

## 在vs2019中使用C++调用C#生成的dll\_向着曙光前进的博客-CSDN博客

成就一亿技术人!

原创

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

### 在vs2019中使用C++调用C#生成的dll

[引言](#)

[C#生成dll](#)

[创建工程](#)

[配置工程属性](#)

[编写代码](#)

[生成dll](#)

[C++使用C#dll](#)

[Successful!](#)

### 引言

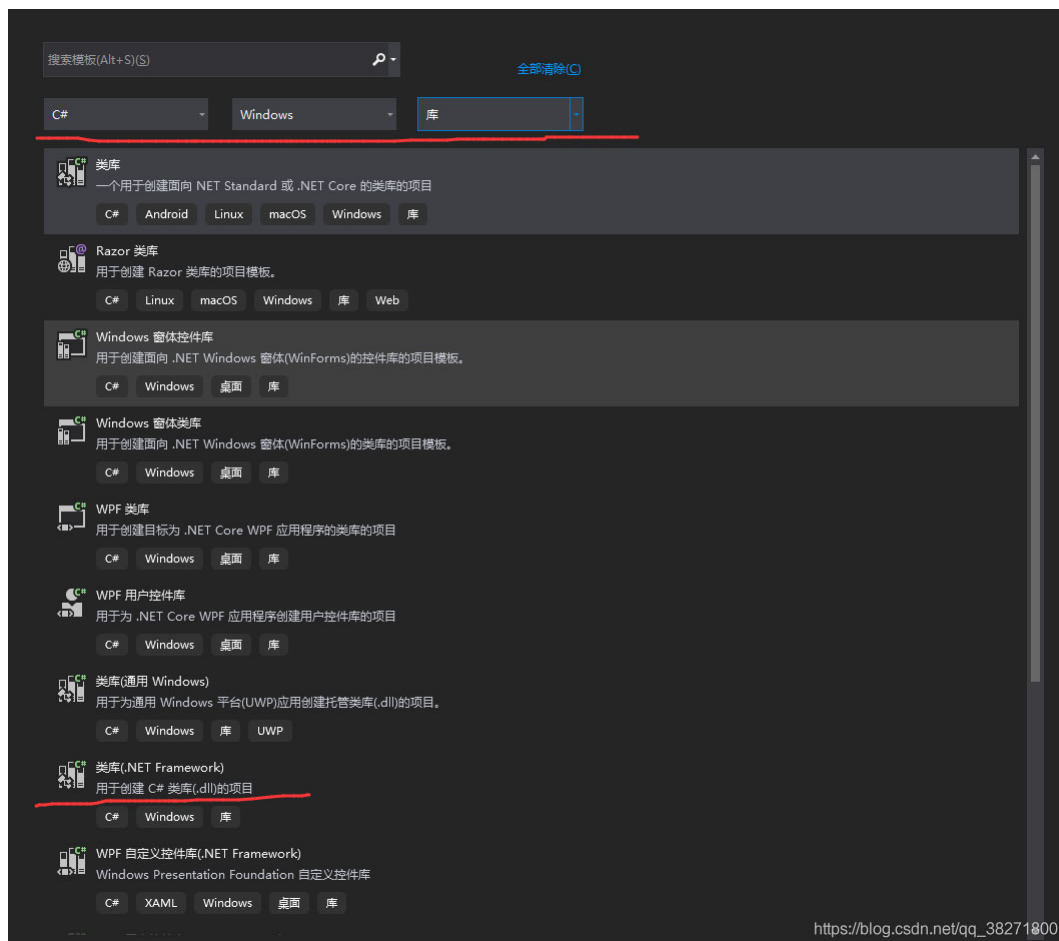
网上有很多篇相关博客，我为什么还要写这篇博客呢？

原因很简单，就是网上的博客写的并不详细，不是给新手学习是用的，而是供老手回忆的。这样并不能让新手快速入手。

接下来我将以一个新手的视角讲解C#如何生成dll,C++又是如何调用。

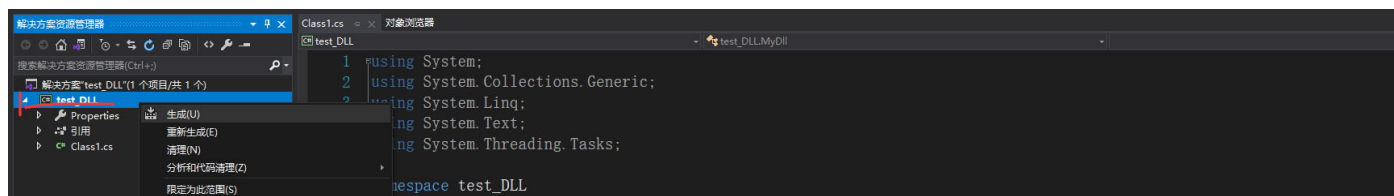
### C#生成dll

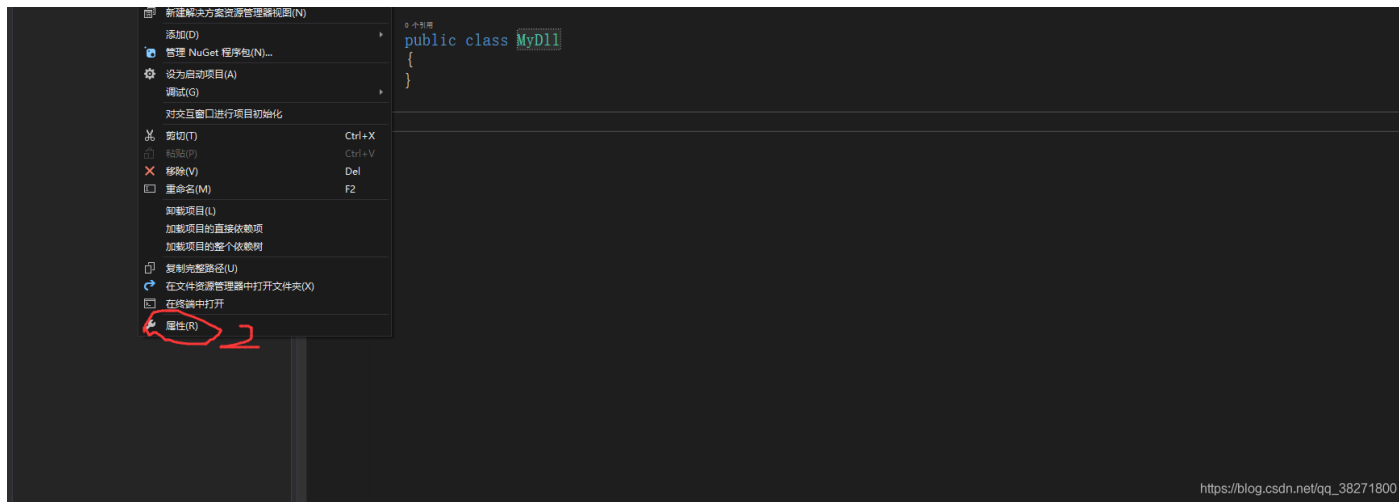
#### 创建工程



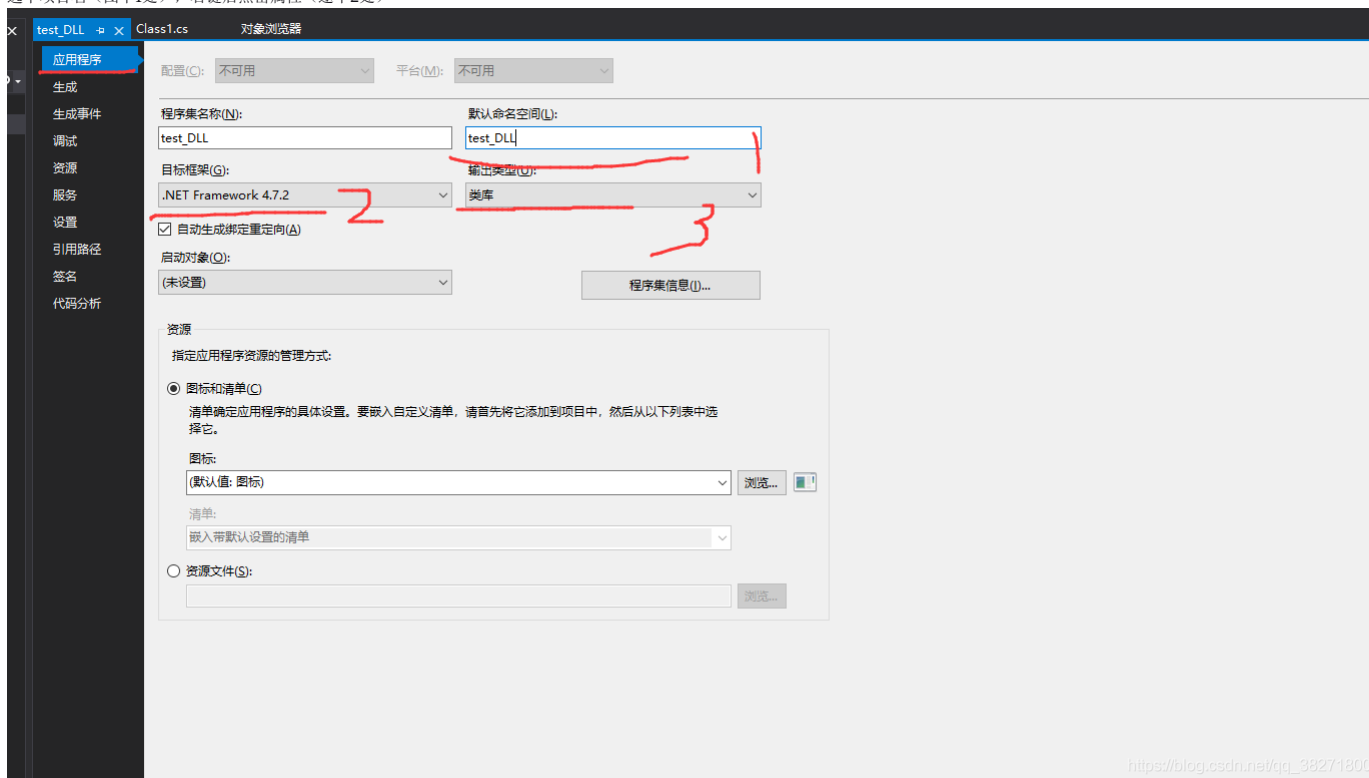
打开vs2019，点击新项目，在上面的筛选条件中选中如图所示的条件，最后在下面选中“类库(.NET framework)”，点击下一步，直至创建成功。

#### 配置工程属性

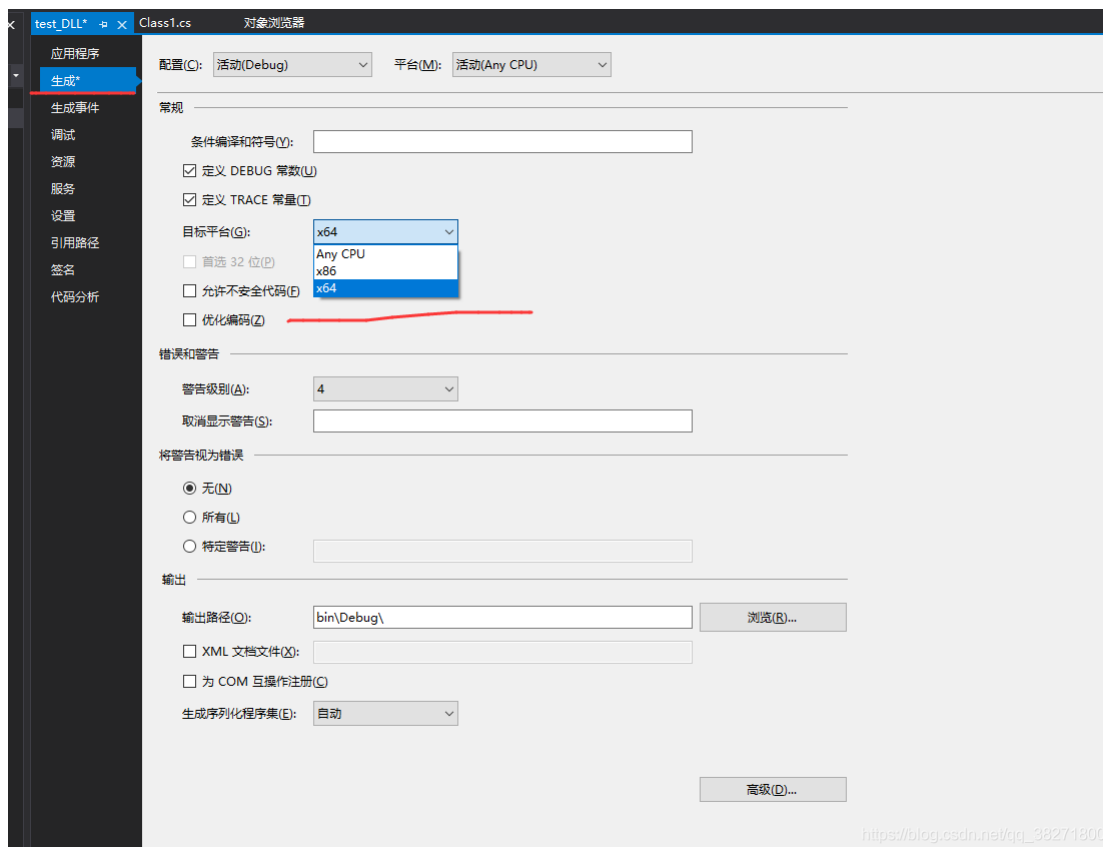




选中项目名（图中1处），右键后点击属性（途中2处）



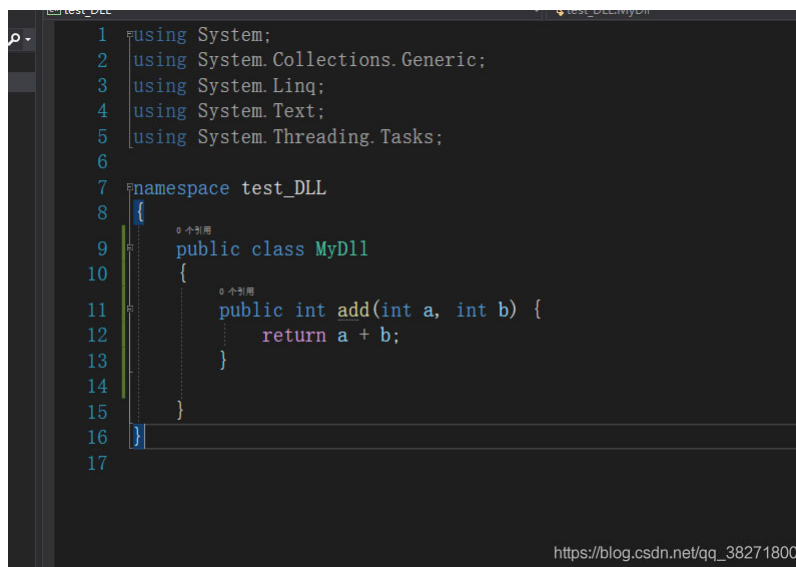
（上图中1处）是名字空间，后续在c++调用时会进行使用。（上图中2处）是.net framework版本，选择适合你想用的版本。（上图中3处）是输出内容，类库代表dll(动态连接库)



选择生成，根据上图中红线指示选择自己想要生成的文件应用于什么平台（这个c++和c#需要统一），我是用的是x64。

Ctrl + s 保存配置

#### 编写代码



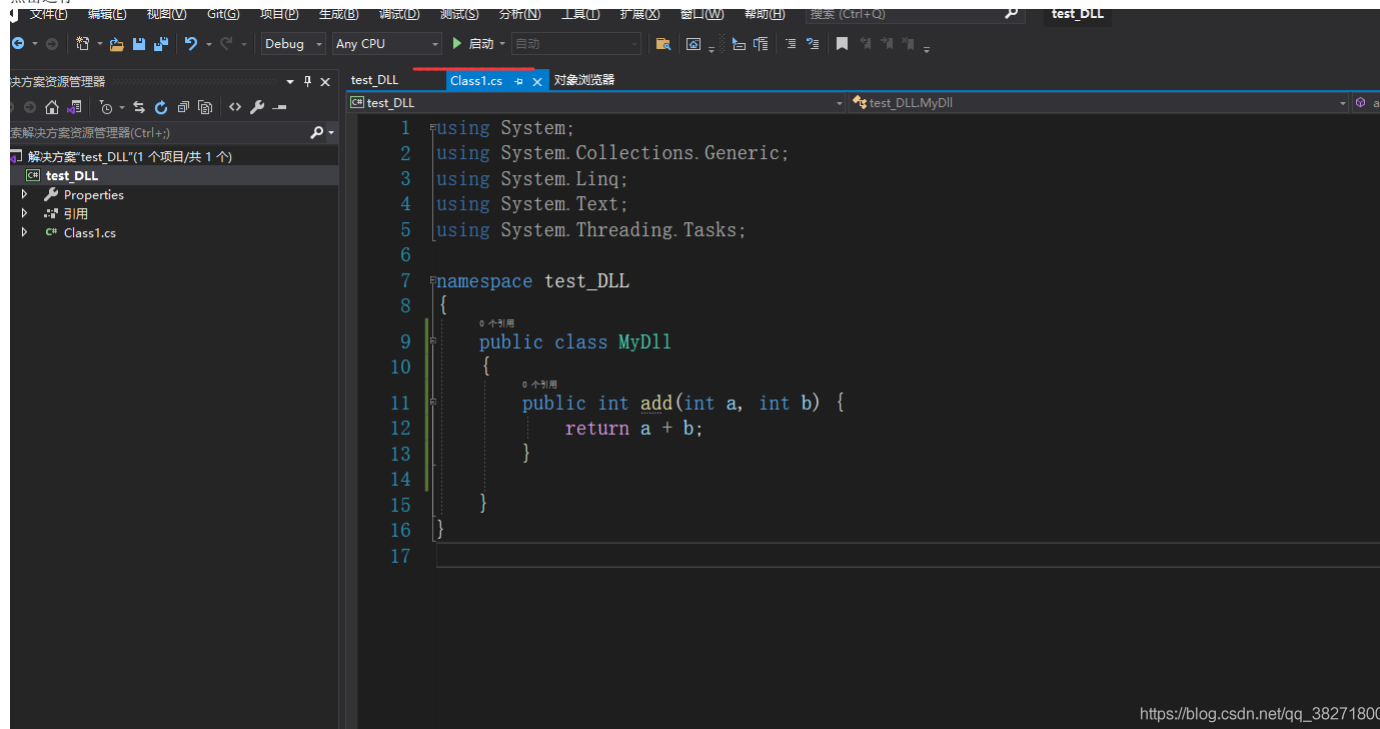
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace test_DLL
{
    public class MyDll
    {
        public int add(int a, int b) {
            return a + b;
        }
    }
}
```

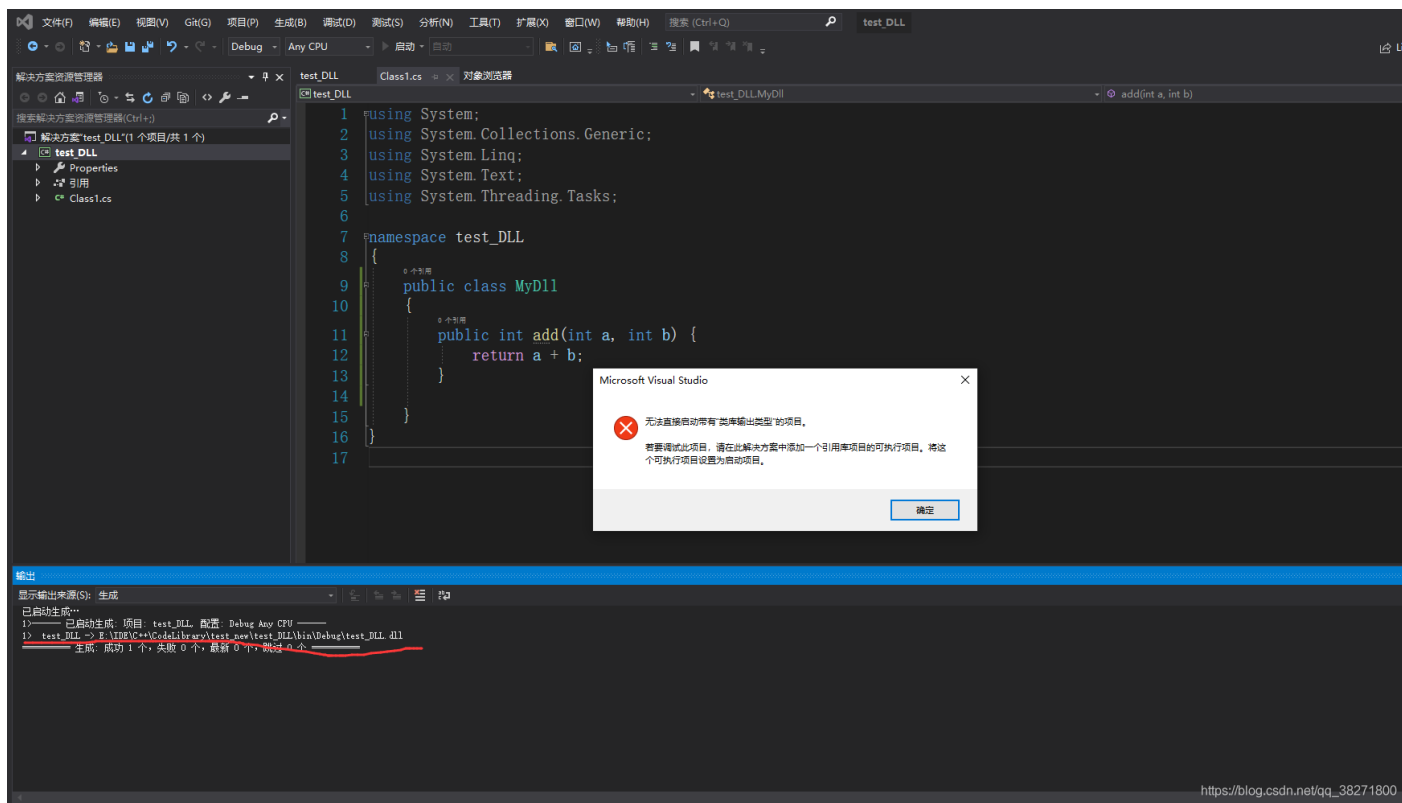
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17

### 生成dll

点击运行

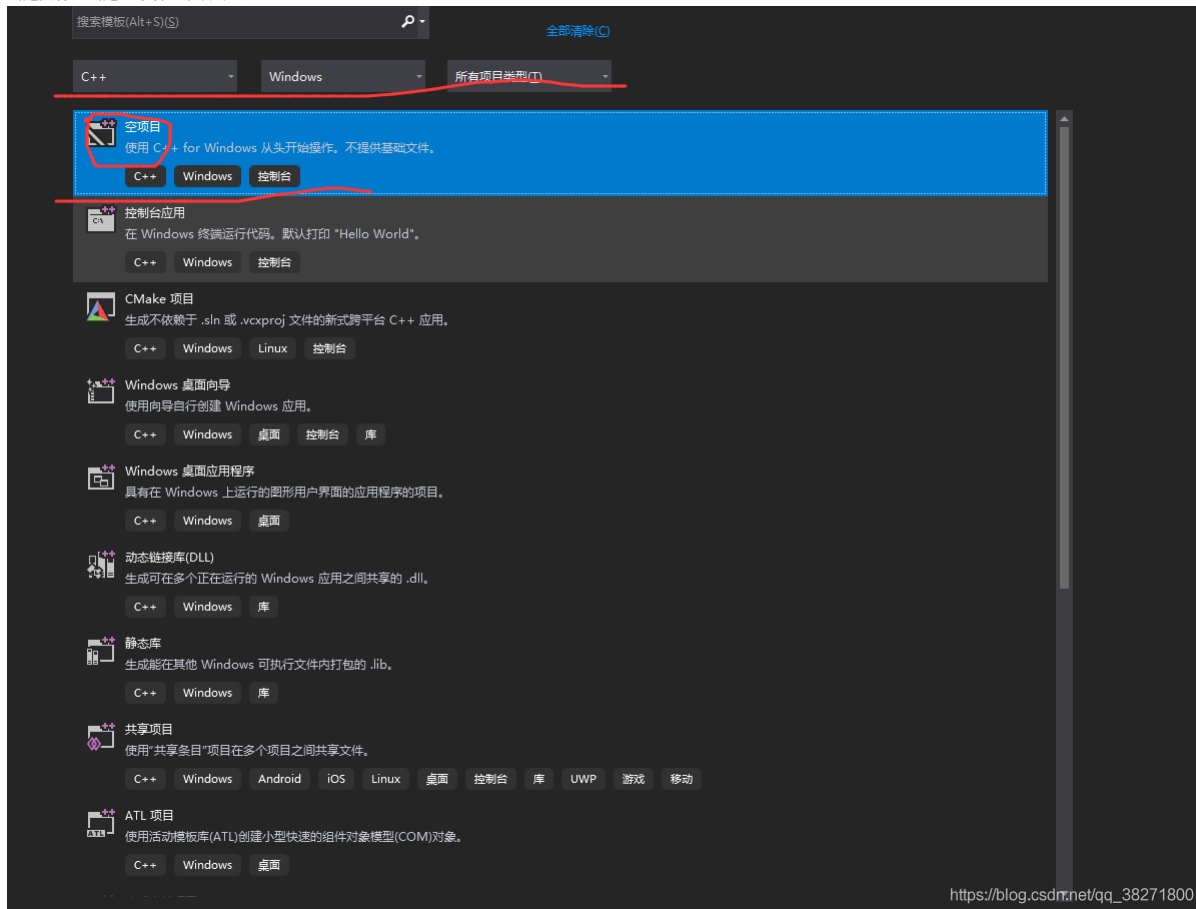


查找dll

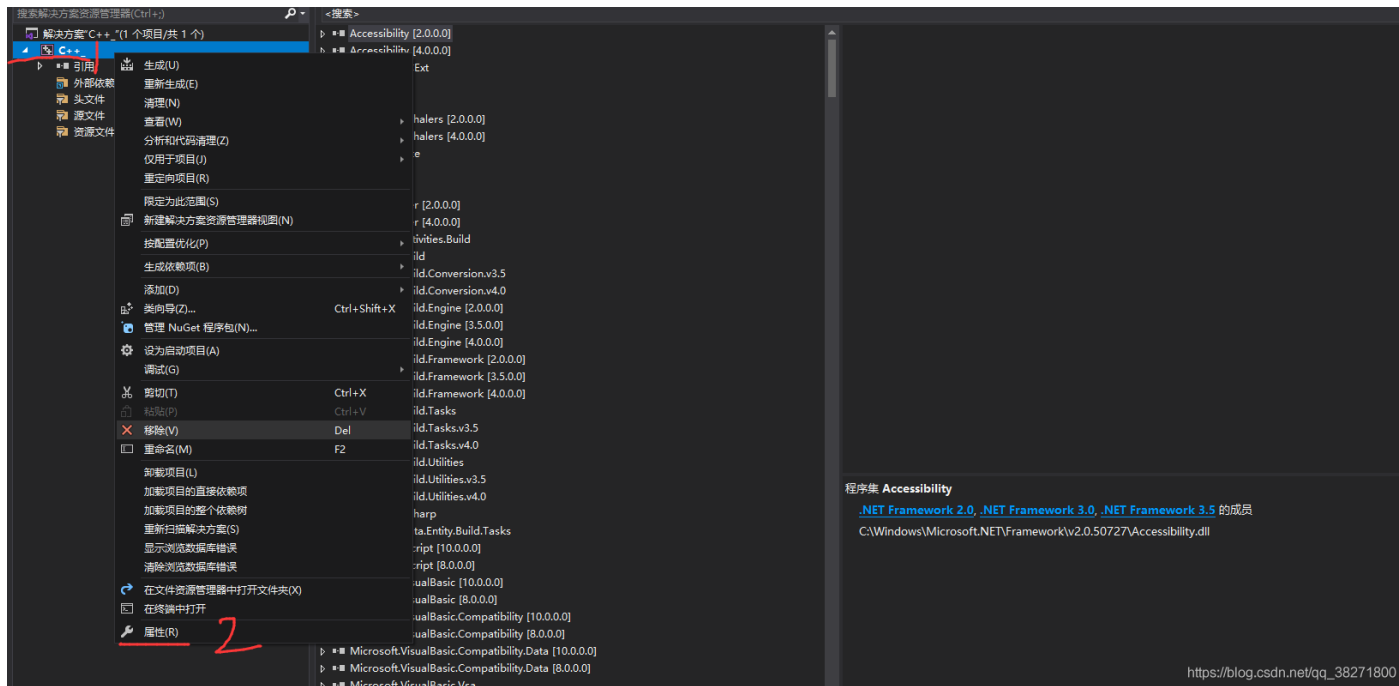


## C++使用C#dll

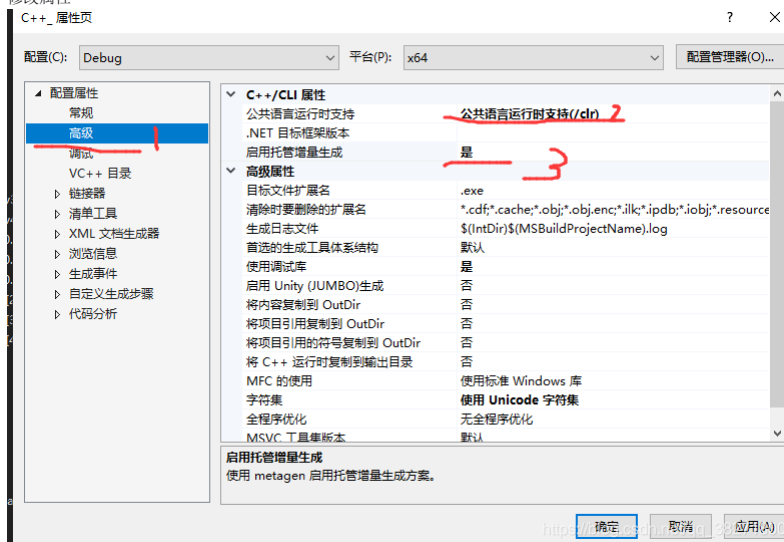
创建文件，创建一个普通项目即可



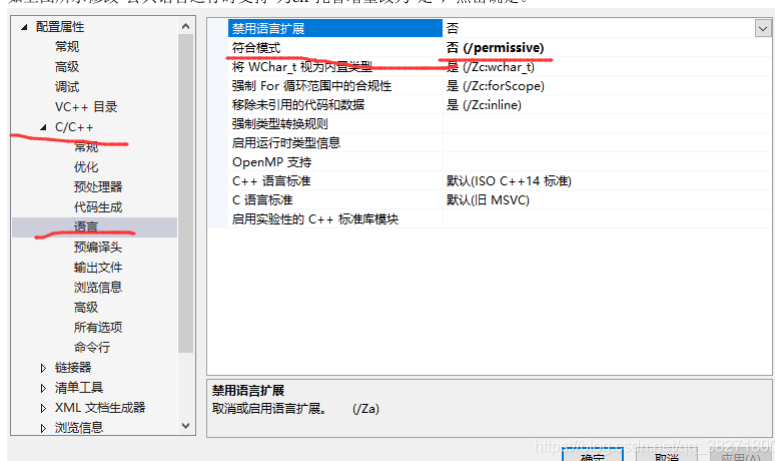
打开属性配置



修改属性

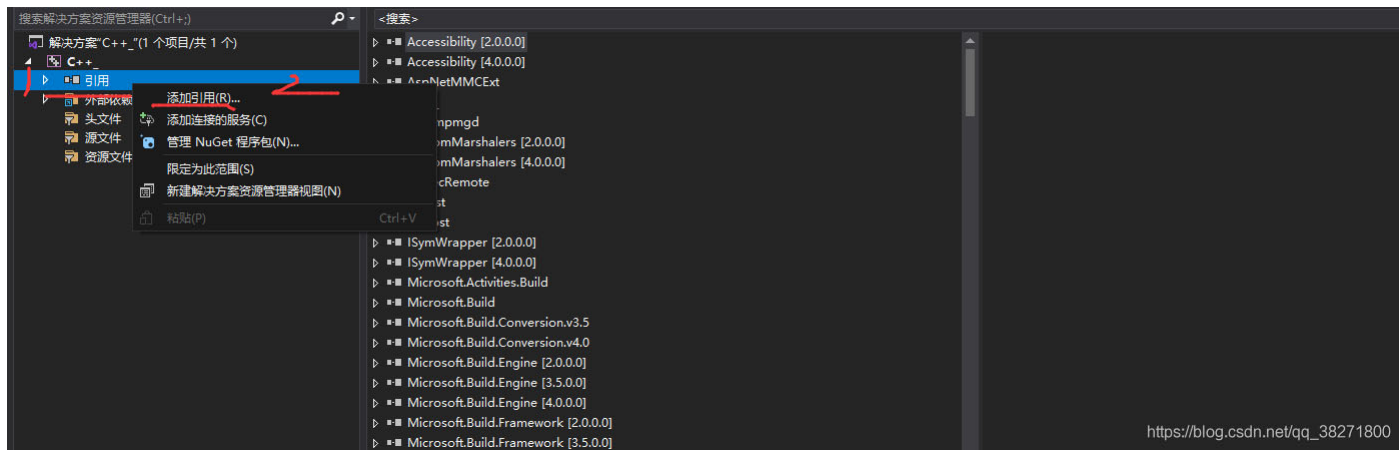


如上图所示修改“公共语言运行时支持”为clr 托管增量改为“是”，点击确定。

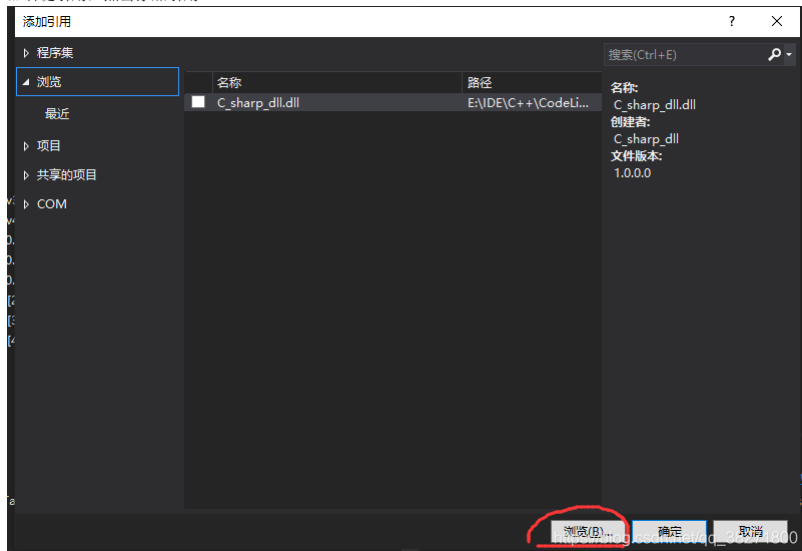


将语言中“符合模式”改为“否”

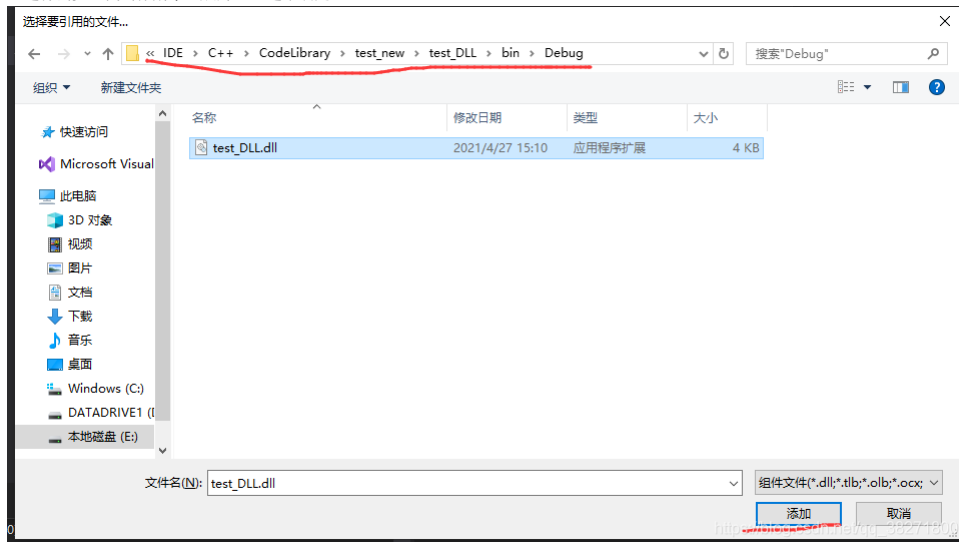
添加C#dll



a. 右键引用，点击添加引用

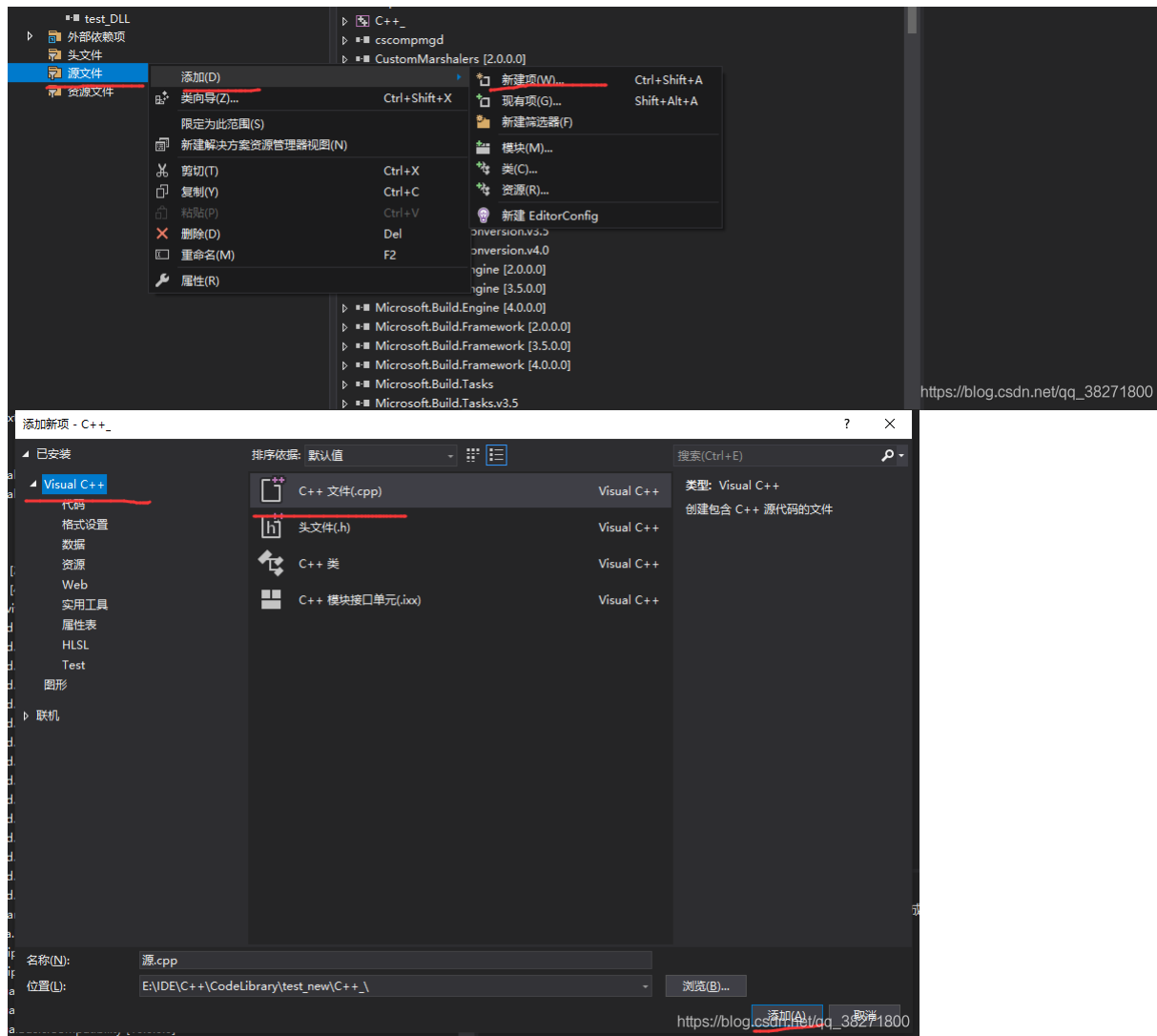


b. 选择浏览，找到你刚才生成的dll，选中确定

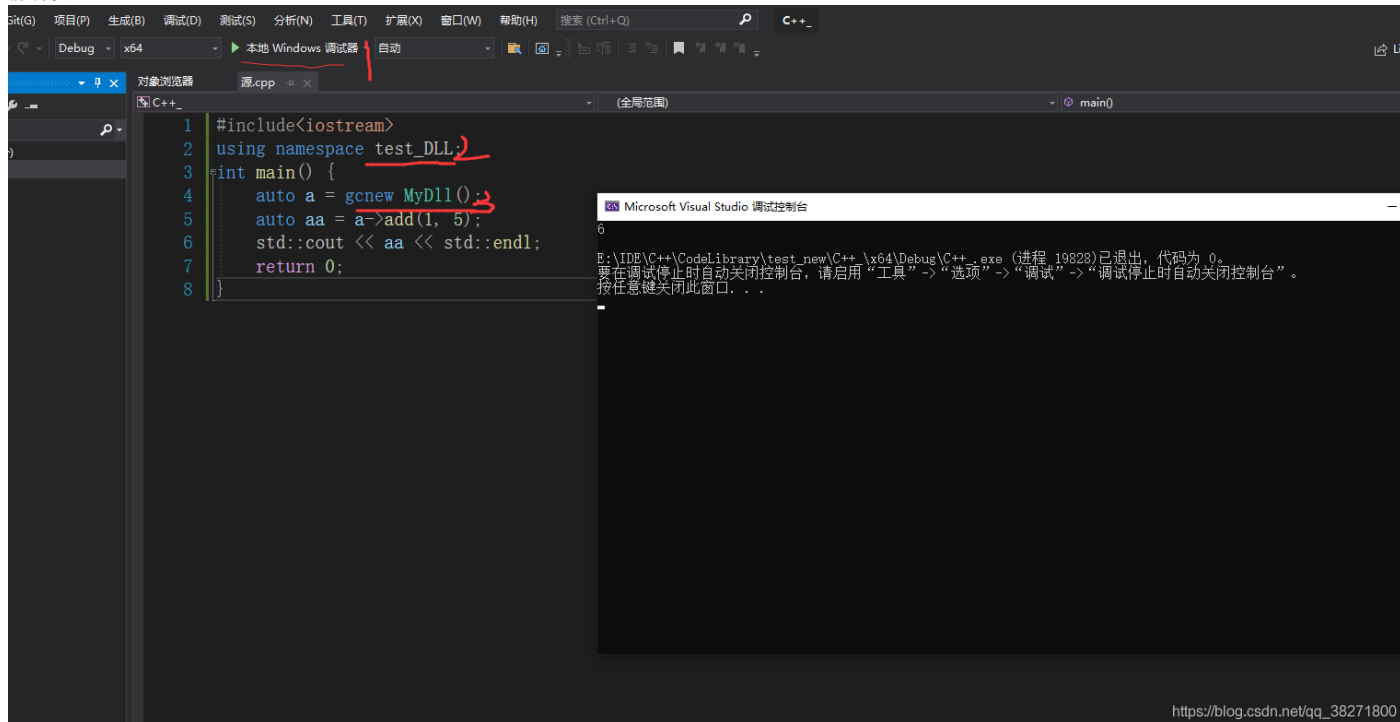


上图为我生成的dll,你们写的需要查找自己生成的dll 在什么位置

添加源文件



编写代码



上图1：运行程序；上图2：C#中你写的名字空间；上图3：C#中你写的类名

```

#include<iostream>
using namespace test_DLL; // C#中名字空间
int main() {
    auto a = gcnew MyDll(); // C#中类名
    auto aa = a->add(1, 5); // C#中函数
    std::cout << aa << std::endl;
}

```



```
// 第二种写法
MyDll^ b = gcnew MyDll();
int bb = b->add(7, 8);
std::cout << bb << std::endl;
return 0;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

**Seccessful!**