

用VC进行COM编程所必须掌握的理论知识 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/270/2021_2022__E7_94_A8VC_E8_BF_9B_E8_A1_c67_270965.htm 这篇文章是给初学者看的，尽量写得比较通俗易懂，并且尽量避免编程细节。完全是根据我自己的学习体会写的，其中若有技术上的错误之处，请大家多多指正。

一、为什么要用COM 软件工程发展到今天，从一开始的结构化编程，到面向对象编程，再到现在的COM编程，目标只有一个，就是希望软件能象积方块一样是累起来的，是组装起来的，而不是一点点编出来的。结构化编程是函数块的形式，通过把一个软件划分成许多模块，每个模块完成各自不同的功能，尽量做到高内聚低耦合，这已经是一个很好的开始，我们可以把不同的模块分给不同的人去做，然后合到一块，这已经有了组装的概念了。软件工程的核心就是要模块化，最理想的情况就是100%内聚0%耦合。整个软件的发展也都是朝着这个方向走的。结构化编程方式只是一个开始。下一步就出现了面向对象编程，它相对于面向功能的结构化方式是一个巨大的进步。我们知道整个自然界都是由各种各样不同的事物组成的，事物之间存在着复杂的千丝万缕的关系，而正是靠着事物之间的联系、交互作用，我们的世界才是有生命力的才是活动的。我们可以认为在自然界中事物做为一个概念，它是稳定的不变的，而事物之间的联系是多变的、运动的。事物应该是这个世界的本质所在。面向对象的着眼点就是事物，就是这种稳定的概念。每个事物都有其固有的属性，都有其固有的行为，这些都是事物本身所固有的东西，而面向对象的方法就是描述出这种

稳定的东西。而面向功能的模块化方法它的着眼点是事物之间的联系，它眼中看不到事物的概念它只注重功能，我们平常在划分模块的时候有没有想过这个函数与哪些对象有关呢？很少有人这么想，一个函数它实现一种功能，这个功能必定与某些事物想联系，我们没有去掌握事物本身而只考虑事物之间是怎么相互作用而完成一个功能的。说白了，这叫本末倒置，也叫急功近利，因为不是我们智慧不够，只是因为我们没有多想一步。面向功能的结构化方法因为它注意的只是事物之间的联系，而联系是多变的，事物本身可能不会发生大的变化，而联系则是很有可能发生改变的，联系一变，那就是另一个世界了，那就是另一种功能了。如果我们用面向对象的方法，我们就可以以不变应万变，只要事先把事物用类描述好，我们要改变的只是把这些类联系起来的方法，只是重新使用我们的类库，而面向过程的方法因为它构造的是一个不稳定的世界，所以一点小小的变化也可能导致整个系统都要改变。然而面向对象方法仍然有问题，问题在于重用的方法。搭积木式的软件构造方法的基础是有许许多多各种各样的可重用的部件、模块。我们首先想到的是类库，因为我们用面向对象的方法产生的直接结果就是许多的类。但类库的重用是基于源码的方式，这是它的重大缺陷。首先它限制了编程语言，你的类库总是用一种语言写的吧，那你就不能拿到别的语言里用了。其次你每次都必须重新编译，只有编译了才能与你自己的代码结合在一起生成可执行文件。在开发时这倒没什么，关键在于开发完成后，你的EXE都已经生成好了，如果这时候你的类库提供厂商告诉你他们又做好了一个新的类库，功能更强大速度更快，而你为之心动又

想把这新版的类库用到你自己的程序中，那你就必须重新编译、重新调试！这离我们理想的积木式软件构造方法还有一定差距，在我们的设想里希望把一个模块拿出来再换一个新的模块是非常方便的事，可是现在不但要重新编译，还要冒着很大的风险，因为你可能要重新改变你自己的代码。另一种重用方式很自然地就想到了是DLL的方式。Windows里到处是DLL，它是Windows的基础，但DLL也有它自己的缺点。总结一下它至少有四点不足。(1)函数重名问题。DLL里是一个一个的函数，我们通过函数名来调用函数，那如果两个DLL里有重名的函数怎么办？(2)各编译器对C++函数的名称修饰不兼容问题。对于C++函数，编译器要根据函数的参数信息为它生成修饰名，DLL库里存的就是这个修饰名，但是不同的编译器产生修饰的方法不一样，所以你在VC里编写的DLL在BC里就可以用不了。不过也可以用extern "C"来强调使用标准的C函数特性，关闭修饰功能，但这样也丧失了C++的重载多态性功能。(3)路径问题。放在自己的目录下面，别人的程序就找不到，放在系统目录下，就可能有重名的问题。而真正的组件应该可以放在任何地方甚至可以不在本机，用户根本不需考虑这个问题。(4)DLL与EXE的依赖问题。我们一般都是用隐式连接的方式，就是编程的时候指明用什么DLL，这种方式很简单，它在编译时就把EXE与DLL绑在一起了。如果DLL发行了一个新版本，我们很有必要重新链接一次，因为DLL里面函数的地址可能已经发生了改变。DLL的缺点就是COM的优点。首先我们要先把握住一点，COM和DLL一样都是基于二进制的代码重用，所以它不存在类库重用时的问题。另一个关键点是，COM本身也是DLL，即使

是ActiveX控件.ocx它实际上也是DLL，所以说DLL在还是有重用上有很大的优势，只不过我们通过制订复杂的COM协议，通COM本身的机制改变了重用的方法，以一种新的方法来利用DLL，来克服DLL本身所固有的缺陷，从而实现更高一级的重用方法。COM没有重名问题，因为根本不是通过函数名来调用函数，而是通过虚函数表，自然也不会有函数名修饰的问题。路径问题也不复存在，因为是通过查注册表来找组件的，放在什么地方都可以，即使在别的机器上也可以。也不用考虑和EXE的依赖关系了，它们二者之间是松散的结合在一起，可以轻松的换上组件的一个新版本，而应用程序混然不觉。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com