

C 指针使用方法解惑 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/133/2021_2022_C___E6_8C_87_E9_92_88_E4_c97_133696.htm

在下列函数声明中，为什么要同时使用*和amp.pBuildingElement)。论坛中经常有人问到这样的问题。本文试图通过一些实际的指针使用经验来解释这个问题。仔细看一下这种声明方式，确实有点让人迷惑。在某种意义上，"*"和"amp.p).void func1(MYCLASS** pMyClass).{*pMyClass = new MYCLASS.....}调用func1之后，p指向新的对象。在COM编程中，你到处都会碰到这样的用法--例如在查询对象接口的QueryInterface函数中：interface ISomeInterface { HRESULT QueryInterface(IID amp.p). 此处，p是SOMEINTERFACE类型的指针，所以amp.pMyClass).{pMyClass = new MYCLASS.}其实，它和前面所讲得指针的指针例子是一码事，只是语法有所不同。传递的时候不用传p的地址amp.，就应该想到**。也就是说这个函数修改或可能修改调用者的指针，而调用者象普通变量一样传递这个指针，不使用地址操作符amp.

GetAt(POSITION position).Cobject* GetAt(POSITION position) const.}.这里有两个GetAt函数，功能都是获取给定位置的元素。区别何在呢？区别在于一个让你修改列表中的对象，另一个则不行。所以如果你写成下面这样：Cobject* pObj = mylist.GetAt(pos).则pObj是列表中某个对象的指针，如果接着改变pObj的值：pObj = pSomeOtherObj.这并改变不了在位置pos处的对象地址，而仅仅是改变了变量pObj。但是，如果你写成下面这样：Cobject*amp.版本可以替代对象。在C/C

中引用是很重要的，同时也是高效的处理手段。所以要想成为C/C 高手，对引用的概念没有透彻的理解和熟练的应用是不行的。 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