

# C++课程设计教程

本课程设计涉及到的教学内容：

- 1.C++连接MySQL数据库。
- 2.使用MFC实现前端。

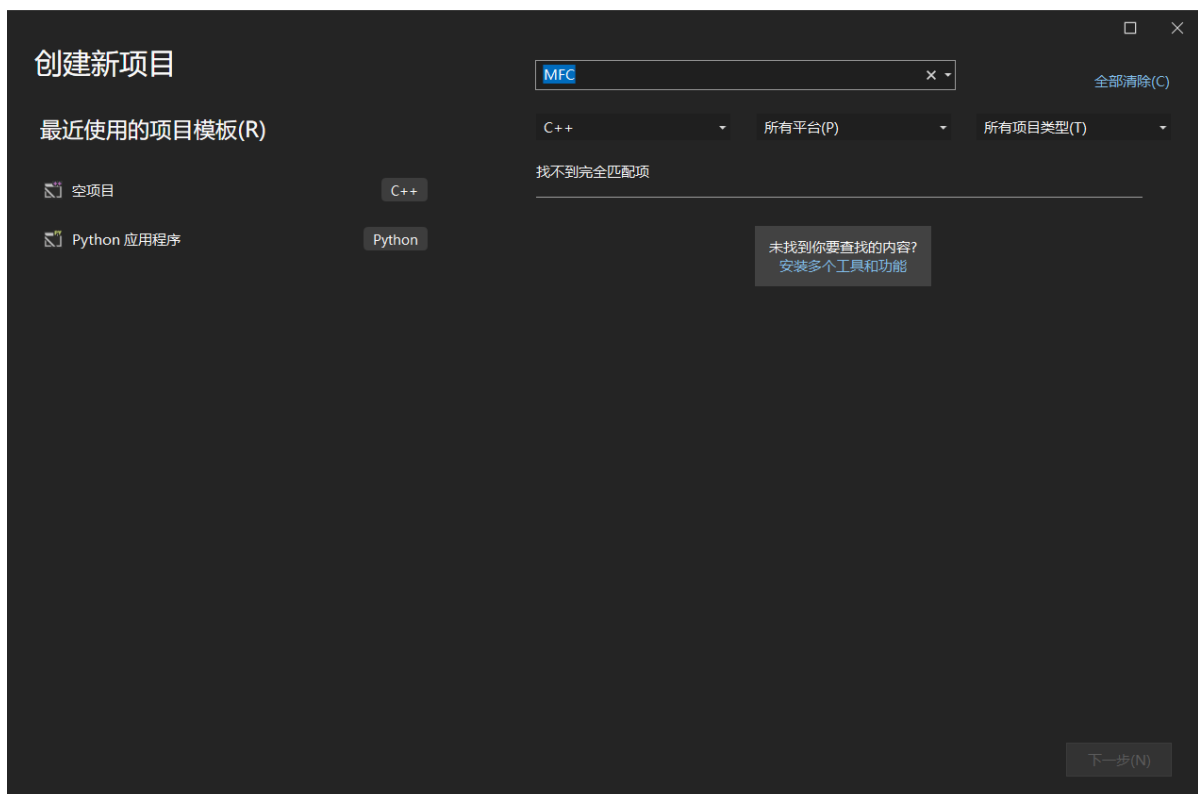
需要预先准备的工作：

- 1.使用的开发工具为 Visual Studio 2022，别的版本大概也可行，可以自己尝试一下。
- 2.使用的数据库为mysql，版本为8.2.0，别的版本也可以。需要提前去看一下自己的mysql装到哪里了，后面配置环境会用到。
- 3.使用的数据库可视化工具为datagrip，这个没有影响，使用一个自己觉得顺手的工具即可。
- 4.VS2022中MFC插件的安装，这个放到下面MFC的地方一起讲。

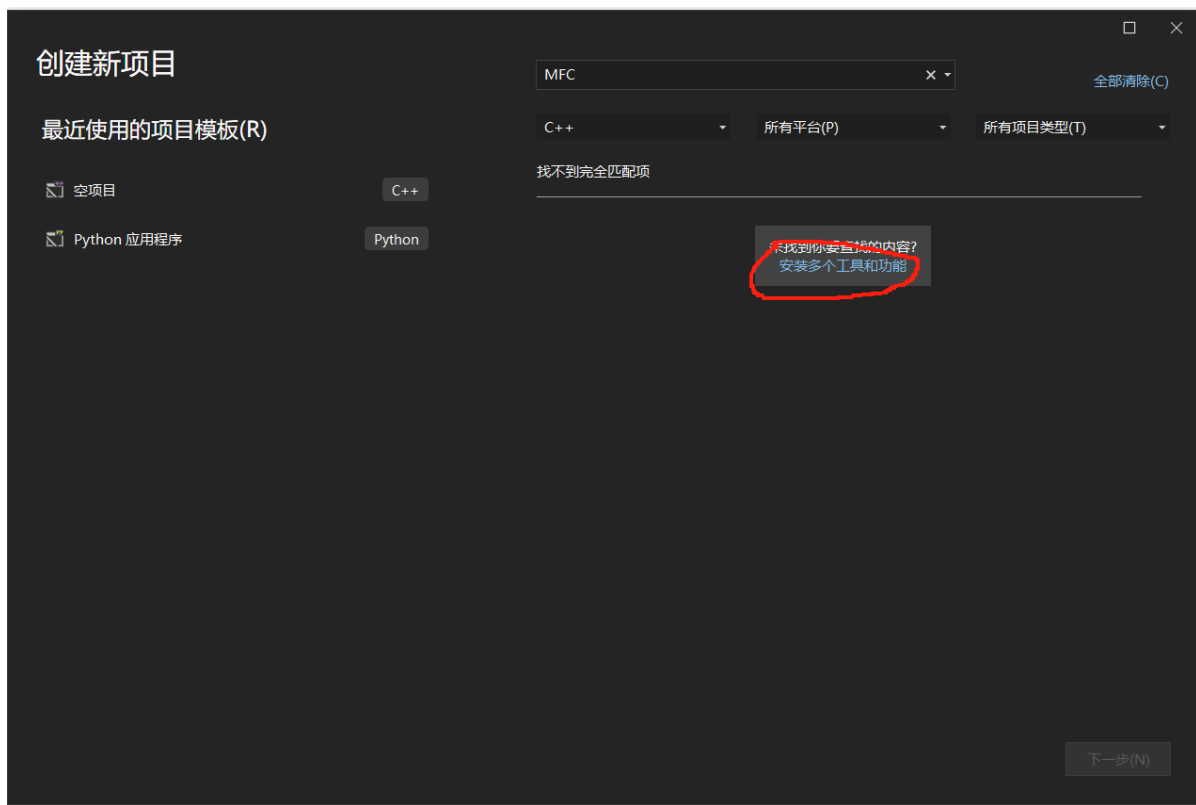
## 一、MFC

### 1.安装VS2022的MFC插件

打开VS2022，选择创建新项目。正常是没有MFC项目的。



点击安装多个工具和功能



勾选VS拓展开发，适用于最新v143生成工具的C++MFC...两项，点击修改

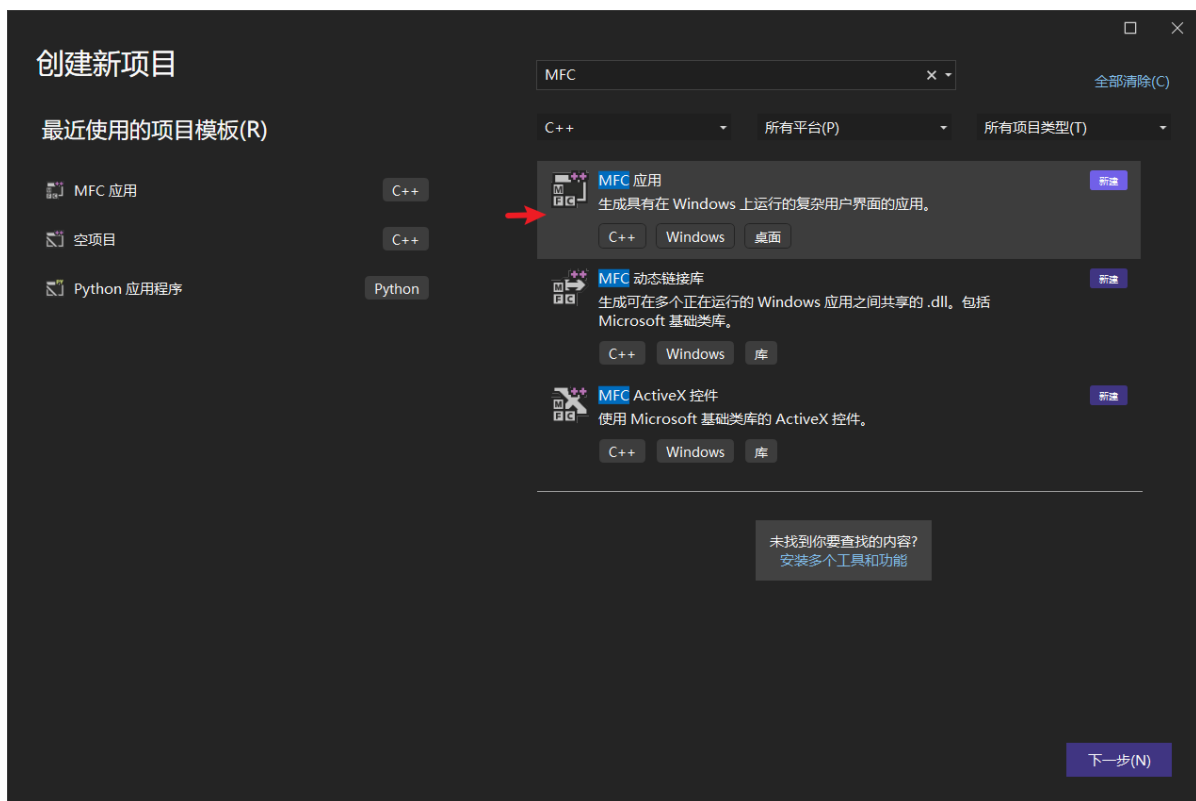


稍等片刻，等待下载安装完成

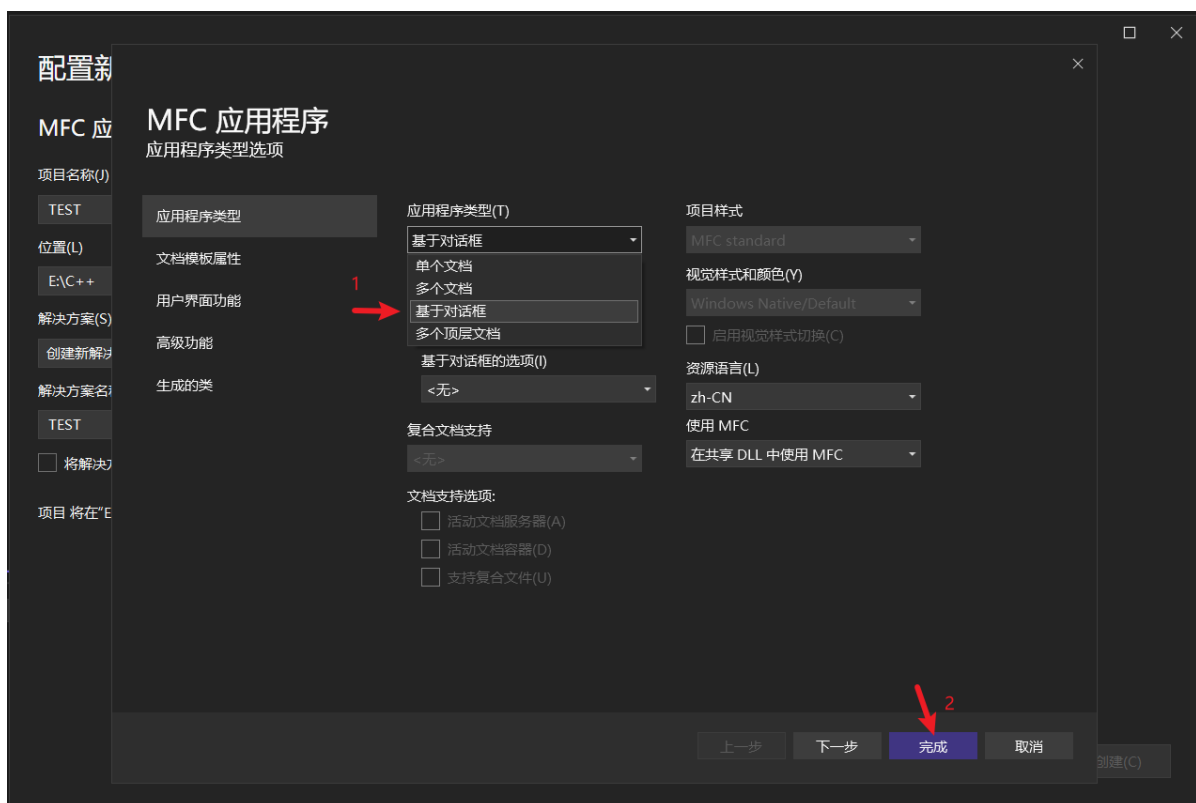
...

## 2.新建MFC项目

回到新建项目界面，就可以新建MFC项目了

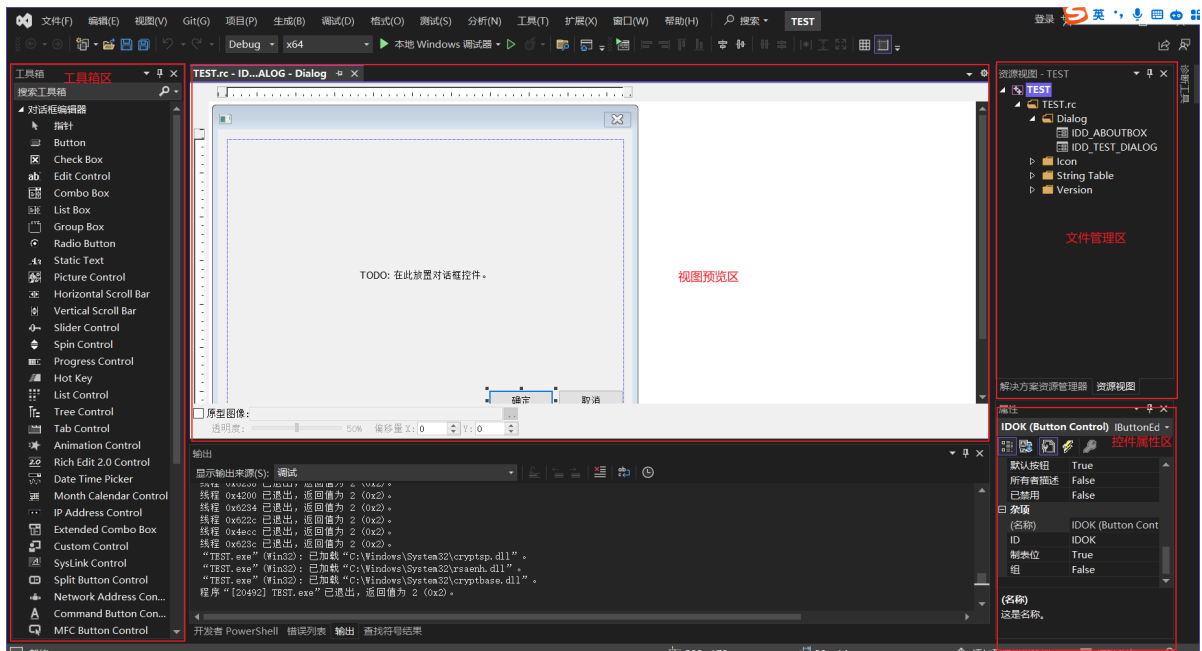


接着往下来到啊这个页面，选择基于多个对话框，点击完成即可。



### 3.页面介绍

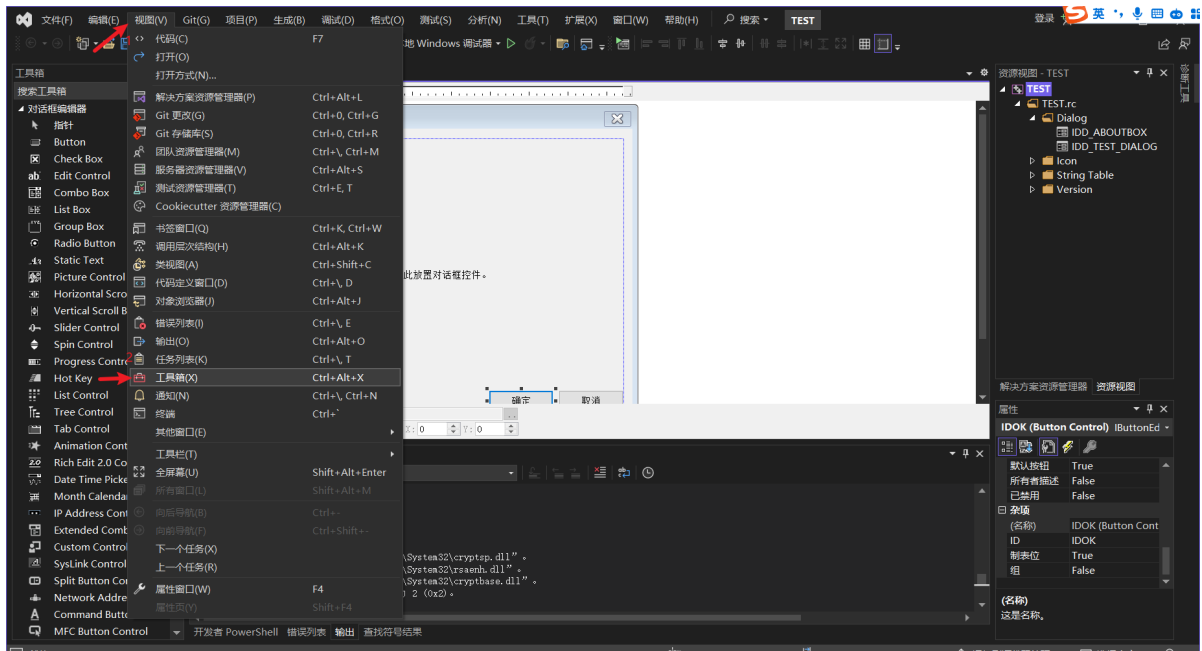
创建完成后就可以到以下页面，改页面就作为运行时系统默认打开的页面：



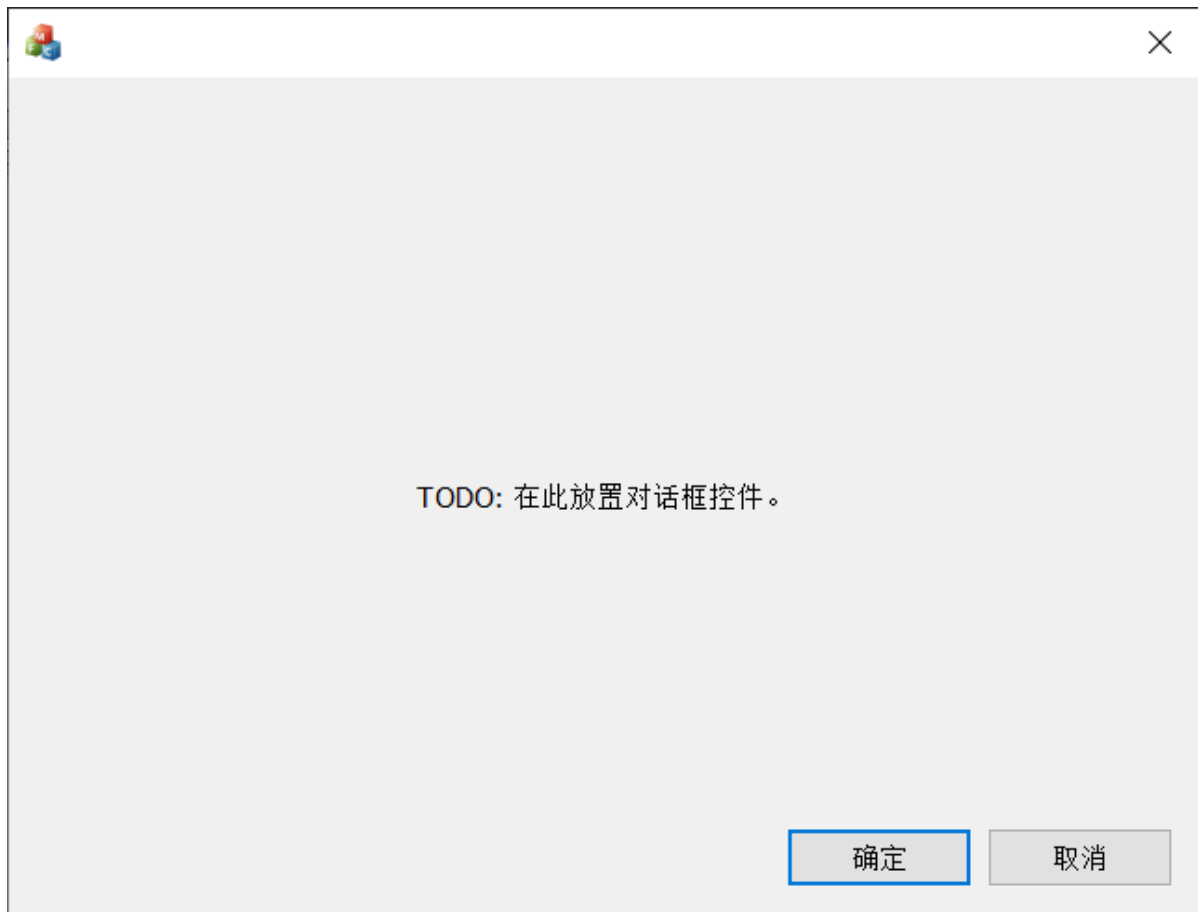
- 1) 工具箱区：包含了常用的一些控件。
- 2) 视图预览区：能够大致浏览你当前制作的页面的样式。
- 3) 文件管理区：能够对当前视图文件，以及C++代码文件等进行管理
- 4) 控件属性区：点击一个控件后在此界面显示该控件。需要注意的是，ID为标识该控件的属性，需要按需求设置，否则代码调用时容易搞混。

tips:

如果没有看到工具箱的话可以点击**视图》工具箱**打开。

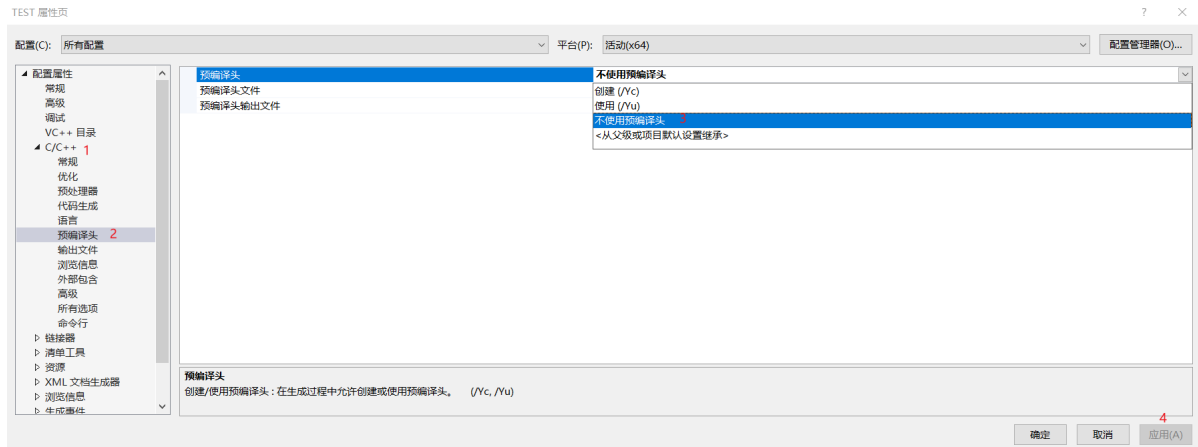


运行一下看看效果：



出来的就是这个页面。

为了减少bug的出现，我们把项目的预编译功能关闭，并删除所有的pch.h头文件。



## 4.控件介绍

现在简单讲解几个我用到的控件。

有以下：

- 1) 按钮button
- 2) 输入文本框Edit Control
- 3) 单选框Radio Button
- 4) 静态文本框 Static Text
- 5) 显示列表List Control
- 6) 页面组合Tab Control

介绍如下：

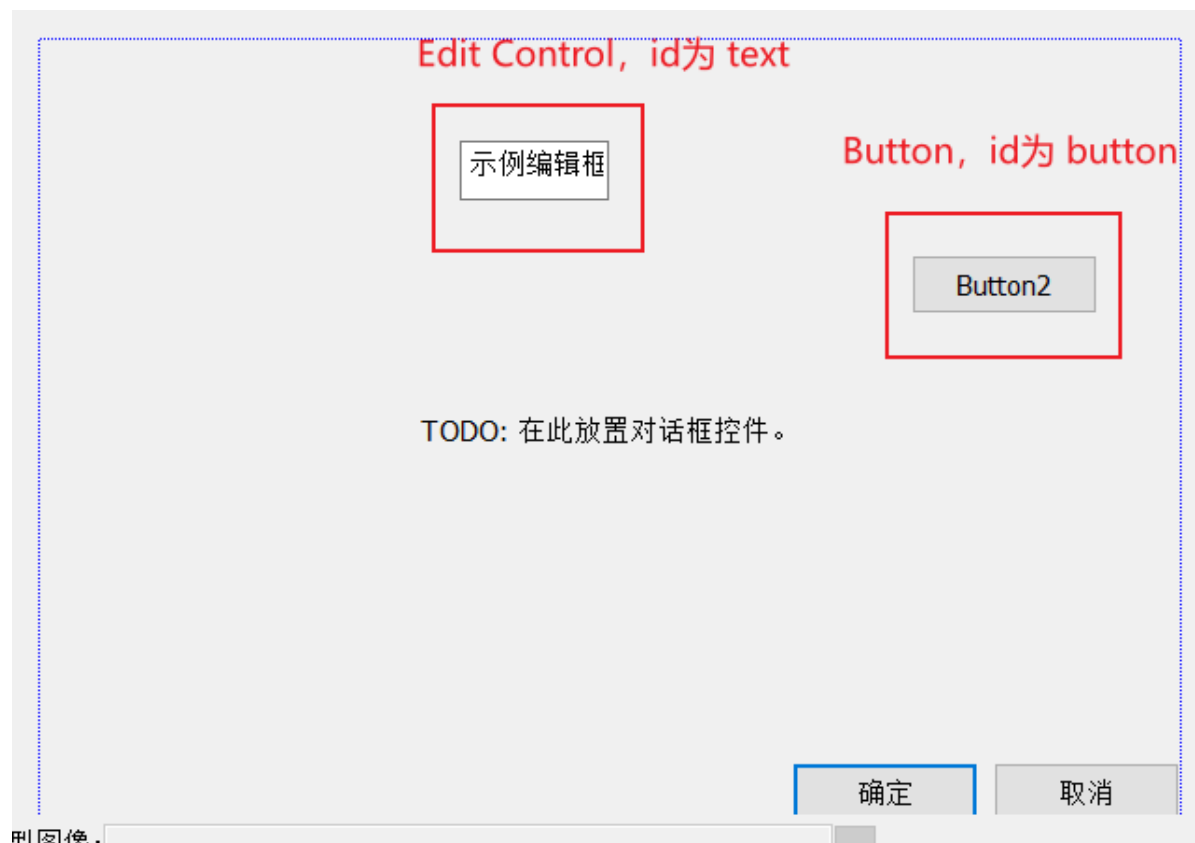
## 1) 按钮button

上手难度低，必不可少的控件。功能是单击后执行绑定的代码。

演示一下：

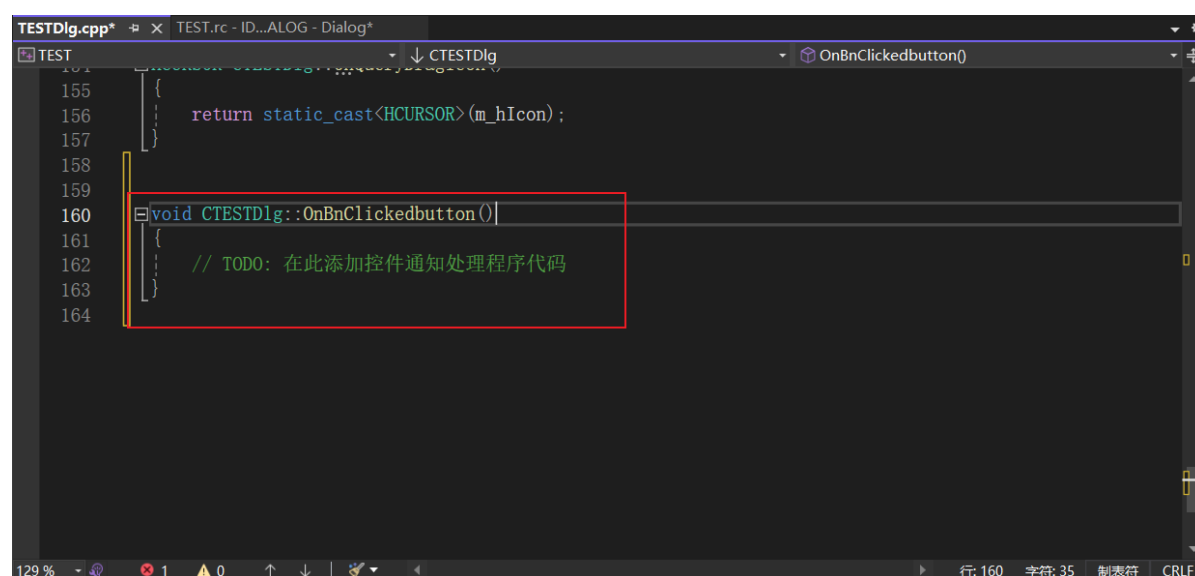
现在我想要实现：单击一个按钮后在文本框中显示“C++课设”。

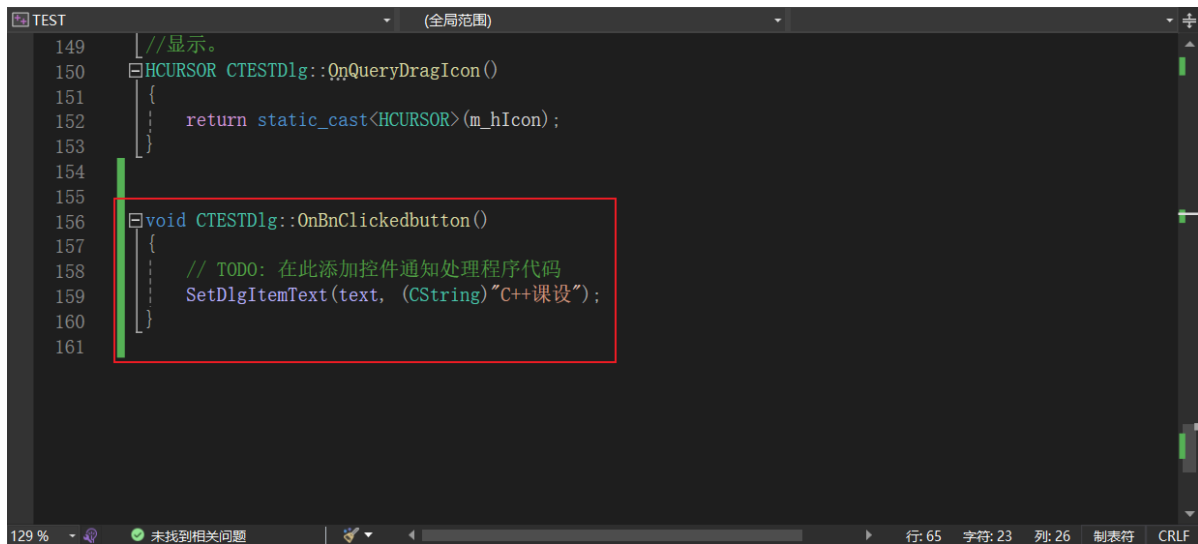
(1) 拖两个控件，Edit Control，id在右下角设置为text；Button，id设为button



型图像。

(2) 双击一下“button2”，自动创建跟这个按钮绑定的函数



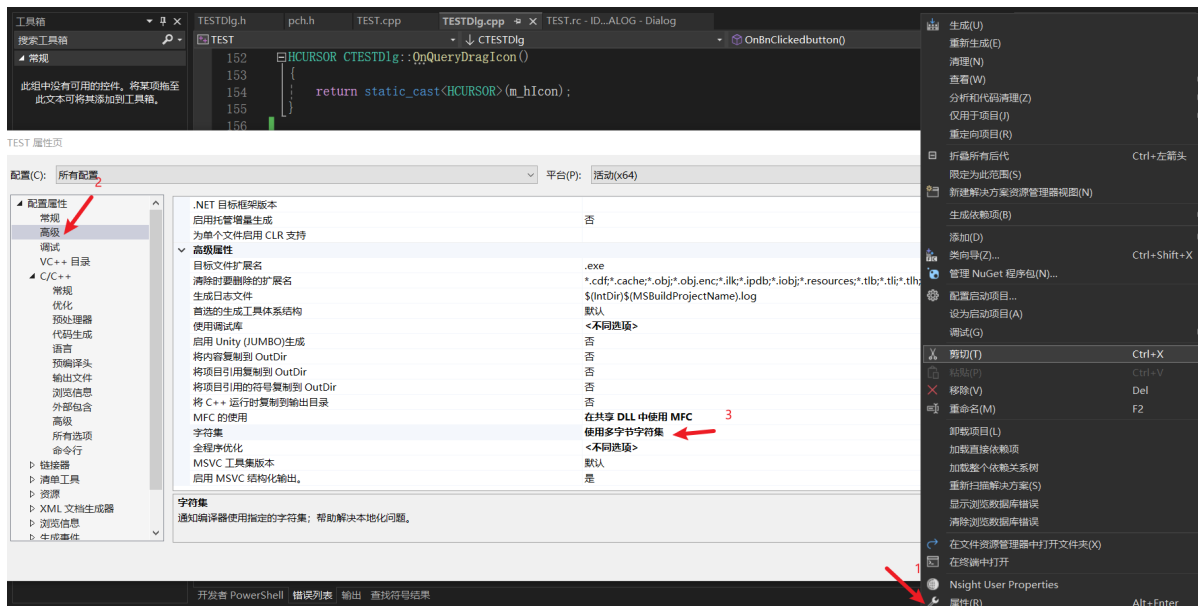


```
1 void CTESTDlg::OnBnClickedbutton()
2 {
3     // TODO: 在此添加控件通知处理程序代码
4     SetDlgItemText(text, (CString)\"C++课设\");
5 }
```

tips: MFC中有专用的字符串变量CString, 与MFC交互时需要转化成该样式才行哦!

先配置下环境

右键项目》属性》高级》使用多字节字符集》应用



```
1 CString c;
2 string s;
3 c=s.GetBuffer();//string->CString
4 s=c.c_str();    //CString->string
```



到这里想必大家对按钮有了初步了解。只要点击按钮就能运行对应函数内的语句。

## 2) 输入文本框Edit Control

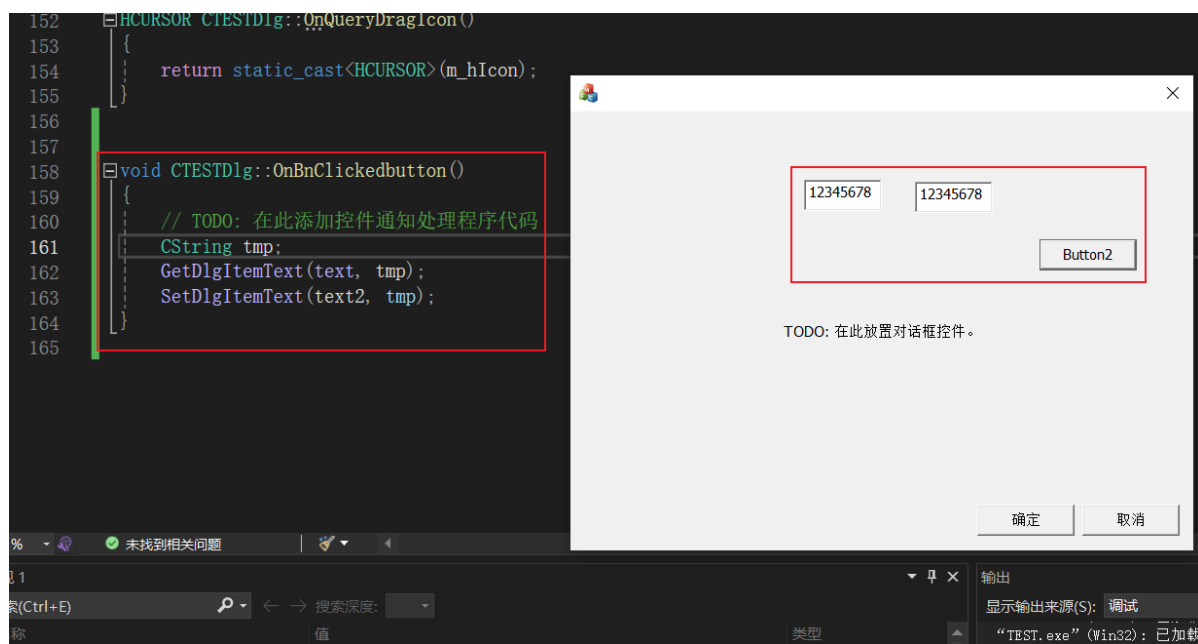
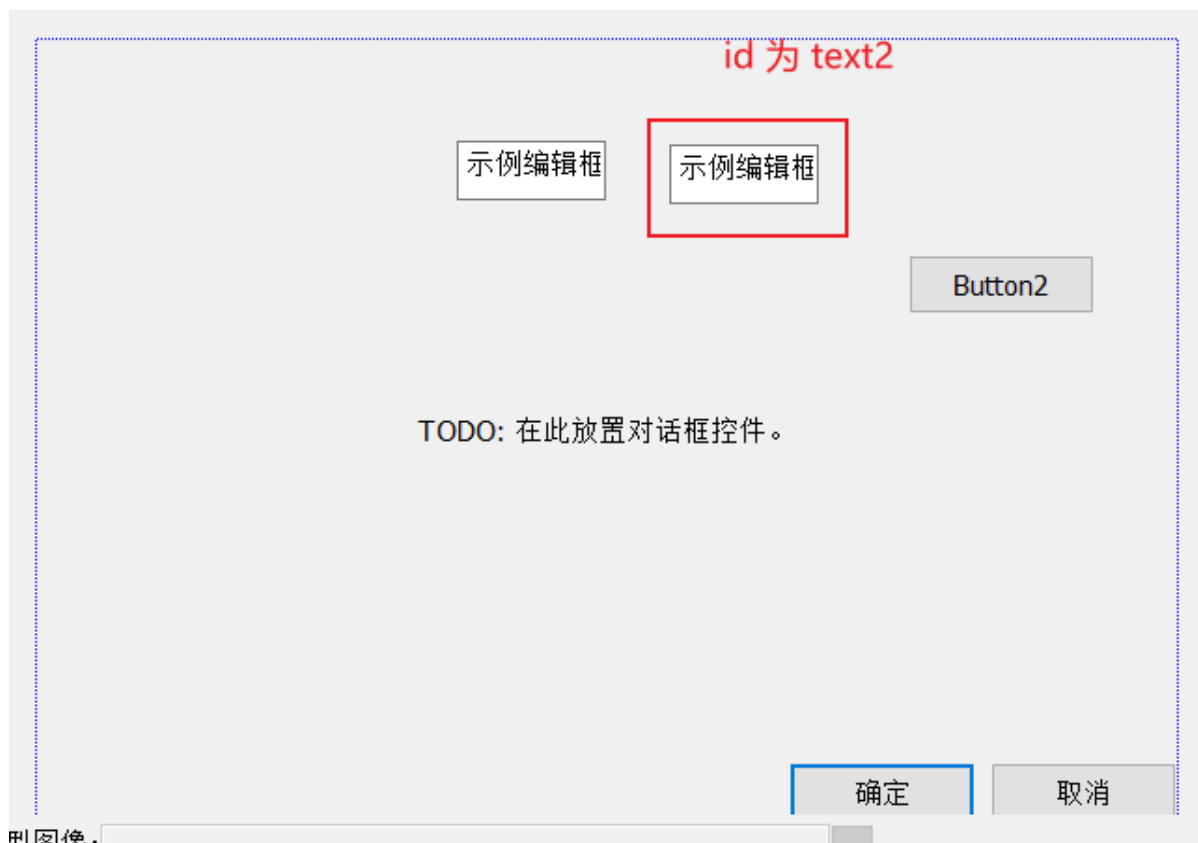
在 1) 中已经讲解了怎么使用文本框显示输出文字。它的主要作用其实是用来输入文字。在右下角的控件属性中还可以更改样式：例如，像输入密码，输入的文字都被隐藏，或者显示不可编辑只可显示等，这个大家自己研究一下。

接下来看下怎么获取到输入的文字：

我们新加一个Edit Control，id为text2

需求为，按下按钮后从左边的文本框中读取输入的字符输出到右边的文本框中。





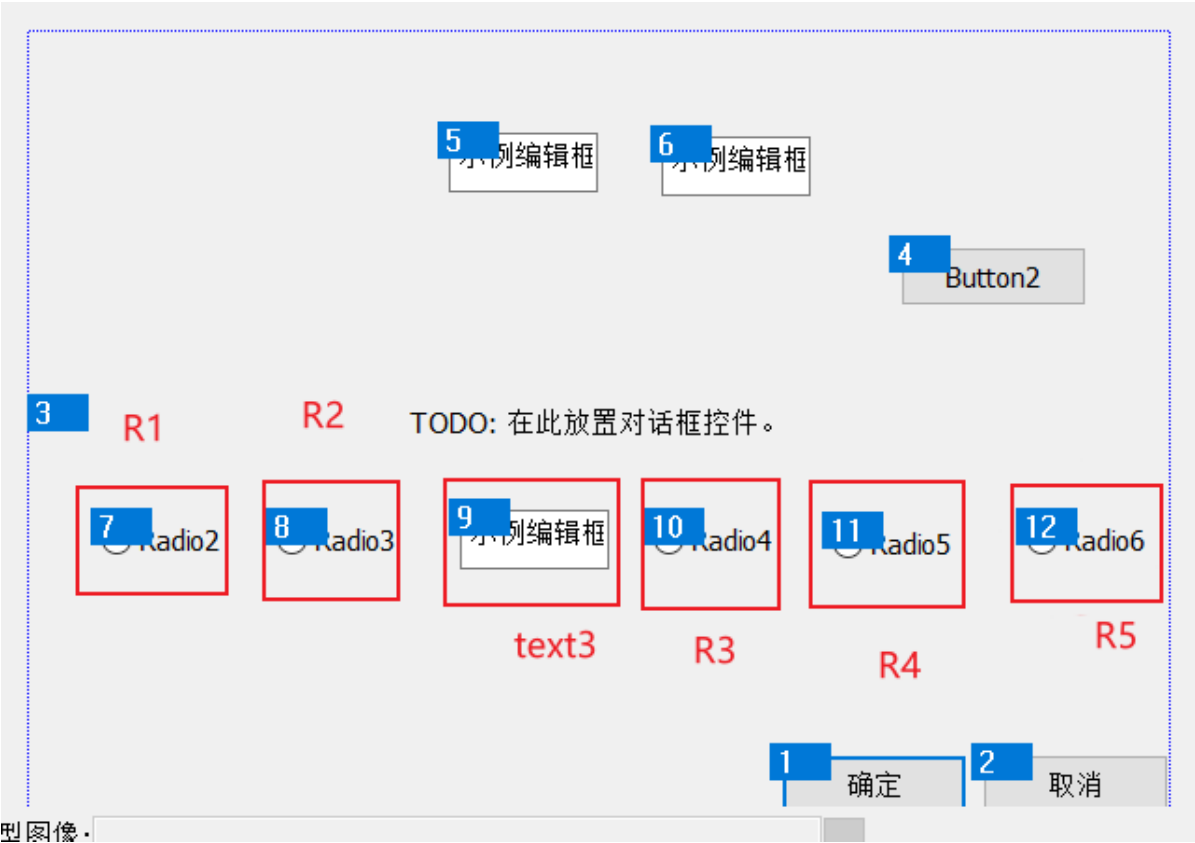
```
1 void CTestDlg::OnBnClickedbutton()  
2 {  
3     // TODO: 在此添加控件通知处理程序代码  
4     CString tmp;  
5     GetDlgItemText(text, tmp);  
6     SetDlgItemText(text2, tmp);  
7 }
```

### 3) 单选框Radio Button

单选框用的相对少些

tips:  
在预览界面按Ctrl+D就能看到控件摆放顺序的编号（注意这个编号和ID是不一样的），如下图蓝框白字。

我们创建5个radio和一个edit，id如下图



radio分组的条件如下：  
我们单击一个单选标签，能看到右下角的一条属性为“组”，这个就是分组的标识



当将“组”状态选为True时，意思为当前组从这个控件开始，并且接下来连续且组状态不为True的单选控件为它的组员。注意这个连续被其他控件打断是没关系的。

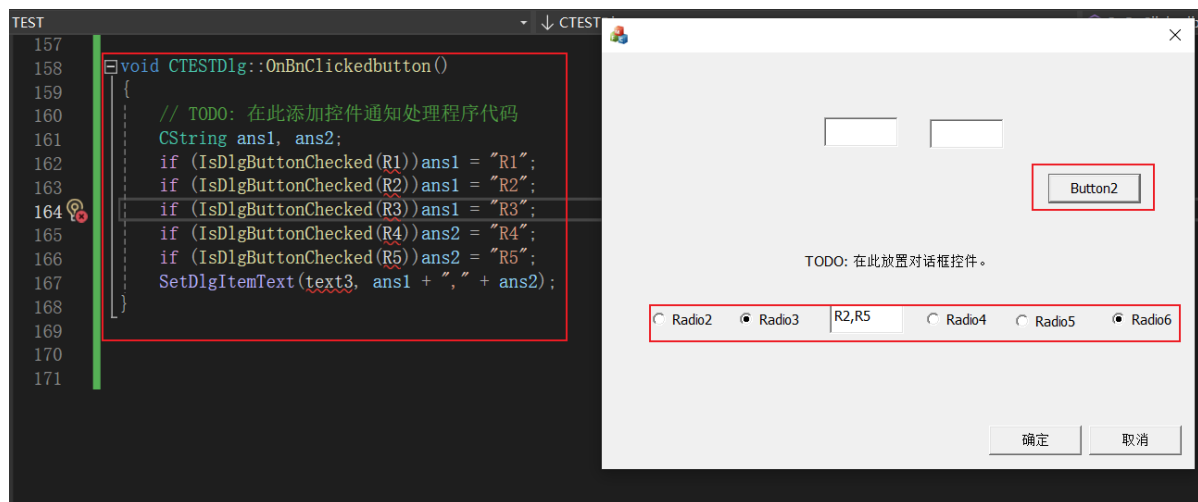
例如，上图中，假如把R1和R4的“组”状态设为True，那么最后分组的情况是：R1，R2，R3为一组，R4，R5为一组。大家可以自己试一下。

接下来来实验一下：

需求：

分组就按以上描述，选中两个组的成员，在框中显示组的ID，格式为XX，XX。

```
1 void CTESTDlg::OnBnClickedbutton()
2 {
3     // TODO: 在此添加控件通知处理程序代码
4     CString ans1, ans2;
5     if (IsDlgButtonChecked(R1)) ans1 = "R1";
6     if (IsDlgButtonChecked(R2)) ans1 = "R2";
7     if (IsDlgButtonChecked(R3)) ans1 = "R3";
8     if (IsDlgButtonChecked(R4)) ans2 = "R4";
9     if (IsDlgButtonChecked(R5)) ans2 = "R5";
10    SetDlgItemText(text3, ans1 + "," + ans2);
11 }
```



编译器报红的地方都没有问题，编译器对于ID的读取速度还是慢了些。

#### 4) 静态文本框 Static Text

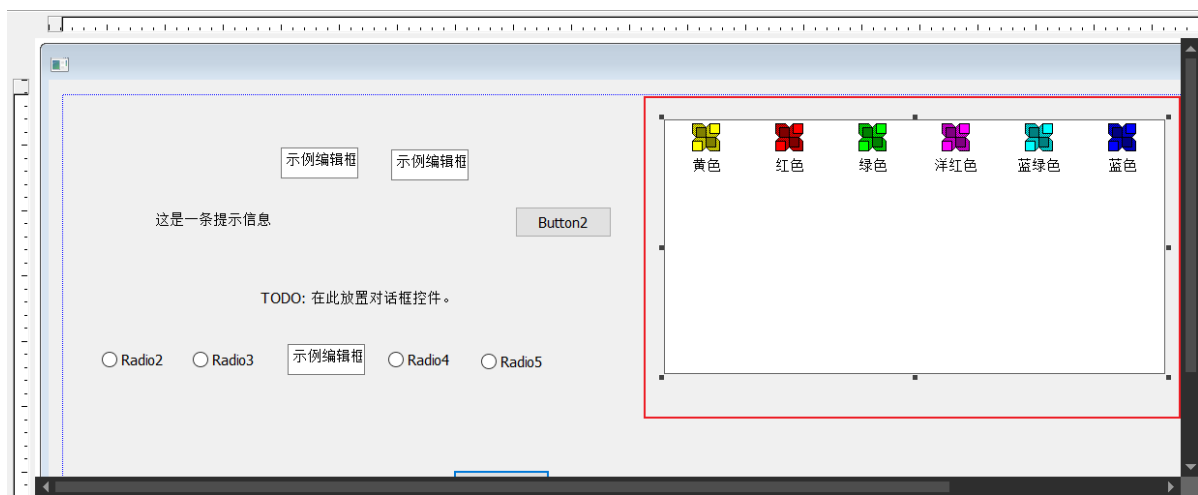
使用很简单，就是用来显示提示信息的，这个信息不可修改。



## 5) 显示列表List Control

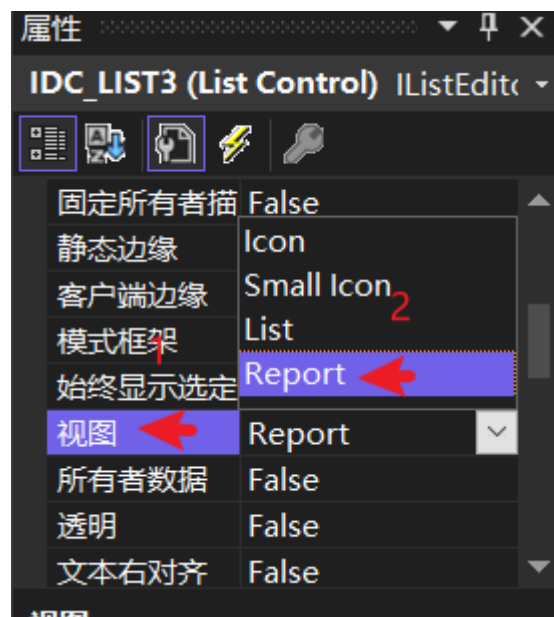
这个控件的上手难度比以上的略高。一般用来显示查询数据的结果

这个东西默认长这样

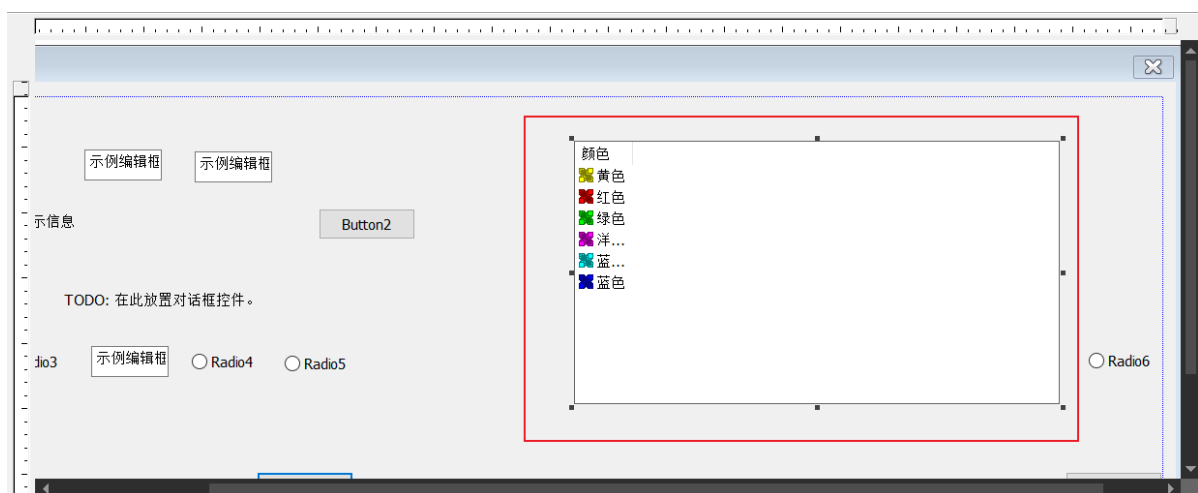


和我们想的不一样，可以改一下样式，单击一下，进入到属性栏。

选择视图》Report



改完之后就变成这样了



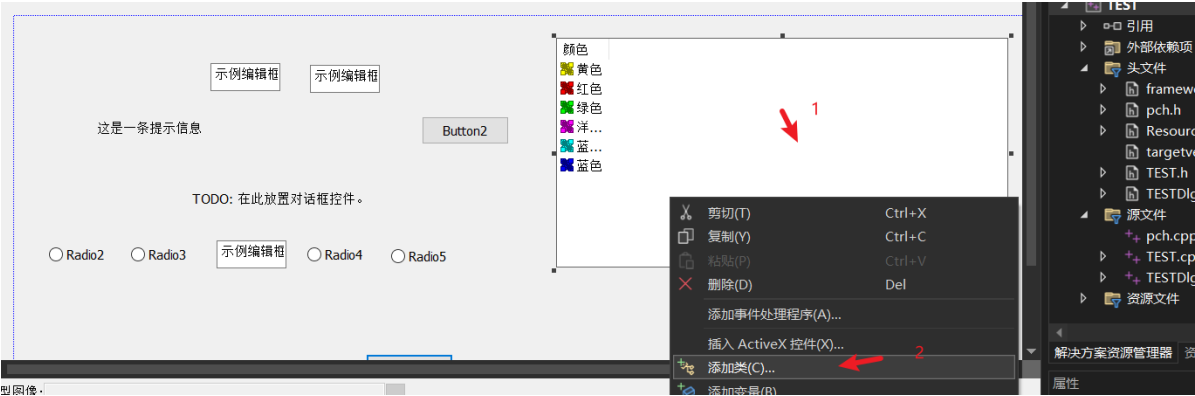
上面是提示信息，下面是数据。

下面来编写一下代码，需求：

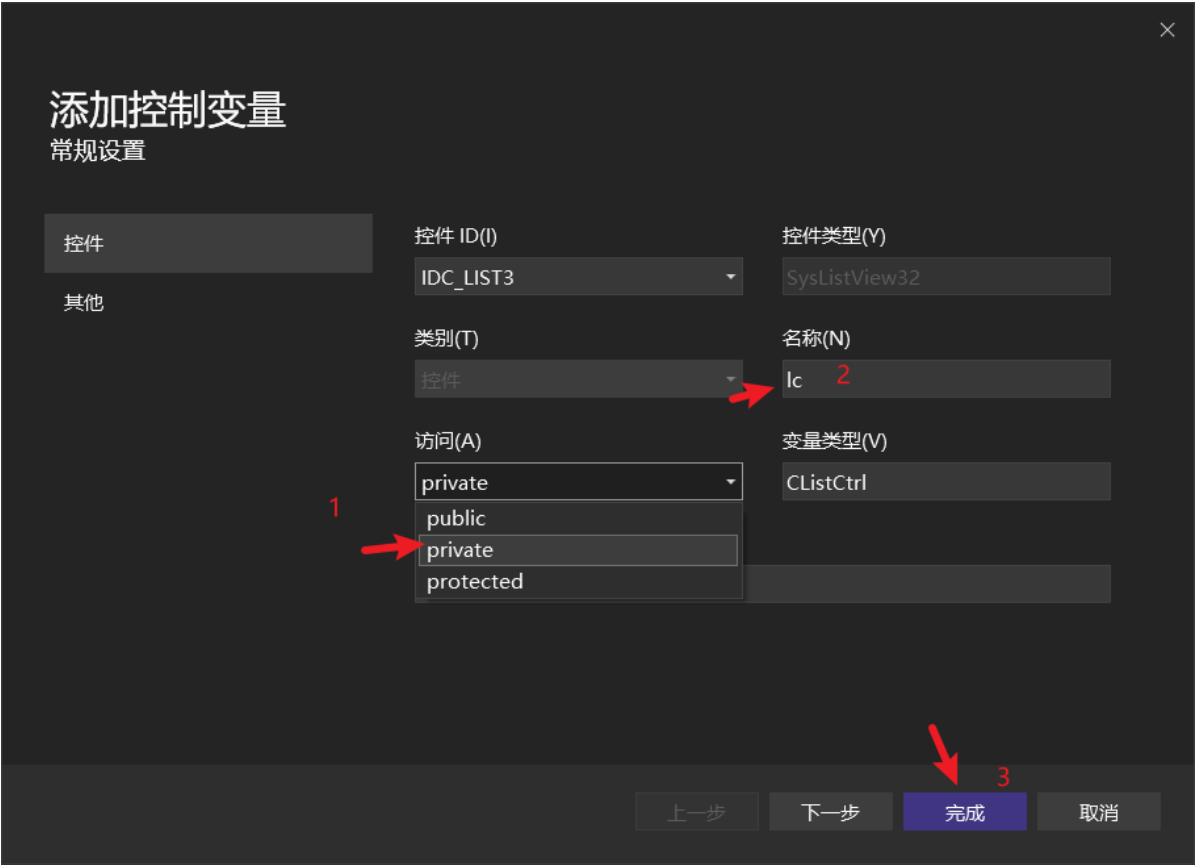
单击按钮后，显示一个表格，表格的提示信息显示这是第几列。5行4列，每行显示行号×列号。

这个操作比较特殊，我们要添加一个用来专门操作这个表格的对象。

右键这个表格》添加变量



把这个变量设置成private的，防止别的页面变量重名。然后将这个表格的变量名设置成ls(都行，自己看吧🐼)



tips:

我在做的时候遇到了这个问题，但是复刻的时候没有，就贴一下网上的图吧。



这种问题有两种方法可以解决。

第一种方法：

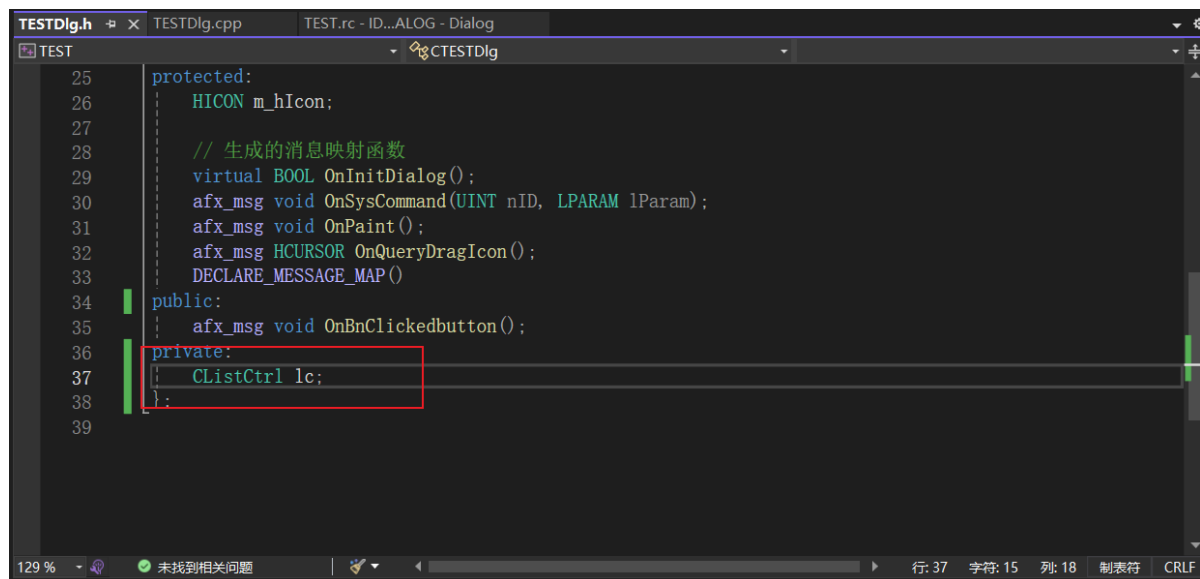
下面是一个网址，对应第一种方法的解决方案，可以去看一下

```
1 https://blog.csdn.net/zahngzhic/article/details/124924803?
ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%8E%A7%E4%BB%B6%E5%A2%9E%
E5%8A%A0%E5%8F%98%E9%87%8F%E5%A4%B1%E8%B4%A5%EF%BC%8C%E6%98%BE%E7%A4%BA%EF%BC%
9A%E6%9C%AA%E8%83%BD%E8%BF%94%E5%9B%9E%E6%96%B0%E4%BB%A3%E7%A0%81%E5%85%83%E7%
B4%A0%20%E5%8F%AF%E8%83%BD%E6%98%AF%E8%AF%AD%E6%B3%95%E9%94%99%E8%AF%AF&utm_me
dium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-4-
124924803.142^v96^pc_search_result_base1&spm=1018.2226.3001.4187
```

第二种方法：

手动添加代码：

在视图对应的头文件下添加下图红框的代码。注意要在private下。



```
1 CListCtrl lc;
```

在视图对应cpp文件下的一个函数下添加以下代码：

```
TESTDlg.h TESTDlg.cpp TEST.rc - ID...ALOG - Dialog
TEST
// CTESTDlg 对话框

CTESTDlg::CTESTDlg(CWnd* pParent /*=nullptr*/)
: CDialogEx(IDD_TEST_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CTESTDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_LIST3, lc);
}

BEGIN_MESSAGE_MAP(CTESTDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()

```

```
1 DDX_Control(pDX, IDC_LIST3, lc);
```

然后就能看见在这个这个视图对应头文件的地方，它帮你创建好了这个成员，手动添加也行，见上。

```
TESTDlg.h TESTDlg.cpp TEST.rc - ID...ALOG - Dialog
TEST
protected:
    HICON m_hIcon;

    // 生成的消息映射函数
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()

public:
    afx_msg void OnBnClickedbutton();

private:
    CListCtrl lc;
};

```

写代码实现功能：

```
void CTESTDlg::OnBnClickedbutton()
{
    // TODO: 在此添加控件通知处理程序代码

    CString title[] = { TEXT("第0列"), TEXT("第1列"), TEXT("第2列"), TEXT("第3列") };
    lc.DeleteAllItems(); //先清空表中的全部元素，不包括表提示信息
    for (int i = 0; i < 4; i++) lc.InsertColumn(i, title[i], LVCFMT_LEFT, 120); //插入提示信息，设置为左对齐，列宽为120
    for (int i = 0; i < 5; i++)
    {
        lc.InsertItem(i, "0"); //每行需要先插入第一个元素
        for (int j = 1; j < 4; j++) lc.SetItemText(i, j, to_string(i * j).c_str()); //插入后续元素
    }
}

```

这是一条提示信息

Button2

TODO: 在此放置对话框控件。

Radio2 Radio3 Radio4 Radio5 Radio6

第0列	第1列	第2列	第3列
0	0	0	0
0	1	2	3
0	2	4	6
0	3	6	9
0	4	8	12



```

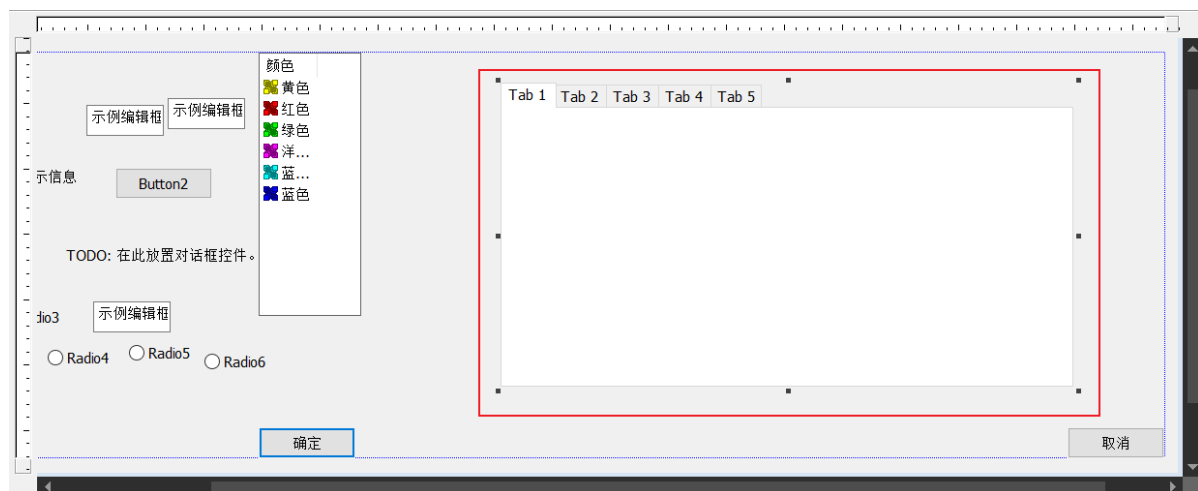
1 void CTESTDlg::OnBnClickedbutton()
2 {
3     // TODO: 在此添加控件通知处理程序代码
4
5     CString title[] = { TEXT("第0列"),TEXT("第1列"), TEXT("第2列"), TEXT("第3
列") };
6     lc.DeleteAllItems();//先清空表中的全部元素，不包括表提示信息
7     for (int i = 0; i < 4; i++)lc.InsertColumn(i, title[i], LVCFMT_LEFT,
120);//插入提示信息，设置为左对齐，列宽为120
8     for (int i = 0; i < 5; i++)
9     {
10         lc.InsertItem(i, "0");//每行需要先插入第一个元素
11         for (int j = 1; j < 4; j++)lc.SetItemText(i, j, to_string(i *
j).c_str());//插入后续元素
12     }
13 }

```

## 6) 页面组合Tab Control

该控件实现起来最难，功能是把好多个页面集成到一个页面中，并可以通过头标签进行切换。

先来看看，它长这样：



是的，只要我们单击上面的tab1和tab2就可以实现不同页面的切换了。

页面集成，并可切换的功能实现起来较为复杂。我们引用两个官方提供的封装文件以实现对多页面的管理。

文件1：

TabSheet.h

```

1 #if
2 !defined(AFX_TABSHEET_H__42EE262D_D15F_46D5_8F26_28FD049E99F4__INCLUDED_)
3 #define AFX_TABSHEET_H__42EE262D_D15F_46D5_8F26_28FD049E99F4__INCLUDED_
4
5 #if _MSC_VER > 1000
6 #pragma once
7 #endif // _MSC_VER > 1000
8 // TabSheet.h : header file
9 //
10 #include "afxdialogex.h"

```

```

10 ///////////////////////////////////////////////////////////////////
11 /
12 // CTabSheet window
13 #define MAXPAGE 16
14
15 class CTabSheet : public CTabCtrl
16 {
17     // Construction
18 public:
19     CTabSheet();
20
21     // Attributes
22 public:
23
24     // Operations
25 public:
26
27     // Overrides
28     // Classwizard generated virtual function overrides
29     //{AFX_VIRTUAL(CTabSheet)
30     //}AFX_VIRTUAL
31
32     // Implementation
33 public:
34     int GetCurSel();
35     int SetCurSel(int nItem);
36     void Show();
37     void SetRect();
38     BOOL AddPage(LPCTSTR title, CDialog* pDialog, UINT ID);
39     virtual ~CTabSheet();
40
41     // Generated message map functions
42 protected:
43     LPCTSTR m_Title[MAXPAGE];
44     UINT m_IDD[MAXPAGE];
45     CDialog* m_pPages[MAXPAGE];
46     int m_nNumOfPages;
47     int m_nCurrentPage;
48     //{AFX_MSG(CTabSheet)
49     afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
50     afx_msg void OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);
51     //}AFX_MSG
52
53     DECLARE_MESSAGE_MAP()
54 };
55
56 ///////////////////////////////////////////////////////////////////
57 /
58 //{{AFX_INSERT_LOCATION}}
59 // Microsoft Visual C++ will insert additional declarations immediately
60 // before the previous line.
61
62 #endif //
63
64 !defined(AFX_TABSHEET_H__42EE262D_D15F_46D5_8F26_28FD049E99F4__INCLUDED_)

```

## TabSheet.cpp

```

1  // TabSheet.cpp : implementation file
2  //
3
4
5  // #include "Property5.h"
6  #include "TabSheet.h"
7
8  #ifdef _DEBUG
9  #define new DEBUG_NEW
10 #undef THIS_FILE
11 static char THIS_FILE[] = __FILE__;
12 #endif
13
14 //////////////////////////////////////
15 //
16 // CTabSheet
17 CTabSheet::CTabSheet()
18 {
19     m_nNumOfPages = 0;
20     m_nCurrentPage = 0;
21 }
22
23 CTabSheet::~CTabSheet()
24 {
25 }
26
27
28 BEGIN_MESSAGE_MAP(CTabSheet, CTabCtrl)
29     //{AFX_MSG_MAP(CTabSheet)
30     ON_WM_LBUTTONDOWN()
31     ON_WM_HSCROLL()
32     //}}AFX_MSG_MAP
33 END_MESSAGE_MAP()
34
35 //////////////////////////////////////
36 //
37 // CTabSheet message handlers
38
39 BOOL CTabSheet::AddPage(LPCTSTR title, CDialog* pDialog, UINT ID)
40 {
41     if (MAXPAGE == m_nNumOfPages)
42         return FALSE;
43
44     m_nNumOfPages++;
45
46     m_pPages[m_nNumOfPages - 1] = pDialog;
47     m_IDD[m_nNumOfPages - 1] = ID;
48     m_Title[m_nNumOfPages - 1] = title;
49
50     return TRUE;

```

```

50 }
51
52 void CTabSheet::SetRect()
53 {
54     CRect tabRect, itemRect;
55     int nX, nY, nXc, nYc;
56
57     GetClientRect(&tabRect);
58     GetItemRect(0, &itemRect);
59
60     nX = itemRect.left;
61     nY = itemRect.bottom + 1;
62     nXc = tabRect.right - itemRect.left - 2;
63     nYc = tabRect.bottom - nY - 2;
64
65     m_pPages[0]->SetWindowPos(&wndTop, nX, nY, nXc, nYc, SWP_SHOWWINDOW);
66     for (int nCount = 1; nCount < m_nNumOfPages; nCount++)
67         m_pPages[nCount]->SetWindowPos(&wndTop, nX, nY, nXc, nYc,
68     SWP_HIDEWINDOW);
69 }
70
71 void CTabSheet::Show()
72 {
73     int i = 0;
74     for (i = 0; i < m_nNumOfPages; i++)
75     {
76         m_pPages[i]->Create(m_IDD[i], this);
77         InsertItem(i, m_Title[i]);
78     }
79
80     m_pPages[0]->ShowWindow(SW_SHOW);
81     for (i = 1; i < m_nNumOfPages; i++)
82         m_pPages[i]->ShowWindow(SW_HIDE);
83
84     SetRect();
85
86 }
87
88 void CTabSheet::OnLButtonDown(UINT nFlags, CPoint point)
89 {
90     CTabCtrl::OnLButtonDown(nFlags, point);
91
92     if (m_nCurrentPage != GetCurFocus())
93     {
94         m_pPages[m_nCurrentPage]->ShowWindow(SW_HIDE);
95         m_nCurrentPage = GetCurFocus();
96         m_pPages[m_nCurrentPage]->ShowWindow(SW_SHOW);
97         // m_pPages[m_nCurrentPage]->SetFocus();
98
99         //AfxMessageBox("选中"); wbm test
100     }
101
102 }
103
104 int CTabSheet::SetCurSel(int nItem)

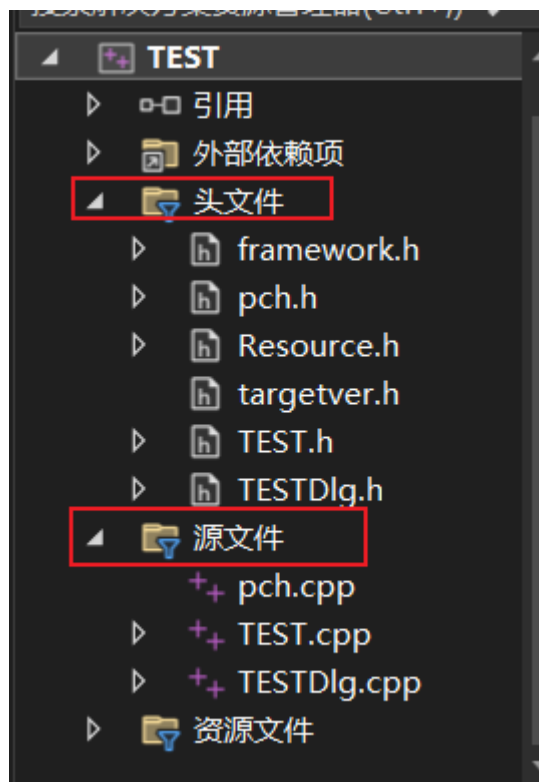
```

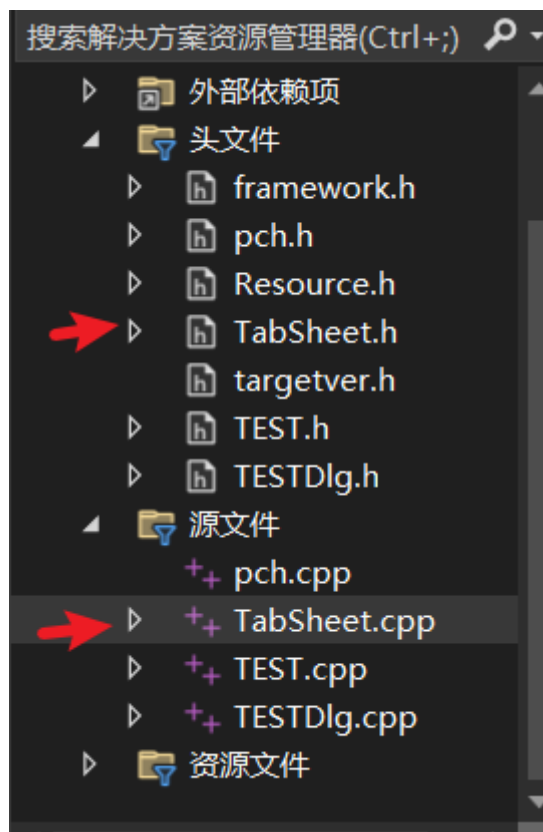
```

105 {
106     if (nItem < 0 || nItem >= m_nNumOfPages)
107         return -1;
108
109     int ret = m_nCurrentPage;
110
111     if (m_nCurrentPage != nItem)
112     {
113         m_pPages[m_nCurrentPage]->ShowWindow(SW_HIDE);
114         m_nCurrentPage = nItem;
115         m_pPages[m_nCurrentPage]->ShowWindow(SW_SHOW);
116         // m_pPages[m_nCurrentPage]->SetFocus();
117         CTabCtrl::SetCurSel(nItem);
118     }
119
120     return ret;
121 }
122
123 int CTabSheet::GetCurSel()
124 {
125     return CTabCtrl::GetCurSel();
126 }
127
128 void CTabSheet::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
129 {
130     // TODO: Add your message handler code here and/or call default
131
132     CTabCtrl::OnHScroll(nSBCode, nPos, pScrollBar);
133 }

```

我们在项目下的头文件和源文件下分别创建这两个文件并将代码粘贴进去

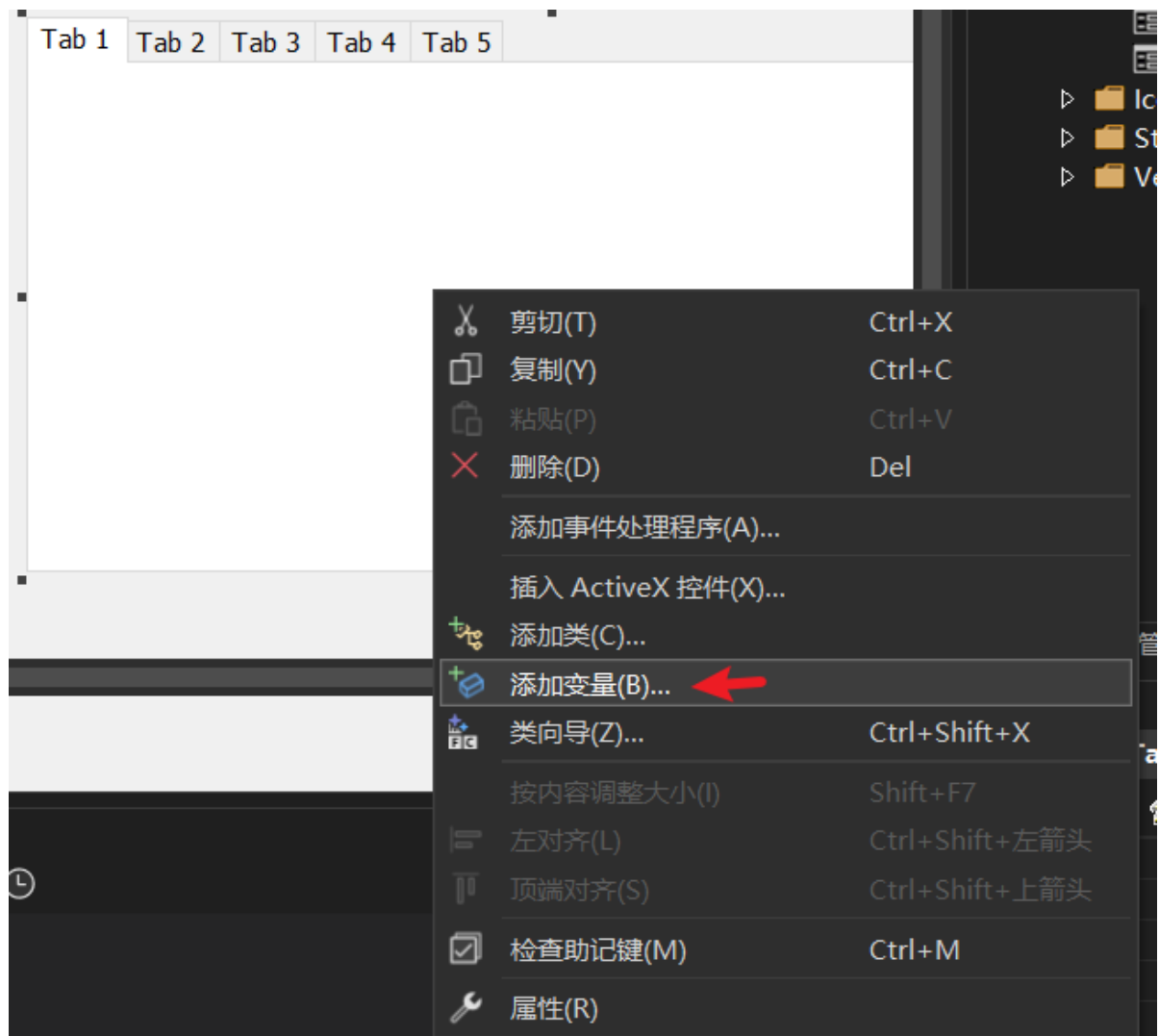




这样我们就可以更加方便快捷地对页面组合控件进行管理了。

我们也需要给这个控件添加控制它的变量：

右键点击这个控件》**添加变量**



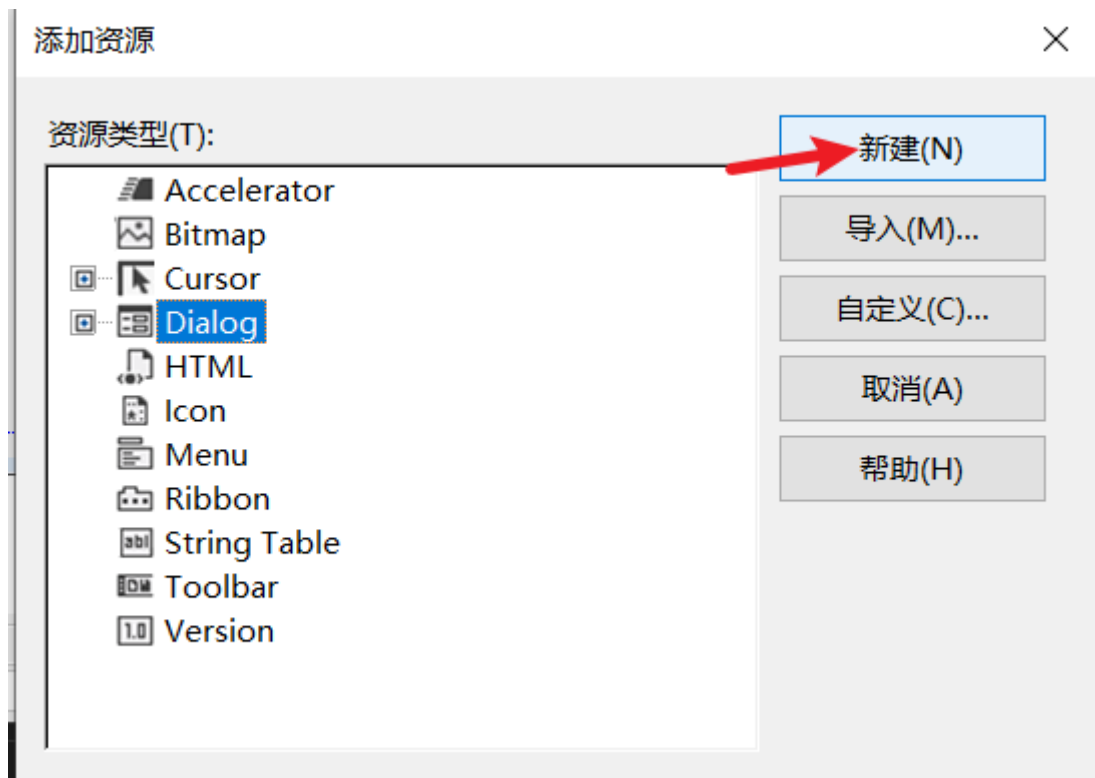
需求：使用Tab Control集成两个页面，两个页面中包含一个静态的文本“页面X”。

首先我们来创建两个页面：

右键点击Dialog》添加资源



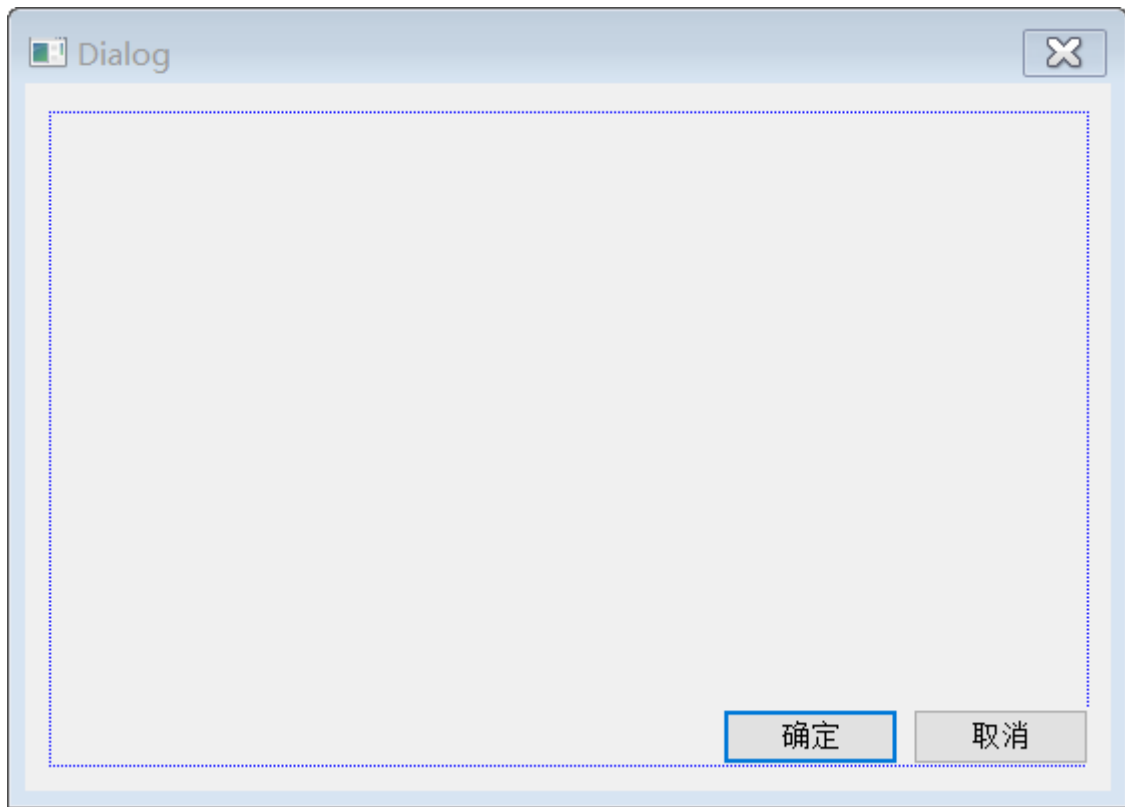
点击新建即可



页面的属性可以在右下角进行修改,我将它们的id分别修改为D1和D2.

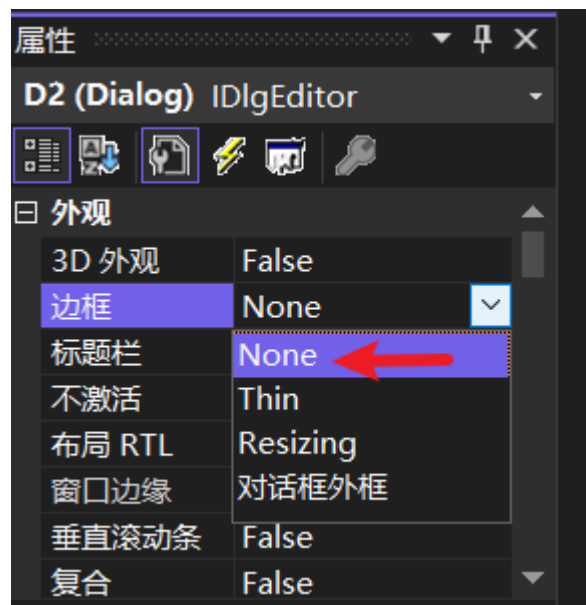
这是其中一个页面的样子:



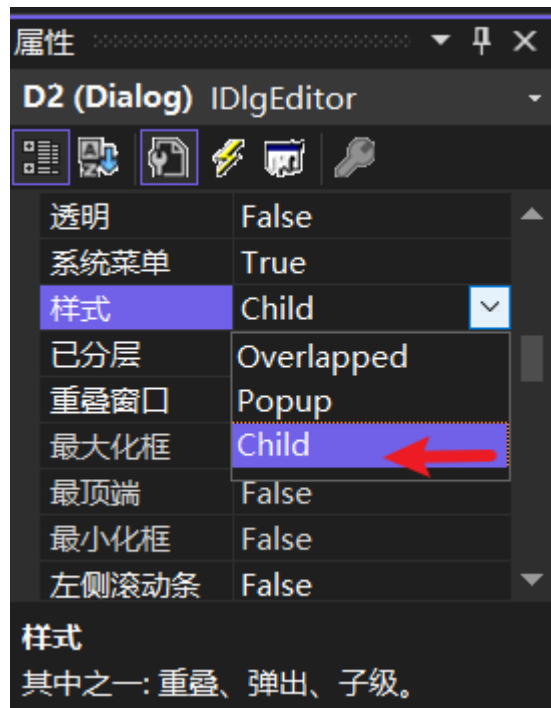


其中，它需要内嵌在另一个页面中，那么应该给它设置非弹窗属性，并且要去掉外圈的栏位。

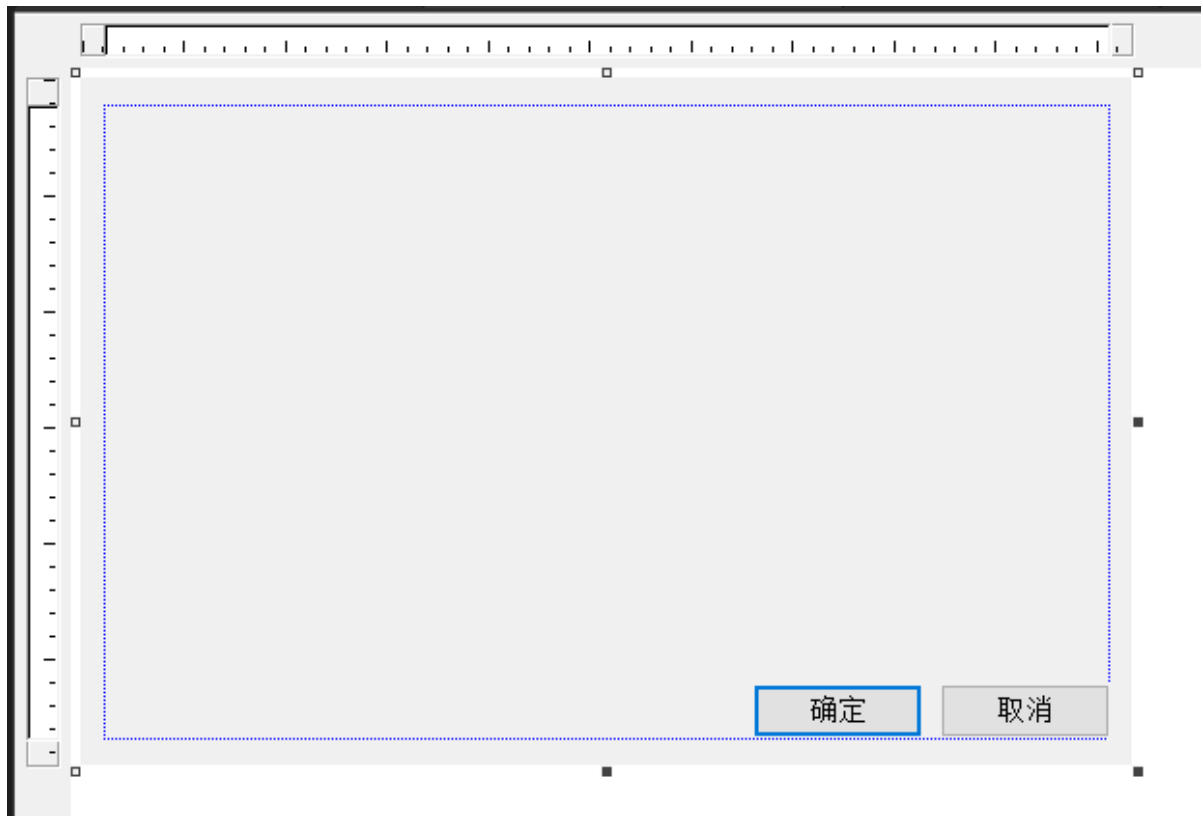
先去掉外框，将边框属性设置为None：



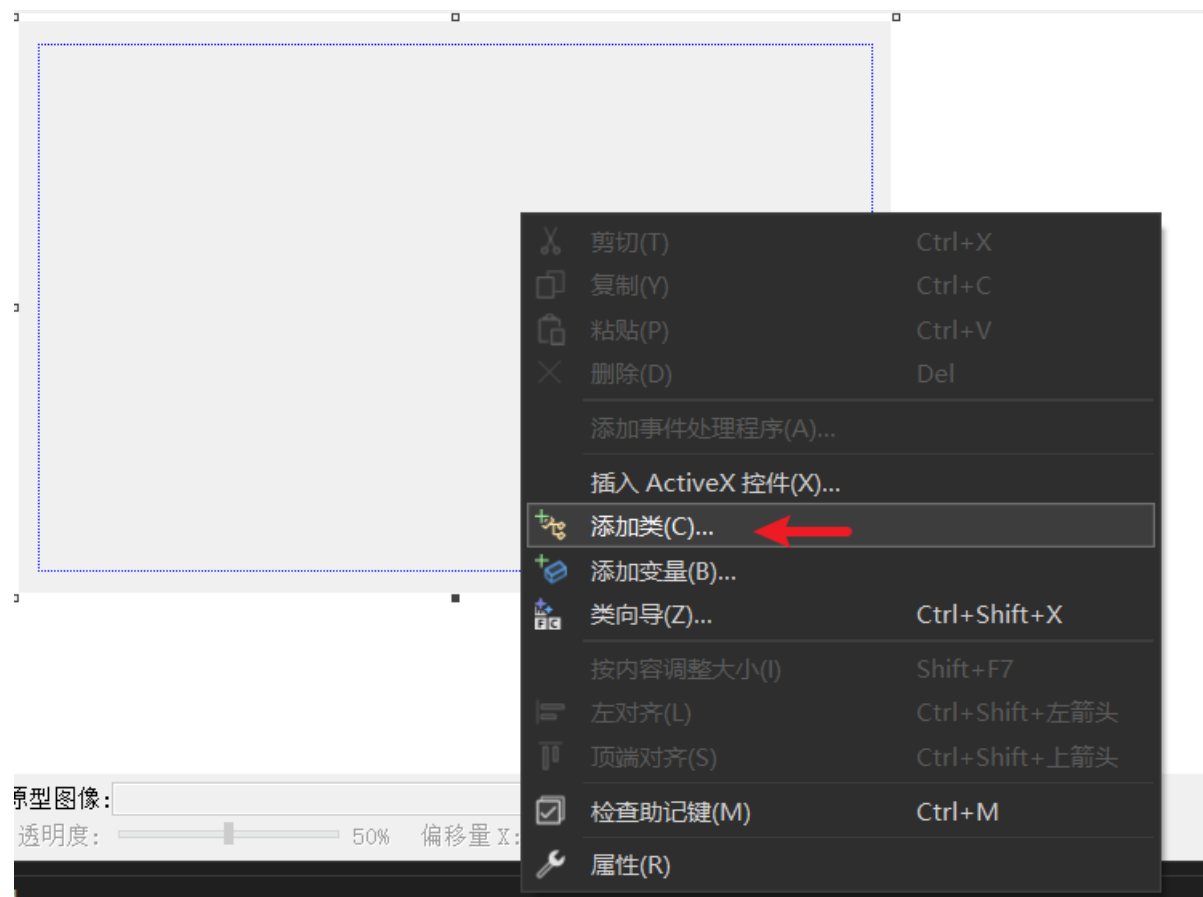
然后将样式设为Child：



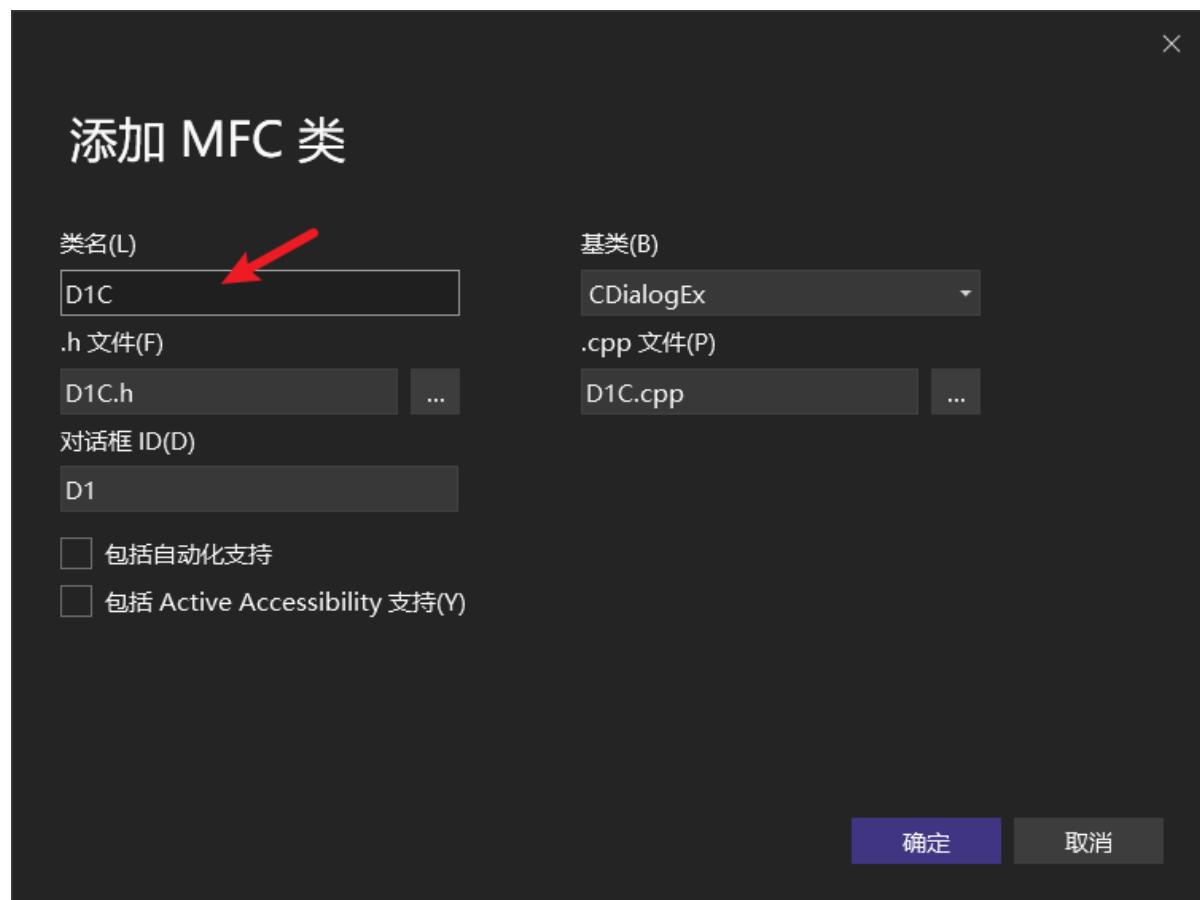
对另一个对话框也进行相同的操作，我们就会得到两个这样的框：



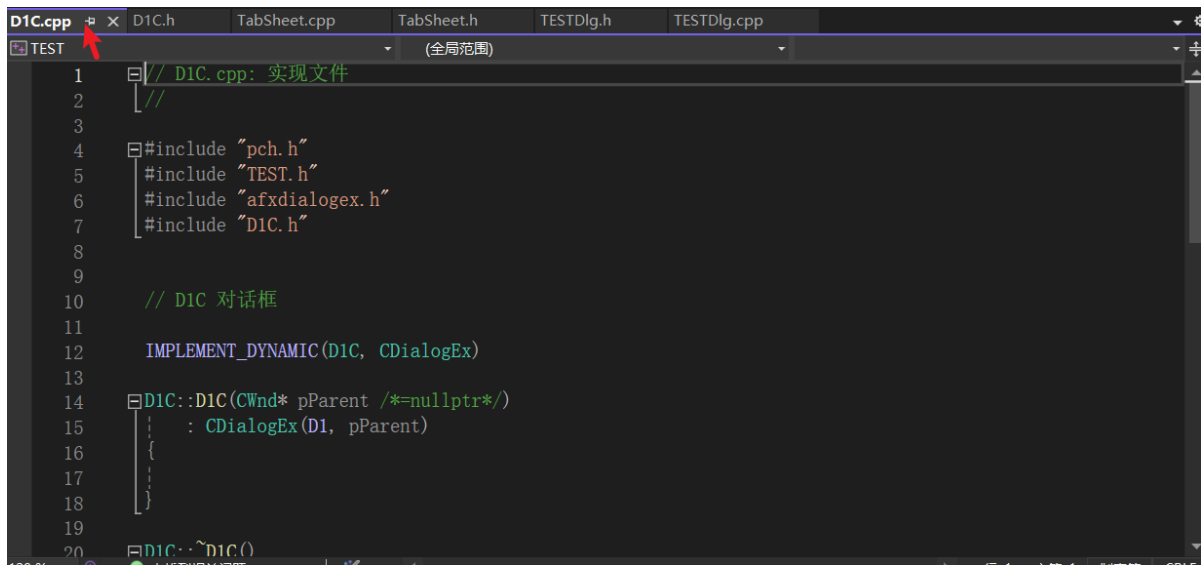
这两个框也都是独立的，我们可以右键这个框为它添加一个类，这样就能用代码实现和前面主框一样的自主开发。



起个名字，确定：



这样我们就创建好了类，可以通过代码进行开发了！



```
1 // D1C.cpp: 实现文件
2 //
3
4 #include "pch.h"
5 #include "TEST.h"
6 #include "afxdialogex.h"
7 #include "D1C.h"
8
9
10 // D1C 对话框
11
12 IMPLEMENT_DYNAMIC(D1C, CDialogEx)
13
14 D1C::D1C(CWnd* pParent /*=nullptr*/)
15     : CDialogEx(ID1, pParent)
16 {
17 }
18
19
20 D1C::~D1C()
```

重复操作，这样我们就添加了对这两个窗口进行管理的两个类

现在我们来给这两个页面设置样式：

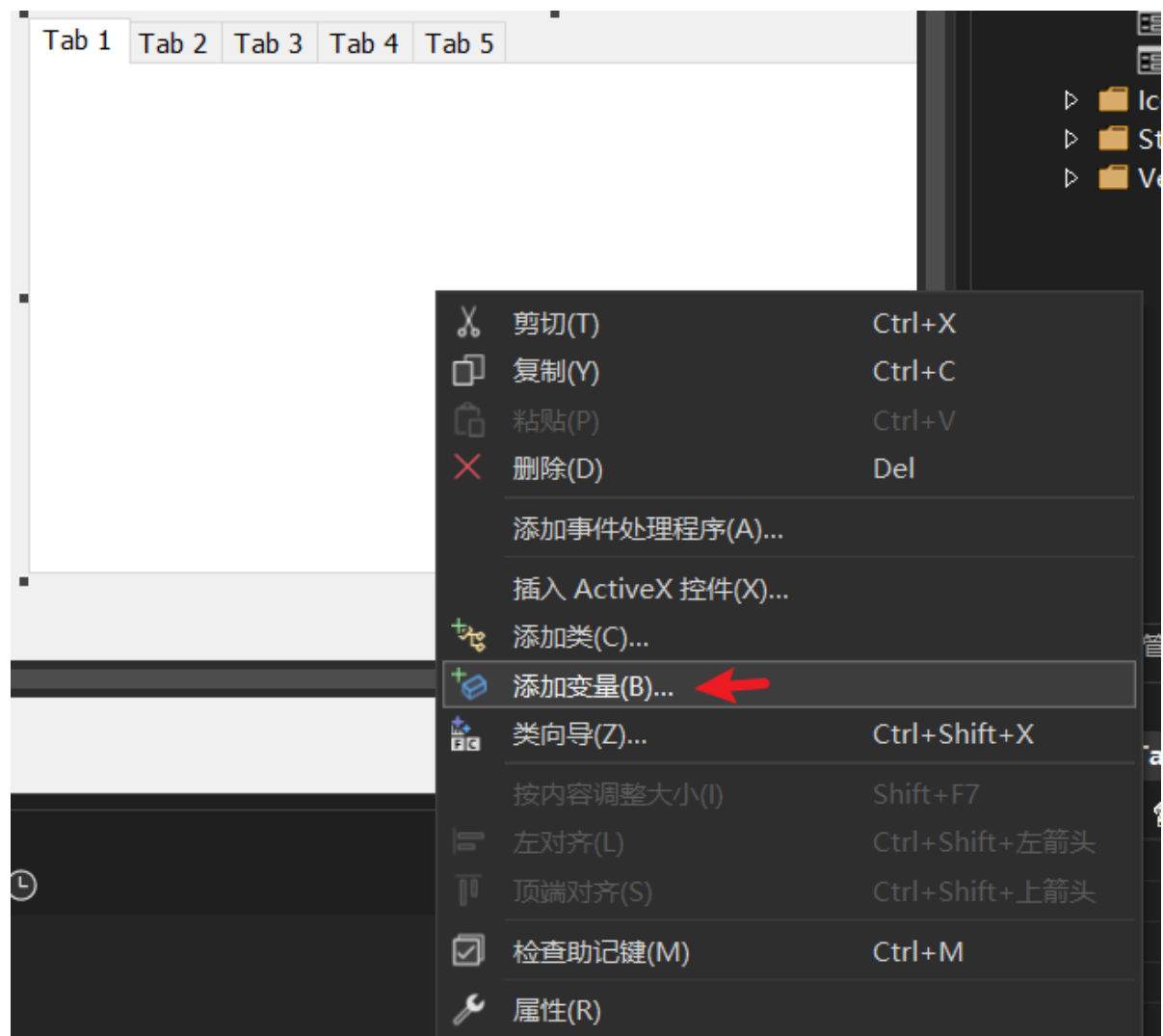




然后，我们把两个页面插入到主页面上去：

首先我们也需要给这个控件添加控制它的变量：

右键点击这个控件》添加变量



这里稍微有点不一样，注意下面的变量类型要填写CTabSheet，就是我们上面引入的两个文件。

添加控制变量

常规设置

控件

其他

控件 ID(I)

IDC\_TAB1

类别(T)

控件

访问(A)

private

控件类型(Y)

SysTabControl32

名称(N)

tc

变量类型(V)

CTabSheet

注释(M)

显示生成的变量将基于的控

上一步

下一步

完成

取消

tip:

如果遇到了变量创建失败，请移步 5) 去看看怎么解决的bug

可以看到变量已经添加上了，但是爆红：

```
TEST
CTESTDlg

22
23
24 // 实现
25 protected:
26     HICON m_hIcon;
27
28     // 生成的消息映射函数
29     virtual BOOL OnInitDialog();
30     afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
31     afx_msg void OnPaint();
32     afx_msg HCURSOR OnQueryDragIcon();
33     DECLARE_MESSAGE_MAP()
34 public:
35     afx_msg void OnBnClickedbutton();
36 private:
37     CListCtrl lc;
38     CTabSheet tc;
39
40
```

我们在这个文件头上引入TabSheet.h就可以了 [OK](#)：

```
TEST
25 protected:
26     HICON m_hIcon;
27
28     // 生成的消息映射函数
29     virtual BOOL OnInitDialog();
30     afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
31     afx_msg void OnPaint();
32     afx_msg HCURSOR OnQueryDragIcon();
33     DECLARE_MESSAGE_MAP()
34 public:
35     afx_msg void OnBnClickedbutton();
36 private:
37     CListCtrl lc;
38     CTabSheet tc;
39
40
```

```
1 #include "TabSheet.h"
```

然后我们引入两个页面的头文件，并在下面创建控制这两个类的成员：

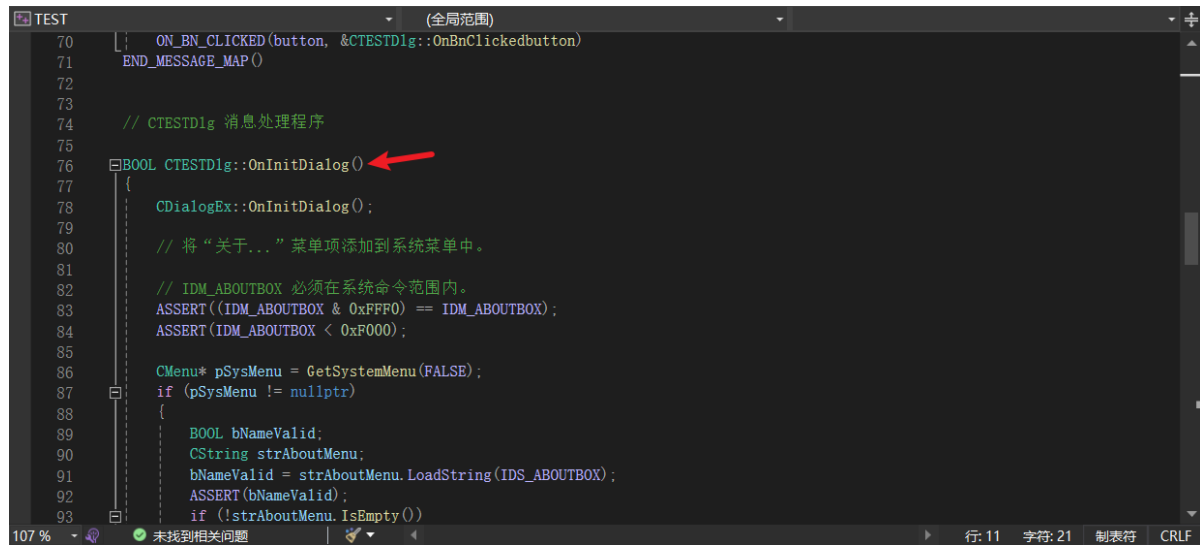
```
D2C.h TESTDlg.h* TESTDlg.cpp TabSheet.cpp
TEST
1 // TESTDlg.h: 头文件
2 //
3
4
5 #pragma once
6 #include "TabSheet.h"
7 #include "DIC.h"
8 #include "D2C.h"
9 // CTESTDlg 对话框
10 class CTESTDlg : public CDialogEx
11 {
12 // 构造
13 public:
14     CTESTDlg(CWnd* pParent = nullptr); // 标准构造函数
15
16 // 对话框数据
17 #ifdef AFX_DESIGN_TIME
18     enum { IDD = IDD_TEST_DIALOG };
19 #endif
20
21 protected:
22     virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV 支持
```

```
D2C.h TESTDlg.h* TESTDlg.cpp TabSheet.cpp
TEST
28
29 // 生成的消息映射函数
30 virtual BOOL OnInitDialog();
31 afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
32 afx_msg void OnPaint();
33 afx_msg HCURSOR OnQueryDragIcon();
34 DECLARE_MESSAGE_MAP()
35 public:
36     afx_msg void OnBnClickedbutton();
37 private:
38     CListCtrl lc;
39     CTabSheet tc;
40     DIC p1;
41     D2C p2;
42
43
```

然后我们进入到下一步：

首先需要大家知道，页面的初始化是写在一个函数中的，这两个页面如果需要显示，则需要再初始化函数中添加这两个页面的信息。

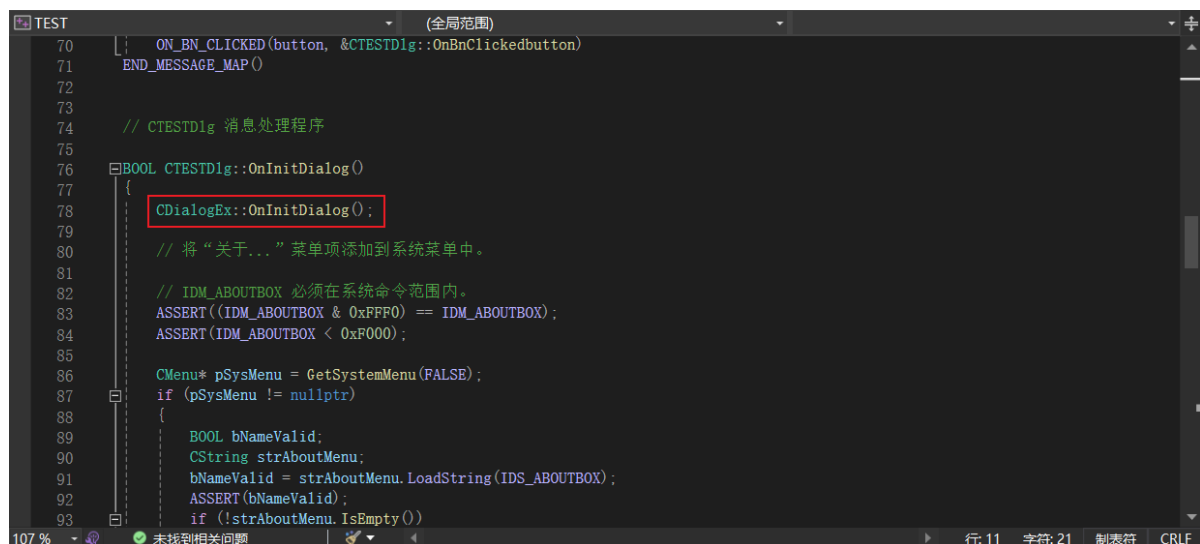
这个叫做OnInitDialog()的函数就是初始化函数。对于初始的页面，这个函数是自动创建好的，但是后创建的页面，就没有这个函数，如果需要则要手动添加（注意大小写别拼错了，要不然它不认😭血与泪的教训）



```
70  [ ON_BN_CLICKED(button, &CTESTDlg::OnBnClickedbutton)
71  END_MESSAGE_MAP()
72
73
74  // CTESTDlg 消息处理程序
75
76  BOOL CTESTDlg::OnInitDialog()
77  {
78      CDialogEx::OnInitDialog();
79
80      // 将“关于...”菜单项添加到系统菜单中。
81
82      // IDM_ABOUTBOX 必须在系统命令范围内。
83      ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
84      ASSERT(IDM_ABOUTBOX < 0xF000);
85
86      CMenu* pSysMenu = GetSystemMenu(FALSE);
87      if (pSysMenu != nullptr)
88      {
89          BOOL bNameValid;
90          CString strAboutMenu;
91          bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
92          ASSERT(bNameValid);
93          if (!strAboutMenu.IsEmpty())
```

可以在这个初始化函数中实现一些功能：包括但不限于，现在正在做的插入页面；多选，单选框的默认选中；表单、文本框中显示的默认信息，除首个功能外，其他的功能交给大家自己去研究😁

我们可以看到这个初始化函数中已经为我们写了好多东西了。下面的一些都不重要，注意这个初始化函数中的父类初始化，一定不能忘记写，特别是自己添加初始化函数的时候，要不然好多函数都会报错的，真的很坑很坑😭（涉及到了野指针的知识）



```
70  [ ON_BN_CLICKED(button, &CTESTDlg::OnBnClickedbutton)
71  END_MESSAGE_MAP()
72
73
74  // CTESTDlg 消息处理程序
75
76  BOOL CTESTDlg::OnInitDialog()
77  {
78      CDialogEx::OnInitDialog();
79
80      // 将“关于...”菜单项添加到系统菜单中。
81
82      // IDM_ABOUTBOX 必须在系统命令范围内。
83      ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
84      ASSERT(IDM_ABOUTBOX < 0xF000);
85
86      CMenu* pSysMenu = GetSystemMenu(FALSE);
87      if (pSysMenu != nullptr)
88      {
89          BOOL bNameValid;
90          CString strAboutMenu;
91          bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
92          ASSERT(bNameValid);
93          if (!strAboutMenu.IsEmpty())
```

中间的这些我们就不管了，来到函数的后面接着写我们的东西就行了

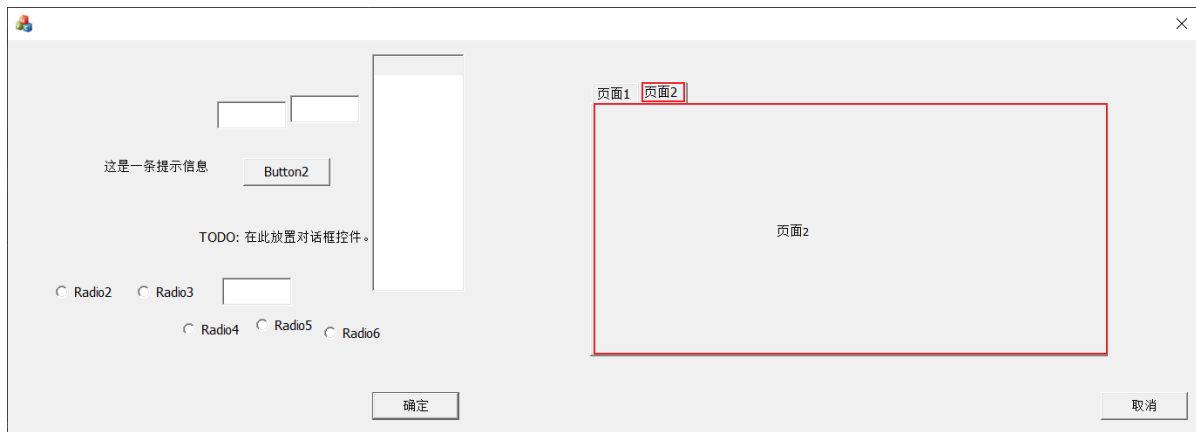


```
D2C.h TESTDlg.h TESTDlg.cpp* TabSheet.cpp
TEST
92 bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
93 ASSERT(bNameValid);
94 if (!strAboutMenu.IsEmpty())
95 {
96     pSysMenu->AppendMenu(MF_SEPARATOR);
97     pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
98 }
99 }
100
101 // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
102 // 执行此操作
103 SetIcon(m_hIcon, TRUE); // 设置大图标
104 SetIcon(m_hIcon, FALSE); // 设置小图标
105
106 // TODO: 在此添加额外的初始化代码
107 tc.AddPage(TEXT("页面1"), &p1, D1); //三个变量分别是 页面标签上显示的文字 页面成员的地址 该页面的ID
108 tc.AddPage(TEXT("页面2"), &p2, D2);
109 tc.Show();
110
111
112 return TRUE; // 除非将焦点设置到控件，否则返回 TRUE
113 }
```

```
1  BOOL CTESTDlg::OnInitDialog()
2  {
3      CDialogEx::OnInitDialog();
4
5      //中间省略
6
7      // TODO: 在此添加额外的初始化代码
8      tc.AddPage(TEXT("页面1"), &p1, D1); //三个变量分别是 页面标签上显示的文字 页面成员的地址 该页面的ID
9      tc.AddPage(TEXT("页面2"), &p2, D2);
10     tc.Show();
11
12
13     return TRUE; // 除非将焦点设置到控件，否则返回 TRUE
14 }
15
```

看看实现的效果：





## 5.一些其他功能的实现

### 1) 实现页面跳转

当我们点击了一个按钮之后我们就从一个页面跳转到另一个页面：

我们可以先弹出窗口，然后显示关闭本窗口。

需求：

实现窗口之间的切换，当点击一个按钮后此窗口不可见，另个窗口出现。

我们就选用上面的主窗口和D1窗口进行操作

先取消D1的Child属性，并注释掉TabControl部分代码。

注意语句的逻辑顺序，应该先让当前页面消失再让另个页面显示。

```

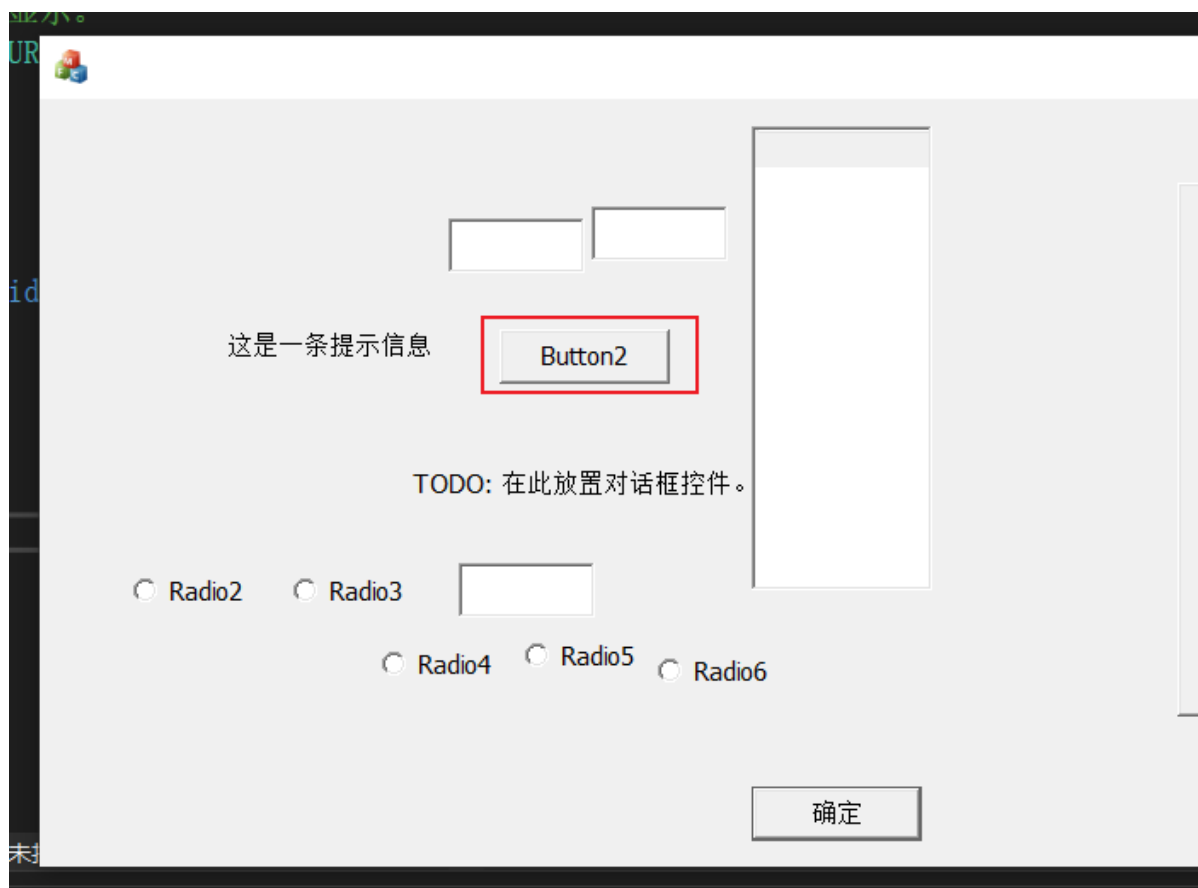
156
157
158 //当用户拖动最小化窗口时系统调用此函数取得光标
159 //显示。
160 HCURSOR CTESTDlg::OnQueryDragIcon()
161 {
162     return static_cast<HCURSOR>(m_hIcon);
163 }
164
165 void CTESTDlg::OnBnClickedbutton()
166 {
167     // TODO: 在此添加控件通知处理程序代码
168     D1C p1;
169     EndDialog(0);
170     p1.DoModal();
171 }
172

```

```

1 void CTESTDlg::OnBnClickedbutton()
2 {
3     // TODO: 在此添加控件通知处理程序代码
4     D1C p1;
5     EndDialog(0);
6     p1.DoModal();
7 }

```



2) 实现全局状态存取，例如保存登录信息并在另一个页面显示  
使用全局变量就行了。

需求：

在第一个页面输入信息，单击按钮后在第二个页面中显示。

在第二个页面中创建一个输出框用于输出信息，id为text4：



使用text3来接收信息，button进行操作：



代码如下：

```
155     }
156
157     //当用户拖动最小化窗口时系统调用此函数取得光标
158     //显示。
159     HCURSOR CTESTDlg::OnQueryDragIcon()
160     {
161         return static_cast<HCURSOR>(m_hIcon);
162     }
163
164     CString globalInfo;//全局变量
165
166     void CTESTDlg::OnBnClickedbutton()
167     {
168         // TODO: 在此添加控件通知处理程序代码
169         GetDlgItemText(text3, globalInfo);
170         D1C p1;
171         EndDialog(0);
172         p1.DoModal();
173     }
174
```

```
1  CString globalInfo;//全局变量
2
3  void CTESTDlg::OnBnClickedbutton()
4  {
5      // TODO: 在此添加控件通知处理程序代码
6      GetDlgItemText(text3, globalInfo);
7      D1C p1;
8      EndDialog(0);
9      p1.DoModal();
10 }
```

在D1头文件下添加初始化函数:

```
7  {
8      DECLARE_DYNAMIC(D1C)
9
10     public:
11         D1C(CWnd* pParent = nullptr); // 标准构造函数
12         virtual ~D1C();
13
14         // 对话框数据
15     #ifdef AFX_DESIGN_TIME
16         enum { IDD = D1 };
17     #endif
18
19     protected:
20         virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV 支持
21
22         DECLARE_MESSAGE_MAP()
23     public:
24         afx_msg void OnBnClickedOk();
25         BOOL OnInitDialog();
26
27 }
```

在D1cpp文件下定义初始化函数:

```
TEST
D1C
OnInitDialog()

28
29
30 BEGIN_MESSAGE_MAP(D1C, CDialogEx)
31 {
32     ON_BN_CLICKED(IDOK, &D1C::OnBnClickedOk)
33 }
34 END_MESSAGE_MAP()
35
36 extern CString globalInfo;
37 // D1C 消息处理程序
38 BOOL D1C::OnInitDialog()
39 {
40     CDialogEx::OnInitDialog();
41     SetDlgItemText(text4, globalInfo);
42     return TRUE;
43 }
44
45 void D1C::OnBnClickedOk()
46 {
47     // TODO: 在此添加控件通知处理程序代码
48 }
49
```

```
1 extern CString globalInfo;
2 // D1C 消息处理程序
3 BOOL D1C::OnInitDialog()
4 {
5     CDialogEx::OnInitDialog();
6
7     SetDlgItemText(text4, globalInfo);
8
9     return TRUE;
10 }
```

运行效果:



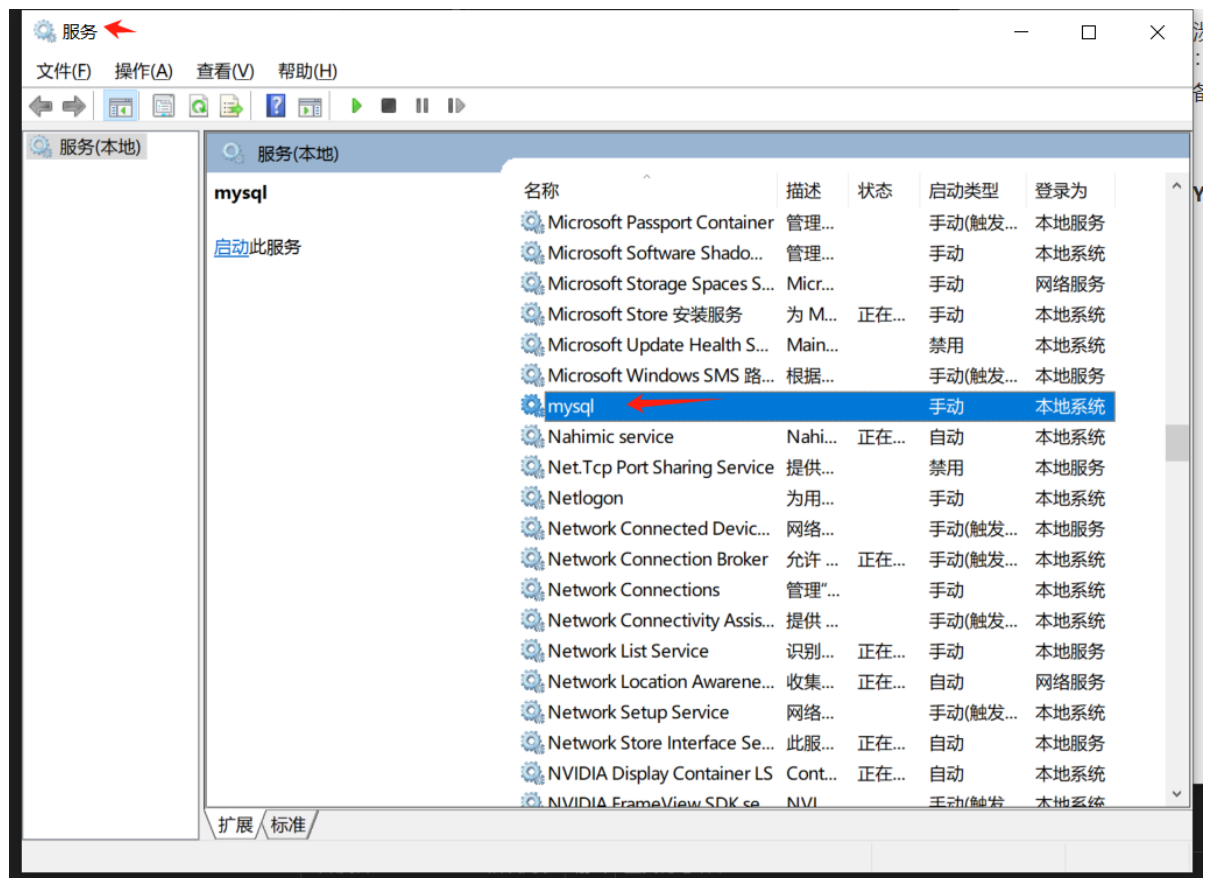


## 二、C++连接MYSQL数据库，并实现简单的查找

### 1.数据库的创建

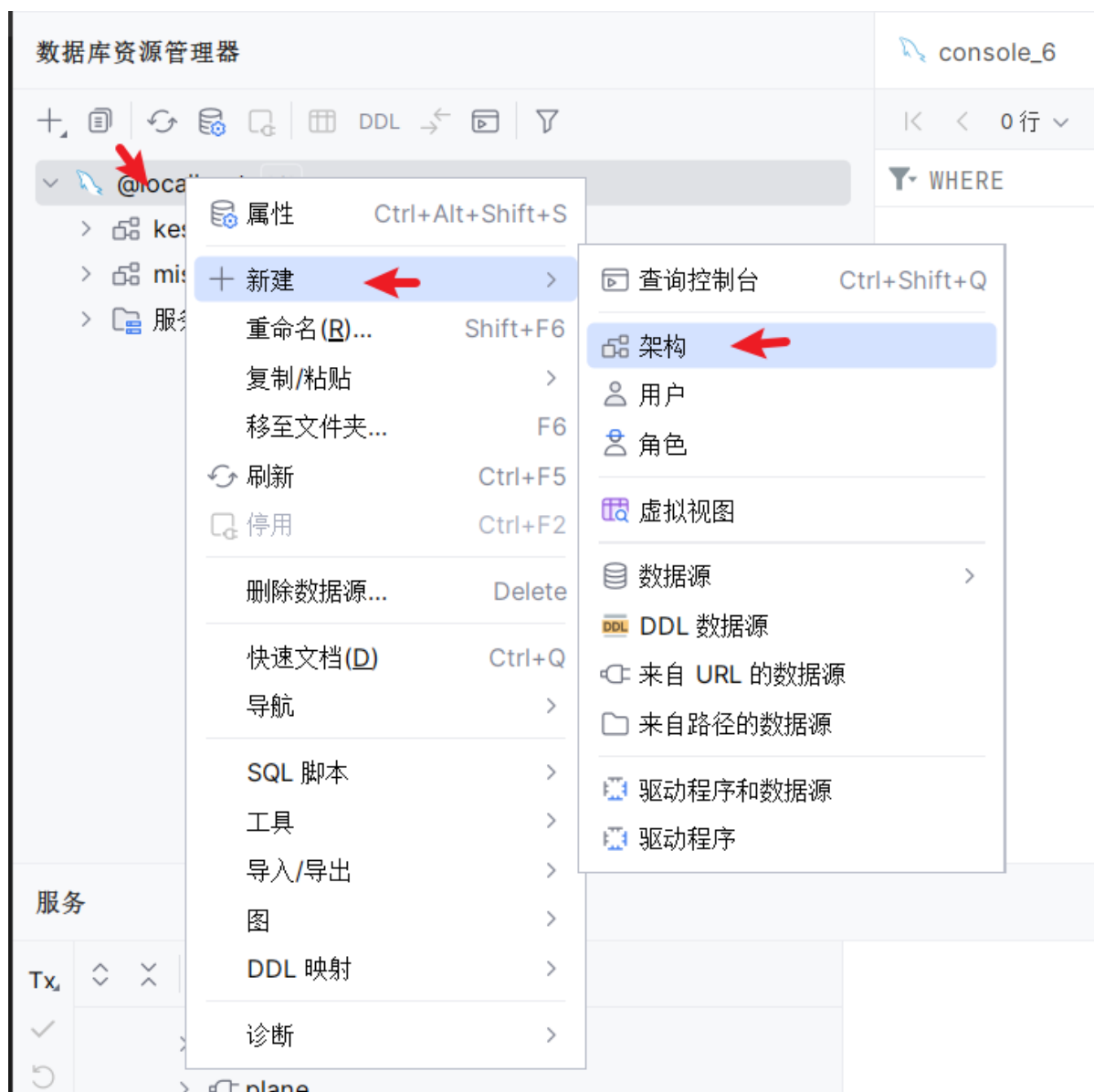
首先默认数据库已经安装完成，并且环境变量配置正确。不会的可以去CSDN上找，教程很多。

首先在电脑的**服务**处右键mysql启动数据库



用自己的账号密码连接到数据库服务，记住自己的账号密码，下面会用到。

右键自己的连接，**新建》架构**来新建一个数据库



起个名字，然后确定



pg 创建

×

名称

TEST

:

排序规则

▼

授权

+

−

↑

↓

没有要显示的内容

▼ 预览(P)

⚙️

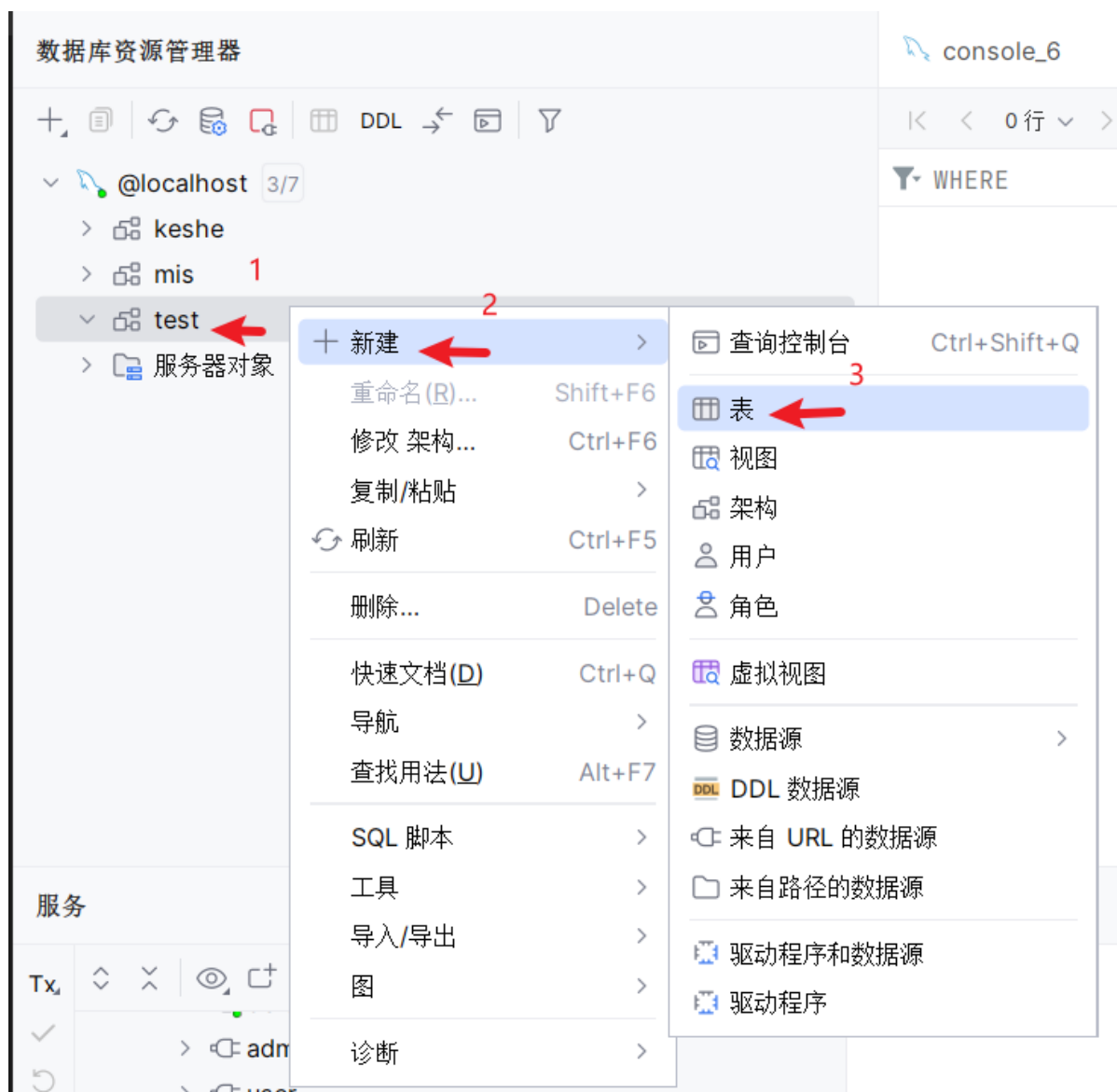
🔗

```
create schema TEST;
```

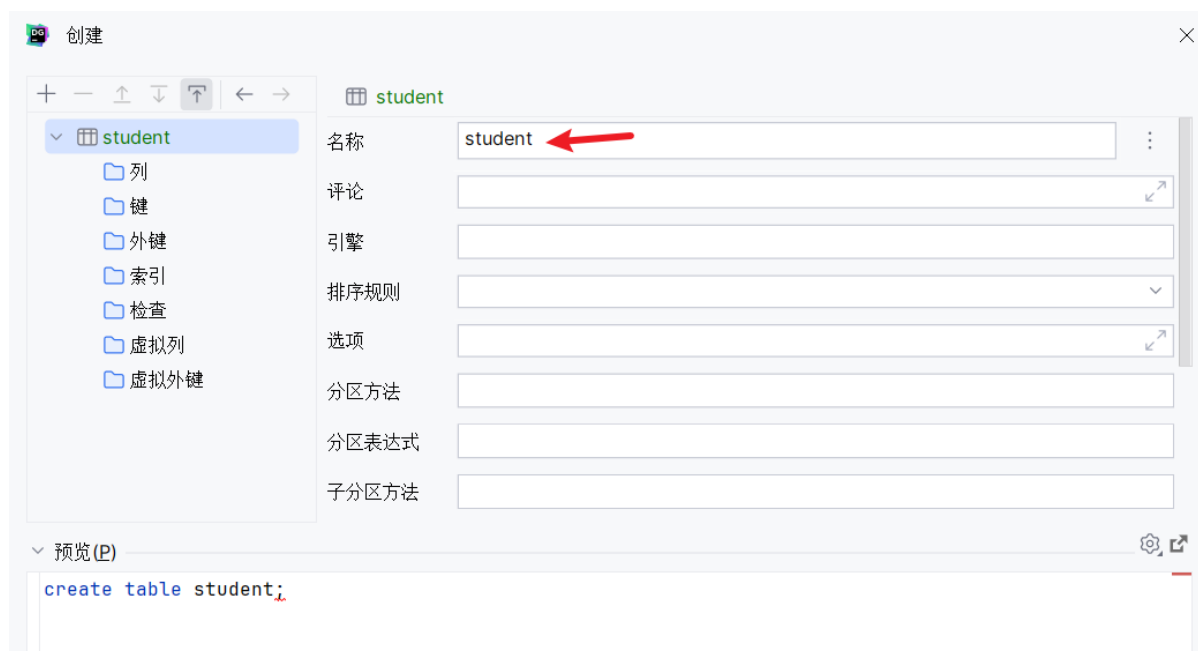
确定

取消

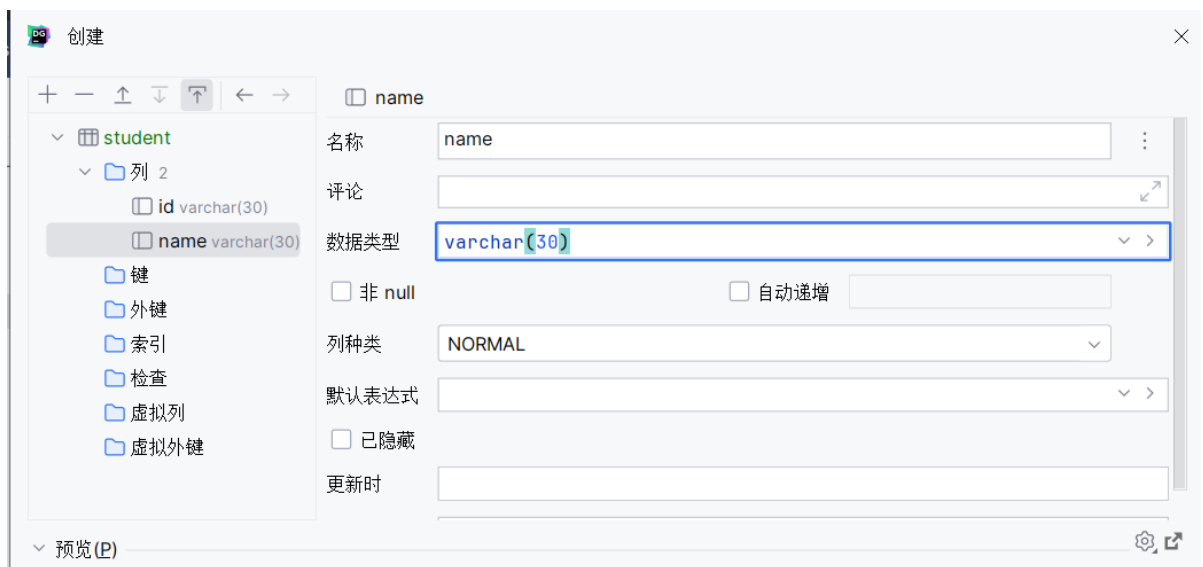
右键自己的数据库》新建》表



起一个表的名字。



为演示方便，这里新建两个列id和name，类型都为varchar，长度为30：



往表中插入一些数据。

student

WHERE ORDER B

	id	name
1	1	zhangsan
2	2	lisi
3	3	wangwu

这样数据库就配置完成了，接着我们来为编译器配置数据库的环境。

## 2.编译器配置数据库环境

首先我们要找到自己安装数据库的地址，如下。

此电脑 > 新加卷 (E:) > Mysql > mysql-8.2.0-winx64

名称	修改日期	类型	大小
bin	2023/10/12 21:28	文件夹	
data	2023/12/20 14:59	文件夹	
docs	2023/10/12 21:22	文件夹	
include	2023/10/12 21:22	文件夹	
lib	2023/10/12 21:28	文件夹	
share	2023/10/12 21:22	文件夹	
LICENSE	2023/10/12 19:40	文件	273 KB
my.ini	2023/11/3 10:17	配置设置	1 KB
README	2023/10/12 19:40	文件	1 KB

然后我们需要记录下两个地址，分别是include文件和lib文件的地址。在我这里是：

```
1 E:\Mysql\mysql-8.2.0-winx64\include //include
2 E:\Mysql\mysql-8.2.0-winx64\lib //lib
```

这个地址可以直接通过点击路径框得到：

include

文件 主页 共享 查看

E:\Mysql\mysql-8.2.0-winx64\include

名称	修改日期	类型	大小
mysql	2023/10/12 21:22	文件夹	
openssl	2023/10/12 21:22	文件夹	
errmsg.h	2023/10/12 19:40	C Header File	6 KB
field_types.h	2023/10/12 19:40	C Header File	4 KB
my_command.h	2023/10/12 19:40	C Header File	5 KB
my_compress.h	2023/10/12 19:40	C Header File	4 KB
my_list.h	2023/10/12 19:40	C Header File	3 KB
mysql.h	2023/10/12 19:40	C Header File	34 KB
mysql_com.h	2023/10/12 19:40	C Header File	42 KB

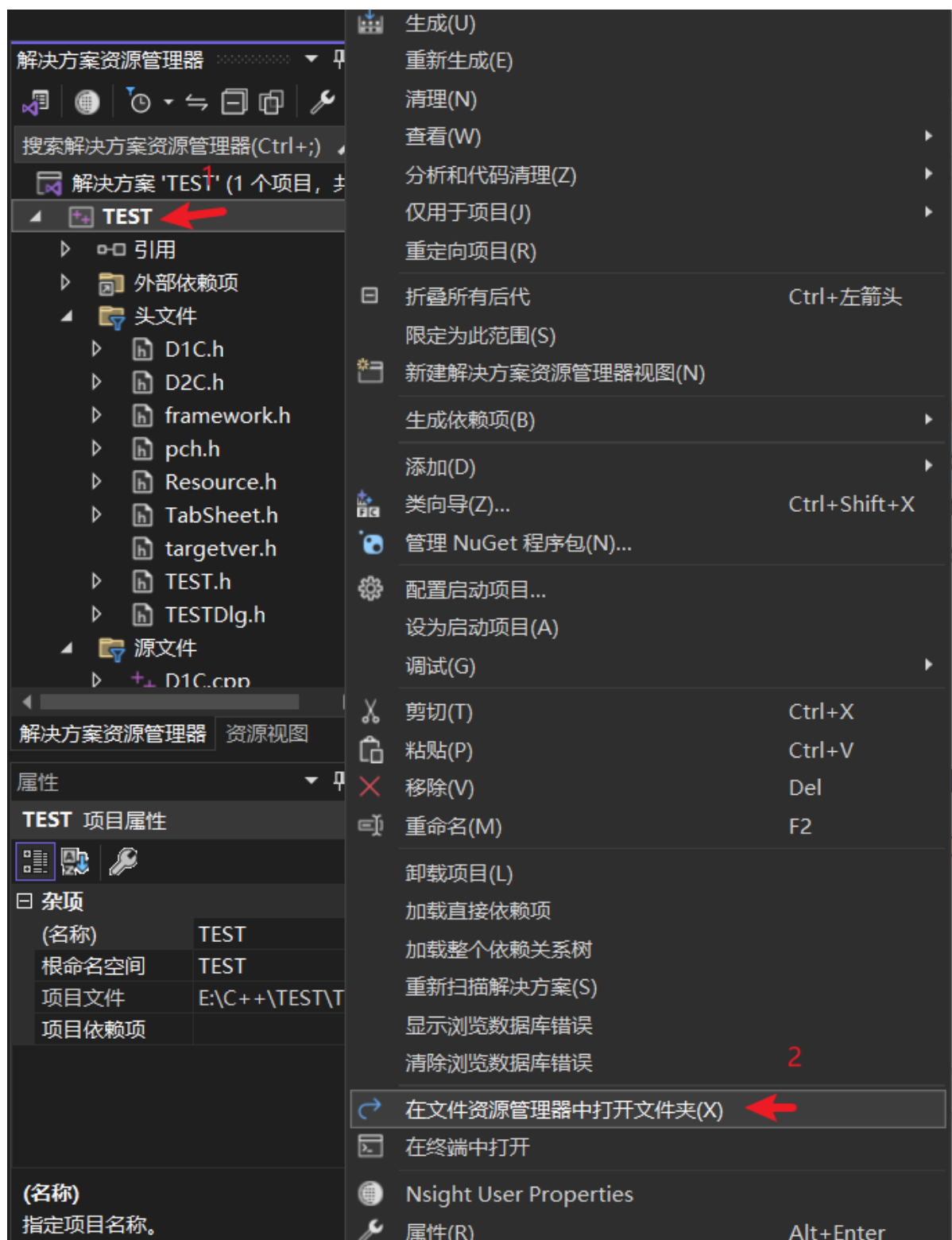
然后，我们进入到lib文件目录下，找到libmysql.dll文件。

此电脑 > 新加卷 (E:) > Mysql > mysql-8.2.0-winx64 > lib

名称	修改日期	类型	大小
mecab	2023/10/12 21:22	文件夹	
plugin	2023/10/12 21:28	文件夹	
private	2023/10/12 21:22	文件夹	
libmysql.dll	2023/10/12 14:27	应用程序扩展	6,850 KB
libmysql.lib	2023/10/12 20:53	Object File Library	32 KB
mysqlclient.lib	2023/10/12 20:53	Object File Library	43,947 KB
mysqlservices.lib	2023/10/12 20:49	Object File Library	75 KB

然后将这个文件复制到我们的项目下。

在编译器中右键自己的项目》在文件资源管理器中打开文件夹就可以打开

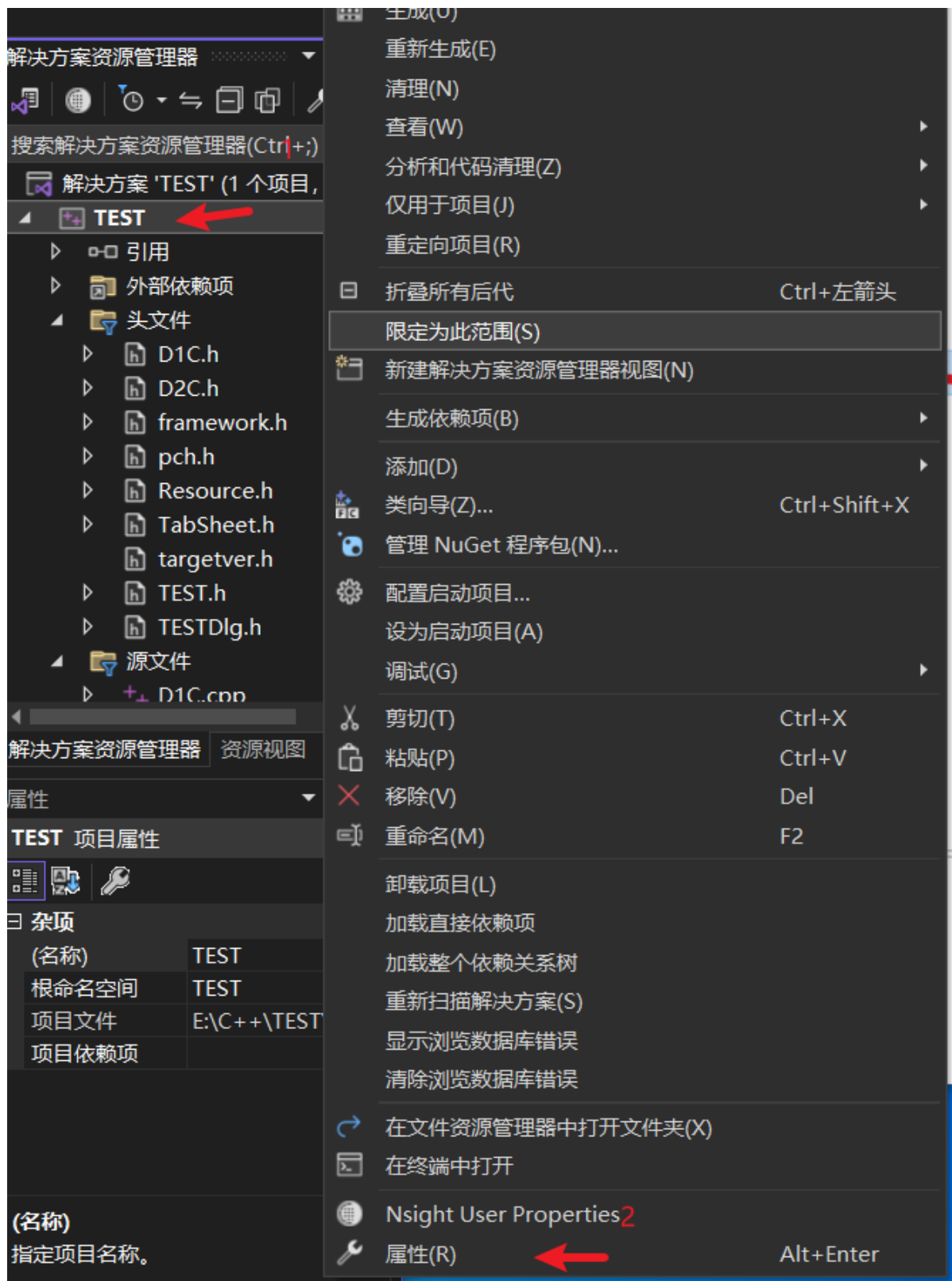


将文件复制进去

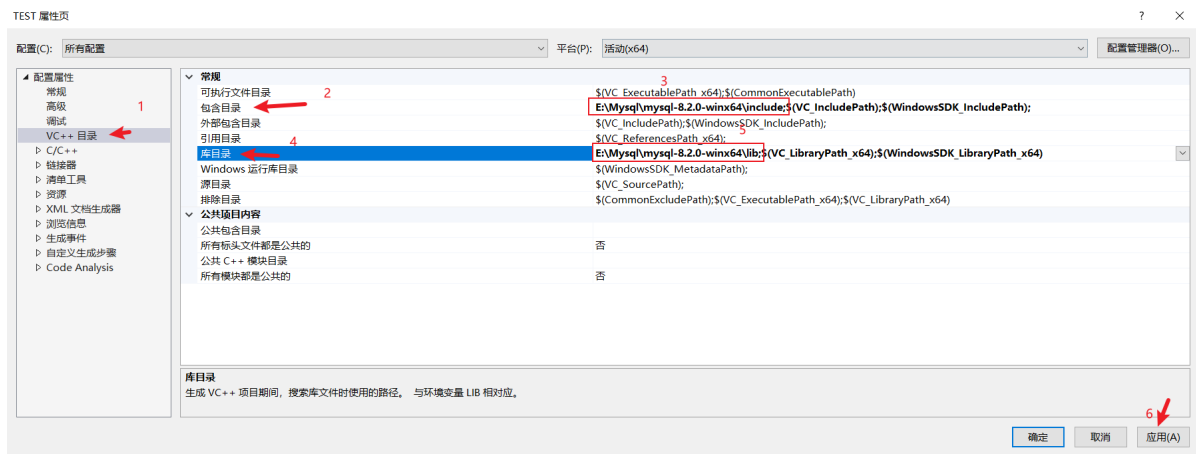
> 此电脑 > 新加卷 (E:) > C++ > TEST > TEST >

	名称	修改日期	类型	大小
SD (	res	2023/12/19 13:32	文件夹	
	x64	2023/12/19 13:33	文件夹	
	Conn.cpp	2023/12/20 15:45	C++ Source File	2 KB
DA	Conn.h	2023/12/20 15:38	C Header File	1 KB
omca	D1C.cpp	2023/12/19 18:41	C++ Source File	1 KB
	D1C.h	2023/12/19 18:30	C Header File	1 KB
	D2C.cpp	2023/12/19 17:54	C++ Source File	1 KB
	D2C.h	2023/12/19 17:54	C Header File	1 KB
+Ke	framework.h	2023/12/19 13:32	C Header File	2 KB
+课	libmysql.dll	2023/10/12 14:27	应用程序扩展	6,850 KB
司库	pch.cpp	2023/12/19 13:32	C++ Source File	1 KB
	pch.h	2023/12/19 14:21	C Header File	1 KB
olica	resource.h	2023/12/19 18:25	C Header File	2 KB
li	Test.cpp	2023/12/19 17:45	C++ Source File	2 KB

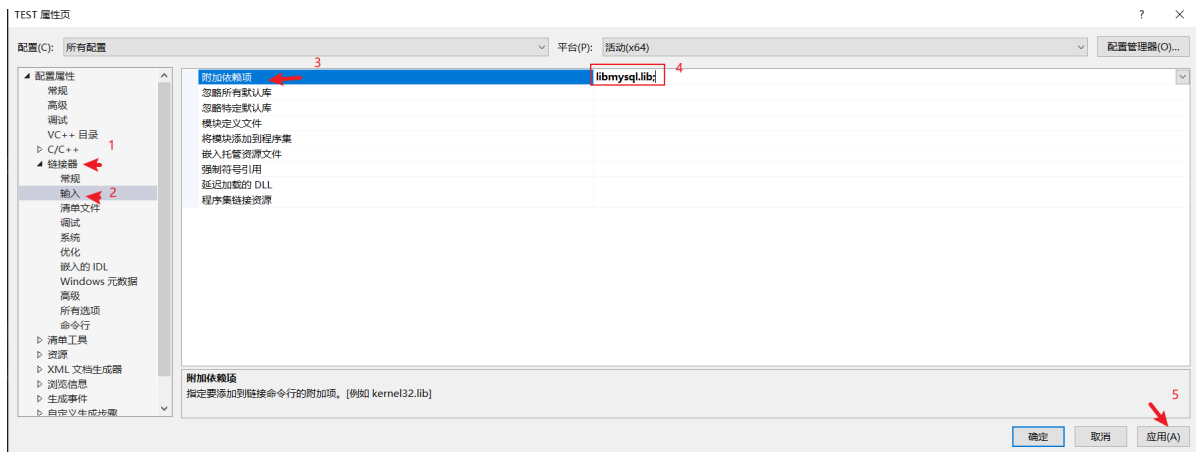
然后回到项目，右键项目》属性



将include和lib的地址分别添加到包含目录和库目录的前面，不要忘记加分号，然后点击应用。



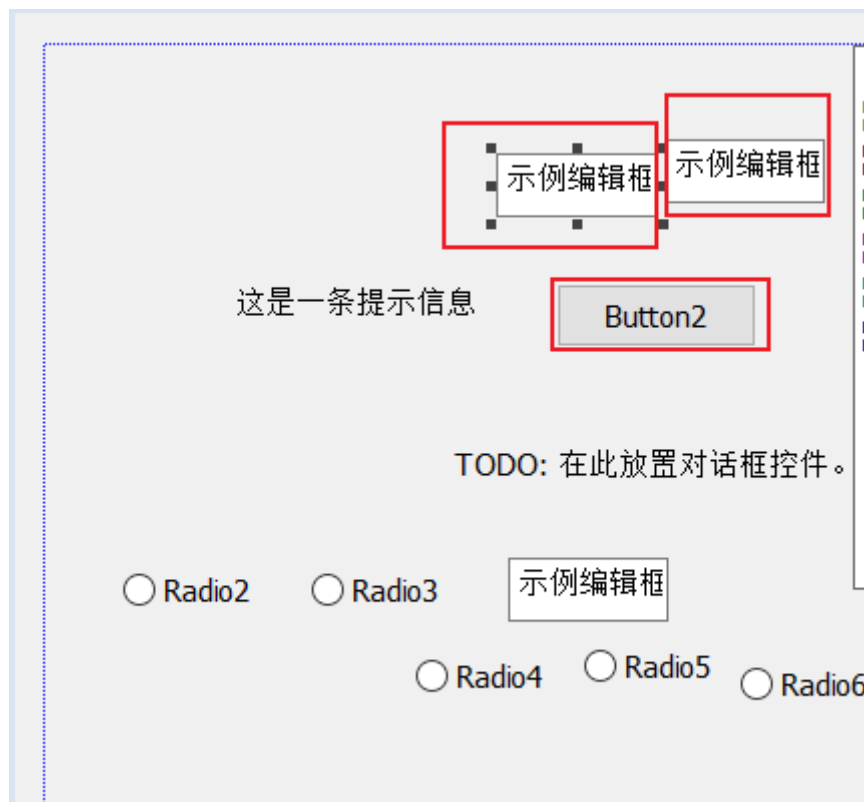
然后点击**链接器》输入**，将libmysql.lib;字段添加到附加依赖项。



这样环境配置就完成了。

### 3.编写代码连接数据库并实现查找

需求，通过左边的文本框输入id，点击按钮后把结果输出到右边的框中，左边框的id为text，右边为text2。





添加Conn.h和Conn.cpp对数据库操作进行封装

Conn.h

```
1  #pragma once
2  #ifndef MY_SQL_H
3  #define MY_SQL_H
4
5  #include<iostream>
6  #include<vector>
7  #include<windows.h>
8  #include<winSock.h>
9  #include<mysql.h>
10 using namespace std;
11
12 class Conn
13 {
14     bool _state;//连接状态 true为已连接
15     MYSQL* _mysql;//mysql连接
16     MYSQL_FIELD* _fd;//字段列数组
17     char _field[32][32];//存字段名二维数组
18     MYSQL_RES* _res;//这个结构代表返回行的一个查询结果集
19     MYSQL_ROW _column;//一个行数据的类型安全(type-safe)的表示，表示数据行的列
20     char _sql[150];//查询语句
21 public:
22     Conn();
23     void Exq(string sql);
24     vector<vector<string>>>getInfo();
25     string getNameById(string id);
26 };
27 #endif
```

Conn.cpp

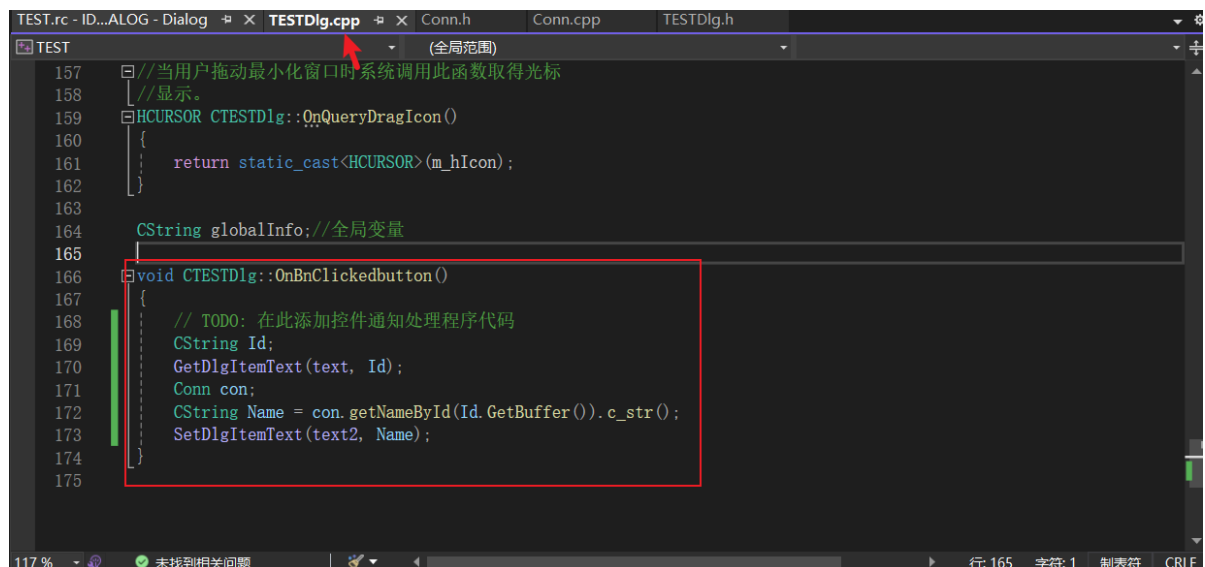
```
1  #include "Conn.h"
2  using namespace std;
3  Conn::Conn()
4  {
5      _state = false;
6      _mysql = new MYSQL;
7      _fd = nullptr;
8      memset(_field, NULL, sizeof(_field));
9      _res = nullptr;
10     _column = nullptr;
11     memset(_sql, NULL, sizeof(_sql));
12     //mysql_init(_mysql);
13     _mysql = mysql_init(NULL);
14     if ((mysql_real_connect(_mysql, "localhost", "root", "123456", "test",
15     3306, NULL, 0))) //中间分别是主机，用户名，密码，数据库名，端口号（可以写默认0或者3306
16     等），根据自己的信息配置
17     {
18         _state = true;
19         //printf("Connected succeed\n");
20     }
21     else cout << "连接失败\n";
```

```

20 }
21
22
23 void Conn::Exq(string sql)//执行当前的sql语句
24 {
25     sprintf_s(_sql, sql.data());
26     mysql_query(_mysql, "set names gbk");
27     mysql_query(_mysql, _sql);
28 }
29
30
31 vector<vector<string>> Conn::getInfo()//得到查询的结果
32 {
33     vector<vector<string>>ans;
34     _res = mysql_store_result(_mysql);
35     int cnt = _res->field_count;
36     while (_column = mysql_fetch_row(_res))
37     {
38         vector<string>tmp;
39         for (int i = 0; i < cnt; i++)tmp.emplace_back(_column[i]);
40         ans.emplace_back(tmp);
41     }
42     return ans;
43 }
44
45 string Conn::getNameById(string id)
46 {
47     string Sql = "select * from student where id = '" + id + "'";
48     Exq(Sql);
49     vector<vector<string>> ans = getInfo();
50     return ans[0][1];
51 }
52

```

在对话框对应文件中引入Conn文件并进行按钮功能的编写：



运行结果如下：

2 lisi

这是一条提示信息

Button2

TODO: 在此放置对话框控件。

☐ Radio2 ☐ Radio3

☐ Radio4 ☐ Radio5 ☐ Radio6

确定

查询成功。