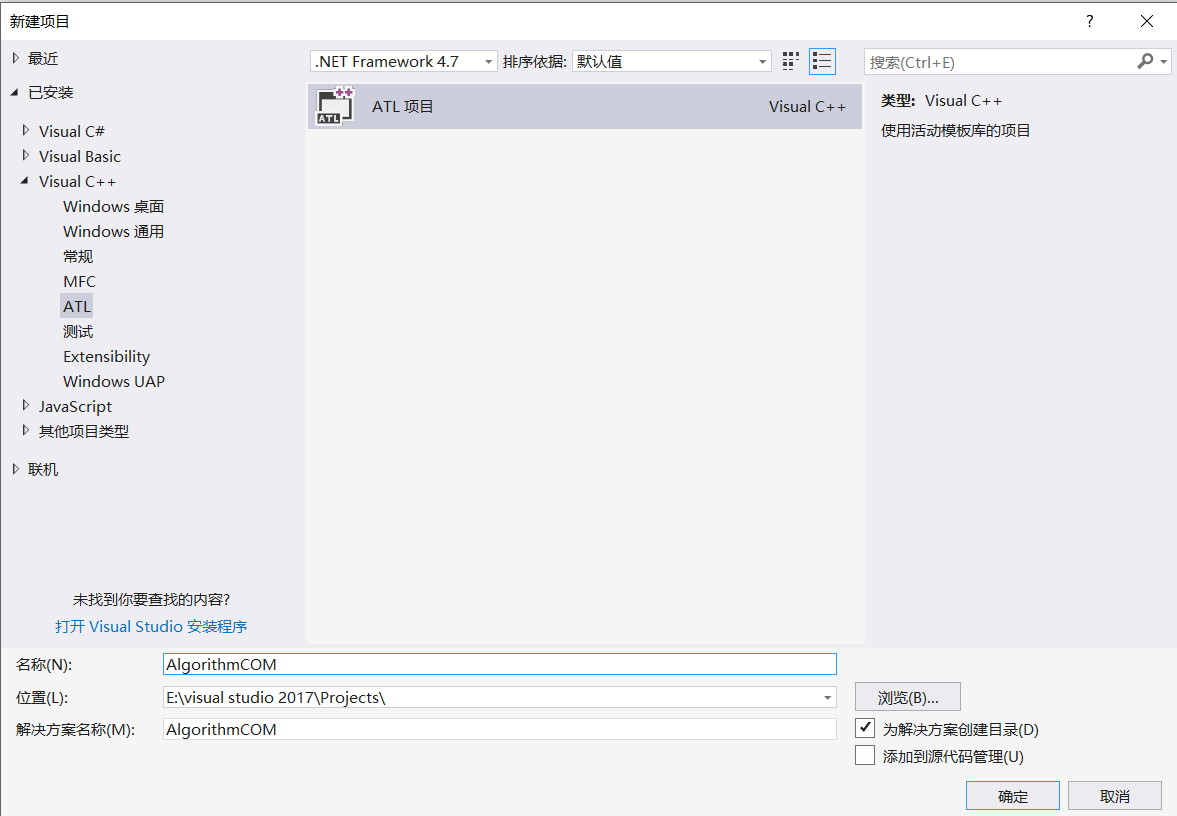
**C++ COM编写和调用**

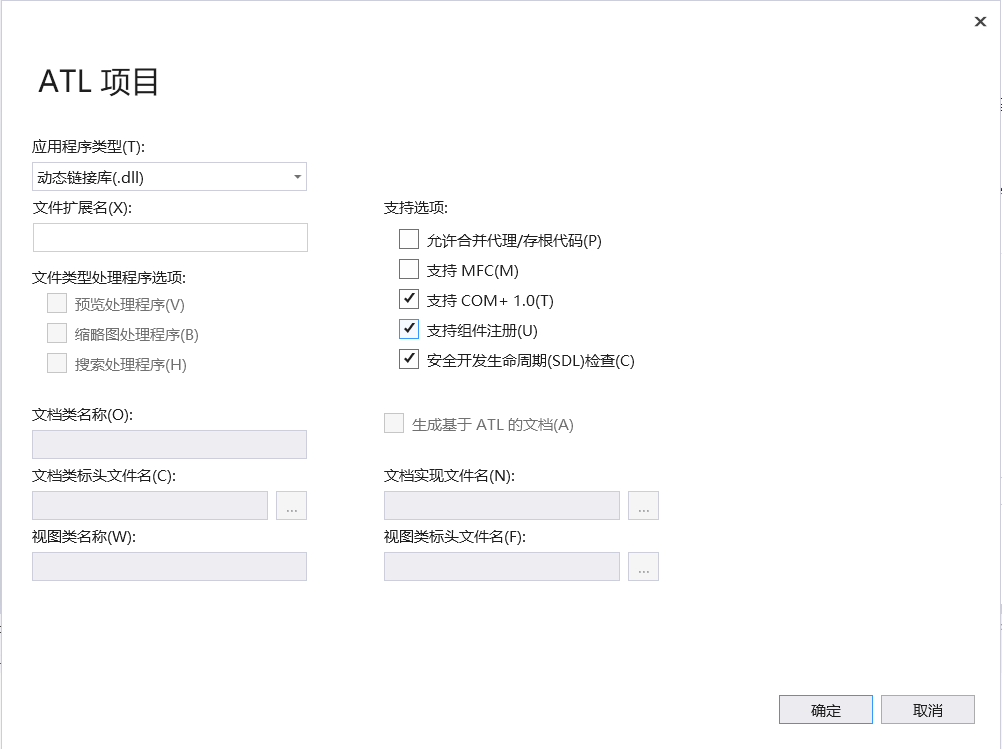
**一、生成COM**

1.新建ATL COM项目

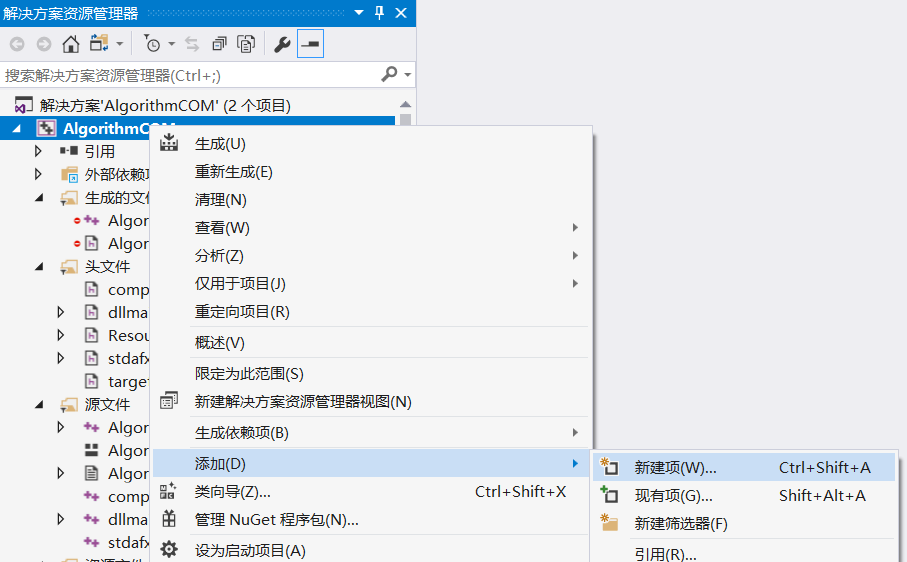
打开VS，文件->新建->项目，选择Visual C++->ATL项目，填写COM组件名称, 例如AlgorithmCOM，确定。



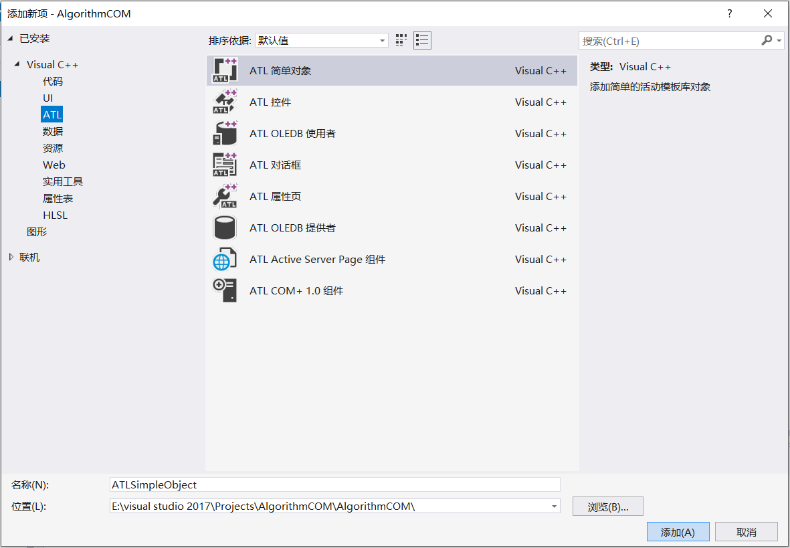
2.ATL 项目向导，勾选【支持COM+ 1.0】和【支持部件注册器】，其他默认，点击完成。



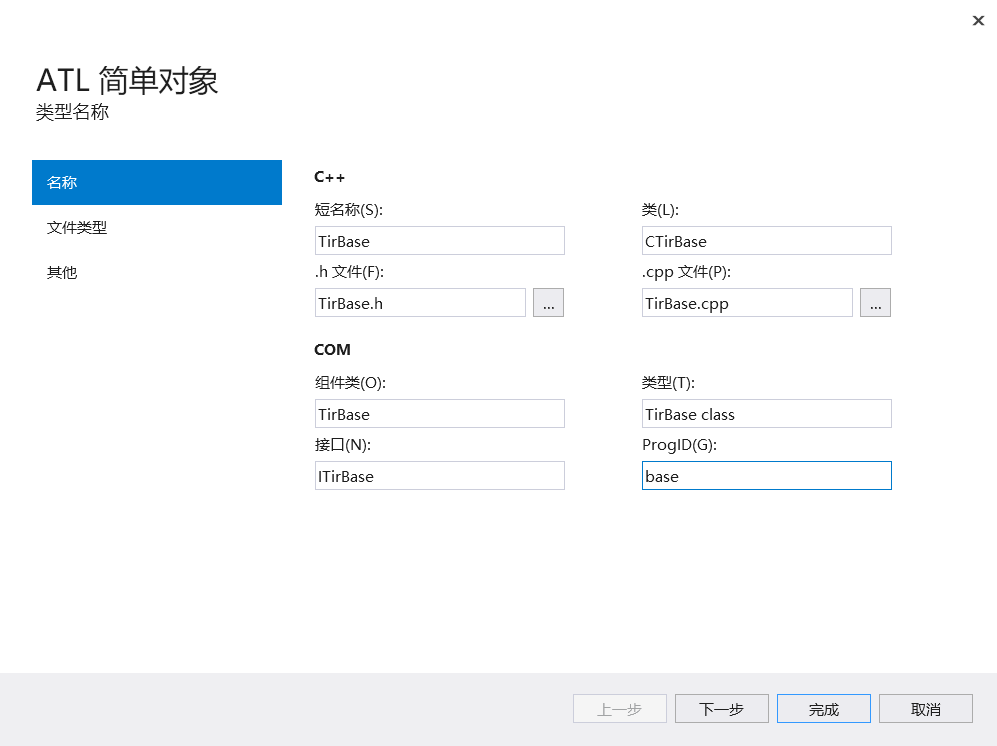
3.右键单击项目“AlgorithmCOM”->添加->新建项。



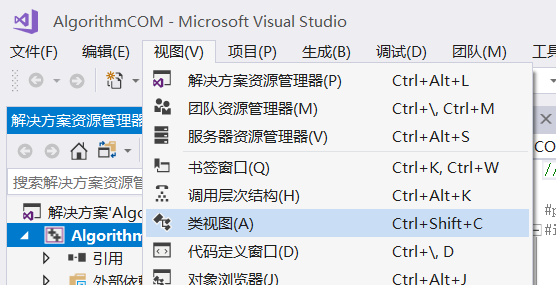
4.选择：ATL-> ATL 简单对象->添加。

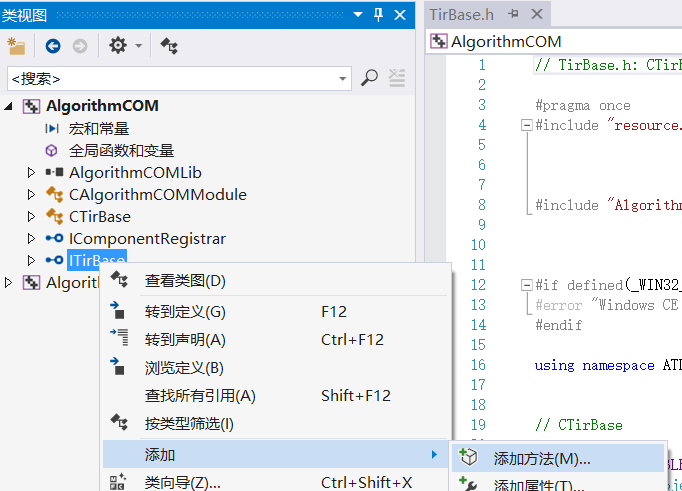


5.ATL  简单对象向导，“简称”中输入类名称，如TirBase，其他输入框会自动填写，最下面的“ProgID”没有自动填写，最好自己写上一个，如base，否则后面无法正确调用组件，点击“完成”。如果出现提示【是否覆盖现有文件】，点击【是】即可。

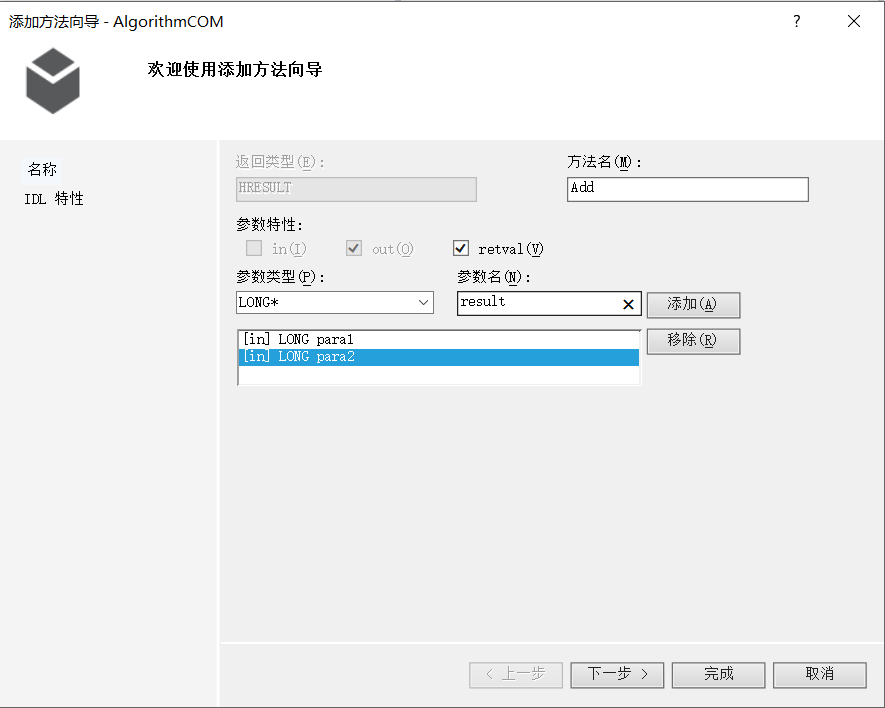


6. 视图，切换到类视图，选中刚才添加的接口类ITirBase，如下，然后右键单击【添加】添加方法(成员函数)。

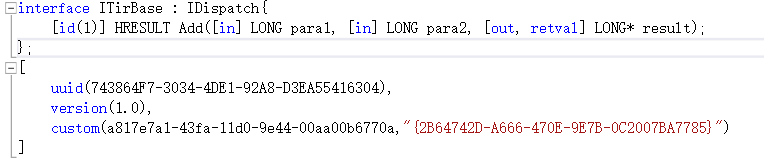




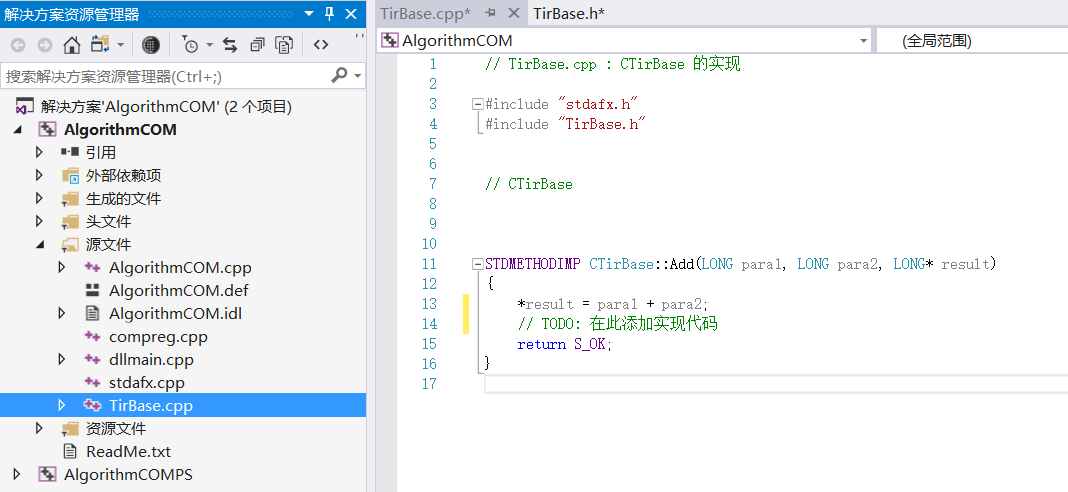
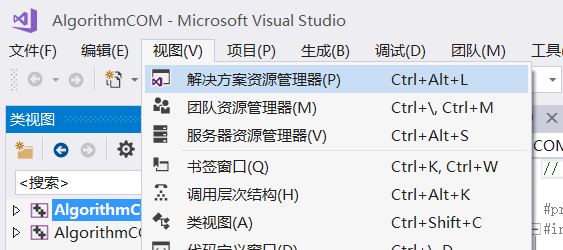
7.添加方法向导，在"方法名" 下输入成员函数名，如Add。设置输入输出参数属性，如Add加法函数，需要增加两个输入参数：勾选【in】，选择参数类型LONG，输入参数名para1，点击“添加”。同样操作增加参数para2。然后增加一个返回参数：先选择参数类型LONG\*，再勾选【retval】(返回值),输入参数名result，点击添加。点击“完成”即可，如果点击下一步，不需要填写信息，点击“完成”即可。



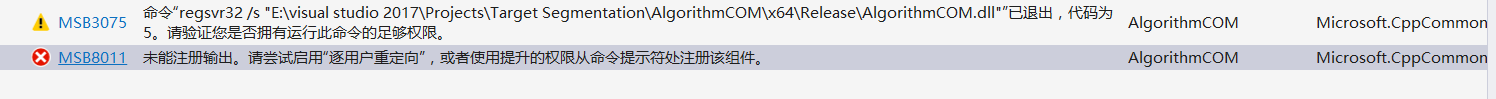
8.切换到解决方案资源管理器，展开项目“源文件”， 双击 TirAlgorithmCom.idl 打开IDL定义文件，看到如下代码，就是定义的COM方法了。(类似于成员函数声明)



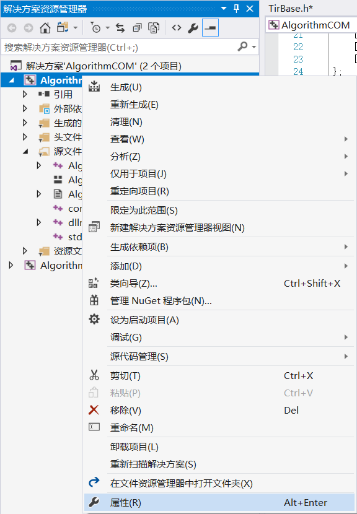
9.切换视图，解决方案资源管理器视图。左侧源文件，双击打开TirBase类，编写成员函数的代码。如Add函数代码：\*result = para1 + para2；



10. 编译运行，会生成dll文件。生成后会报如下错误：

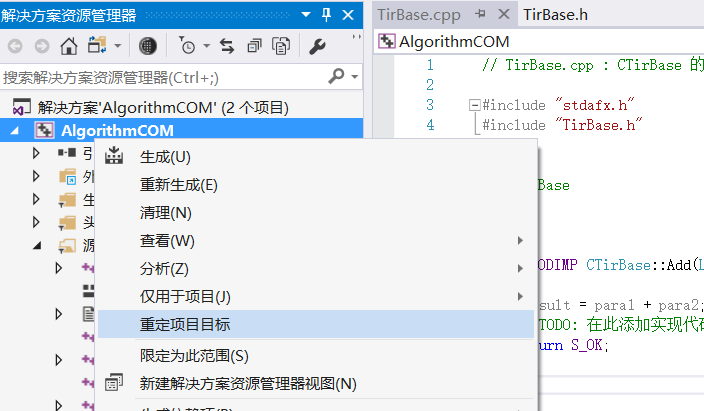


选择项目，右键，属性。



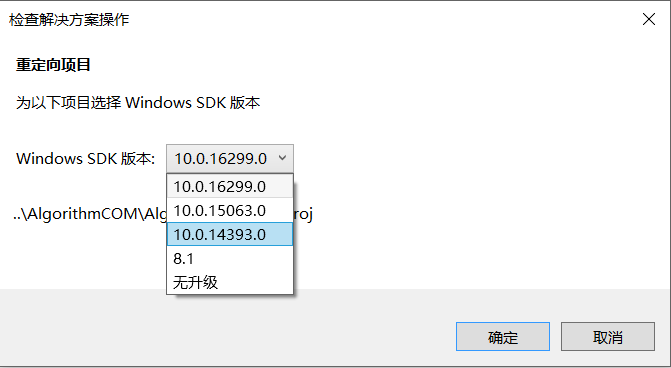
链接器->常规->注册输出改为“是”，确定保存。

左侧AlgorithmCOM工程目录，右键，选择重定项目目标



**SDK版本必须选择10.0.15及以下的版本。10.0.16不支持ATL对象。**

**比如选择10.0.15063.0（硬盘里面有）**



重新编译后会出现如下错误命令框



说明已经生成AlgorithmCOM的dll文件，在其工程目录下的子文件夹中可以找到dll文件。

比如我的是64位release编译的，在x64\release文件夹下。

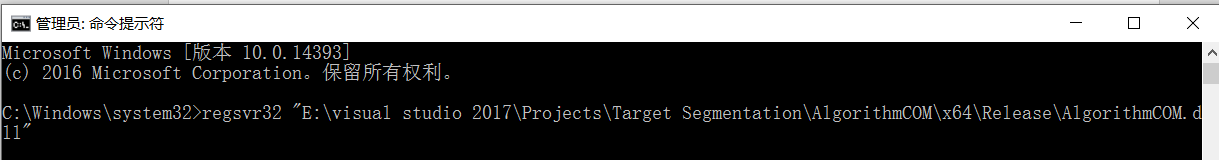


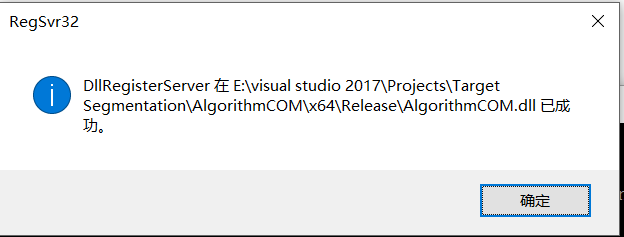
上述错误的原因是未注册COM组件。

11.注册COM：以管理员身份打开cmd命令窗口，输入：

regsvr32 “E:\vs2017\Projects \AlgorithmCOM\x64\Release \ AlgorithmCOM.dll”

也就是dll文件的绝对路径，进行注册即可。





**二、调用COM**

1.新建一个VS工程，win32控制台程序

输入如下代码：

#include "stdafx.h"

#include<iostream>

#include "atlcomcli.h"

#import "E:\\visual studio 2017\\Projects\\AlgorithmCOM\\x64\\Release\\AlgorithmCOM.dll" no\_namespace

using namespace std;

int main()

{

CoInitialize(NULL);

CLSID clsid;//CLSID指针

CLSIDFromProgID(OLESTR("base.1"), &clsid);//ProgID指针，classname.rgs中可以找到

//rgs是组件注册的脚本文件，当使用Regsvr32.exe注册组件时，组件内部便是调用了这个文件。rgs文件是以资源的形式存于DLL内的。

CComPtr<ITirBase> pFirstClass;//智能指针，指向COM中的类

pFirstClass.CoCreateInstance(clsid);

long ret = pFirstClass->Add(1, 2);//函数名

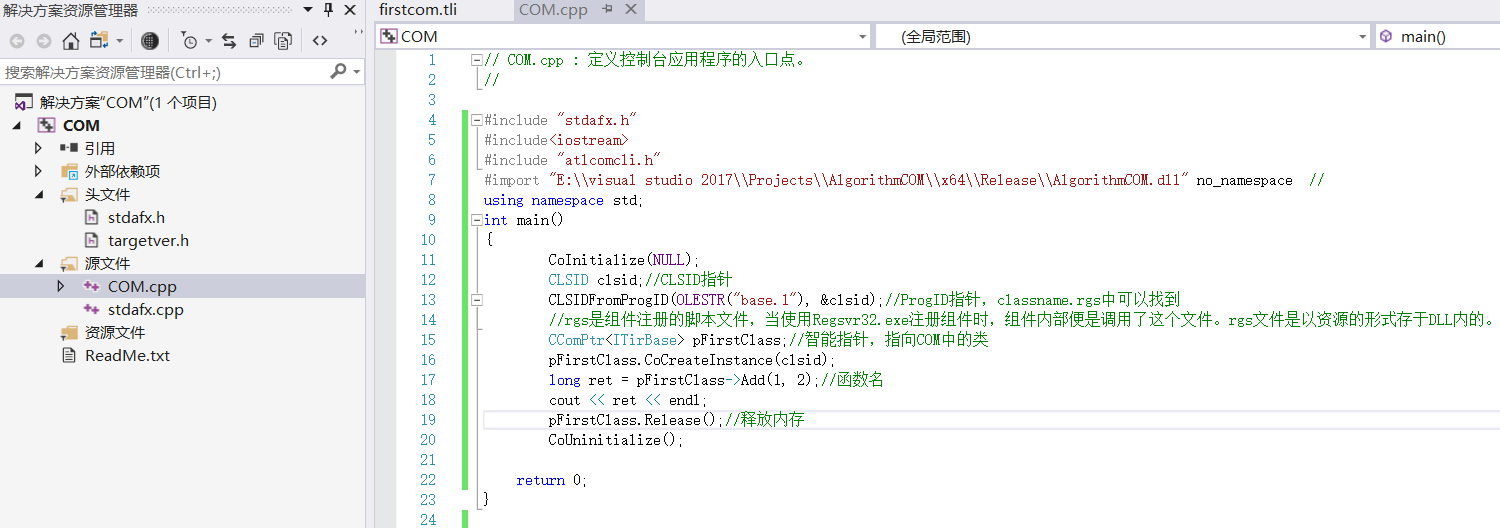
cout << ret << endl;

pFirstClass.Release();//释放内存

CoUninitialize();

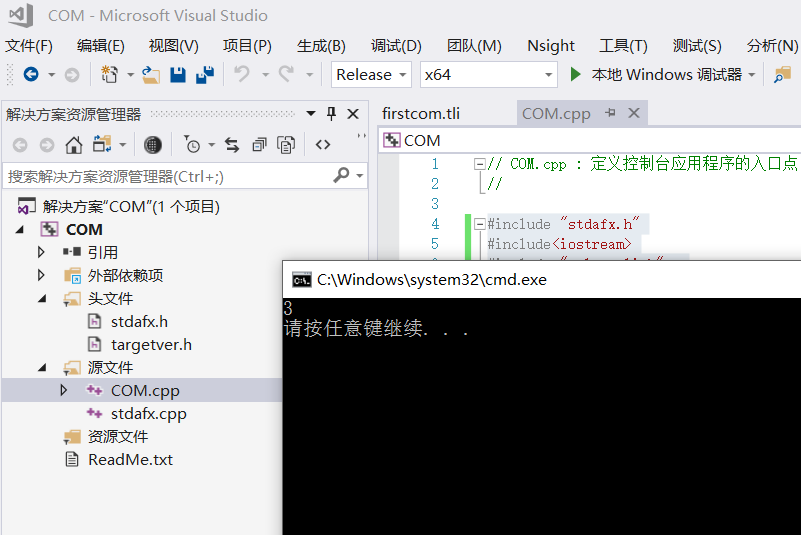
return 0;

}



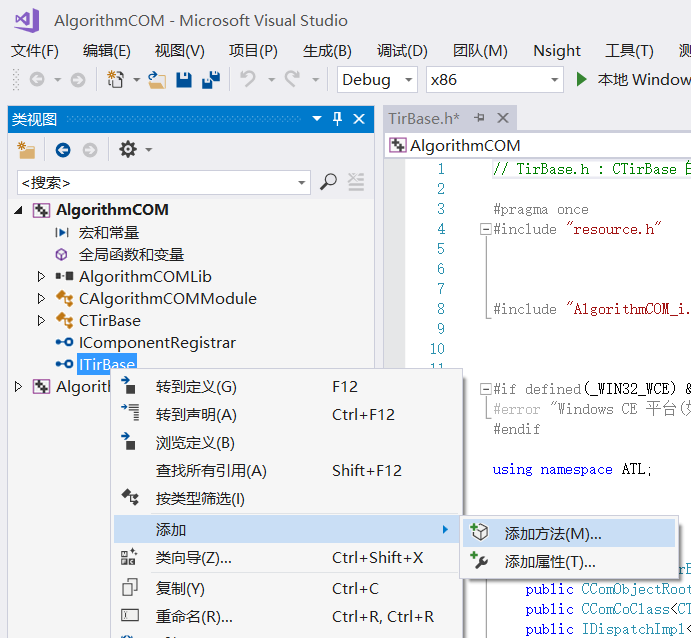
即可实现调用AlgorithmCOM.dll中的Add函数。

如Add(1,2)的测试结果如下：



三、COM中多个函数的添加

将AlgorithmCOM工程切换到类视图，右键ITirBase，添加->添加方法。按照之前添加方法的步骤添加新的函数add2，并在TirBase.cpp中添加add2函数的代码。执行程序生成新的dll文件，并注册。（具体实现方法如一、生成COM）



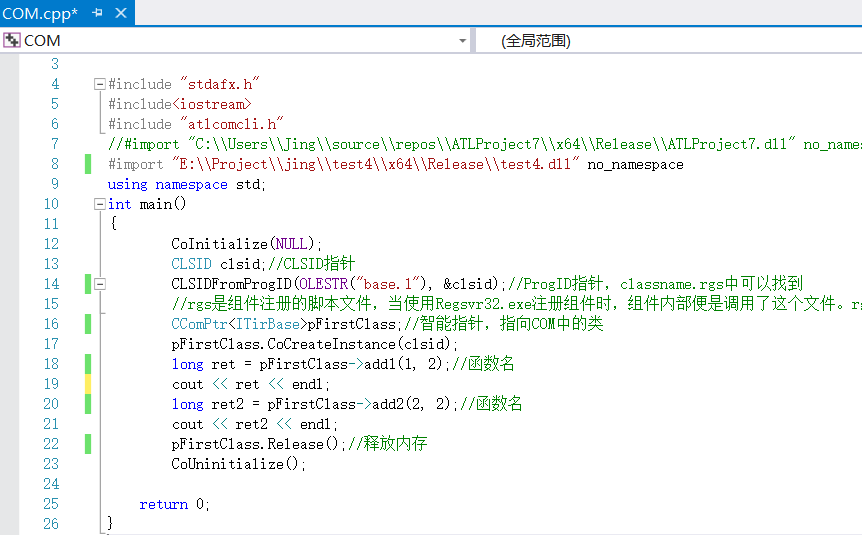


四、C++中多函数COM的调用

只需增加新的对应函数的调用语句，如：

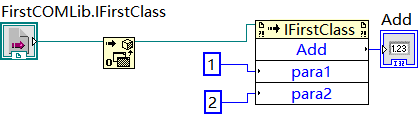
long ret2 = pFirstClass->add2(2, 2);

cout << ret2 << endl;



五、LabVIEW调用COM

在LabVIEW中添加自动打开控件、调用方法节点。在自动打开控件的输入端右键选择输入ActiveX类，选择刚生成的COM组件（dll文件）。将控件间连接并运行即可实现该COM组件的功能。如下图为简单的两数相加的LabVIEW程序：



如果该类中含有多个函数，在调用方法节点处可以进行函数的选择，如下图：

