

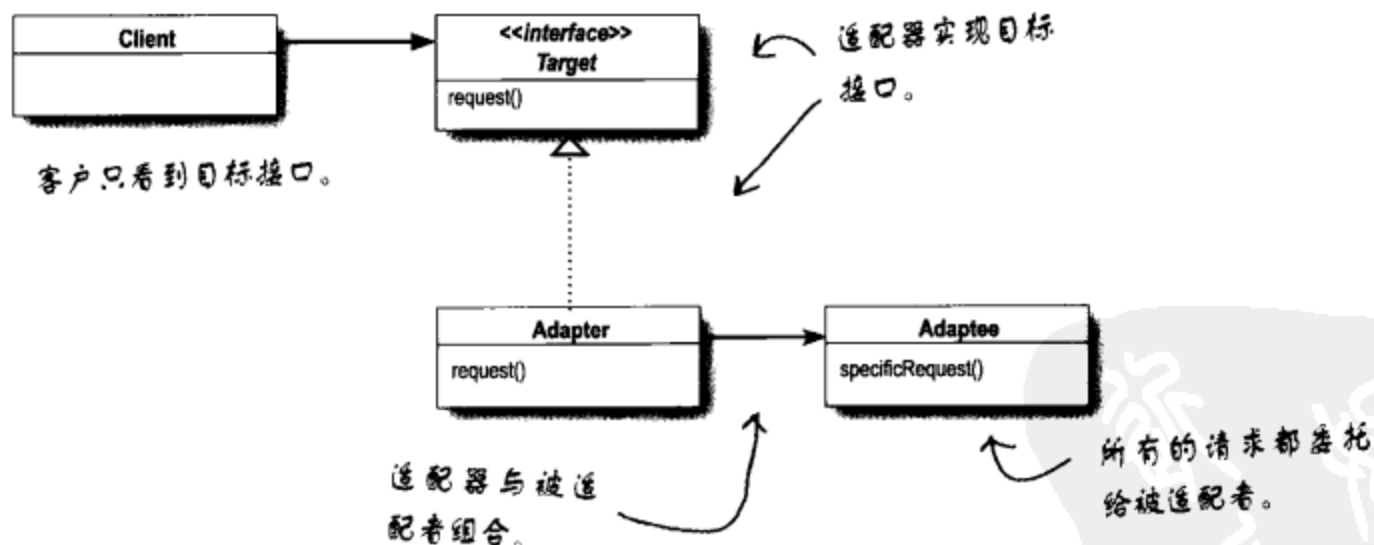
## 定义适配器模式

玩够了鸭子、火鸡和交流电适配器，现在让我们进入真实世界，并看看适配器模式的正式定义：

**适配器模式** 将一个类的接口，转换成客户期望的另一个接口。适配器让原本接口不兼容的类可以合作无间。

现在，我们知道，这个模式可以通过创建适配器进行接口转换，让不兼容的接口变成兼容。这可以让客户从实现的接口解耦。如果在一段时间之后，我们想要改变接口，适配器可以将改变的部分封装起来，客户就不必为了应对不同的接口而每次跟着修改。

我们已经看过了这个模式的运行时行为，现在来看它的类图：



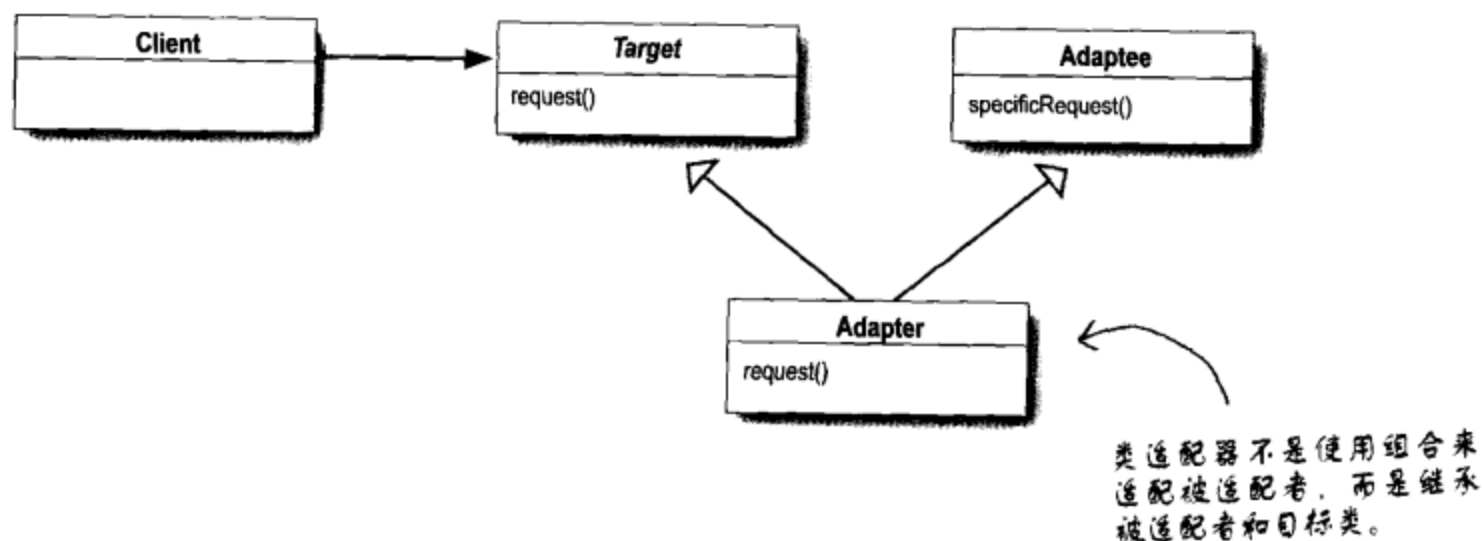
这个适配器模式充满着良好的OO设计原则：使用对象组合，以修改的接口包装被适配者：这种做法还有额外的优点，那就是，被适配者的任何子类，都可以搭配着适配器使用。

也请留意，这个模式是如何把客户和接口绑定起来，而不是和实现绑定起来的。我们可以使用数个适配器，每一个都负责转换不同组的后台类。或者，也可以加上新的实现，只要它们遵守目标接口就可以。

## 对象和类的适配器

现在，尽管已经定义了这个模式，但其实我们还没有告诉你有关的一切。实际上有“两种”适配器：“对象”适配器和“类”适配器。本章涵盖了对象适配器和类适配器。前一页是对象适配器的图。

究竟什么是“类”适配器？为什么我们还没告诉你这种适配器？因为你需要多重继承才能够实现它，这在Java中是不可能的。但是当你在使用多重继承语言的时候，还是可能遇到这样的需求。让我们看看多重继承的类图。



看起来很熟悉吗？没错，唯一的差别就在于适配器继承了Target和Adaptee。而对象适配器利用组合的方式将请求传送给被适配者。



对象适配器和类适配器使用两种不同的适配方法（分别是组合与继承）。这两种实现的差异如何影响适配器的弹性？