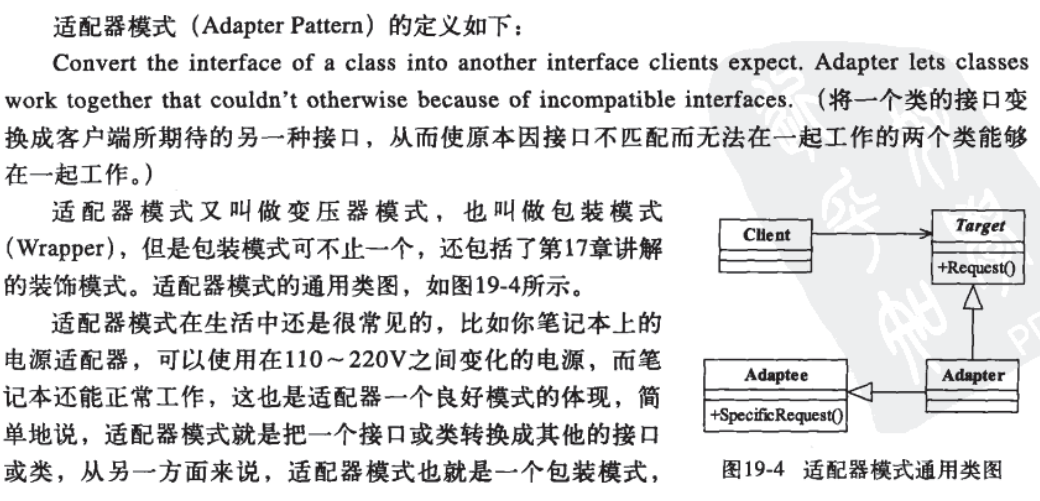
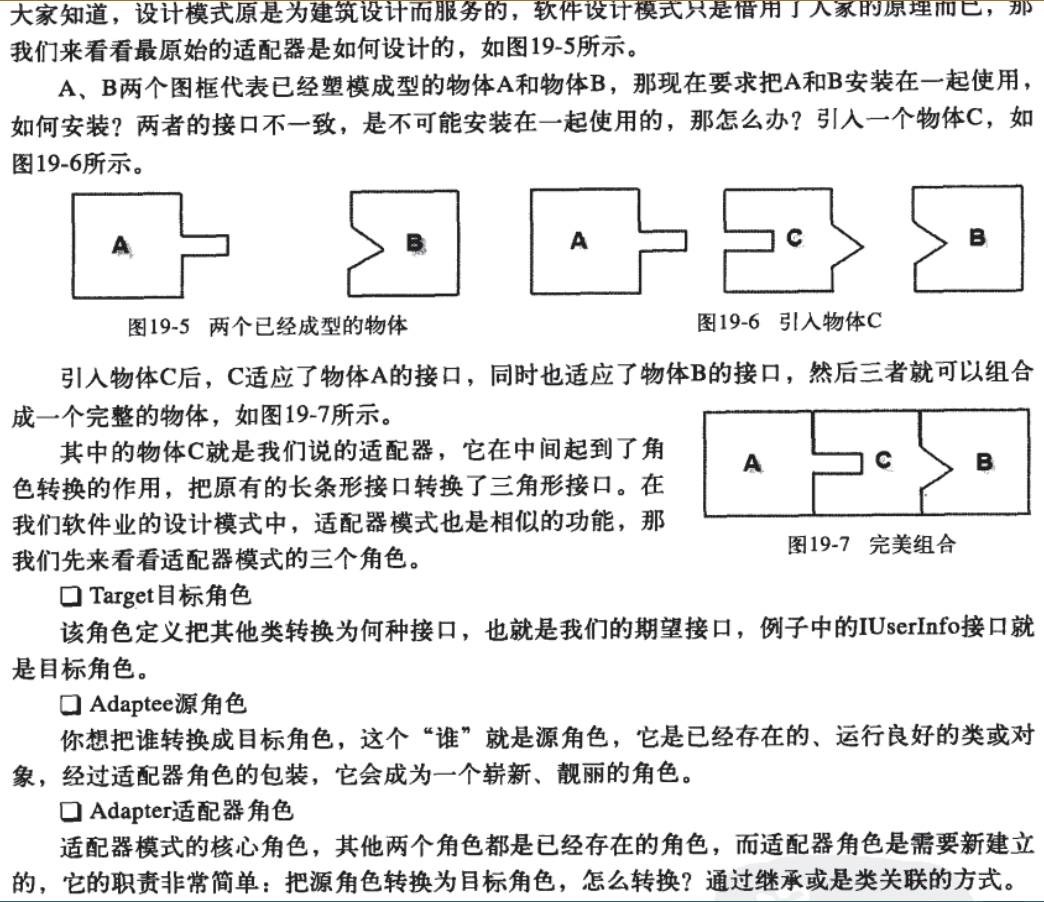
# 适配器模式

1. 定义

Convert the interface of a class into another interface clients expect. Adapter lets class work together that couldn’t otherwise because of incompatible interfaces.(将一个接口转换成客户端期望的另外一个接口，从而使原本因为接口不兼容而无法一起工作的多个类能一起工作)

1. 类图

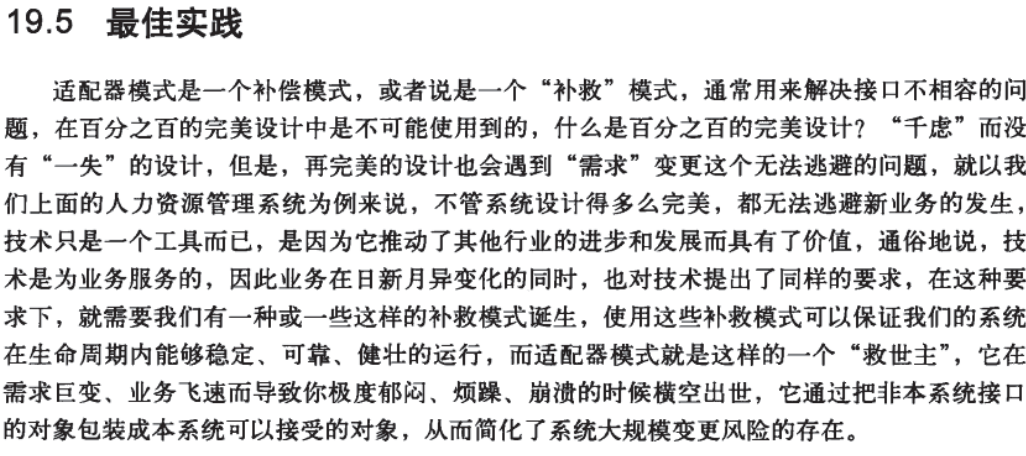




1. 模式的应用
2. 模式的优点
3. **适配器能让两个没有任何关系的类在一起运行，**只要适配器这个角色能够搞定他们就成。
4. **增加了类的透明度**：我们访问的Target对象，但具体实现委托给了源角色，而对这些高层次模块是透明的，也是它不关心的。
5. **提高了类的复用度**：源角色在原有的系统中还是可以正常使用，目标对象可以充当新的演员；
6. **灵活性非常好**：如果不想用适配器，删掉适配器类就好，其他代码不用修改。
7. 模式的缺点
8. 最好在详细设计阶段不要考虑使用适配器模式，它不是为了解决开发阶段问题，而是解决正在服役的项目问题。
9. 实际应用场景

策略模式又称为：政策模式。

1. 多个类只有在算法和行为上稍有不同的场景；
2. 算法需要自由切换场景：算法是由使用者决定或者算法在进化；
3. 需要屏蔽算法的场景：通过传递一个参数，然后反馈一个结果。
4. 最佳实践



1. 注意事项

