

一、创建 MFC 应用程序

1. 在 VS2017 里按“文件-》新建-》项目”选择菜单项，弹出“新建项目”对话框，如图 1.1 所示。在对话框左侧的选择创建的“应用程序类型”区域选择“Visual C++”下的“MFC/ATL”，然后在中间区域选择“MFC 应用程序”来创建 MFC 应用程序项目，在下面的“名称(N):”输入框里输入应用程序项目名称（此处的项目名称会作为生成的类的名称的一部分，所以要使用英文名称，不要用中文），最后点击“确定”按钮，开始创建 MFC 应用程序。

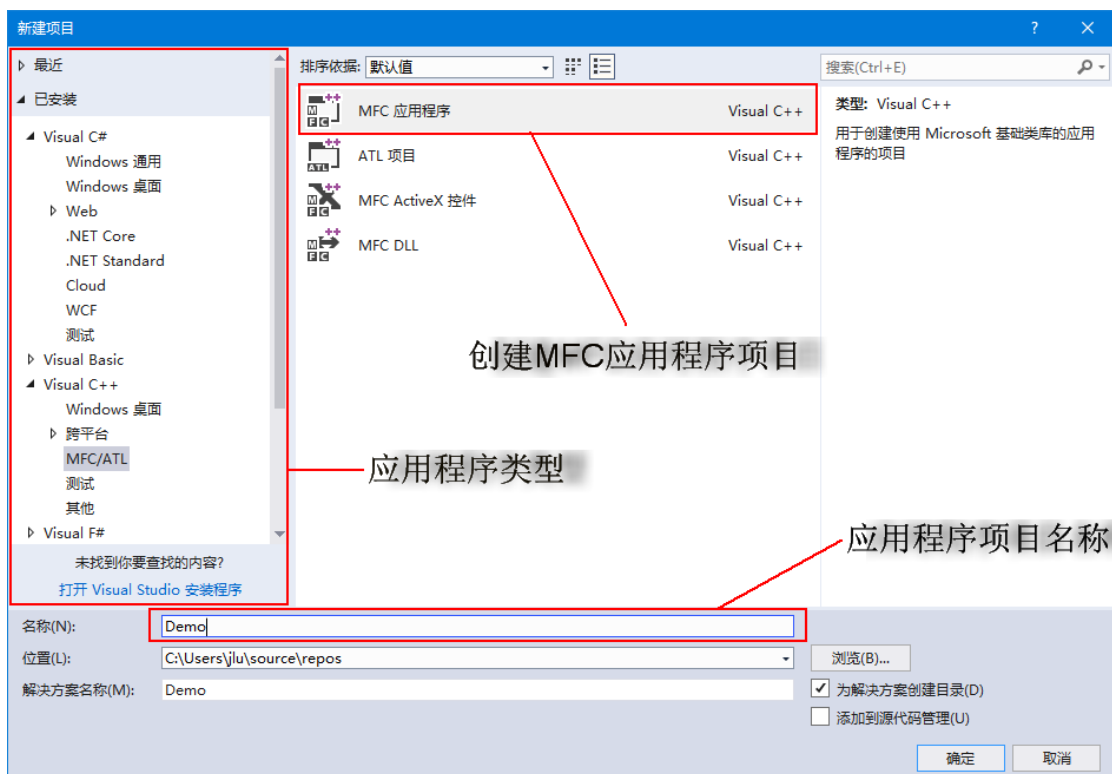


图 1.1 新建项目

2. 点击“确定”按钮后，出现“MFC 应用程序-应用程序类型选项”对话框，用于设置要创建的应用程序的类型，如图 1.2 所示。在本对话框中可以设置要创建的 MFC 应用程序的类型，针对创建满足实验课要求的应用程序，只需在“应用程序类型(T):”下拉框中选择“单个文档”，在“项目样式:”下拉框中选择“MFC standard”即可，其他的所有选项都使用默认值。点击“完成”按钮，开始创建 MFC 应用程序。

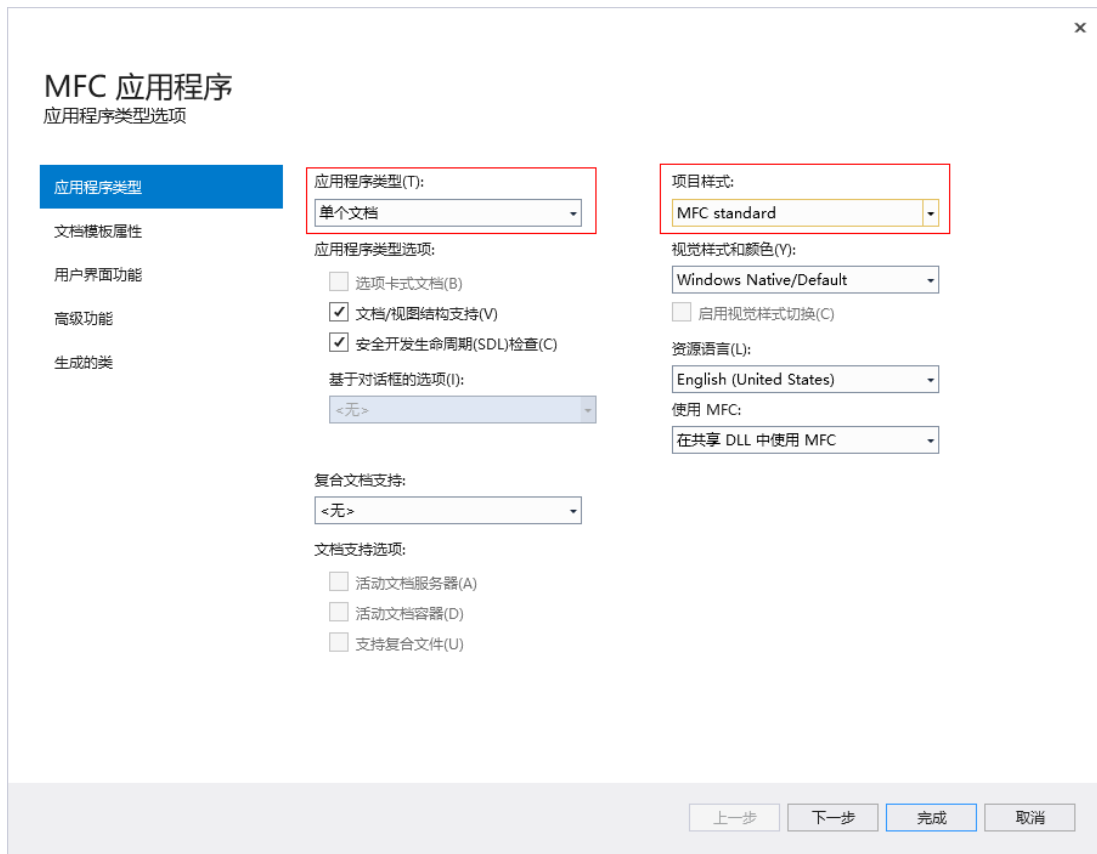


图 1.2 MFC 应用程序-应用程序类型选项

3. 创建 MFC 应用程序项目后，还需要打开两个视图，用于编辑 MFC 应用程序（编程环境可能没有打开这两个视图，如果已经打开，可忽略此步骤）。按“视图-》类视图”选择菜单项，打开“类视图”，如图 1.3 所示；按“视图-》其他视图-》资源视图”选择菜单项，打开“资源视图”，如图 1.4 所示。“类视图”中可以看到当前应用程序中都有哪些类，可以看到创建的 MFC 应用程序中自动生成了 5 个类，“CAboutDlg”是对话框，对应一个“关于”对话框；“CDemoApp”是应用程序入口类，程序从该类开始执行；“CDemoDoc”为文档类，用于存储数据；“CDemoView”为视图类，用于处理应用程序显示；“CMainFrame”为主窗体框架类，包括了应用程序窗口的菜单，工具条，状态栏等部分。图形学实验课主要是为了绘图，所以只需编辑视图类，或者新建类编写代码，其他四个类不用修改。“资源视图”列出了应用程序中所包含的所有资源，包括了“Dialog”对话框，“Menu”菜单，“Toolbar”工具条，这三项是实验课上会用到的。

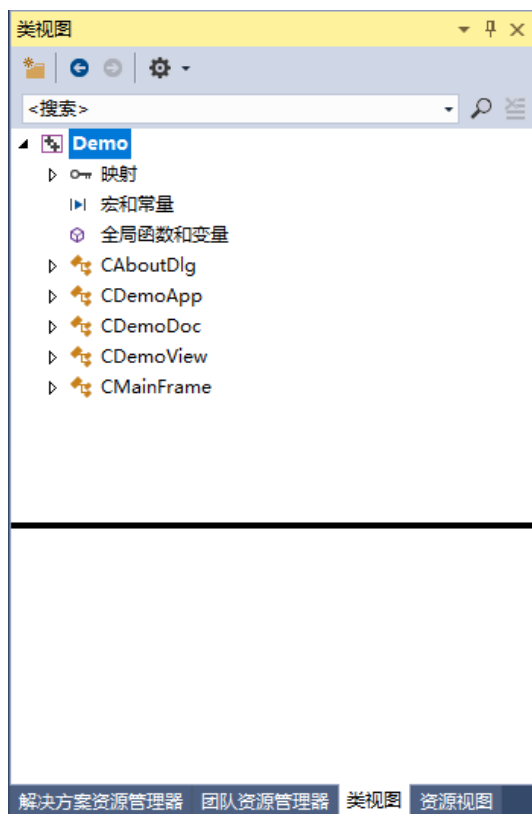


图 1.3 类视图

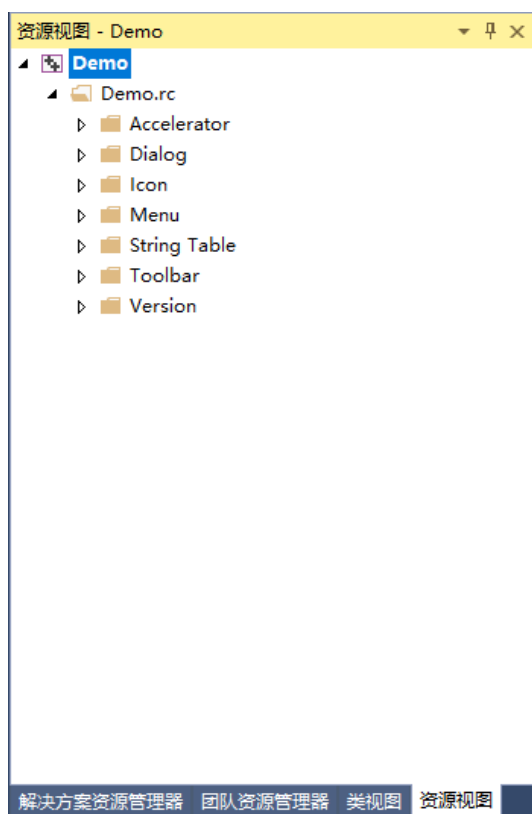


图 1.4 资源视图

4. 此时的 MFC 应用程序已经可以运行，运行结果如图 1.5 所示，窗口中间的白色区域为视图区，绘图即在该区域完成。

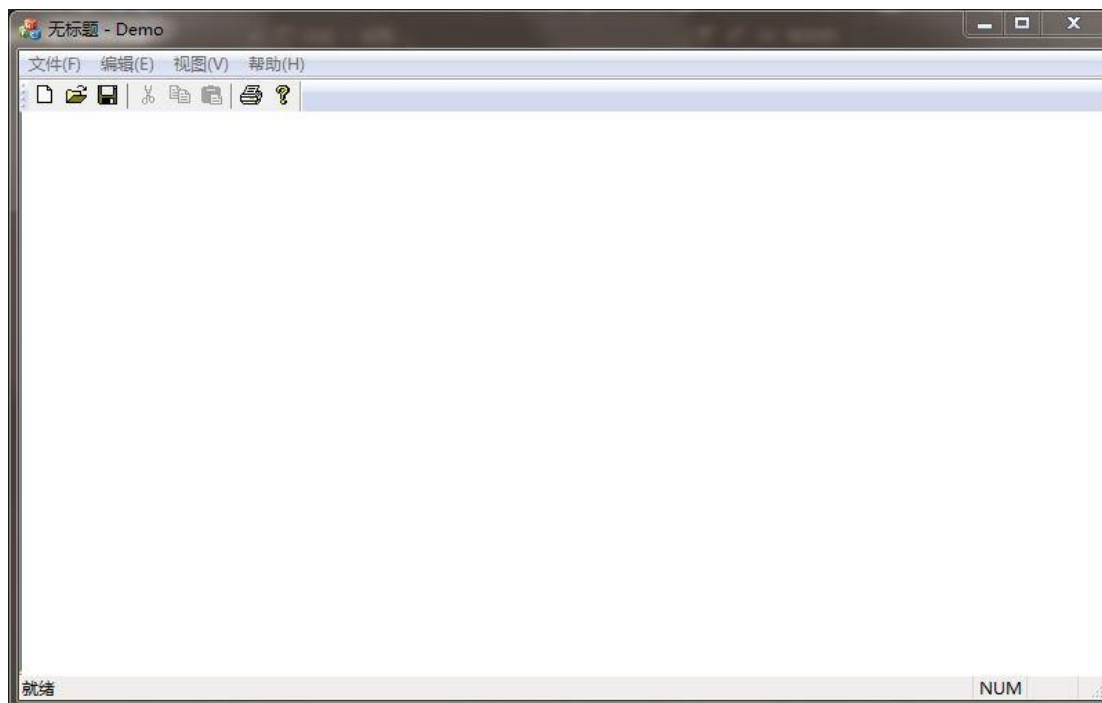


图 1.5 MFC 应用程序窗口

二、添加菜单项并连接处理函数

1. 选择“资源视图”，展开“Menu”节点，可以看到应用程序的初始菜单“IDR_MAINFRAME”，如图 2.1 所示。

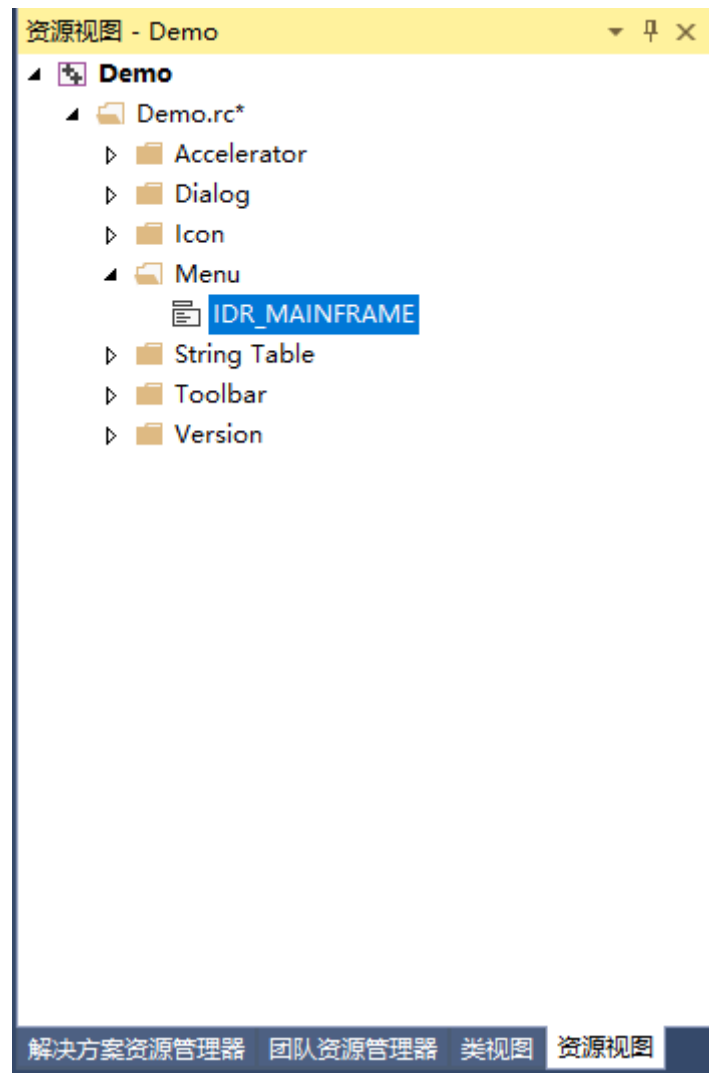


图 2.1 资源视图，初始菜单

2. 双击“IDR_MAINFRAME”，可以打开初始菜单，如图 2.2 所示。菜单里写着“请在此处键入”的地方可以添加菜单项。

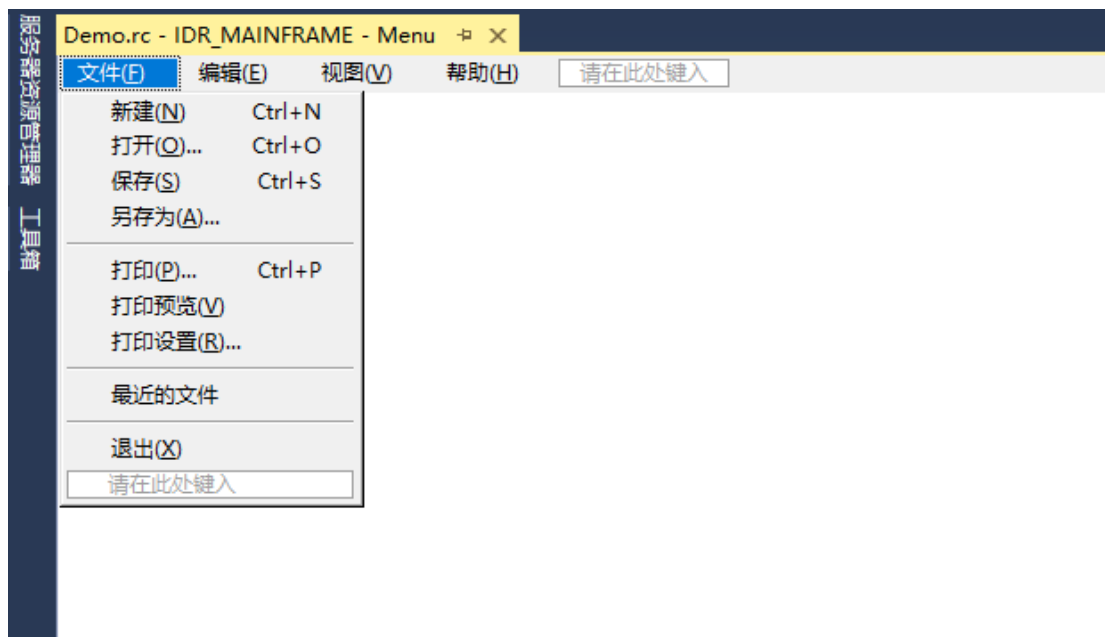


图 2.2 初始菜单

3. 用鼠标点击菜单里右侧的“请在此处键入”，输入菜单项显示内容，比如输入“Draw”，如图 2.3 所示。

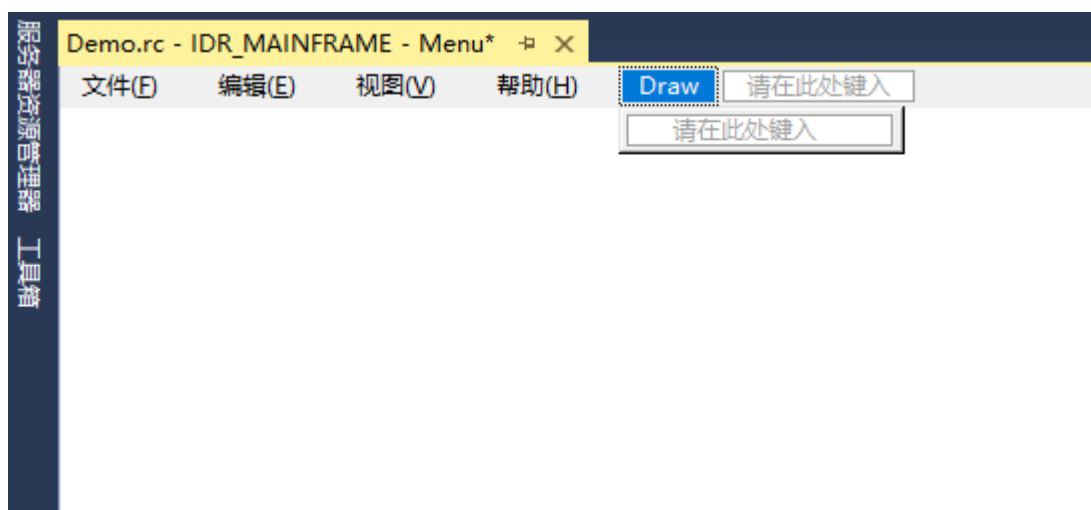


图 2.3 输入菜单项名称

4. 此时可在“属性”视图中看到菜单项的属性，如图 2.4 所示。其中属性“Popup”值为“True”，代表该菜单项为弹出菜单项，可包含子菜单。弹出菜单不能连接处理函数。

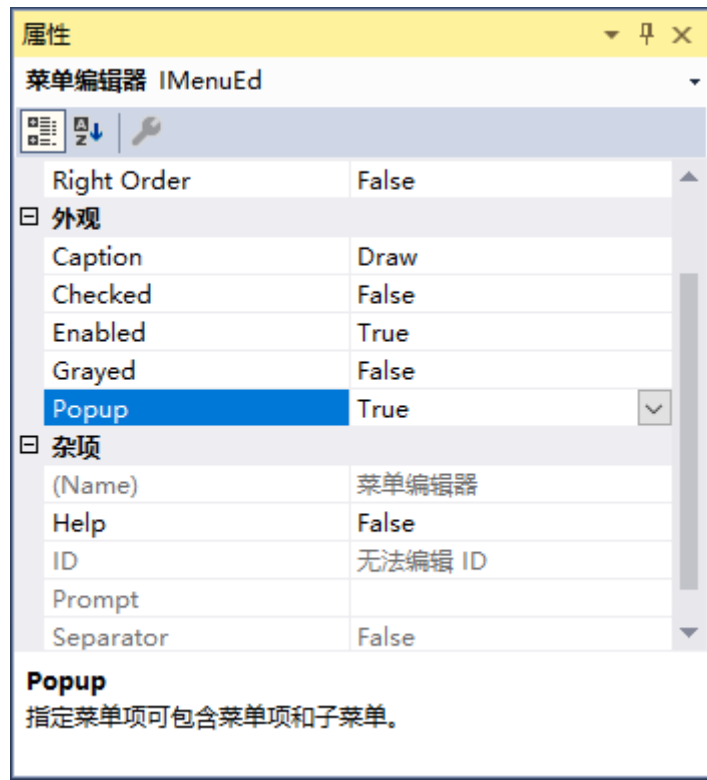


图 2.4 菜单项属性

5. 在“Draw”菜单项下一级添加一个菜单项“Set Color”，如图 2.5 所示。修改该菜单项的属性“Popup”值为“False”，设置菜单项的“ID”为“ID_DRAW_SETCOLOR”，如图 2.6 所示。该菜单项可以连接处理函数。“ID”属性是菜单项的标识符，应使用有描述意义的 ID。

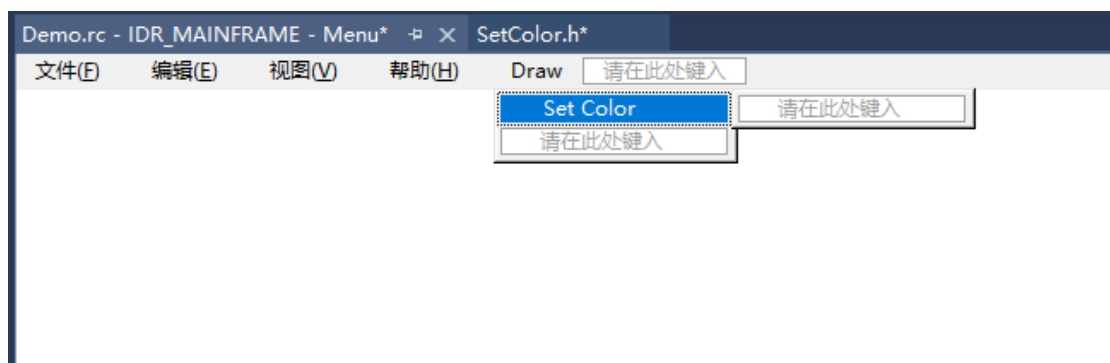
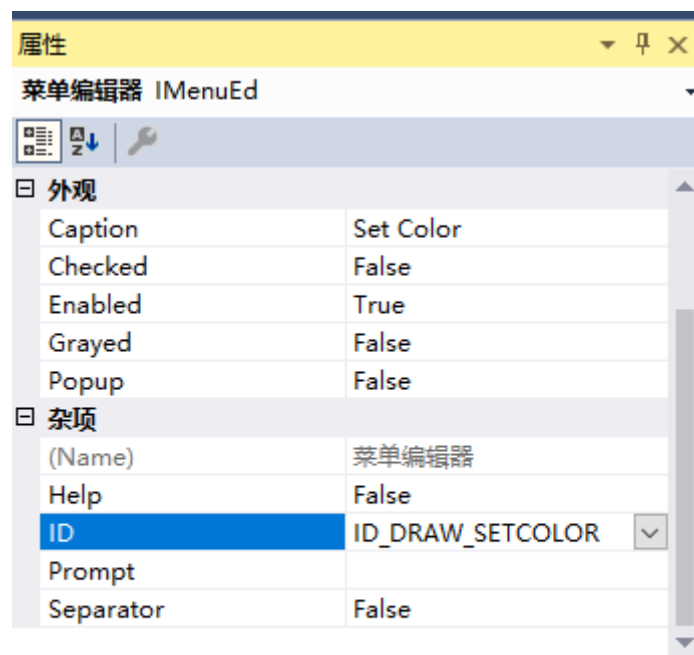


图 2.5 添加二级菜单



ID

指定菜单项或菜单资源的标识符。

图 2.6 设置菜单项 ID

6. 为菜单项连接处理函数，需要使用到“类向导”。在 VS2017 中按“项目-》类向导”选择菜单项，可以打开“类向导”对话框，如图 2.7 所示。“类名(N):”下拉框中选择视图类“CDemoView”，中间的选项页选择“命令”页。在“对象 ID(B):”列表框中找到刚才添加的菜单项 ID “ID_DRAW_SETCOLOR”，此时右侧的“消息(S):”列表框中会列出菜单项所支持的所有消息。其中“COMMAND”消息是程序运行时，用户用鼠标选择菜单项时所产生的消息。

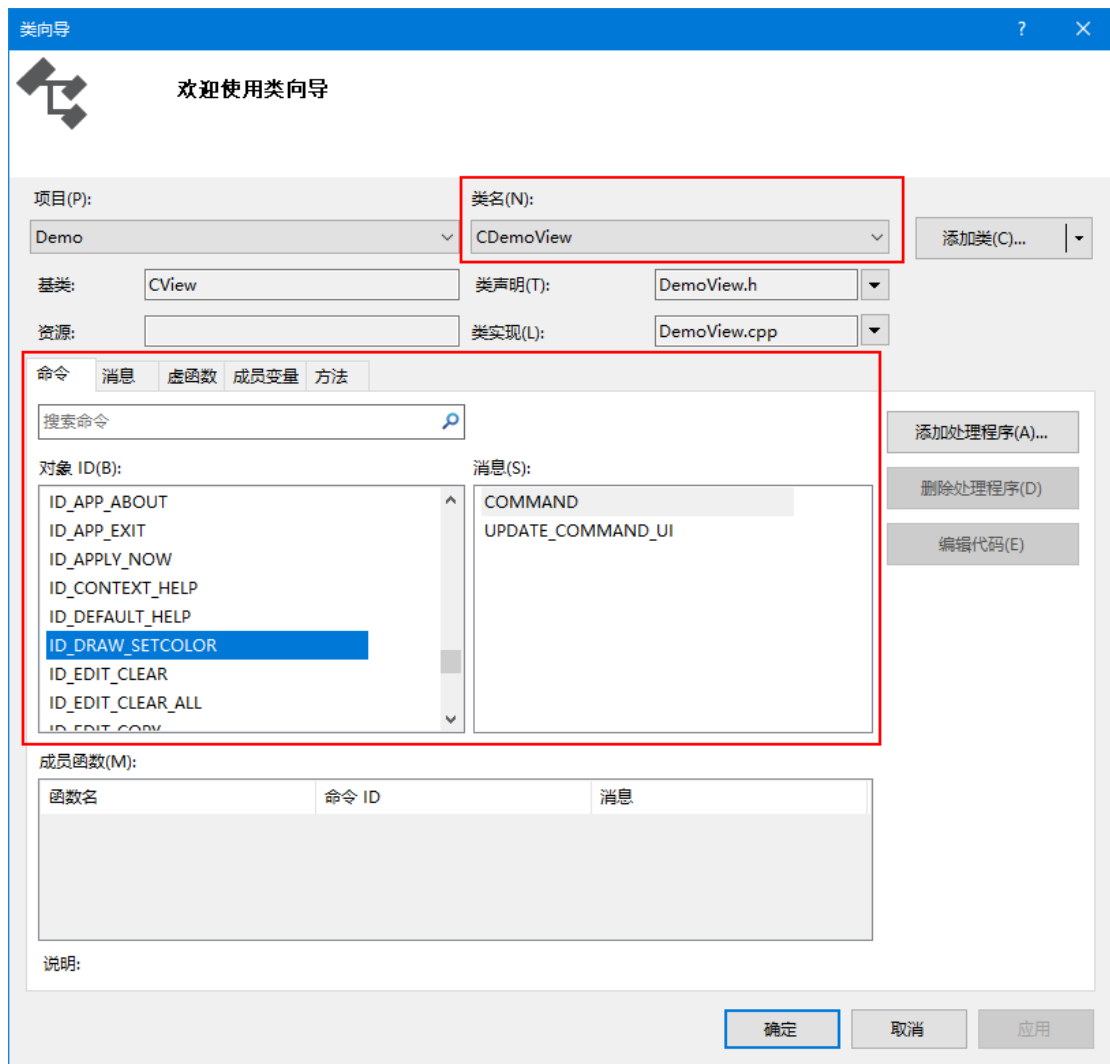


图 2.7 类向导-添加菜单项处理函数

7. 鼠标左键双击“COMMAND”，弹出“添加成员函数”对话框，如图 2.8 所示。弹出对话框时，会给该菜单项的处理函数一个默认的处理函数名，可以修改该函数名，但是建议使用该默认函数名。

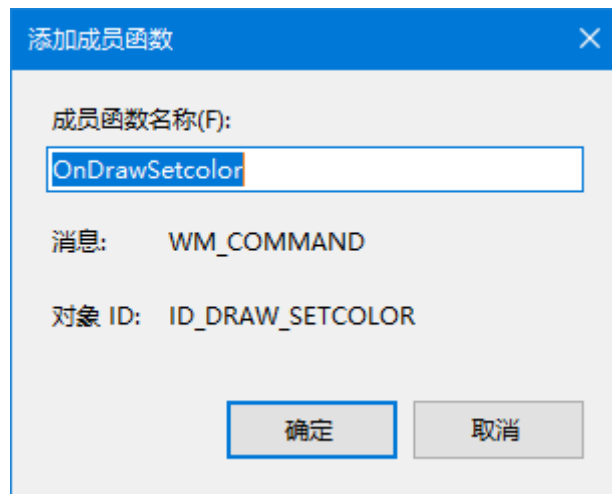


图 2.8 添加成员函数-菜单项处理函数

8. 在“添加成员函数”对话框中点击“确定”按钮，添加该处理函数，如图 2.9 所示。
在“类向导”对话框中的“成员函数(M):”列表框中可以看到添加的处理函数。

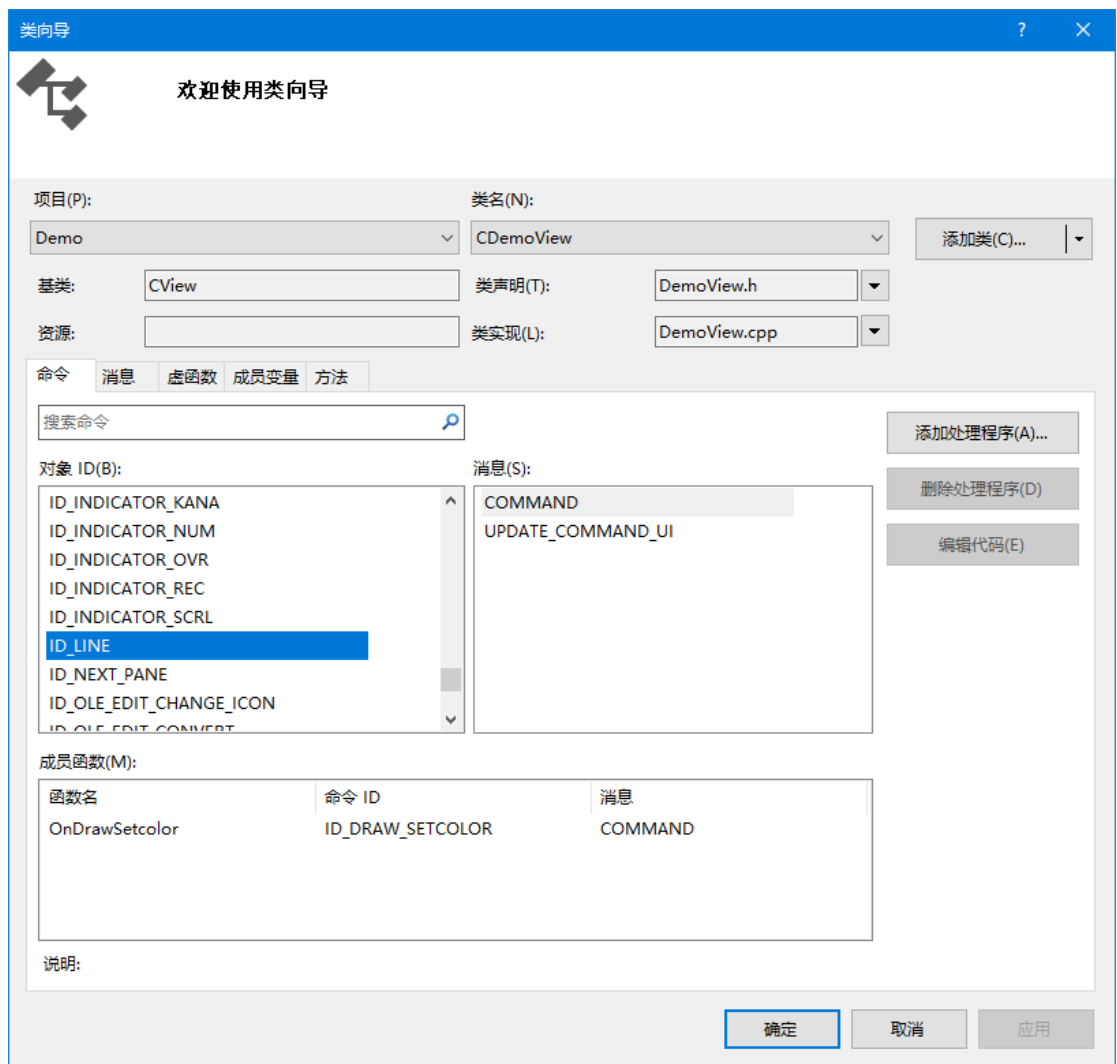


图 2.9 类向导-添加了菜单项处理函数 OnDrawSetcolor

9. 点击“类向导”对话框的“确定”按钮，可以编辑该处理函数，如图 2.10 所示。

`// CDemoView 消息处理程序`

```
void CDemoView::OnDrawSetcolor()
{
    // TODO: 在此添加命令处理程序代码
}
```

图 2.10 菜单项 COMMAND 消息处理函数

三、添加工具条按钮并连接处理函数

1. 选择“资源视图”，展开“Toolbar”节点，可以看到应用程序的初始工具条“IDR_MAINFRAME”，如图 3.1 所示。

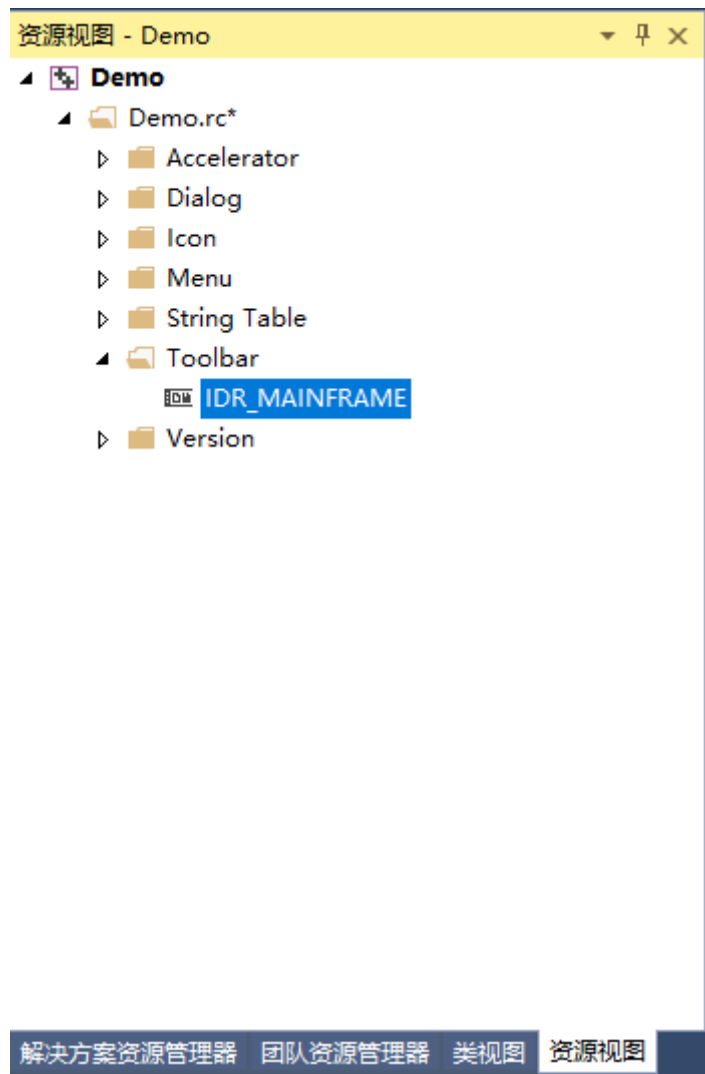


图 3.1 资源视图，初始工具条

2. 双击“IDR_MAINFRAME”，可以打开初始工具条，如图 3.2 所示。最右侧有一个矩形虚线框的按钮可以用于添加新按钮。下面左侧是选中按钮的显示样式，右侧用于绘制按钮。

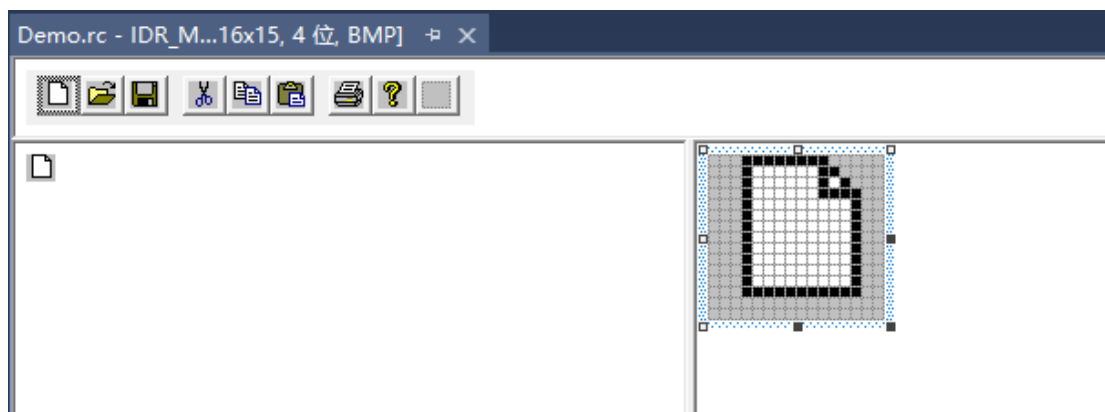


图 3.2 初始工具条

3. 打开工具条后，在 VS2017 的工具条中会有个用于绘制按钮的工具条，如图 3.3 所示。



图 3.3 工具条按钮绘制工具条

- 在图 3.1 的初始工具条中选择最右侧的按钮，用 3.3 工具条中的直线工具在按钮的绘制区中绘制一条直线段，如图 3.4 所示。该按钮就成为初始工具条上的一个新按钮。

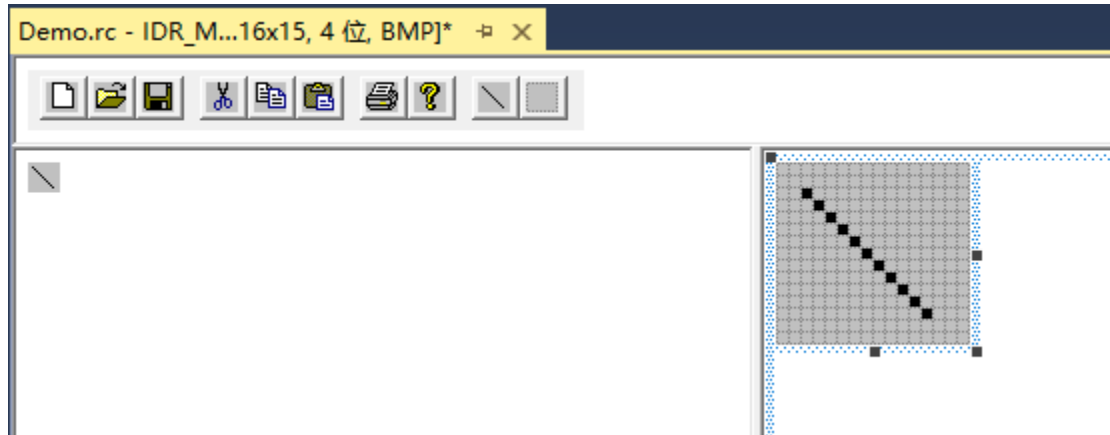


图 3.4 绘制工具条按钮

- 修改工具条按钮“属性”视图中“ID”属性值为有描述意义的值，如图 3.5 所示。不要使用自动生成的 ID 值。从工具条按钮的样式及 ID 可以看到，该按钮用于执行绘制直线段功能。

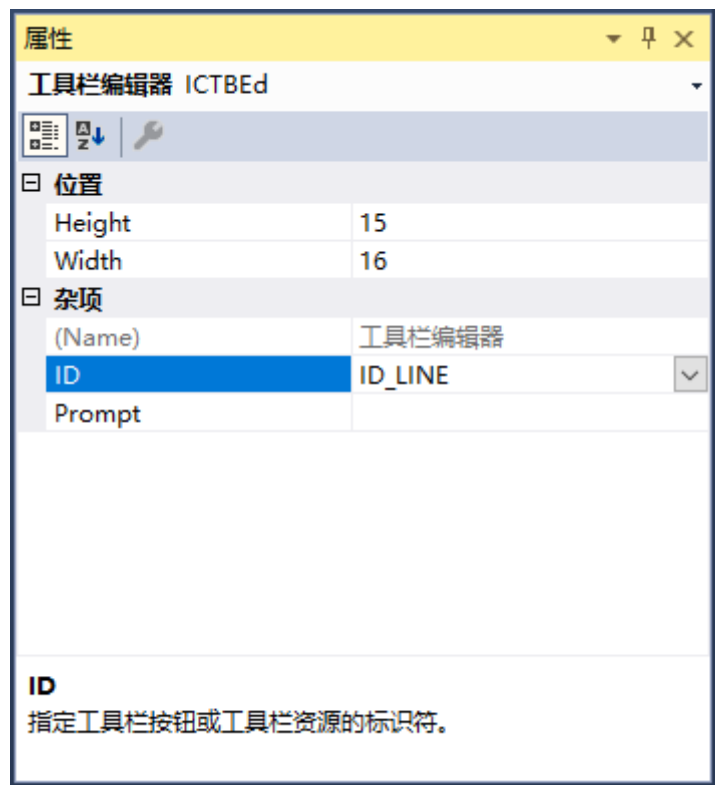


图 3.5 工具条按钮属性

- 为工具条按钮添加处理函数的过程与菜单项相同。打开“类向导”对话框，如图 3.6 所示。“类名(N):”下拉框中选择视图类“CDemoView”，中间的选项页选择“命令”页。在“对象 ID(B):”列表框中找到刚才添加的工具条 ID “ID_LINE”，此时右侧的“消息(S):”列表框中会列出工具条按钮所支持的所有消息。其中“COMMAND”消息是程序运行时，用户用鼠标点击工具条按钮时所产生的消息。

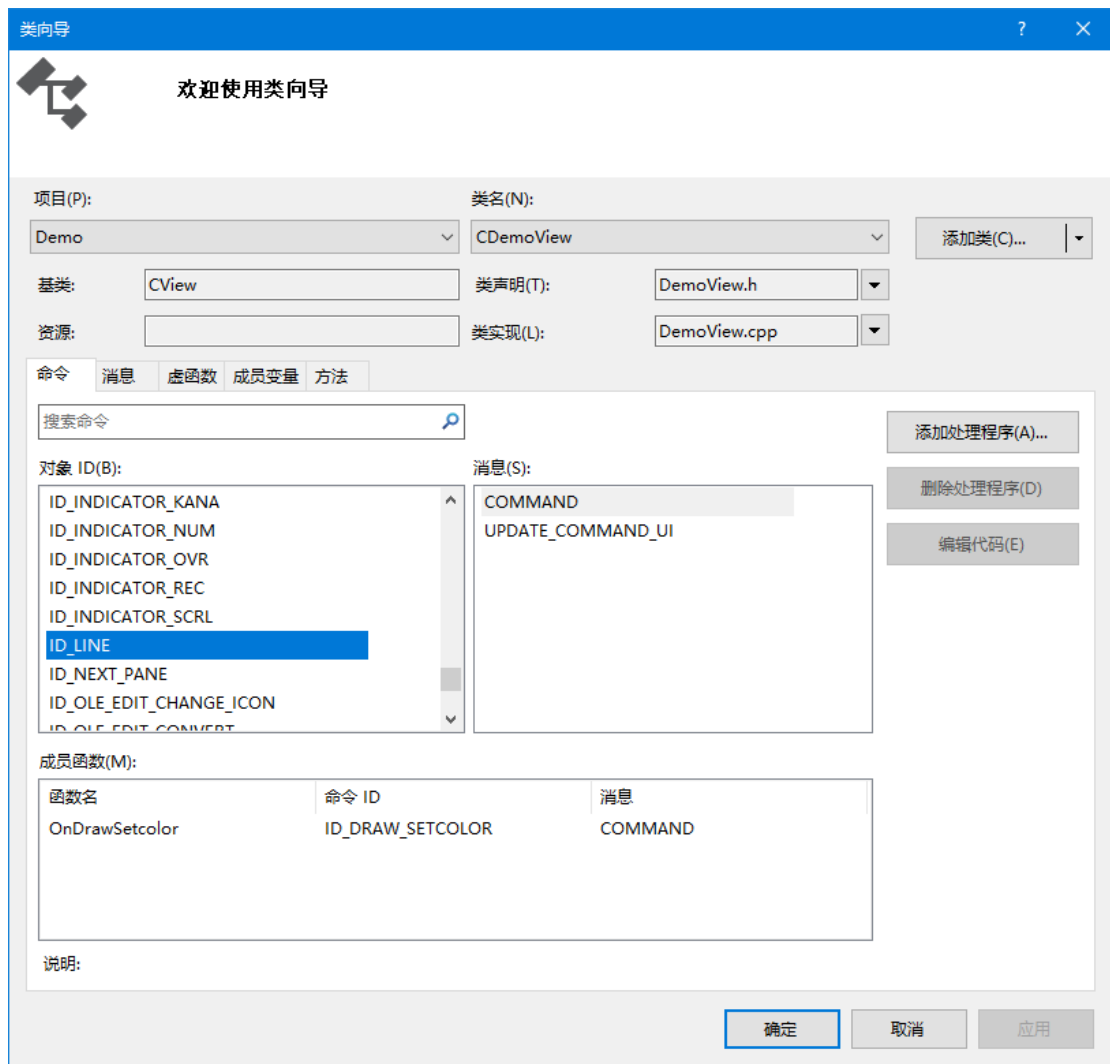


图 3.6 类向导-添加工具条按钮处理函数

- 鼠标左键双击“COMMAND”，弹出“添加成员函数”对话框，如图 3.7 所示。使用生成的默认处理函数名。

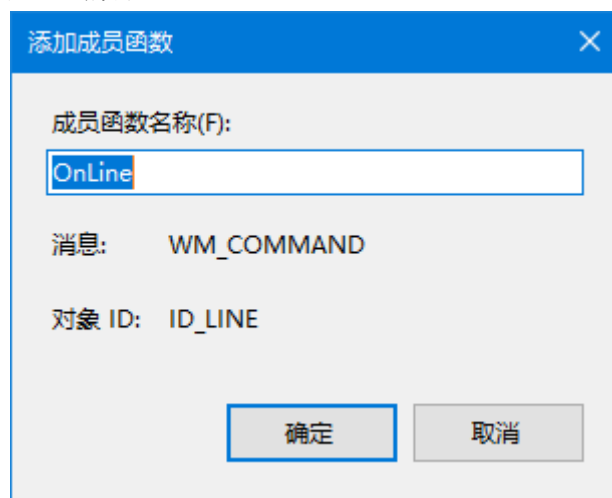


图 3.7 添加成员函数-工具条按钮处理函数

- 点击“添加成员函数”对话框中的“确定”按钮，添加处理函数，在“类向导”对话框的“成员函数(M):”列表框中可以看到添加的处理函数，如图 3.8 所示。

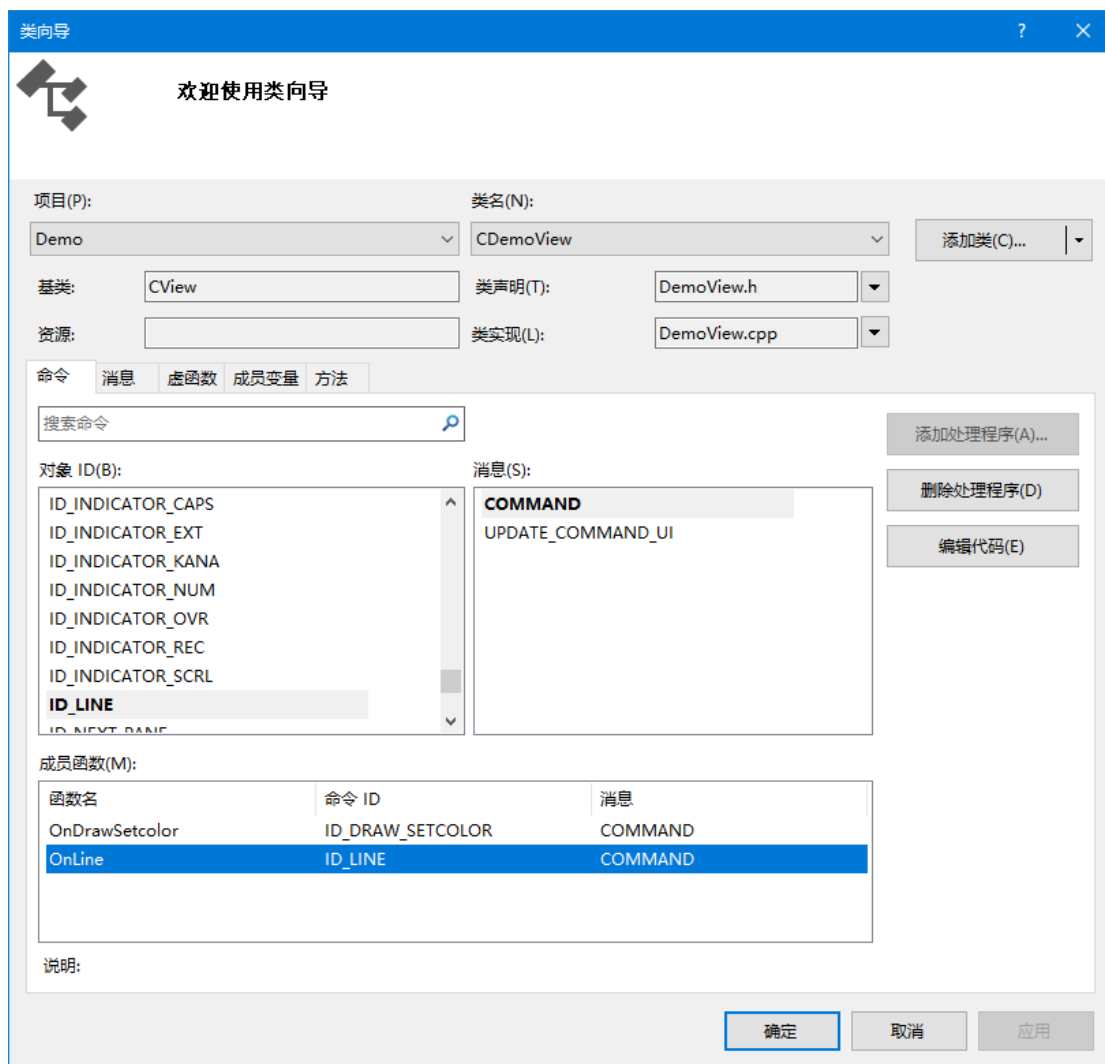


图 3.8 类向导-添加了工具条按钮处理函数 OnLine

9. 点击“类向导”的“确定”按钮，可以编辑该处理函数，如图 3.9 所示。

```
void CDemoView::OnLine()
{
    // TODO: 在此添加命令处理程序代码
}
```

图 3.9 工具条按钮 COMMAND 消息处理函数

四、创建并使用对话框

1. 使用对话框需要首先创建对话框资源。选择“资源视图”，展开“Dialog”节点，可以看到应用程序已有的“关于”对话框资源“IDD_ABOUTBOX”。用鼠标右键点击“Dialog”节点，弹出快捷菜单，如图 4.1 所示。

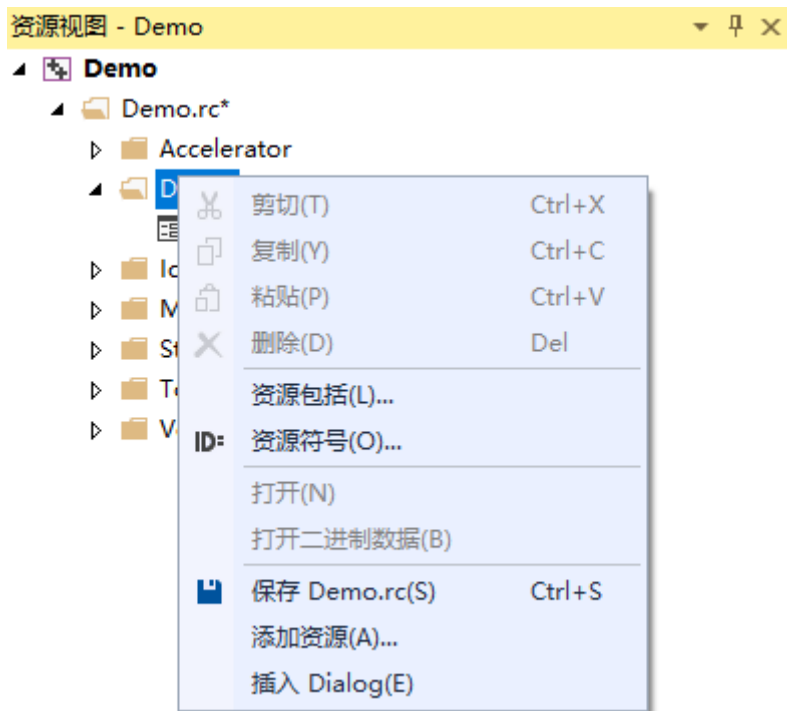


图 4.1 资源视图-创建对话框资源

- 在弹出的快捷菜单中点击“插入 Dialog(E)”，会插入一个新的对话框资源，默认 ID 为“IDD_DIALOG1”，如图 4.2 所示。

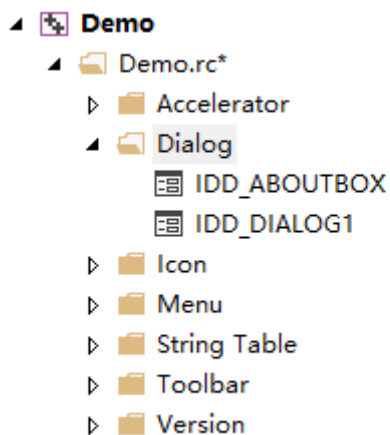


图 4.2 创建新的对话框资源

- 为了从 ID 能够看出该对话框是什么对话框，可以在对话框的“属性”视图中修改“ID”的值，如图 4.3 所示。从 ID 值可以看出，要创建的对话框是用来设置颜色的。

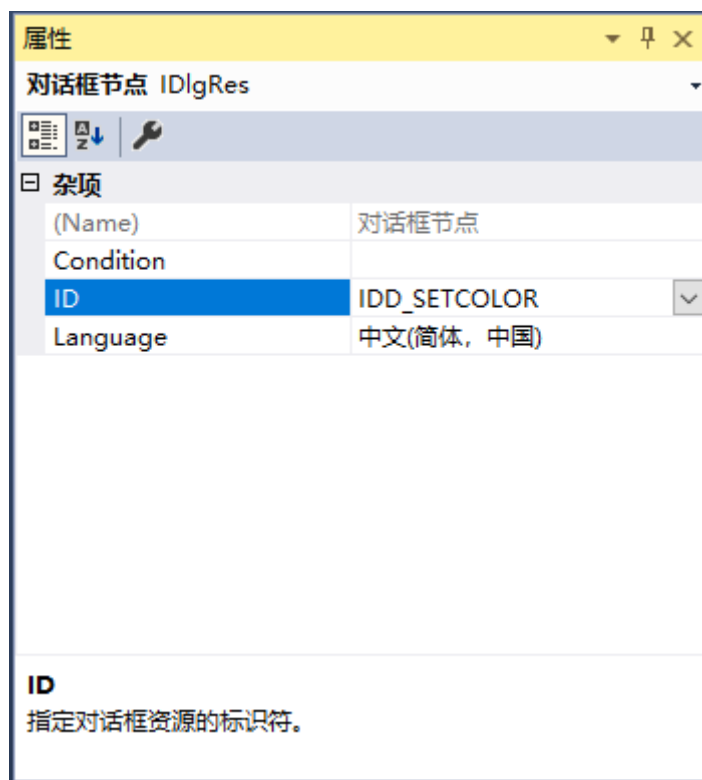


图 4.3 设置对话框 ID

4. 双击新创建的对话框资源节点，可以打开对话框资源编辑区，对对话框资源进行编辑，如图 4.4 所示。对话框中会默认添加“确定”和“取消”按钮。

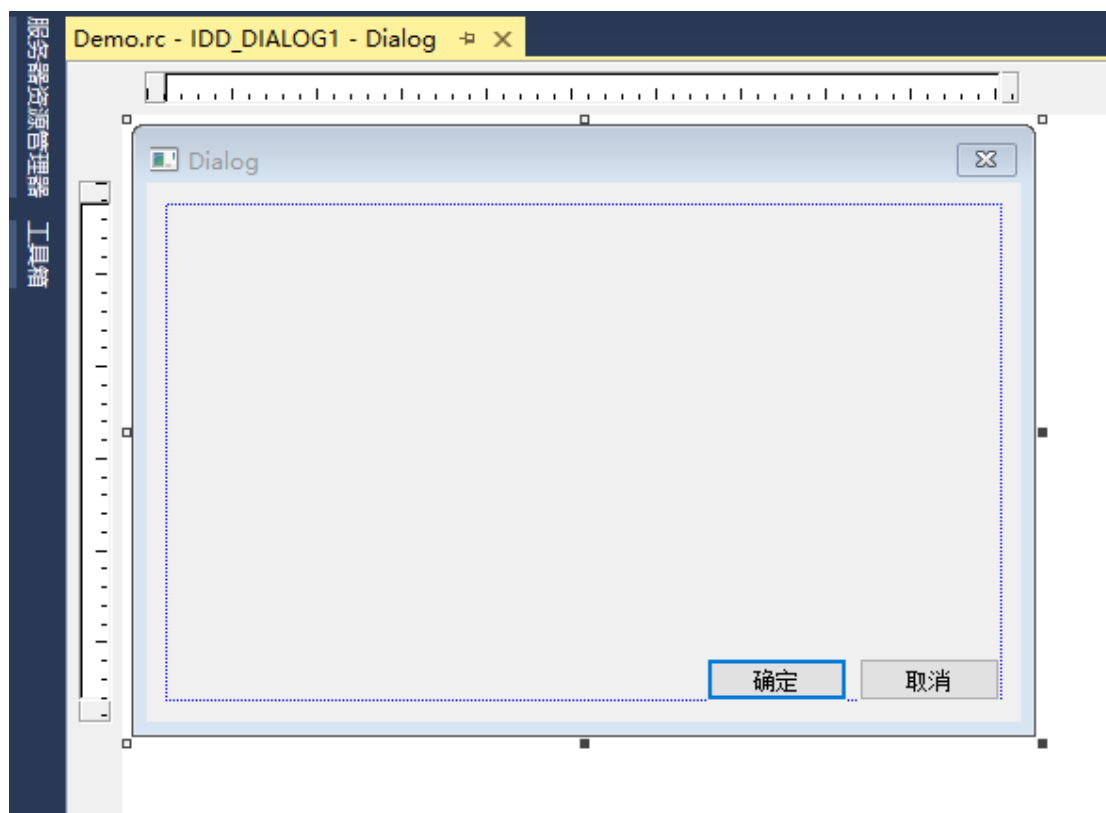


图 4.4 对话框资源编辑区

5. 点击对话框资源编辑区左侧的“工具箱”，可以打开“工具箱”视图，如图 4.5 所示。该视图中列出了 MFC 对话框所支持的所有控件。因为 VS2017 编程环境里的各窗口采用了动态停靠设计，所以工具箱位置可能与图 4.4 不同。

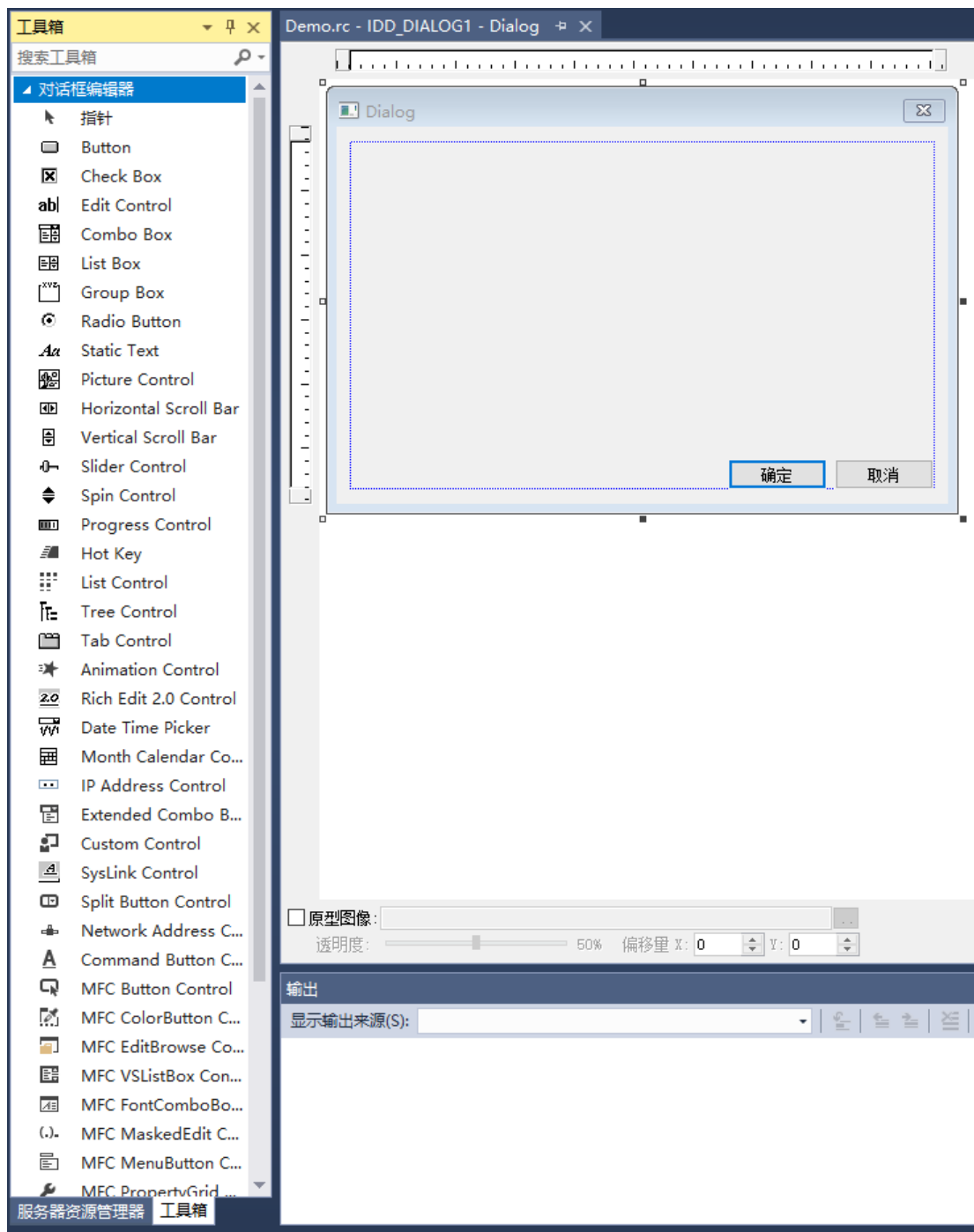


图 4.5 工具箱

6. 创建的对话框用于输入颜色。MFC 中颜色可以利用 RGB 函数获得，RGB 函数有三个整形参数，分别代表了红、绿、蓝三颜色分量，每个分量的取值范围都是 0 到 255。所以，需要使用对话框输入三分量的值，在对话框中添加三个“Static Text”（静态文本）控件（用于说明编辑框应输入的内容）和三个“Edit Control”（编辑框）控件（用于接收用户输入的数据），如图 4.6 所示。

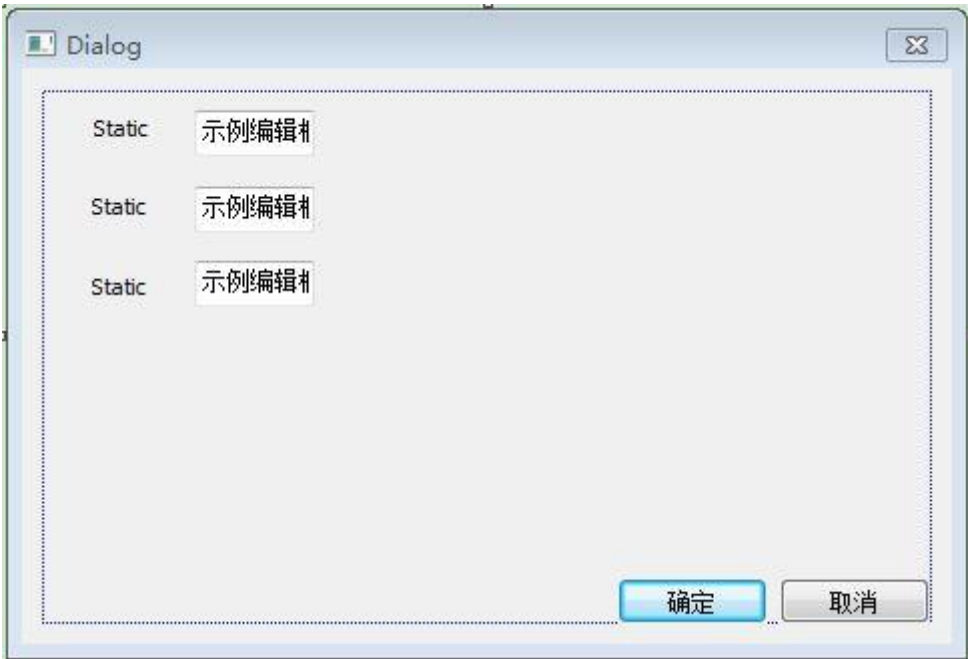


图 4.6 添加了控件的对话框

7. 用鼠标选择一个“Static Text”控件，在“属性”视图中修改“Caption”属性值，即设置静态文本所显示的内容，如图 4.7 所示。选择一个“Edit Control”控件，在“属性”视图中修改“ID”属性值，如图 4.8 所示。默认 ID 为“IDC_EDIT1”，修改为有描述作用的“IDC_R”。其余两个“Static Text”控件的“Caption”属性值修改为“G”和“B”。其余两个“Edit Control”控件的“ID”属性值修改为“IDC_G”和“IDC_B”。静态文本与编辑框一一对应。

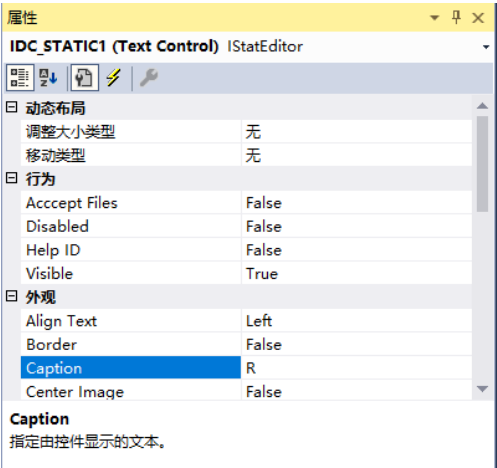


图 4.7 “Static Text” 控件属性

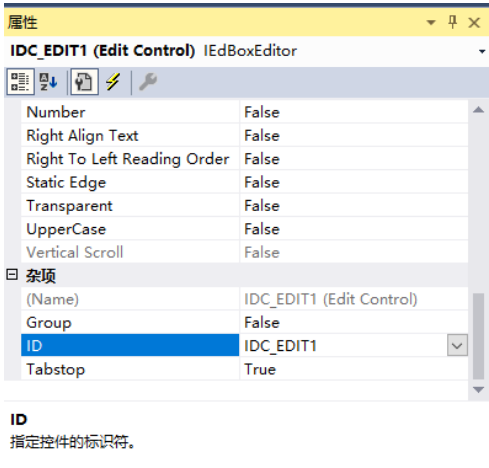


图 4.8 “Edit Control” 控件属性

8. 到此对话框资源创建完毕，需要创建一个类与资源相关联。用鼠标左键在对话框中没有控件的地方双击，弹出“添加 MFC 类”对话框，如图 4.9 所示。“类名(L):”输入框中输入对话框类名，在“基类(B):”下拉框中选择“CDialog”，该类是 MFC 已经编写好的对话框基类。“.h 文件(F):”和“.cpp 文件(P):”使用自动生成的文件名。“对话框 ID(D):”里是当前类所关联的对话框资源的 ID。



图 4.9 添加 MFC 类-添加对话框类

9. 点击“确定”按钮，就会创建该类，在“类视图”中可以看到该类，如图 4.10 所示。

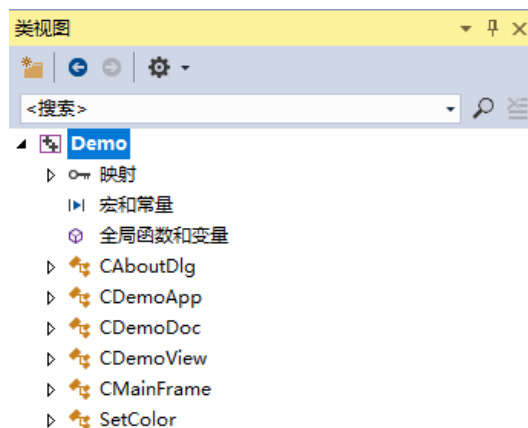


图 4.10 创建了对话框类 SetColor

10. 为了获得用户在对话框中输入的数值，还需要给对话框中的编辑框连接成员变量。需要使用“类向导”，如图 4.11 所示。“类名(N):”下拉框中选择刚创建的对话框类“SetColor”，中间的选项页选择“成员变量”页。在“成员变量(V):”列表框中列出了当前对话框可连接变量的控件 ID，可以看到添加的三个编辑框 ID。

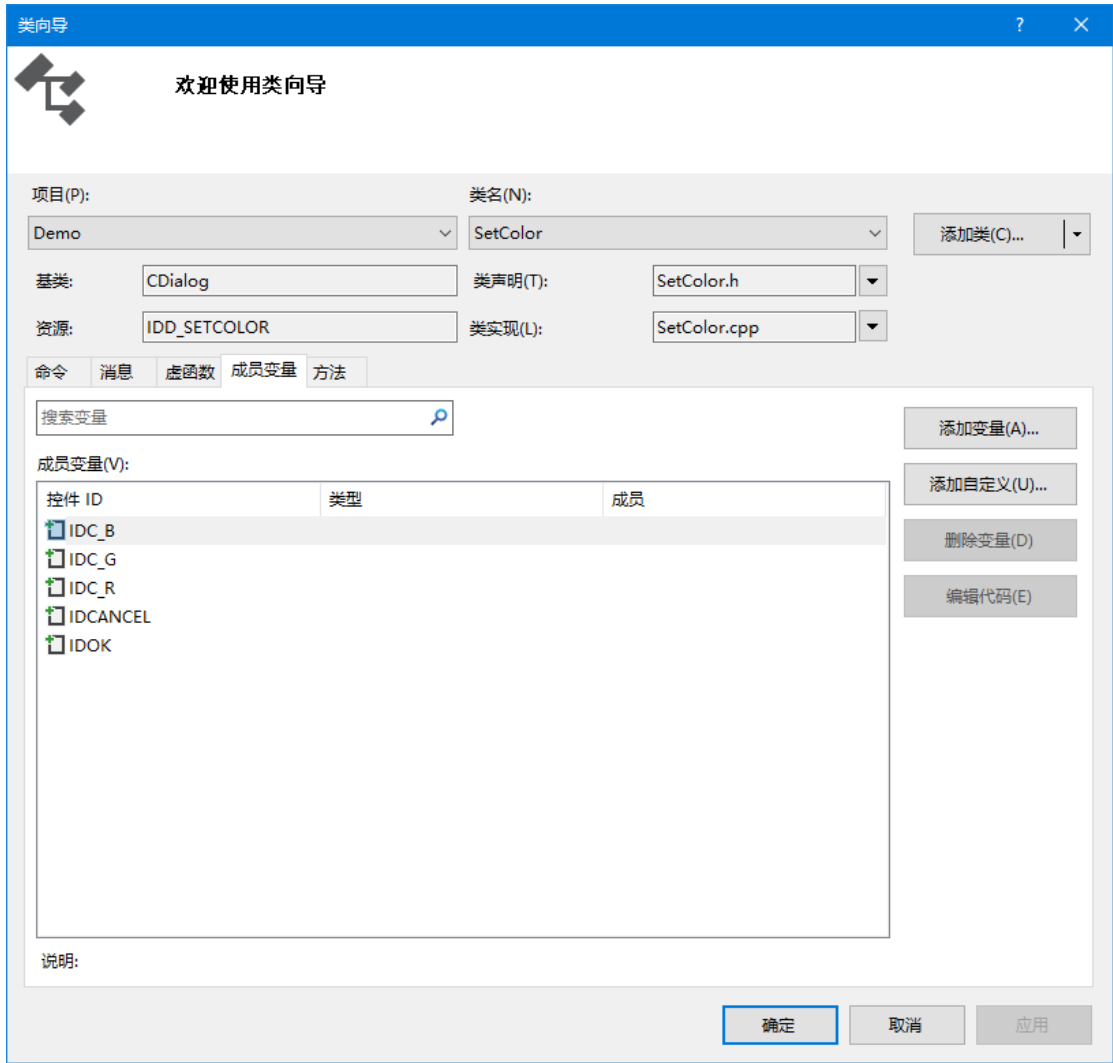


图 4.11 类向导-为对话框控件连接成员变量

11. 用鼠标左键双击“IDC_B”，弹出“添加控件变量”对话框，如图 4.12 所示。在“类别(T):”下拉框中选择“值”，代表需要获得的是控件中输入的值。“名称(N):”输入框中输入的是变量名。“变量类型(V):”输入框中输入变量类型为整型 int。

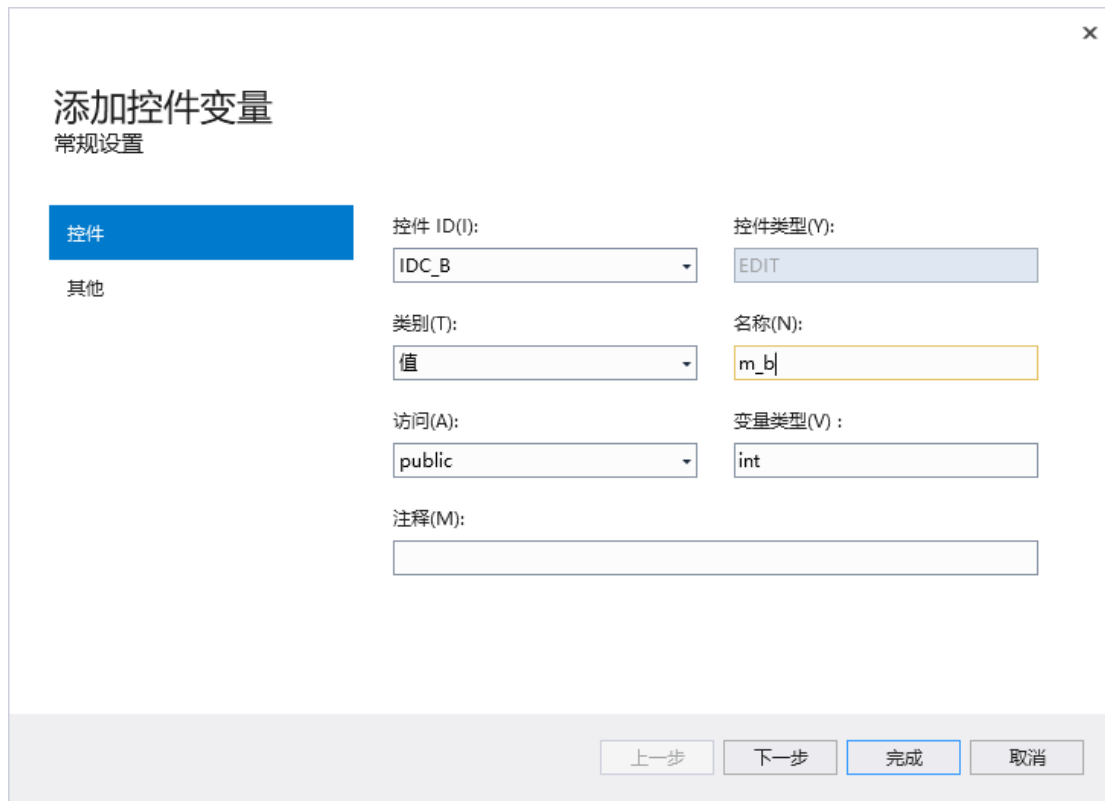


图 4.12 添加控件变量

12. 点击“完成”按钮，在“类向导”对话框的“成员变量(V):”列表框中可以看到控件“IDC_B”与 int 类型的变量 m_b 相关联。相同方法可以给“IDC_G”和“IDC_R”控件关联变量 m_g 和 m_r，如图 4.13 所示。

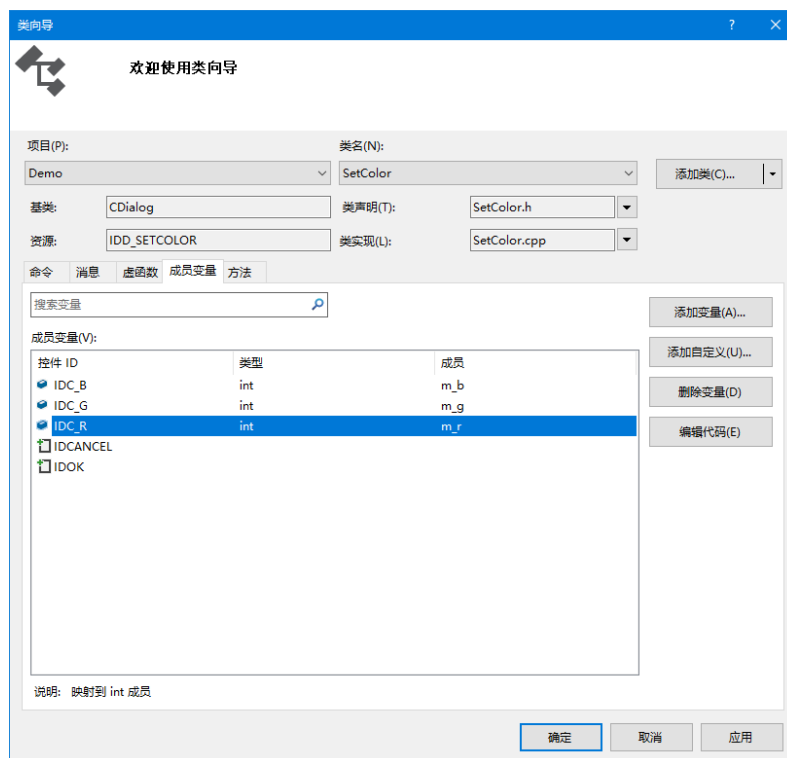


图 4.13 类向导-对话框控件变量

13. 最后在 MFC 程序中调用该对话框，并获得用户输入的数据。代码可如图 4.14 所示，在前面添加的菜单项处理函数中调用对话框，变量 r，b，g 是视图类中定义的成员变量，用于存储当前颜色值。

```
// CDemoView 消息处理程序

#include "SetColor.h"
void CDemoView::OnDrawSetColor()
{
    // TODO: 在此添加命令处理程序代码
    SetColor dia; //构造对话框对象
    dia.m_r = r; dia.m_g = g; dia.m_b = b; //设置当前值到对话框中
    if (dia.DoModal() == IDOK) { //显示对话框，当点击对话框“确定”按钮关闭对话框时，返回IDOK
        r = dia.m_r; g = dia.m_g; b = dia.m_b; //设置对话框中用户输入的值当前值
    }
}
```

图 4.14 调用对话框

五、添加鼠标和键盘消息处理函数

1. 利用“类向导”添加鼠标消息，如图 5.1 所示。“类名(N):”下拉框中选择视图类“CDemoView”，中间的选项页选择“消息”页。在“消息(S):”列表框中列出了视图类所支持的所有消息。常用的鼠标消息有 7 种：“WM_LBUTTONDOWN”（鼠标左键按下），“WM_LBUTTONUP”（鼠标左键抬起），“WM_LBUTTONDBLCLK”（鼠标左键双击），“WM_RBUTTONDOWN”（鼠标右键按下），“WM_RBUTTONUP”（鼠标右键抬起），“WM_RBUTTONDBLCLK”（鼠标右键双击），“WM_MOUSEMOVE”（鼠标移动）。双击要处理的鼠标消息，即可在“现有处理程序(H):”列表框中看到添加的消息处理函数。点击“确定”按钮即可编辑处理函数。

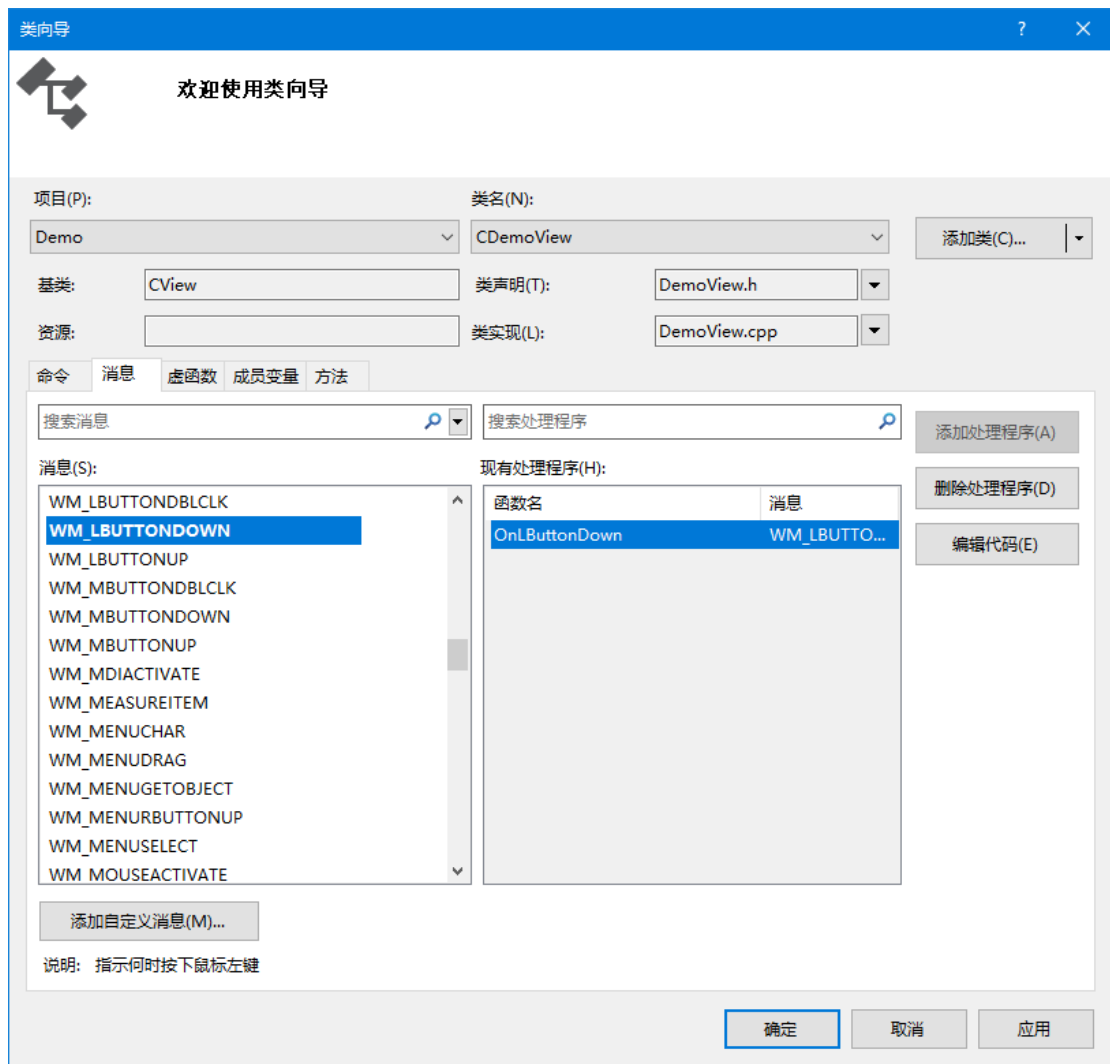


图 5.1 类向导-添加鼠标消息处理函数

2. 利用“类向导”添加键盘消息，如图 5.2 所示。“类名(N):”下拉框中选择视图类“CDemoView”，中间的选项页选择“消息”页。在“消息(S):”列表框中列出了视图类所支持的所有消息。常用的键盘消息有 2 种：“WM_KEYDOWN”（键盘按键按下），“WM_KEYUP”（键盘按键抬起）。双击要处理的键盘消息，即可在“现有处理程序(H):”列表框中看到添加的消息处理函数。点击“确定”按钮即可编辑处理函数。

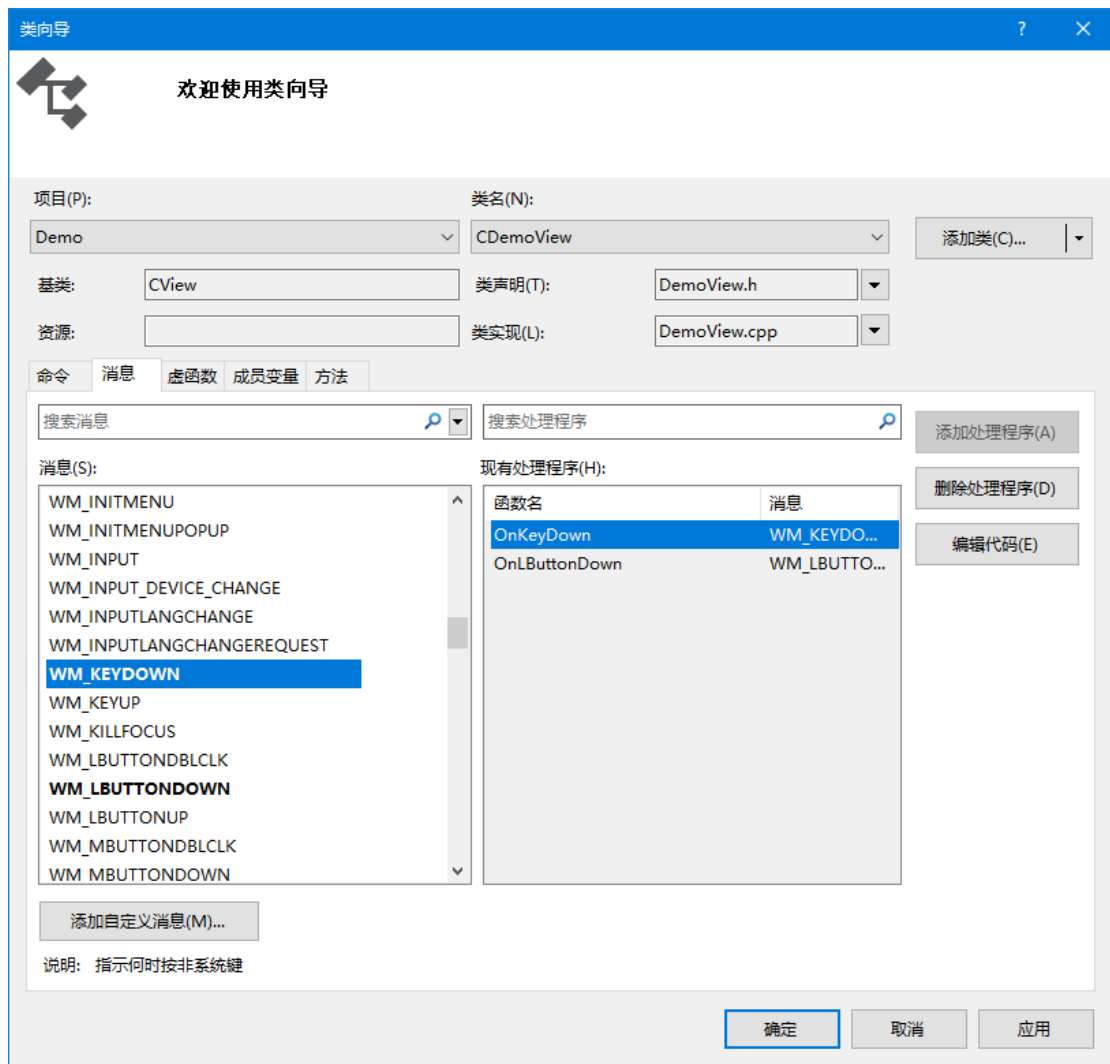


图 5.2 类向导-添加键盘消息处理函数

六、在 MFC 应用程序中添加类

1. 用鼠标右键点击“类视图”的根节点，在弹出的快捷菜单中选择“添加-》类”，会弹出“添加类”对话框，如图 6.1 所示。在“类名(L):”输入框中输入类名，如果有基类，在“基类(B):”输入框中输入。

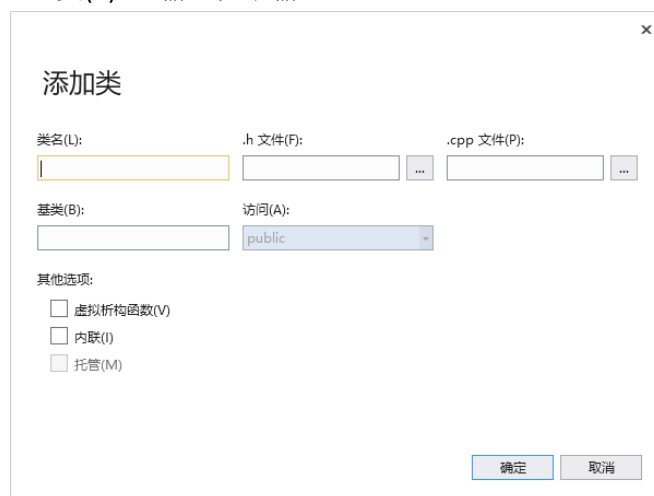



图 6.1 添加类

七、在 MFC 应用程序中添加函数

1. 在视图类中添加函数，可以用鼠标右键点击“类视图”中视图类节点，在弹出的快捷菜单中选择“添加-》函数”，弹出“添加函数”对话框，如图 7.1 所示。“函数名(U):”输入框中输入函数名加上参数列表，比如“DDALine(CDC* pDC, int x1, int y1, int x2, int y2, COLORREF)”。 “返回类型(Y):”输入函数的返回值，无返回值输入“void”。



添加函数

函数名(U):

返回类型(Y):

int

访问(A):

public

.cpp 文件(F):

DemoView.cpp

注释(M):

其他选项:

☐ 内联(I)

☐ 静态(S)

☐ 虚拟(V)

☐ 纯(P)

确定 取消

图 7.1 添加函数