实验二

### 实验目的

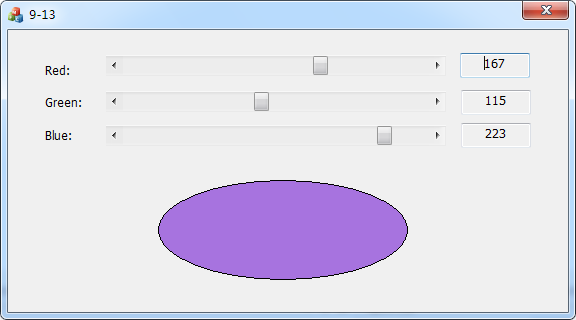
熟练MFC中各种控件的使用

## 实验内容

完成课本9-13、9-17、9-20的程序

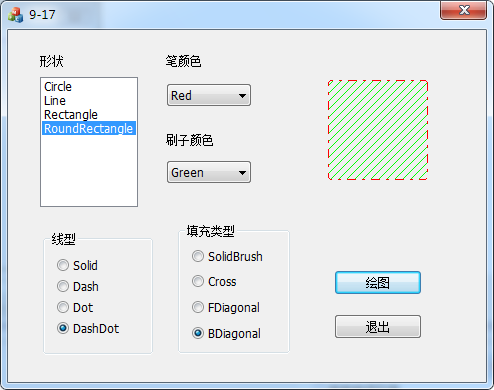
### 图例与描述

1. 9-13



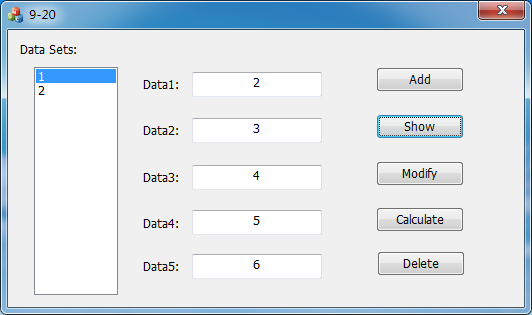
|  |  |  |
| --- | --- | --- |
| 控件ID | 动作 | 描述 |
| IDC\_SCROLLBAR1 | 拖动 | 改变滚动条值，并将改变写入到编辑框 |
| 单击 |
| IDC\_SCROLLBAR2 | 拖动 |
| 单击 |
| IDC\_SCROLLBAR3 | 拖动 |
| 单击 |
| IDC\_EDIT1 |  | 显示R值 |
| IDC\_EDIT2 | 显示G值 |
| IDC\_EDIT3 | 显示B值 |

1. 9-17



|  |  |  |
| --- | --- | --- |
| 控件ID | 动作 | 描述 |
| IDC\_LIST1 | 单击选择条目 | 单纯的改变列表控件中列表项的选定状态 |
| IDC\_COMBO1 | 下拉并选择 | 单纯的改变下拉菜单的控件状态 |
| IDC\_COMBO2 |
| IDC\_RADIO1 | 单击选择 | 单纯的改变radiobtn组的控件状态 |
| IDC\_RADIO2 |
| IDC\_RADIO3 |
| IDC\_RADIO4 |
| IDC\_RADIO5 | 单击选择 |
| IDC\_RADIO6 |
| IDC\_RADIO7 |
| IDC\_RADIO8 |
| IDC\_BUTTON1 | 单击 | 读取list、combobox、各个radiobutton控件的状态值，决定选择的画笔、画刷种类，并用它们在预设的矩形位置画出相应图案 |
| IDC\_BUTTON2 | 单击 | 释放各画笔、画刷申请的DC内存空间，并调用exit(0)函数退出程序 |

1. 9-20



|  |  |  |
| --- | --- | --- |
| 控件ID | 动作 | 描述 |
| IDC\_LIST1 | 单击选择条目 | 单纯的改变列表控件中列表项的选定状态 |
| IDC\_EDIT1 | 输入数字 | 显示输入的数字 |
| IDC\_EDIT2 |
| IDC\_EDIT3 |
| IDC\_EDIT4 |
| IDC\_EDIT5 |
| IDC\_BUTTON1 | 单击 | 验证五个输入框控件都不为空，出现为空则报错，若无为空则将输入的五个数据转为数值型数据存入array中，再将array存入list中，之后在列表控件中新增显示新的数据组合编号 |
| IDC\_BUTTON2 | 检测列表控件是否有列表项是选中状态，若无则报错，否则读取被选中条目，将被选中条目的windowtext作为数据组合编号在list中查询，找到对应数据组合后在各个Edit中显示 |
| IDC\_BUTTON3 | 验证五个输入框控件都不为空，出现为空则报错，若无为空则检测列表控件是否有列表项是选中状态，若无则报错，否则读取被选中条目，将被选中条目的windowtext作为数据组合编号在list中查询，找到对应数据组合后将对应数据组合的值更新为五个Edit中的值 |
| IDC\_BUTTON4 | 检测列表控件是否有列表项是选中状态，若无则报错，否则读取被选中条目，将被选中条目的windowtext作为数据组合编号在list中查询，找到对应数据组合后计算数据组合中的最小数和最大数，用msgbox将两个数及其下标显示出来 |
| IDC\_BUTTON5 | 检测列表控件是否有列表项是选中状态，若无则报错，否则读取被选中条目，将被选中条目的windowtext作为数据组合编号在list中查询，找到对应数据组合后将其从list中删除，并将列表控件中被选中条目删除 |

### 主要代码

1. 9-13

// 9-13Dlg.cpp: 实现文件

//

#include "stdafx.h"

#include "9-13.h"

#include "9-13Dlg.h"

#include "afxdialogex.h"

#define RGB\_MIN 0

#define RGB\_MAX 255

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

// CMy913Dlg 对话框

CMy913Dlg::CMy913Dlg(CWnd\* pParent /\*=nullptr\*/)

: CDialogEx(IDD\_MY913\_DIALOG, pParent)

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CMy913Dlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Control(pDX, IDC\_SCROLLBAR1, m\_RedScroll);

DDX\_Control(pDX, IDC\_SCROLLBAR2, m\_GreenScroll);

DDX\_Control(pDX, IDC\_SCROLLBAR3, m\_BlueScroll);

DDX\_Control(pDX, IDC\_EDIT1, m\_RedShow);

DDX\_Control(pDX, IDC\_EDIT2, m\_GreenShow);

DDX\_Control(pDX, IDC\_EDIT3, m\_BlueShow);

}

BEGIN\_MESSAGE\_MAP(CMy913Dlg, CDialogEx)

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

// ON\_NOTIFY(NM\_THEMECHANGED, IDC\_SCROLLBAR1, &CMy913Dlg::OnNMThemeChangedScrollbar1)

//ON\_NOTIFY(NM\_THEMECHANGED, IDC\_SCROLLBAR1, &CMy913Dlg::OnNMThemeChangedScrollbar1)

ON\_WM\_HSCROLL()

END\_MESSAGE\_MAP()

// CMy913Dlg 消息处理程序

BOOL CMy913Dlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

// 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

// 执行此操作

SetIcon(m\_hIcon, TRUE); // 设置大图标

SetIcon(m\_hIcon, FALSE); // 设置小图标

initScrollBars(); //初始化各个水平滚动条

//RGB值初始为0

m\_RedShow.SetWindowTextW(\_T("0"));

m\_GreenShow.SetWindowTextW(\_T("0"));

m\_BlueShow.SetWindowTextW(\_T("0"));

return TRUE; // 除非将焦点设置到控件，否则返回 TRUE

}

void CMy913Dlg::initScrollBars()

{

m\_RedScroll.SetScrollRange(RGB\_MIN, RGB\_MAX);

m\_GreenScroll.SetScrollRange(RGB\_MIN, RGB\_MAX);

m\_BlueScroll.SetScrollRange(RGB\_MIN, RGB\_MAX);

}

void CMy913Dlg::DrawOval()

{

CPaintDC dc(this);

CBrush brush(RGB(R, G, B));

dc.SelectObject(&brush);//选择画刷

dc.Ellipse(&OvalRect);//画椭圆

}

// 如果向对话框添加最小化按钮，则需要下面的代码

// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，

// 这将由框架自动完成。

void CMy913Dlg::OnPaint()

{

DrawOval(); //画椭圆

if (IsIconic())

{

CPaintDC dc(this); // 用于绘制的设备上下文

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作区矩形中居中

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialogEx::OnPaint();

}

}

//当用户拖动最小化窗口时系统调用此函数取得光标

//显示。

HCURSOR CMy913Dlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CMy913Dlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar\* pScrollBar)

{

UINT nowPos = pScrollBar->GetScrollPos();//获取当前滚动条轴位置

switch (nSBCode)

{

//拖动事件

case SB\_THUMBTRACK:

pScrollBar->SetScrollPos(nPos);

UpdateData(FALSE);

break;

//单击左箭头

case SB\_LINEUP:

if (nowPos > RGB\_MIN)

{

pScrollBar->SetScrollPos(nowPos - 1);

}

break;

//单击右箭头

case SB\_LINEDOWN:

if (nowPos < RGB\_MAX)

{

pScrollBar->SetScrollPos(nowPos + 1);

}

break;

//单击左空白

case SB\_PAGEUP:

if (nowPos > RGB\_MIN)

{

pScrollBar->SetScrollPos(nowPos - 1);

}

break;

//单击右空白

case SB\_PAGEDOWN:

if (nowPos < RGB\_MAX)

{

pScrollBar->SetScrollPos(nowPos + 1);

}

break;

default:

break;

}

//某RGB的字符串形式

CString str\_RGB;

//转换UINT2CString

str\_RGB.Format(\_T("%d"), nowPos);

//设置EditBox

//本次设置了红色

if (pScrollBar == &m\_RedScroll)

{

R = nowPos;

m\_RedShow.SetWindowTextW(str\_RGB);

}

//本次设置了绿色

else if (pScrollBar == &m\_GreenScroll)

{

G = nowPos;

m\_GreenShow.SetWindowTextW(str\_RGB);

}

//本次设置了蓝色

else if (pScrollBar == &m\_BlueScroll)

{

B = nowPos;

m\_BlueShow.SetWindowTextW(str\_RGB);

}

InvalidateRect(&OvalRect);//重绘椭圆区域防止整个界面闪烁

CDialogEx::OnHScroll(nSBCode, nPos, pScrollBar);

}

// 9-13Dlg.h: 头文件

//

#pragma once

// CMy913Dlg 对话框

class CMy913Dlg : public CDialogEx

{

// 构造

public:

CMy913Dlg(CWnd\* pParent = nullptr); // 标准构造函数

// 对话框数据

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_MY913\_DIALOG };

#endif

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持

// 实现

protected:

HICON m\_hIcon;

UINT R = 0, G = 0, B = 0;//RGB参数

RECT OvalRect = { 150, 150, 400, 250 };//椭圆绘图区域

// 生成的消息映射函数

virtual BOOL OnInitDialog();

afx\_msg void OnPaint();

afx\_msg HCURSOR OnQueryDragIcon();

//初始化各个水平滚动条

void initScrollBars();

//绘制填充的椭圆

void DrawOval();

DECLARE\_MESSAGE\_MAP()

public:

CScrollBar m\_RedScroll;

CScrollBar m\_GreenScroll;

CScrollBar m\_BlueScroll;

CEdit m\_RedShow;

CEdit m\_GreenShow;

CEdit m\_BlueShow;

afx\_msg void OnHScroll(UINT nSBCode, UINT nPos, CScrollBar\* pScrollBar);

};

1. 9-17

// 9-17Dlg.cpp: 实现文件

//

#include "stdafx.h"

#include "9-17.h"

#include "9-17Dlg.h"

#include "afxdialogex.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

// CMy917Dlg 对话框

CMy917Dlg::CMy917Dlg(CWnd\* pParent /\*=nullptr\*/)

: CDialogEx(IDD\_MY917\_DIALOG, pParent)

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CMy917Dlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Control(pDX, IDC\_LIST1, m\_List);

DDX\_Control(pDX, IDC\_COMBO1, m\_PenColor);

DDX\_Control(pDX, IDC\_COMBO2, m\_BrushColor);

DDX\_Control(pDX, IDC\_RADIO1, m\_LineType[0]);

DDX\_Control(pDX, IDC\_RADIO2, m\_LineType[1]);

DDX\_Control(pDX, IDC\_RADIO3, m\_LineType[2]);

DDX\_Control(pDX, IDC\_RADIO4, m\_LineType[3]);

DDX\_Control(pDX, IDC\_RADIO5, m\_FillType[0]);

DDX\_Control(pDX, IDC\_RADIO6, m\_FillType[1]);

DDX\_Control(pDX, IDC\_RADIO7, m\_FillType[2]);

DDX\_Control(pDX, IDC\_RADIO8, m\_FillType[3]);

}

BEGIN\_MESSAGE\_MAP(CMy917Dlg, CDialogEx)

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BUTTON1, &CMy917Dlg::OnBnClickedButton1)

ON\_BN\_CLICKED(IDC\_BUTTON2, &CMy917Dlg::OnBnClickedButton2)

END\_MESSAGE\_MAP()

// CMy917Dlg 消息处理程序

BOOL CMy917Dlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

// 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

// 执行此操作

SetIcon(m\_hIcon, TRUE); // 设置大图标

SetIcon(m\_hIcon, FALSE); // 设置小图标

//radio初始

m\_LineType[0].SetCheck(1);

m\_FillType[0].SetCheck(1);

//ListView填充

m\_List.AddString(\_T("Line"));

m\_List.AddString(\_T("Circle"));

m\_List.AddString(\_T("Rectangle"));

m\_List.AddString(\_T("RoundRectangle"));

//Combo\_PenColor填充

m\_PenColor.AddString(\_T("Red"));

m\_PenColor.AddString(\_T("Green"));

m\_PenColor.AddString(\_T("Blue"));

//Combo\_BrushColor填充

m\_BrushColor.AddString(\_T("Red"));

m\_BrushColor.AddString(\_T("Green"));

m\_BrushColor.AddString(\_T("Blue"));

//初始化画笔

pen[0][0].CreatePen(PS\_SOLID, 1, RGB(0, 0, 255));

pen[1][0].CreatePen(PS\_SOLID, 1, RGB(0, 255, 0));

pen[2][0].CreatePen(PS\_SOLID, 1, RGB(255, 0, 0));

pen[0][1].CreatePen(PS\_DASH, 1, RGB(0, 0, 255));

pen[1][1].CreatePen(PS\_DASH, 1, RGB(0, 255, 0));

pen[2][1].CreatePen(PS\_DASH, 1, RGB(255, 0, 0));

pen[0][2].CreatePen(PS\_DOT, 1, RGB(0, 0, 255));

pen[1][2].CreatePen(PS\_DOT, 1, RGB(0, 255, 0));

pen[2][2].CreatePen(PS\_DOT, 1, RGB(255, 0, 0));

pen[0][3].CreatePen(PS\_DASHDOT, 1, RGB(0, 0, 255));

pen[1][3].CreatePen(PS\_DASHDOT, 1, RGB(0, 255, 0));

pen[2][3].CreatePen(PS\_DASHDOT, 1, RGB(255, 0, 0));

//初始化画刷

brush[0][0].CreateSolidBrush(RGB(0, 0, 255));

brush[1][0].CreateSolidBrush(RGB(0, 255, 0));

brush[2][0].CreateSolidBrush(RGB(255, 0, 0));

brush[0][1].CreateHatchBrush(HS\_CROSS, RGB(0, 0, 255));

brush[1][1].CreateHatchBrush(HS\_CROSS, RGB(0, 255, 0));

brush[2][1].CreateHatchBrush(HS\_CROSS, RGB(255, 0, 0));

brush[0][2].CreateHatchBrush(HS\_FDIAGONAL, RGB(0, 0, 255));

brush[1][2].CreateHatchBrush(HS\_FDIAGONAL, RGB(0, 255, 0));

brush[2][2].CreateHatchBrush(HS\_FDIAGONAL, RGB(255, 0, 0));

brush[0][3].CreateHatchBrush(HS\_BDIAGONAL, RGB(0, 0, 255));

brush[1][3].CreateHatchBrush(HS\_BDIAGONAL, RGB(0, 255, 0));

brush[2][3].CreateHatchBrush(HS\_BDIAGONAL, RGB(255, 0, 0));

return TRUE; // 除非将焦点设置到控件，否则返回 TRUE

}

// 如果向对话框添加最小化按钮，则需要下面的代码

// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，

// 这将由框架自动完成。

void CMy917Dlg::OnPaint()

{

CPaintDC myDrawDC(this);

//选择画笔

myDrawDC.SelectObject(pen[m\_PenColor.GetCurSel()][m\_LineType[1].GetState() +

m\_LineType[2].GetState() \* 2 +

m\_LineType[3].GetState() \* 3]);

//选择画刷

myDrawDC.SelectObject(brush[m\_BrushColor.GetCurSel()][m\_FillType[1].GetState() +

m\_FillType[2].GetState() \* 2 +

m\_FillType[3].GetState() \* 3]);

switch (m\_List.GetCurSel())

{

//画圆

case 0:

myDrawDC.Ellipse(&RedrawRt);

break;

//画线

case 1:

myDrawDC.MoveTo(320, 150);

myDrawDC.LineTo(420, 50);

break;

//画矩形

case 2:

myDrawDC.Rectangle(&RedrawRt);

break;

//画圆角矩形

case 3:

myDrawDC.RoundRect(&RedrawRt, RRCT);

break;

default:

break;

}

if (IsIconic())

{

CPaintDC dc(this); // 用于绘制的设备上下文

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作区矩形中居中

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialogEx::OnPaint();

}

}

//当用户拖动最小化窗口时系统调用此函数取得光标

//显示。

HCURSOR CMy917Dlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CMy917Dlg::OnBnClickedButton1()

{

InvalidateRect(&RedrawRt);

}

void CMy917Dlg::OnBnClickedButton2()

{

//清除画笔画刷DC

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 4; j++)

{

pen[i][j].DeleteObject();

brush[i][j].DeleteObject();

}

}

exit(0);

}

// 9-17Dlg.h: 头文件

//

#pragma once

// CMy917Dlg 对话框

class CMy917Dlg : public CDialogEx

{

// 构造

public:

CMy917Dlg(CWnd\* pParent = nullptr); // 标准构造函数

// 对话框数据

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_MY917\_DIALOG };

#endif

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持

// 实现

protected:

HICON m\_hIcon;

RECT RedrawRt = { 320, 50, 420, 150 };//重绘区

CPen pen[3][4];//3颜色\*4线型

CBrush brush[3][4];//3颜色\*4填充类型

POINT RRCT = { 5, 5 };//圆角矩形参数

// 生成的消息映射函数

virtual BOOL OnInitDialog();

afx\_msg void OnPaint();

afx\_msg HCURSOR OnQueryDragIcon();

DECLARE\_MESSAGE\_MAP()

public:

CListBox m\_List;

CComboBox m\_PenColor;

CComboBox m\_BrushColor;

CButton m\_LineType[4];

CButton m\_FillType[4];

afx\_msg void OnBnClickedButton1();

afx\_msg void OnBnClickedButton2();

};

1. 9-20

// 9-20Dlg.cpp: 实现文件

//

#include "stdafx.h"

#include "9-20.h"

#include "9-20Dlg.h"

#include "afxdialogex.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

// CMy920Dlg 对话框

CMy920Dlg::CMy920Dlg(CWnd\* pParent /\*=nullptr\*/)

: CDialogEx(IDD\_MY920\_DIALOG, pParent)

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CMy920Dlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Control(pDX, IDC\_EDIT1, m\_Data[0]);

DDX\_Control(pDX, IDC\_EDIT2, m\_Data[1]);

DDX\_Control(pDX, IDC\_EDIT3, m\_Data[2]);

DDX\_Control(pDX, IDC\_EDIT4, m\_Data[3]);

DDX\_Control(pDX, IDC\_EDIT5, m\_Data[4]);

DDX\_Control(pDX, IDC\_LIST1, m\_List);

}

BEGIN\_MESSAGE\_MAP(CMy920Dlg, CDialogEx)

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

// ON\_BN\_CLICKED(IDOK, &CMy920Dlg::OnBnClickedOk)

ON\_BN\_CLICKED(IDC\_BUTTON1, &CMy920Dlg::OnBnClickedButton1)

ON\_BN\_CLICKED(IDC\_BUTTON2, &CMy920Dlg::OnBnClickedButton2)

ON\_BN\_CLICKED(IDC\_BUTTON4, &CMy920Dlg::OnBnClickedButton4)

ON\_BN\_CLICKED(IDC\_BUTTON5, &CMy920Dlg::OnBnClickedButton5)

ON\_BN\_CLICKED(IDC\_BUTTON3, &CMy920Dlg::OnBnClickedButton3)

END\_MESSAGE\_MAP()

// CMy920Dlg 消息处理程序

BOOL CMy920Dlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

// 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

// 执行此操作

SetIcon(m\_hIcon, TRUE); // 设置大图标

SetIcon(m\_hIcon, FALSE); // 设置小图标

// TODO: 在此添加额外的初始化代码

return TRUE; // 除非将焦点设置到控件，否则返回 TRUE

}

// 如果向对话框添加最小化按钮，则需要下面的代码

// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，

// 这将由框架自动完成。

void CMy920Dlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // 用于绘制的设备上下文

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作区矩形中居中

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialogEx::OnPaint();

}

}

//当用户拖动最小化窗口时系统调用此函数取得光标

//显示。

HCURSOR CMy920Dlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

//设置临时数据

BOOL CMy920Dlg::SetTempDatas()

{

CString InputStrs[5] = { 0 };//输入的5个数据字符串形式

//检查空输入

for (int i = 0; i < 5; i++)

{

m\_Data[i].GetWindowTextW(InputStrs[i]);

if (InputStrs[i] == "")

{

CString ErrTip;//错误提示信息

ErrTip.Format(\_T("Data %d is blank!"), i);

MessageBox(ErrTip, \_T("No Blank Data"));

return false;

}

}

//没有空输入

//数据临时保存

for (int i = 0; i < 5; i++)

{

tDatas[i + 1] = \_wtol(InputStrs[i]);

}

return true;

}

//单击新增按钮

void CMy920Dlg::OnBnClickedButton1()

{

//设置临时数据数组成功

if (SetTempDatas() == TRUE)

AddDatas();//添加数据到list

return;

}

//添加数据到list

void CMy920Dlg::AddDatas()

{

//第一个被插入的数据组合

if (DataSets.empty() == true)

{

tDatas[0] = 0;

}

//不是第一个被插入的数据组合

else

{

tDatas[0] = DataSets.back()[0] + 1;

}

DataSets.push\_back(tDatas);//插入list

CString str\_Pos;//字符串形式的序号

str\_Pos.Format(\_T("%d"), tDatas[0]);

m\_List.AddString(str\_Pos);//添加序号到列表

return;

}

//根据选定的列表框项目设置控件数据到临时数组

BOOL CMy920Dlg::SetTempDatasByListView()

{

LONG index;//列表框中选定的下标

CString str\_Pos;//字符串形式序号

index = m\_List.GetCurSel();//获取列表框选定下标

//没有选定列表框的项目

if (index == CB\_ERR)

{

MessageBox(\_T("You haven't selected one item!"), \_T("Error"));

return false;

}

m\_List.GetText(index, str\_Pos);//获取选定的数据组合序号

for (list<array<LONG, 6>>::iterator i = DataSets.begin(); i != DataSets.end(); i++)

{

//序号相等，临时数组赋值并跳出

if (\_wtol(str\_Pos) == (\*i)[0])

{

tDatas = \*i;

break;

}

}

return true;

}

//单击显示按钮

void CMy920Dlg::OnBnClickedButton2()

{

if (SetTempDatasByListView() == TRUE)

{

//EditBox中显示信息

CString OutStrs[5] = { 0 };//待输出的5个数据字符串形式

for (int i = 1; i <= 5; i++)

{

OutStrs[i - 1].Format(\_T("%ld"), tDatas[i]);//LtoA

m\_Data[i - 1].SetWindowTextW(OutStrs[i - 1]);//显示

}

}

return;

}

//单击计算按钮

void CMy920Dlg::OnBnClickedButton4()

{

if (SetTempDatasByListView() == TRUE)

{

LONG min = LONG\_MAX, max = LONG\_MIN;

INT pos\_min = 0, pos\_max = 0;

for (int i = 1; i <= 5; i++)

{

if (min > tDatas[i])

{

pos\_min = i;

min = tDatas[i];

}

if (max < tDatas[i])

{

pos\_max = i;

max = tDatas[i];

}

}

CString ShowTip;//用于显示计算结果

ShowTip.Format(\_T("Max[ %d ] = %ld\nMin[ %d ] = %ld"), pos\_min, min, pos\_max, max);

MessageBox(ShowTip, \_T("Result"));

}

return;

}

//单击删除按钮

void CMy920Dlg::OnBnClickedButton5()

{

if (SetTempDatasByListView() == TRUE)

{

for (list<array<LONG, 6>>::iterator i = DataSets.begin(); i != DataSets.end(); i++)

{

//序号相等，删除数据组合并跳出

if (tDatas[0] == (\*i)[0])

{

DataSets.erase(i);

m\_List.DeleteString(m\_List.GetCurSel());

break;

}

}

}

}

//单击修改按钮

void CMy920Dlg::OnBnClickedButton3()

{

//设置临时数据数组

if (SetTempDatas() == TRUE)

{

for (list<array<LONG, 6>>::iterator i = DataSets.begin(); i != DataSets.end(); i++)

{

//序号相等，赋值数据组合并跳出

if (tDatas[0] == (\*i)[0])

{

\*i = tDatas;

break;

}

}

MessageBox(\_T("Modified!"), \_T("Tip"));

}

return;

}

// 9-20Dlg.h: 头文件

//

#pragma once

#include<list>

#include<array>

using namespace std;

// CMy920Dlg 对话框

class CMy920Dlg : public CDialogEx

{

// 构造

public:

CMy920Dlg(CWnd\* pParent = nullptr); // 标准构造函数

// 对话框数据

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_MY920\_DIALOG };

#endif

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持

// 实现

protected:

HICON m\_hIcon;

list<array<LONG, 6>> DataSets;//数据组合集（第一位为序号）

array<LONG, 6> tDatas;//临时保存某一数据集的5个数据（第一位为序号）

// 生成的消息映射函数

virtual BOOL OnInitDialog();

afx\_msg void OnPaint();

afx\_msg HCURSOR OnQueryDragIcon();

//添加数据到list

void AddDatas();

//设置控件数据到临时数组

BOOL SetTempDatas();

//根据选定的列表框项目设置控件数据到临时数组

BOOL SetTempDatasByListView();

DECLARE\_MESSAGE\_MAP()

public:

afx\_msg void OnBnClickedButton1();

CEdit m\_Data[5];//输入框控件数组

CListBox m\_List;

afx\_msg void OnBnClickedButton2();

afx\_msg void OnBnClickedButton4();

afx\_msg void OnBnClickedButton5();

afx\_msg void OnBnClickedButton3();

};