# 电子信息工程学院

实验报告书

|  |  |
| --- | --- |
| 课程名称： | VC++ |
| 题 目： | MFC应用程序-菜单工具栏与状态栏案例 |
| 实验类别： | 实验操作类 |
| 班 级： | 22级计算机科学与技术专升本班 |
| 学 号： | 2022020312 |
| 姓 名： | 魏志杰 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 评语： | | | | | | |
| 实验态度： | 认真 | （ ） | 一般 | （ ） | 差 | （ ） |
| 实验结果： | 正确 | （ ） | 部分正确 | （ ） | 错 | （ ） |
| 实验理论： | 熟悉 | （ ） | 了解 | （ ） | 不懂 | （ ） |
| 操作技能： | 掌握 | （ ） | 一般 | （ ） | 差 | （ ） |
| 实验报告： | 规范 | （ ） | 一般 | （ ） | 差 | （ ） |
| 成绩： |  | | 指导教师： |  | | |
| 批阅时间： |  | | |

**实验五 MFC应用程序-菜单工具栏与状态栏案例**

**1. 实验简介**

编写一个能用不同的颜色及线宽绘制正弦与余弦曲线的应用程

序。程序运行后，首先在视图窗口中显示坐标轴，然后根据用户所选

择的曲线类型、颜色及线宽绘制曲线，并在状态栏中显示相关的提示

信息。

**2. 实验环境**

1）Windows系列操作系统。

2）VS2010编程环境

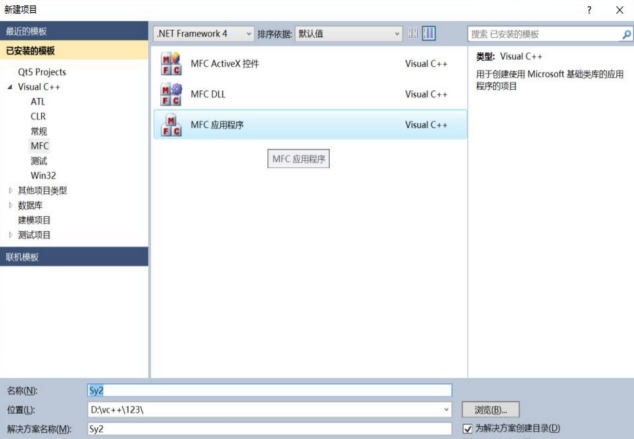
**3. 实验内容与步骤**

1. 创建工程

使用 MFC AppWizard[exe]生成一个单文档应用程序 Sy5。

文件→新建项目→MFC 应用程序，出现窗口按“下一步”，出现

下面窗口，选择单文档。

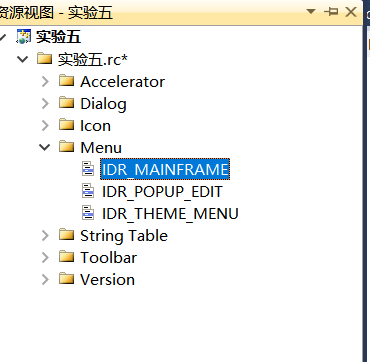




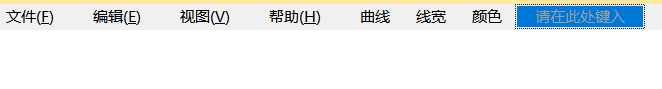
2. 添加菜单项

选择项目工作区中的资源视图，展开 Menu 文件夹，双击菜单资

源 IDR\_MAINFRAME，打开菜单编辑器

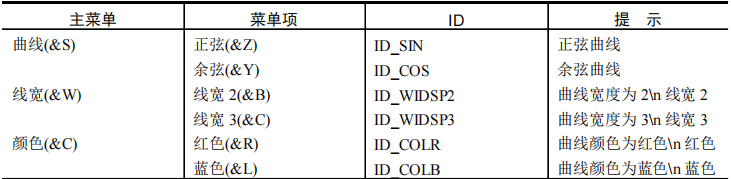


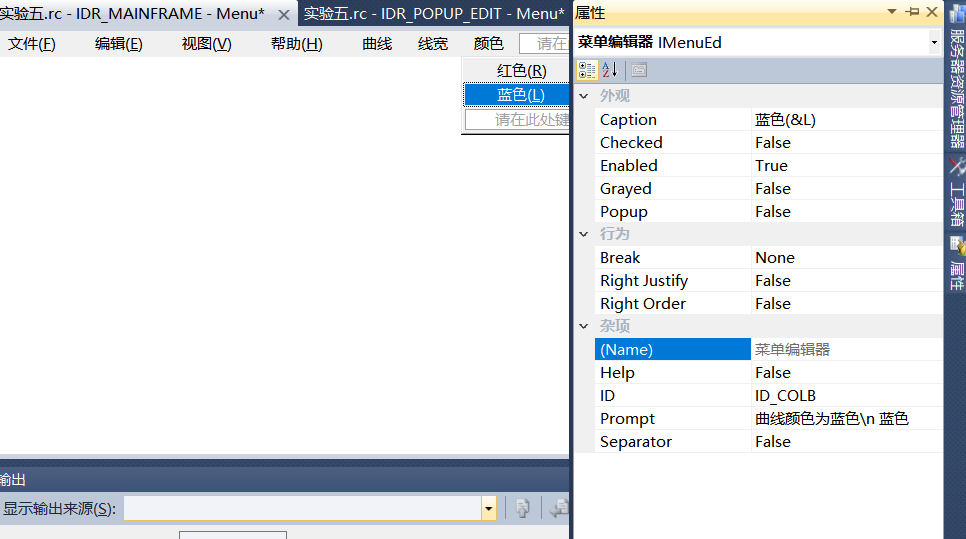
在主菜单【帮助】的后面添加 3 个菜单项【曲线】、【线宽】和【颜色】



并分别给它们添加子菜单，见下表。

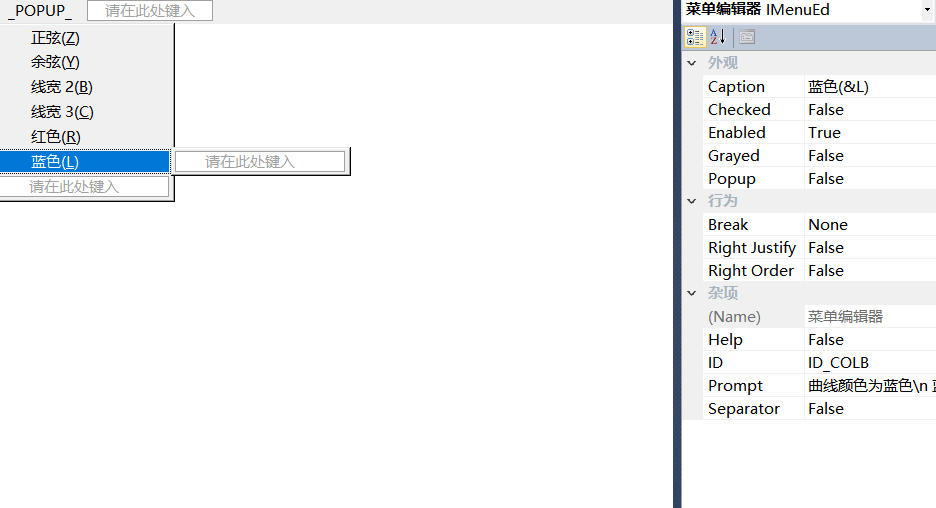
菜单项属性





3.创建快捷菜单

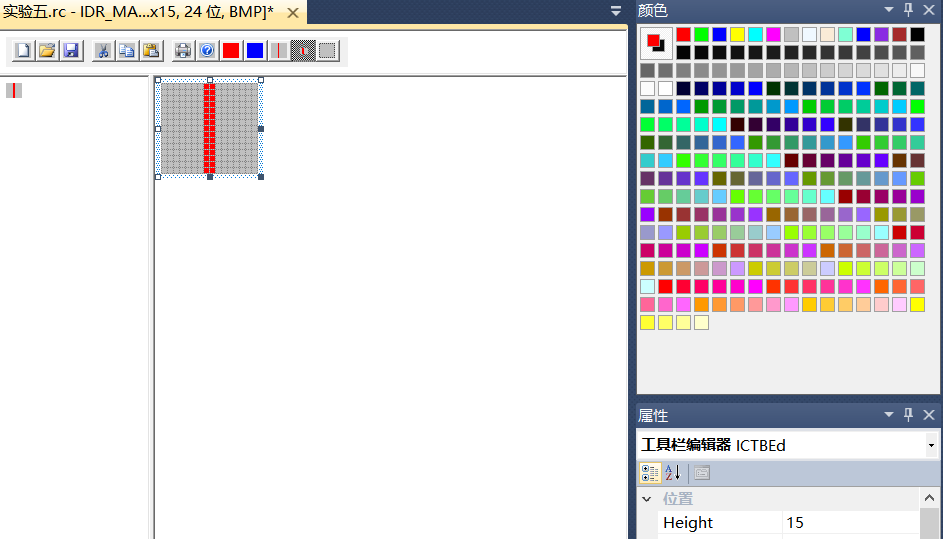
为应用程序添加一个快捷菜单，并将快捷菜单的菜单项设置为菜单栏中的相应项，如下图所示。



4.创建工具栏

为应用程序创建一个新的工具栏，并使工具栏上的各按钮与菜单

栏中的菜单项相对应，如下图所示。



ID 属性：ID\_WIDSP2。不知大家是否还记得，菜单

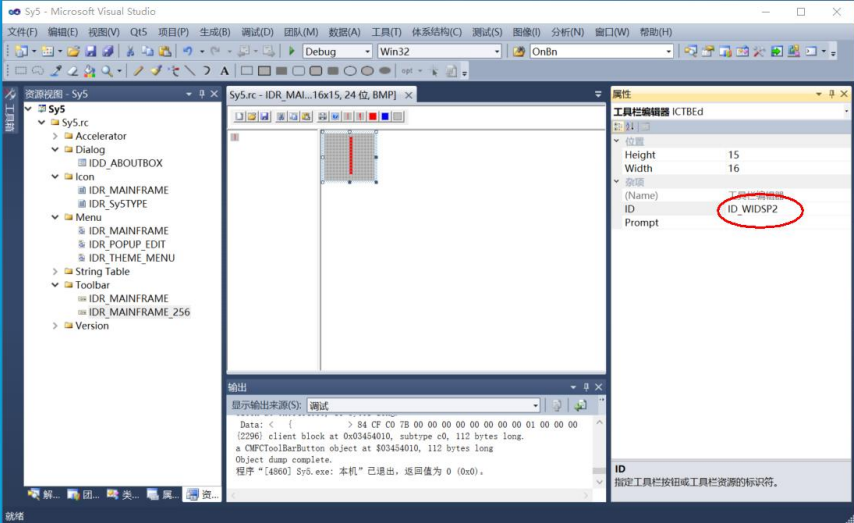
IDR\_MAINFRAME 的菜单项“线宽->线宽 2”的 ID 也是 ID\_WIDSP2，

两者 ID 相同，正是如此才使得工具栏第一个按钮与菜单项“线宽->

线宽 2”能实现相同的功能。所以大家一定要记住，如果想让工具栏

某个按钮与菜单栏某个菜单项点击后执行的操作相同，就要为两者设

置相同的 ID

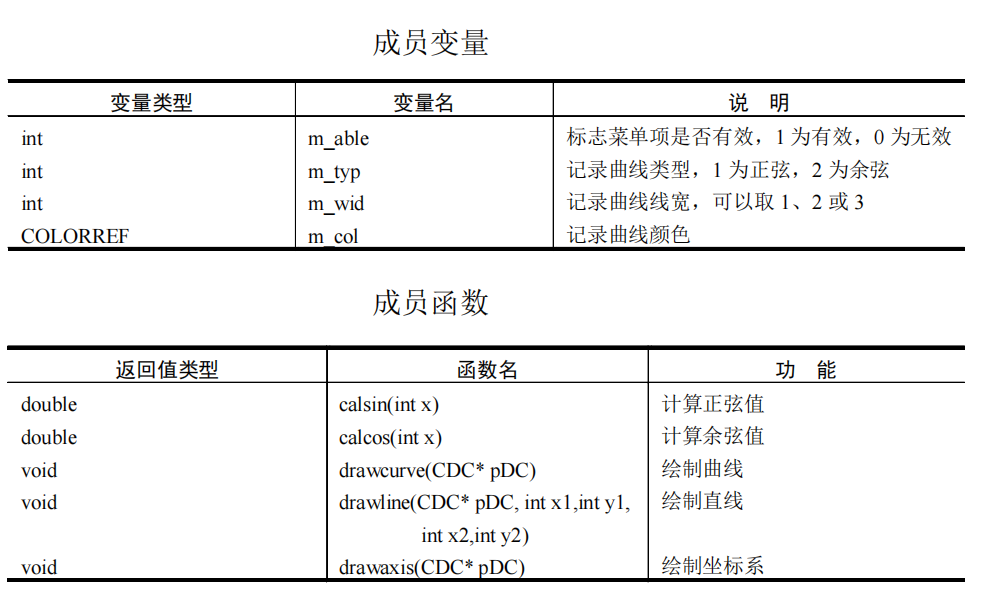


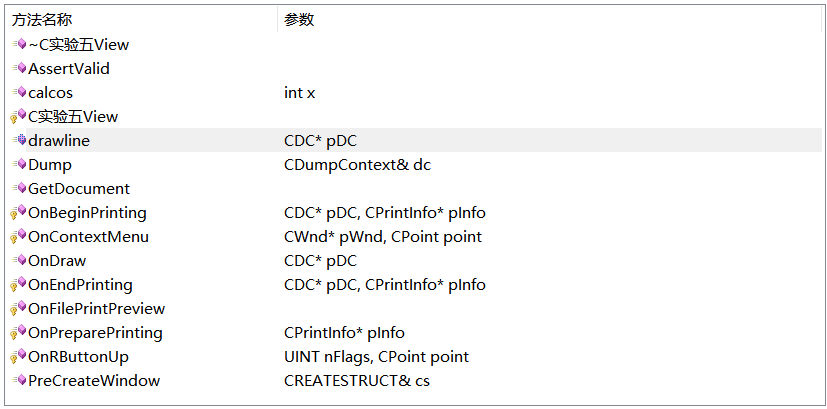
5.添加代码

(1)添加成员变量及成员函数

选择项目工作区中的类视图，右击 CSy5View，在弹出的快捷菜

单中分别选择Add Member Variable和Add Member Function添加成员变量及成员函数，见下表。





在构造函数中初始化成员变量：

CSy5View::CSy5View ()

{

m\_typ=0;

//初始时窗口中不显示曲线

m\_col=RGB(0,0,0); //颜色初始为黑色

m\_wid=1;

//线宽初始为 1

m\_able=0;

//初始时除“正弦”和“余弦”菜单项

外，新增的其他菜单项均为不可用

}

(2) 添加成员函数代码

打开 Sy5View.h 文件，在类的定义前面添加代码：const double PI=3.1416;

打开 Sy5View.cpp 文件，添加如下文件包含代码：

#include "math.h"

#include "MainFrm.h"

为各成员函数添加代码：

double CSy5View::calsin(int x)

{

double y;

y=sin(x\*PI/180); //计算正弦

return y;

}

double CSy5View::calcos(int x)

{

double y;

y=cos(x\*PI/180); //计算余弦

return y;

}

void CSy5View::drawline(CDC\* pDC,int x1, int y1, int x2, int y2)

{pDC->MoveTo(x1,y1);

pDC->LineTo(x2,y2);

}

void CSy5View::drawaxis(CDC\* pDC)

{

CString str;

CFont myfont,\*oldfont;

myfont.CreatePointFont(80,\_T("Arial"),pDC); //定义坐标刻度字体

oldfont=pDC->SelectObject(&myfont);

drawline(pDC,50,10,50,290);

//绘制 Y 轴

drawline(pDC,45,150,780,150); //绘制 X 轴

for(int i=50;i<780;i=i+90)

{

drawline(pDC,i,145,i,150);

//绘制 X 轴上刻度线

str.Format(\_T("%d"),i-50);

if((i-50)!=0) pDC->TextOut(i,155,str);

//输出 X 轴上刻度值

}

for(int i=0;i<11;i++) //绘制 Y 轴上刻度线,输出刻度值

{

drawline(pDC,50,50+20\*i,55,50+20\*i);str.Format(\_T("%.2f"),1-0.2\*i);

if(i!=5) pDC->TextOut(15,45+20\*i,str);

else pDC->TextOut(15,45+20\*i,\_T("0"));

}

pDC->SelectObject(oldfont);

myfont.DeleteObject();

}

void CSy5View::drawcurve(CDC\* pDC)

{

drawaxis(pDC);

//调用成员函数绘制坐标系

CPoint point[750];

CPen mypen,\*oldpen;

mypen.CreatePen(PS\_SOLID,m\_wid,m\_col);

//创建实线画笔。

线宽及颜色由参数 m\_wid 和 m\_col 确定

oldpen=pDC->SelectObject(&mypen);

for(int i=0;i<722;i++)

{

point[i].x=i+50;

if(m\_typ==1)

//根据参数 m\_typ 的

值确定曲线类型

point[i].y=(int)(150-calsin(i)\*100);else if(m\_typ==2)

point[i].y=(int)(150-calcos(i)\*100);

}

for(i=0;i<721;i++)

drawline(pDC,point[i].x,

point[i].y,point[i+1].x,point[i+1].y);

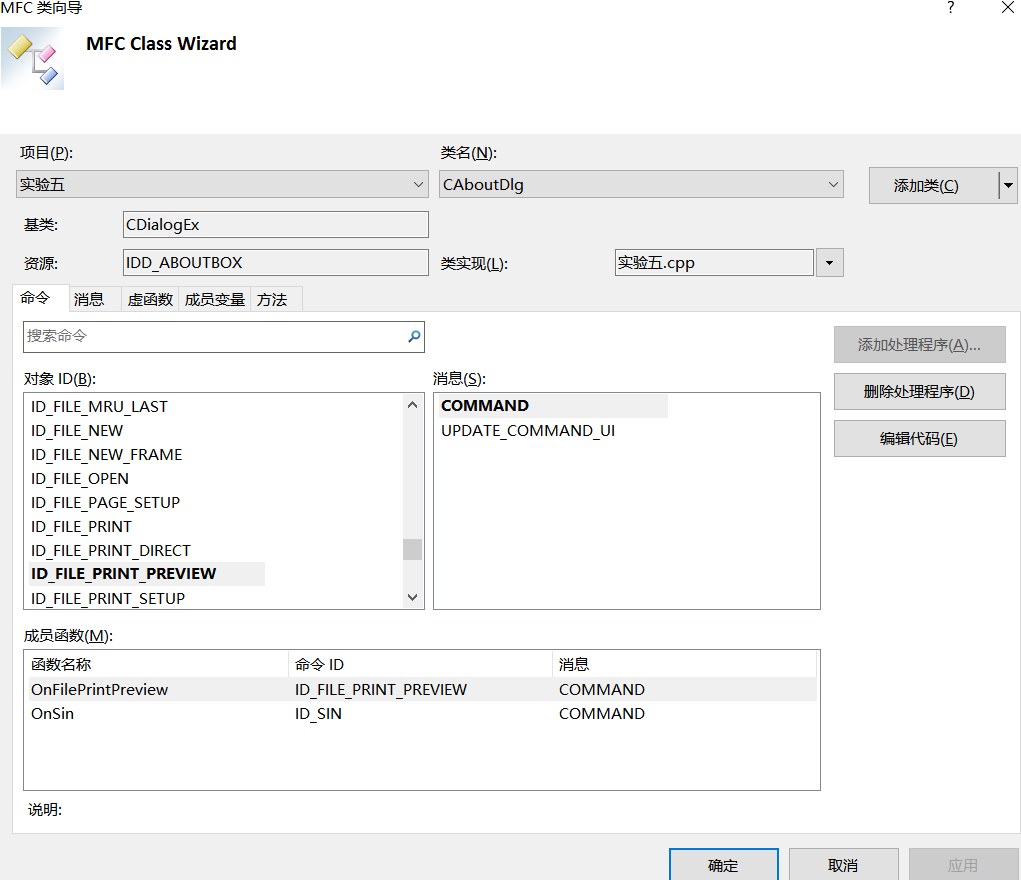
//调用函数绘制曲线

pDC->SelectObject(oldpen);

mypen.DeleteObject();

}

(3)建立消息映射

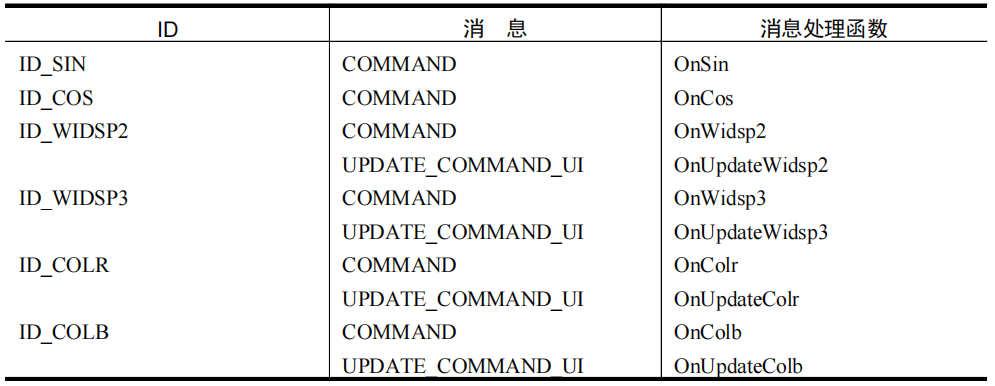


打开类向导对话框，分别为菜单项【正弦】、【余弦】、【线宽

2】 、【 线宽 3】 、【 红色 】和 【蓝 色】 添 加 COMMAND 及

UPDATE\_COMMAND\_UI 消息处理函数。

菜单项的消息处理函数



在消息处理函数中添加如下代码。

void CSy5View::OnSin()

{

m\_typ=1;

//选择正弦曲线

m\_able=1;

//不可用菜单项变为可用

Invalidate();

//更新视窗}

void CSy5View::OnCos()

{

m\_typ=2;

m\_able=1;

Invalidate();

}

void CSy5View::OnColr()

{

m\_col=RGB(255,0,0);

//选择红色

Invalidate();

}

void CSy5View::OnColb()

{

m\_col=RGB(0,0,255);

Invalidate();

}

void CSy5View::OnWidsp2()

{

m\_wid=2;

//设置线宽为 2

Invalidate();

}void CSy5View::OnWidsp3()

{

m\_wid=3;

Invalidate();

}

void CSy5View::OnUpdateColb(CCmdUI\* pCmdUI)

{

pCmdUI->Enable(m\_able); //“红色”菜单项变为可用

if(m\_col==RGB(0,0,255))

pCmdUI->SetCheck(true); //“红色”菜单项加标记

else

pCmdUI->SetCheck(false);

}

void CSy5View::OnUpdateColr(CCmdUI\* pCmdUI)

{

pCmdUI->Enable(m\_able);

if(m\_col==RGB(255,0,0))

pCmdUI->SetCheck(true);

else

pCmdUI->SetCheck(false);

}

void CSy5View::OnUpdateWidsp2(CCmdUI\* pCmdUI){

pCmdUI->Enable(m\_able);

if(m\_wid==2)

pCmdUI->SetCheck(true);

else

pCmdUI->SetCheck(false);

}

void CSy5View::OnUpdateWidsp3(CCmdUI\* pCmdUI)

{

pCmdUI->Enable(m\_able);

if(m\_wid==3)

pCmdUI->SetCheck(true);

else

pCmdUI->SetCheck(false);

}

(4) 修改视图类实现文件 CSy5View.cpp 的成员函数 OnDraw()。

void CCurveView::OnDraw(CDC\* pDC)

{

CCurveDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

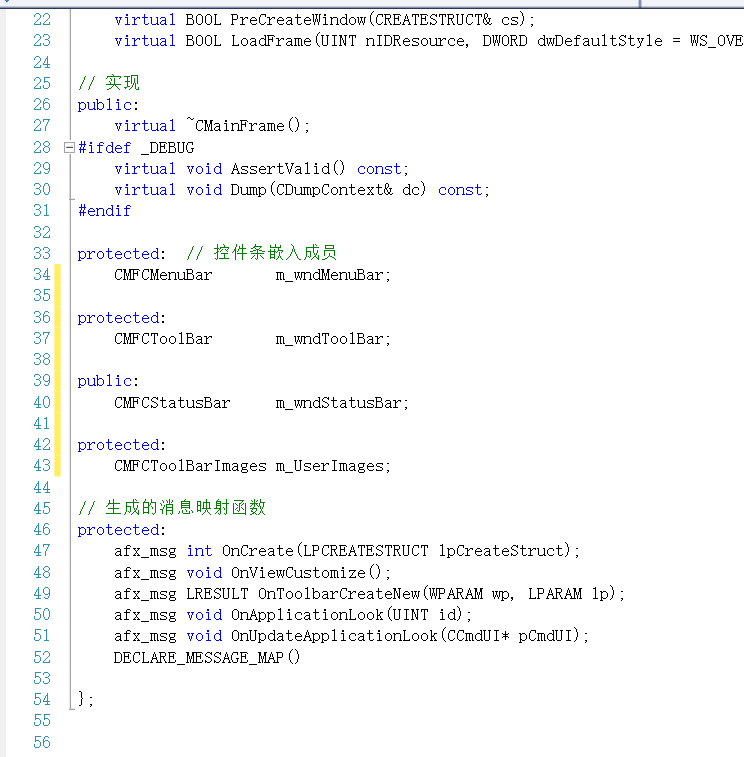
// TODO: 在此处为本机数据添加绘制代码

drawcurve(pDC);}

(5)添加状态栏操作代码

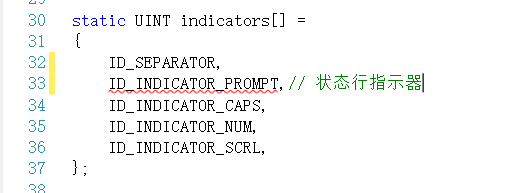
打开 MainFrm.h 文件，将系统默认的状态栏对象 m\_wndStatusBar

的访问类型改为 Public。

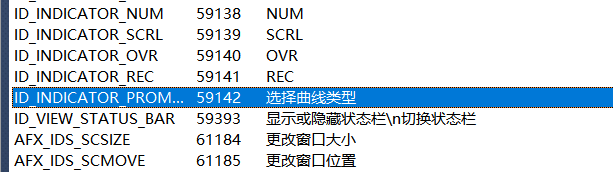


打开 MainFrm.cpp，在 indicators 数组（在状态栏静态数组中）中

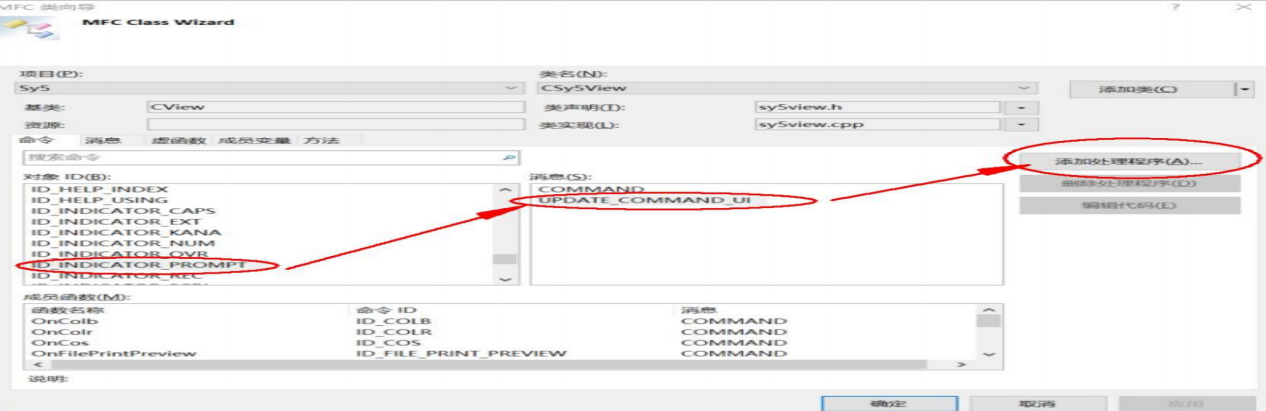
的第二个位置添加 ID\_INDICATOR\_PROMPT，



并在字符串编辑器中设置其属性，



首先，选择项目工作区的类视图，双击 CSy5View 类，右键—》类向导。添加状态栏更新事件



在Sy5View.cpp文件中给消息处理函数OnUpdatePrompt( )添加代

码。

void CSy5View::OnUpdateIndicatorPrompt(CCmdUI \*pCmdUI)

{

// TODO: 在此添加命令更新用户界面处理程序代码

CMainFrame \*pMainFrame=(CMainFrame\*)AfxGetMainWnd();

CMFCStatusBar \*pStatusBar=&pMainFrame->m\_wndStatusBar;

//获取状态栏指针

CString str,str1,str2,str3;

str2="宽 1,";

str3="黑色";

str1=(m\_typ==1)?"正弦,":"余弦,";

if(m\_wid==2)

//以下代码根据用户选择确定提示信息

str2="宽 2,";

else if(m\_wid==3)

str2="宽 3,";

if(m\_col==RGB(255,0,0))

str3="红色";

else if(m\_col==RGB(0,0,255))

str3="蓝色";

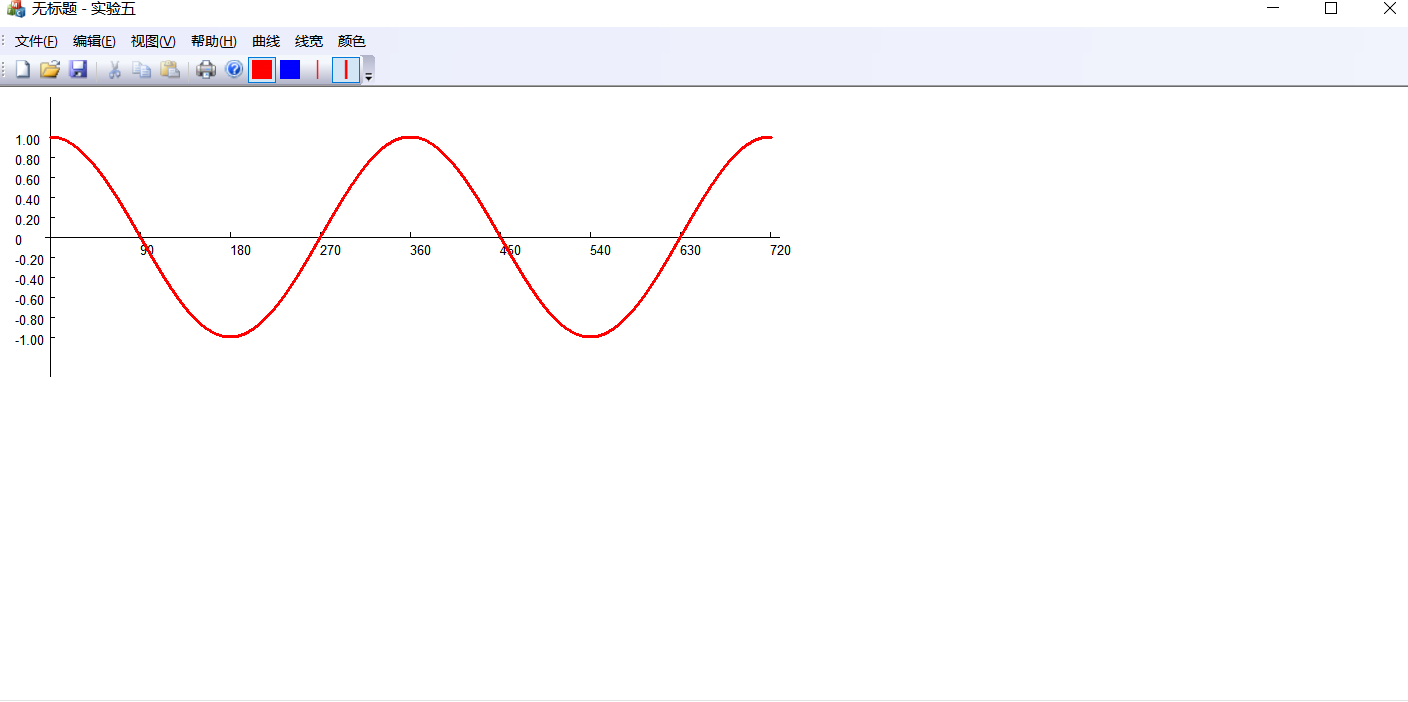
str=str1+str2+str3;

if(m\_able==0) str="选择曲线类型"; //初始时状态栏中提示信息

pStatusBar->SetPaneText(1,str);//显示提示信息

}

6.编译、链接并运行程序。



总结：

MFC是Microsoft Foundation Class的缩写，是一种用于Windows操作系统上GUI应用程序开发的框架。在MFC应用程序中，菜单、工具栏和状态栏可以为用户提供方便的界面交互体验。

工具栏是包含控件位图图像的控件条。 这些图像的作用类似于按键、复选框或单选按钮。 MFC 提供用于管理工具栏的 CToolbar 类。

如果你启用了它，MFC 工具栏的用户可以将它们停靠在窗口的边缘，或者使其“浮动”在应用程序窗口的任何位置。 MFC 不支持可自定义工具栏，如开发环境中的工具栏。

MFC 还支持工具提示：当你把鼠标悬停在工具栏按钮上时，会弹出描述该按钮用途的小型弹出窗口。 默认情况下，当用户按下工具栏按钮时，状态栏中会显示状态字符串（如果有）。 你可以激活“越过”状态栏更新以在鼠标悬停在按钮上而不按下按钮时显示状态字符串。