目录

1. 引入场景
2. 使用委托解决
3. 使用反射解决
4. 使用接口解决

一．场景

假设有一个类A，它里面有两个方法一个是MethodA\_One()而另外一个是MethodA\_Two()。于此同时，有另外一个类B，它里面有两个方法MethodB\_One()个MethodB\_Two()。

遇到这样一个业务场景类A中的MethodA\_One()要调用类B中的MethodB\_One()，而类B中的MethodB\_Two()要调用A中的MethodA\_Two()中的方法。

明显可知，如果类A和类B分别在两个程序集中，如果彼此引用会出现循环引用的问题。那么下面的解决方法都以这个列子进行处理。

二. 委托解决

一般人遇到这个问题的第一反应就是委托。因为委托很像“传递一个函数指针”给另一

个类，往往在类与类之间可以不通过直接的对象引用，来使用另一个类的方法。

三. 使用反射解决

反射也不失为一个很好的方法。在一些项目中常用反射来进行解耦（解除类与类之间中

引用）。当然使用反射的话，不但会造成一定的性能的损失，而且会让代码变的更加的难懂，会让别人琢磨半天，才发现你这里的反射仅仅是用来解决循环引用的。

1. 使用接口解决

这个是比较简单而且是比较推荐的做法。