

学生学号	0121610870910	实验课成绩	
------	---------------	-------	--

武汉理工大学

学 生 实 验 报 告 书

课 程 名 称 可视化编程

开 课 学 院 计算机科学与技术学院

指 导 老 师 胡伦

学 生 姓 名 冯钢果

专 业 班 级 软件 1604 班

2017 — 2018 学年 第 二 学期

实验课程名称： 可视化编程

实验项目名称	基本窗口创建、基于 SDK 编程			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	(无)			实验日期	2018 年 5 月 10 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

实验一内容：基本窗口创建

- 1.1 熟悉 VC++集成开发环境，创建一个 VC 工程，熟悉工程的工作空间；
- 1.2 创建 cpp 文件，编写基本 Windows 应用程序的两个组成部分：WinMain 函数以及窗口函数；
- 1.3 完成 WinMain 函数中的函数说明、窗口类的定义、窗口类的注册、窗口的创建、窗口的显示以及消息循环；
- 1.4 完成窗口函数中的基本消息处理过程。
- 1.5 在例子的基础上，使用定时器，每隔 1 秒，交替的用红色、蓝色和绿色来填充整个窗口的用户区。

实验二内容：基于 SDK 编程（一）

- 2.1 创建一个图形绘制的 VC 工程，创建 cpp 文件；
- 2.2 编写 WinMain 函数进行本工程窗口定义与设计等任务，并完成窗口函数中的基本消息处理过程；
- 2.3 获取设备环境，设置映射模式，创建画笔，编写代码绘制风车叶片，并让风车旋转五、实验过程及结果描述。
- 2.4 在此基础上，添加如下功能：
 - (1) 在窗口左上角显示当前位置，即 nNum 的值；
 - (2) 添加菜单资源（主菜单包括控制和退出，其中控制的子菜单是开始和停止），按下“开始”时风车开始转动，按下“停止”时风车停止转动。

实验三内容：基于 SDK 编程（二）

- 3.1 创建一个键盘操作的 VC 工程，创建 cpp 文件；
- 3.2 编写 WinMain 函数进行本工程窗口定义与设计等任务，并完成窗口函数中的基本消息处理过程；
- 3.3 按照要求对“ctrl”键、“shift”键、下左或右箭头、Home 及 End 键、PageUp 及 PageDown 键编写相应的消息响应代码，以实现单击这些按钮时，分别可以完成不同图形的绘制以及移动等操作。
- 3.4 在此基础上，添加如下功能：
 - (1) 按下“shift”键时，进行矩形绘制；
 - (2) 按下左或右箭头，矩形或椭圆高度或长度加 10；
 - (3) 按下 PageUp 或 PageDown 时，矩形或椭圆向上或向下移动。

二、实验基本原理与设计

实验过程步骤如下：

实验步骤：基本窗口创建

- 1.1 在 VC++ 6.0 环境下，选择“新建”，再选择“新建”下的“工程”中的“Win32 Application”，选择要创建的工程所在文件夹，输入要创建的工程文件名；
- 1.2 创建成功后，继续选择“新建”下的“文件”中的“中的“C++ Source File”，输入本工程的 cpp 文件名；
- 1.3 在本工程的 cpp 文件中输入附录中的代码。

实验二步骤：基于 SDK 编程（一）

- 2.1 依照实验一步骤创建一个基于 SDK 的工程及相应其它文件；
- 2.2 在本工程的 cpp 文件中输入附录中的代码。

实验三步骤：基于 SDK 编程（二）

- 3.1 依照实验一步骤创建一个基于 SDK 的工程及相应其它文件；
- 3.2 在本工程的 cpp 文件中输入附录中代码。

三、主要仪器设备及耗材

1. 一台 windows 7 的 PC 机；
2. 软件 Microsoft Visual Studio 2010；
3. 草稿纸和笔；
4. 参考资料：Visual C++面向对象与可视化程序设计。

第二部分：实验调试与结果分析（可加页）

一、调试过程

实验一调试方法：

- (1) 按照附录中代码输入后，然后先编译后，再点击“开始执行”，发现执行的结果不能先显示红色和绿色；
- (2) 结局方法：可以更改源程序，设置全局变量 `int colorFlag`，但添加 `case WM_TIMER` 事件，然后每捕捉一次，`colorFlag+1` 对 3 求余即可解决问题。

实验二调试过程：

- (1) 按照实验要求中代码输入后，然后先编译后，再点击“开始执行”，发现风车正常运行，但不能达到额外要求；
- (2) 发现风车不能停止，先添加菜单项实现功能，然后添加代码使用 `KillTimer(hwnd, 9999)` 风车停止，`SetTimer(hwnd, 9999, 100, NULL)` 风车开始运行。

实验三调试过程：

- (1) 按照实验要求中代码输入后，然后先编译后，再点击“开始执行”，发现要实现矩形的操作需要添加 `shift` 事件捕捉；
- (2) 可以使用 `int Circle` 和 `int s` 分别标记图形为圆形还是矩形，在进行移动时然后判断图形类型，使用 `Ellipse(hdc, rect.left, rect.top, rect.right, rect.bottom)` 画圆，使用 `Rectangle(hdc, rect.left, rect.top, rect.right, rect.bottom)` 画矩形。

二、实验结果及分析

实验一结果：



图 1.1



图 1.2

实验二结果:

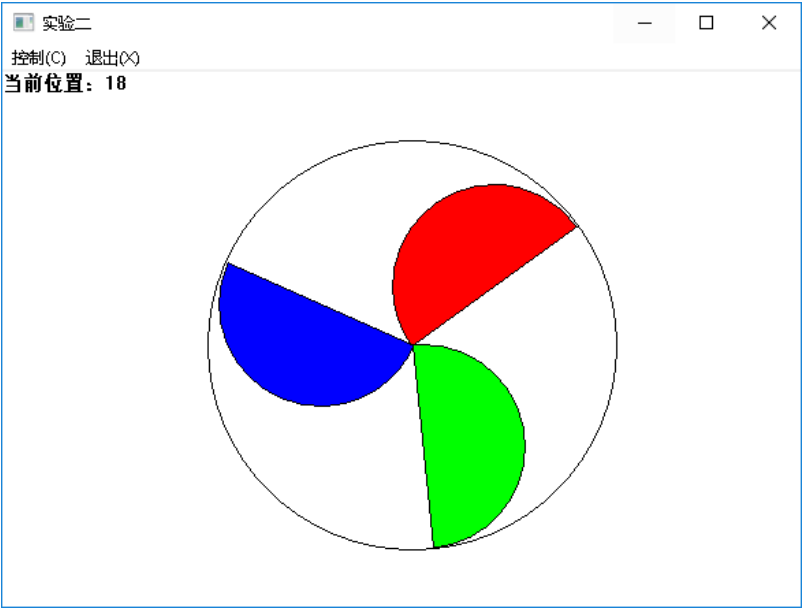


图 2.1

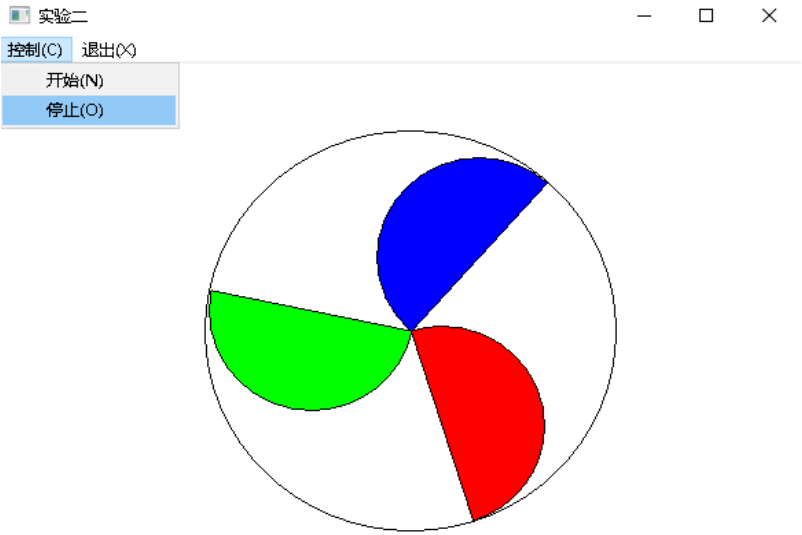


图 2.2

实验三结果:

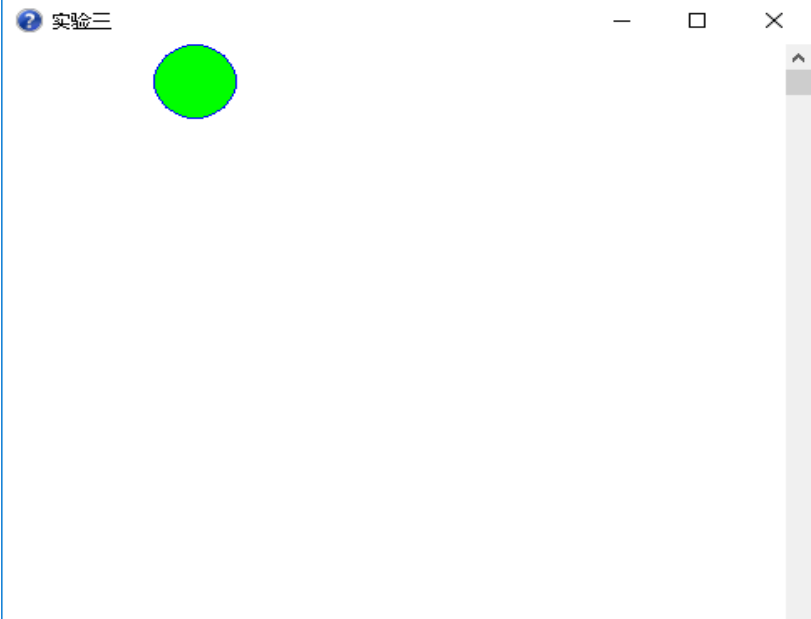


图 3.1



图 3.2

结果分析：

图 1.1 为实验一刚开始运行时的红绿蓝三种颜色变换时的结果，符合题目要求；

图 1.2 为实验一点击关闭按钮时的运行截图，如果选择“是”则程序退出，选择“否”程序继续运行；

图 2.1 为实验二风车旋转是的状态运行截图风车正常运行；

图 2.2 为实验二点击“停止”选项时的截图，转动的风车停止旋转；

图 3.1 为实验三按下 control 键后显示圆形，然后按下 home 和 end 键实现左右移动；

图 3.2 为实验三按下 shift 键后显示的矩形，然后按下 left 或 right 键实现矩形扩展，然后按下 PgUp 或 PgDn 键实现上下移动。

三、实验小结、建议及体会

通过本次实验，我掌握了 Windows 编程基础知识和 Windows 应用程序的组成和基本结构，熟悉了 VC++ 集成开发环境，学会获取设备环境的方法以及映射模式的设置、绘图工具的使用和绘图函数的应用和设备驱动程序翻译的常用虚拟码。

但学到的不仅仅是一些理论知识，更是开发的经验，通过实验的一次次出错，我深刻的意识到自己的动手能力远远不足，我可以通过实验弥补自己开发经验的不足，但要真正成为一个合格的程序员，这些工作量和知识量是远远不够的，因此课下我也在不断地努力的充电，不断地学习更多的编程知识和经验，有时与一些“大牛”们交谈，还在实践中不断成长。

第三部分：附录（实验代码）

//实验一代码:

//Main.cpp(主函数入口)

#include <windows.h>

int colorFlag = 0;

LRESULT CALLBACK WinSunProc(

 HWND hwnd,

 UINT uMsg,

 WPARAM wParam,

 LPARAM lParam

);

int WINAPI WinMain(

 HINSTANCE hInstance,

 HINSTANCE hPrevInstance,

 LPSTR lpCmdLine,

 int nCmdShow

)

{

 WNDCLASS wndcls;

 wndcls.cbClsExtra=0;

 wndcls.cbWndExtra=0;

 HBRUSH hB;

 hB=CreateSolidBrush(RGB(50, 20, 200));

 wndcls.hbrBackground=hB;

 wndcls.hCursor=LoadCursor(NULL, IDC_CROSS);

 wndcls.hIcon=LoadIcon(NULL, IDI_ERROR);

 wndcls.hInstance=hInstance;

 wndcls.lpfnWndProc=WinSunProc;

 wndcls.lpszClassName = "WHUT";

 wndcls.lpszMenuName=NULL;

 wndcls.style = CS_HREDRAW | CS_VREDRAW;

 RegisterClass(&wndcls);

 HWND hwnd;

 hwnd=CreateWindow("WHUT"," 武 汉 理 工 大 学 计 算 机 学 院", WS_OVERLAPPEDWINDOW, 0, 0, 600, 400, NULL, NULL, hInstance, NULL);

 SetTimer(hwnd, 1000, 1000, NULL);

 ShowWindow(hwnd, SW_SHOWNORMAL);

 UpdateWindow(hwnd);

 MSG msg;

 while(GetMessage(&msg, NULL, 0, 0))

 {

 TranslateMessage(&msg);

 DispatchMessage(&msg);

 }

 return 0;

```

}

LRESULT CALLBACK WinSunProc(
    HWND hwnd,
    UINT uMsg,
    WPARAM wParam,
    LPARAM lParam
)
{
    HDC hdc;
    HBRUSH hb1, hb2, hb3;
    PAINTSTRUCT ps;
    switch(uMsg)
    {
    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        SetMapMode(hdc, MM_TEXT);
        hb1 = CreateSolidBrush( RGB(255, 0, 0) );
        hb2 = CreateSolidBrush( RGB(0, 255, 0) );
        hb3 = CreateSolidBrush( RGB(0, 0, 255) );
        if(colorFlag == 0)
        {
            SelectObject(hdc, hb1);
            Rectangle(hdc, 0, 0, 800, 800);
        } else if(colorFlag == 1)
        {
            SelectObject(hdc, hb2);
            Rectangle(hdc, 0, 0, 800, 800);
        } else if(colorFlag == 2)
        {
            SelectObject(hdc, hb3);
            Rectangle(hdc, 0, 0, 800, 800);
        }
        EndPaint(hwnd, &ps);
        DeleteObject(hb1);
        DeleteObject(hb2);
        DeleteObject(hb3);
        break;
    case WM_TIMER:
        if(wParam == 1000)
        {
            colorFlag = (colorFlag + 1)%3;
            InvalidateRect(hwnd, NULL, TRUE);
        }
        break;
    case WM_CLOSE:
        if(IDYES == MessageBox(hwnd, "是否真的结束?", "whut", MB_YESNO))

```



```

        {
            DestroyWindow(hwnd);
        }
        break;
    case WM_DESTROY:
        KillTimer(hwnd, 1000);
        PostQuitMessage(0);
        break;
    default:
        return DefWindowProc(hwnd, uMsg, wParam, lParam);
    }
    return 0;
}

```

//实验二代码:

```

//Main.cpp
#include <Windows.h>
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <afx.h>
#include "resource.h"
#define pi 3.1415926
LRESULT CALLBACK WinProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam );
int nNum=0, nMaxNum=20;

int WINAPI WinMain(
    HINSTANCE hInstance,
    HINSTANCE hPrevInstace,
    LPSTR lpCmdLine,
    int nCmdShow
)
{
    char lpszClassName[]="窗口";
    char lpszTitle[]="实验二";
    WNDCLASS wndcls;
    wndcls.cbWndExtra=0;
    wndcls.cbClsExtra=0;
    wndcls.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
    wndcls.hCursor=LoadCursor(NULL, IDC_ARROW);
    wndcls.hIcon=LoadIcon(NULL, IDI_APPLICATION);
    wndcls.hInstance=hInstance;
    wndcls.lpfnWndProc=WinProc;
    wndcls.lpszClassName=lpszClassName;
    wndcls.lpszMenuName = "MYMENU";
    wndcls.style=CS_HREDRAW | CS_VREDRAW;
    RegisterClass(& wndcls);
}

```

```

    HWND hwnd;
    HMENU hmenu;
    hmenu = LoadMenu(hInstance, "MYMENU");
    hwnd=CreateWindow(lpszClassName, lpszTitle, WS_OVERLAPPEDWINDOW, CW_USEDEFAULT
, 0, 600, 450, NULL, hmenu, hInstance, NULL);
    ShowWindow(hwnd , SW_SHOWNORMAL);
    UpdateWindow(hwnd);
    MSG msg;
    while(GetMessage(&msg , NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

```

```

LRESULT CALLBACK WinProc(
    HWND hwnd,
    UINT uMsg,
    WPARAM wParam,
    LPARAM lParam
)
{
    HBRUSH hBrush;
    HPEN hPen;
    HMENU hmenu;
    CString temp;
    LPCTSTR lpsz;
    char lpszTitle[]="My_windows";
    int x,y;
    double fAngle;
    switch(uMsg)
    {
        case WM_COMMAND:
            switch(LOWORD(wParam))
            {
                case IDM_BEGIN:
                    SetTimer(hwnd, 9999, 100, NULL);
                    break;
                case IDM_STOP:
                    KillTimer(hwnd, 9999);
                    break;
                case IDM_EXIT:
                    SendMessage(hwnd, WM_DESTROY, 0, 0);
                    break;
            }
            break;
    }
}

```

```

case WM_CREATE:
    SetTimer(hwnd, 9999, 100, NULL);
    break;
case WM_PAINT:
    HDC hDC;
    HFONT hf;
    PAINTSTRUCT ps;
    hDC=BeginPaint(hwnd, &ps);
    SelectObject(hDC, hf);
    TEXTMETRIC tm;
    GetTextMetrics(hDC, &tm);
    temp.Format(_T("%d"), nNum);
    temp = "当前位置: " + temp;
    lpsz = LPCTSTR(temp);
    SetTextColor(hDC, RGB(0, 0, 0));
    TextOut(hDC, 0, 0, lpsz, _tcslen(lpsz));
    SetMapMode(hDC, MM_ANISOTROPIC);
    SetWindowExtEx(hDC, 400, 300, NULL);
    SetViewportExtEx(hDC, 600, 450, NULL);
    SetViewportOrgEx(hDC, 300, 200, NULL);
    hPen=(HPEN)GetStockObject(RGB(10, 100, 20));
    SelectObject(hDC, hPen);
    Ellipse(hDC, -100, -100, 100, 100);
    hBrush=CreateSolidBrush(RGB(255, 0, 0));
    SelectObject(hDC, hBrush);
    fAngle=2*pi/nMaxNum*nNum;
    x=(int)(50*cos(fAngle));
    y=(int)(50*sin(fAngle));
    Pie(hDC, x-50, y-50, x+50, y+50,

        (int)(x+50*cos(fAngle)), (int)(y+50*sin(fAngle)), (int)(x+50*cos(fAngle+pi)),
        (int)(y+50*sin(fAngle+pi)) );
    hBrush=CreateSolidBrush(RGB(0, 255, 0));
    SelectObject(hDC, hBrush);
    fAngle=2*pi/nMaxNum*nNum;
    x=(int)(50*cos(fAngle+2*pi/3));
    y=(int)(50*sin(fAngle+2*pi/3));
    Pie(hDC, x-50, y-50, x+50, y+50,

        (int)(x+50*cos(fAngle+2*pi/3)),
        (int)(y+50*sin(fAngle+2*pi/3)),

        (int)(x+50*cos(fAngle+pi+2*pi/3)), (int)(y+50*sin(fAngle+pi+2*pi/3)));
    hBrush=CreateSolidBrush(RGB(0, 0, 255));
    SelectObject(hDC, hBrush);
    fAngle=2*pi/nMaxNum*nNum;
    x=(int)(50*cos(fAngle+4*pi/3));
    y=(int)(50*sin(fAngle+4*pi/3));

```

```

        Pie(hDC, x-50, y-50, x+50, y+50,
            (int) (x+50*cos(fAngle+4*pi/3)),
            (int) (y+50*sin(fAngle+4*pi/3)),
            (int) (x+50*cos(fAngle+pi+4*pi/3)),
            (int) (y+50*sin(fAngle+pi+4*pi/3)) );
    EndPaint(hwnd, &ps);
    DeleteObject(hBrush);
    DeleteObject(hPen);
    return 0;
case WM_TIMER:
    if (wParam==9999)
    {
        nNum++;
        if (nNum > nMaxNum)    nNum = 0;
        InvalidateRect(hwnd, NULL, 1);
    }
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    return 0;
default:
    return DefWindowProc(hwnd, uMsg, wParam, lParam);
}
return 0;
}

```

// resource.h(头文件)

```

#define IDM_BEGIN                15
#define IDM_STOP                 16
#define IDM_EXIT                 17
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 102
#define _APS_NEXT_COMMAND_VALUE 40004
#define _APS_NEXT_CONTROL_VALUE 1001
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

// Menu.rc(资源文件)

#include "resource.h"

MYMENU MENU

BEGIN

POPUP "控制(&C)"

BEGIN

MENUITEM "开始(&N)",

IDM_BEGIN

MENUITEM "停止(&O)",

IDM_STOP

```

        END
        POPUP "退出(&X)"
        BEGIN
            MENUITEM "退出(&E)",                                IDM_EXIT
        END
    END
END

```

//实验三代码:

```

//Main.cpp
#include<windows.h>
int nMode;
RECT rect;
LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM) ;

int WINAPI WinMain
(
    HINSTANCE hinstaance,
    HINSTANCE pre, LPSTR lpszCmdLine,
    int nCmdShow
)
{
    HWND hwnd;
    MSG msg;
    WNDCLASS wndclass;
    char lpszClassName[]=" ";
    char Title[]="实验三";
    wndclass.cbClsExtra=0;
    wndclass.cbWndExtra=0;
    wndclass.hbrBackground=CreateSolidBrush (RGB (255, 255, 255)) ;
    wndclass.hCursor=LoadCursor (NULL, IDC_ARROW) ;
    wndclass.hIcon=LoadIcon (NULL, IDI_QUESTION) ;
    wndclass.hInstance=hinstaance;
    wndclass.lpfWndProc=WndProc;
    wndclass.lpszClassName=lpszClassName;
    wndclass.lpszMenuName=NULL;
    wndclass.style=0;
    RegisterClass (&wndclass);

    hwnd=CreateWindow (lpszClassName, Title, WS_OVERLAPPEDWINDOW|WS_HSCROLL|WS_VSC
    ROLL, 100, 100, 500, 500, NULL, NULL, hinstaance, NULL ) ;
    ShowWindow (hwnd, nCmdShow) ;
    UpdateWindow (hwnd) ;
    while (GetMessage (&msg, NULL, 0, 0))
    {
        TranslateMessage (&msg) ;
        DispatchMessage (&msg) ;
    }
}

```

```

        return msg.wParam;
    }

LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    HPEN hp;
    HBRUSH hb;
    static int Circle;
    static int s;
    PAINTSTRUCT ps;
    switch(message) {
        case WM_KEYDOWN:
            if(wParam==VK_CONTROL)
            {
                Circle=1;
                s = 0;
                rect.left=0;
                rect.right=50;
                rect.top=0;
                rect.bottom=50;
                InvalidateRect(hwnd, NULL, 1);
            }else if(wParam == VK_SHIFT)
            {
                Circle=0;
                s = 1;
                rect.left=0;
                rect.right=50;
                rect.top=0;
                rect.bottom=50;
                InvalidateRect(hwnd, NULL, 1);
            }
            else if(wParam==VK_HOME)
            {
                rect.left=rect.left-10;
                rect.right=rect.right-10;
                if(Circle==1)
                {
                    InvalidateRect(hwnd, NULL, 1);
                }
            }
            else if(wParam==VK_END)
            {
                rect.left=rect.left+10;
                rect.right=rect.right+10;
                if(Circle==1)
                {

```

```

        InvalidateRect (hwnd, NULL, 1);
    }
} else if (wParam == VK_RIGHT)
{
    rect.right = rect.right + 10;
    if (s == 1)
    {
        InvalidateRect (hwnd, NULL, 1);
    }
} else if (wParam == VK_LEFT)
{
    rect.bottom = rect.bottom + 10;
    if (s == 1)
    {
        InvalidateRect (hwnd, NULL, 1);
    }
} else if (wParam == VK_PRIOR)
{
    rect.top = rect.top - 10;
    rect.bottom = rect.bottom - 10;
    if (s == 1)
    {
        InvalidateRect (hwnd, NULL, 1);
    }
} else if (wParam == VK_NEXT)
{
    rect.top = rect.top + 10;
    rect.bottom = rect.bottom + 10;
    if (s == 1)
    {
        InvalidateRect (hwnd, NULL, 1);
    }
}
break;
case WM_PAINT:
    hdc=BeginPaint (hwnd, &ps);
    SetMapMode (hdc, MM_ISOTROPIC);
    SetWindowExtEx (hdc, 1, 1, NULL);
    SetViewportExtEx (hdc, 1, 1, NULL);
    hp=CreatePen (PS_SOLID, 0, RGB (0, 0, 255));
    hb=CreateSolidBrush (RGB (0, 255, 0));
    SelectObject (hdc, hb);
    SelectObject (hdc, hp);
    if (Circle==1)
    {
        Ellipse (hdc, rect.left, rect.top, rect.right, rect.bottom);
    } else if (s == 1)

```

```
    {
        Rectangle(hdc, rect. left, rect. top, rect. right, rect. bottom);
    }
    EndPaint (hwnd, &ps);
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
default:
    return(DefWindowProc(hwnd, message, wParam, lParam));
}
return 0;
}
```


实验课程名称： 可视化编程

实验项目名称	基于 MFC 编程（一）			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	(无)			实验日期	2018 年 5 月 17 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

- 1. 创建一个基于 MFC 的工程，利用 MFC 应用程序向导创建一个基本对话框；
- 2. 在对话框里添加加减乘除运算的相关控件，并设置各个控件的名称、属性等；
- 3. 实现加减乘除运算，并能给出正确运算结果。

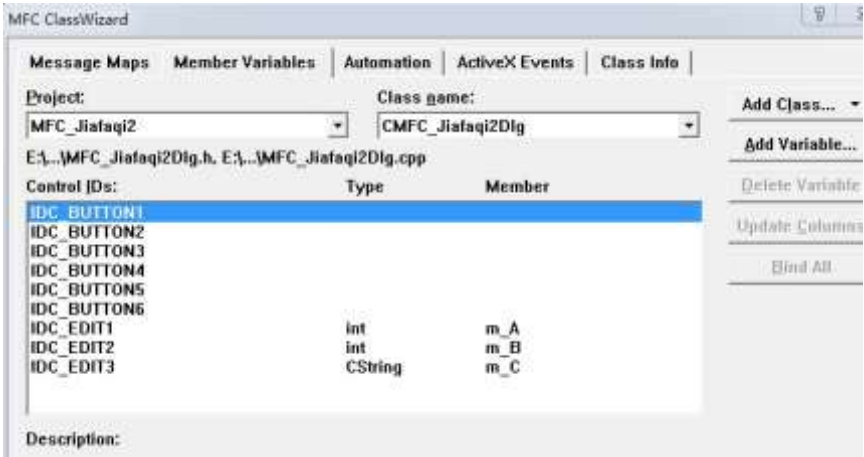
二、实验基本原理与设计

实验过程步骤：

- 1. 在 VC++ 6.0 环境下，选择“新建”，再选择“新建”下的“工程”中的“MFC Appwizard[exe]”，选择要创建的工程所在文件夹，输入要创建的工程文件名；
- 2. 在 MFC 应用程序向导第一步中选择“基本对话框”；
- 3. 依次按照 MFC 应用程序向导步骤创建一个包含对话框的 MFC 工程；
- 4. 在对话框资源中添加相应的静态文本框、编辑框与按钮控件，如下图所示：



- 5. 同时按下 Ctrl+W 键，打开 MFC ClassWizard, 给相应控件对象添加相应的成员变量，设置各成员变量的类型，如下图所示：



- 6. 对六个下压式按钮分别添加单击消息响应函数，并各自键入相应代码，代码如下：

```
(1) void CMFC_Jiafaqi2Dlg::OnButton1()  
{
```

```

        UpdateData(TRUE);
        int i=m_A+m_B;
        m_C.Format("%d", i);
        UpdateData(FALSE);
    }
(2) void CMFC_Jiafaqi2Dlg::OnButton2()
{
    UpdateData(TRUE);
    int i=m_A-m_B;
    m_C.Format("%d", i);
    UpdateData(FALSE);
}
(3) void CMFC_Jiafaqi2Dlg::OnButton3()
{
    UpdateData(TRUE);
    int i=m_A*m_B;
    m_C.Format("%d", i);
    UpdateData(FALSE);
}
(4) void CMFC_Jiafaqi2Dlg::OnButton4()
{
    UpdateData(TRUE);
    float i=float(m_A)/float(m_B);
    m_C.Format("%6f", i);
    UpdateData(FALSE);
}
(5) void CMFC_Jiafaqi2Dlg::OnButton5()
{
    OnOK();
}
(6) void CMFC_Jiafaqi2Dlg::OnButton6()
{
    m_A=0; m_B=0;
    m_C.Format("%d", 0);
    UpdateData(FALSE);
}

```

三、主要仪器设备及耗材

1. 一台 windows 7 的 PC 机;
2. 软件 Microsoft Visual Studio 2010;
3. 草稿纸和笔;
4. 参考资料: Visual C++面向对象与可视化程序设计。

第二部分：实验调试与结果分析（可加页）

一、调试过程

调试方法：

- (1) 按照实验步骤操作输入代码，然后先编译后，再点击“开始执行”；
- (2) 在编辑框中分别按照以下数据输入：
操作数一：2016 操作数二：2 点击“+”按钮；
操作数一：1009 操作数二：2 点击“-”按钮；
- (3) 然后按照实验结果记录数据，并对实验结果分析。

二、实验结果及分析

按照实验步骤，然后调试程序，记录数据如下：

实验结果截图：

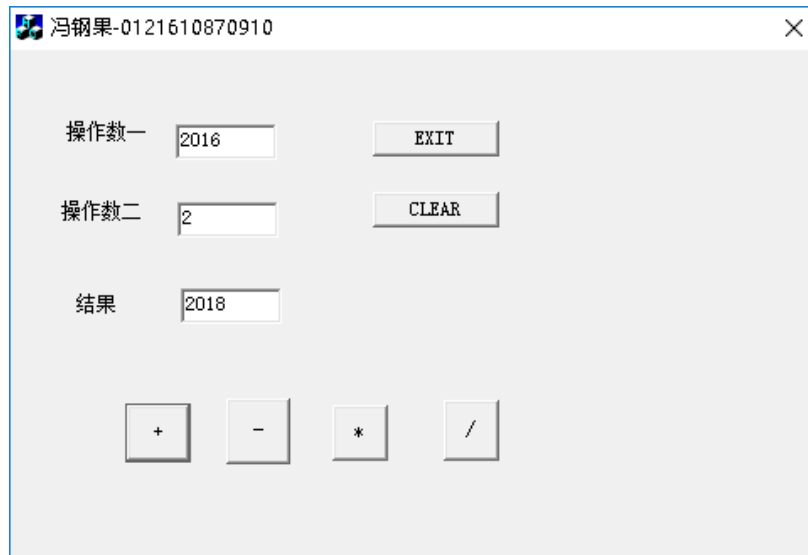


图 1

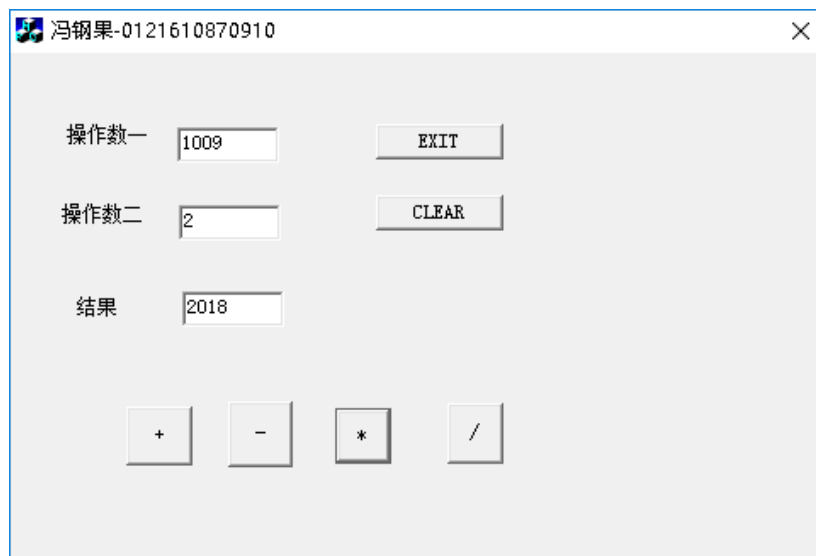


图 2



图 3

实验结果分析：

图 1 为实验输入 2017 和 1，然后执行加法功能，得到结果“2018”，符合结果；

图 2 为实验输入 1009 和 2，然后执行乘法功能，得到结果“2018”，符合结果；

图 3 为实验按下按钮“CLEAR”，实现计算器清零功能。

三、实验小结、建议及体会

通过本次实验，我掌握了 MFC 类的组织结构和主要类，熟悉了 VC++集成开发环境，学会了 MFC 下标准控件的使用，每类控件的属性、事件及方法。

在这次实验中，我学到的不仅仅是一些理论知识，还有开发的经验，通过实验的一次次出错，我深刻的意识到自己的动手能力远远不足。我可以通过实验弥补自己开发经验的不足，但实验中的工作量和知识量是远远不够的，因此课下我也在不断地努力的学习，不断地学习更多的编程技巧，也在和同学组队开发一些小的项目，这次实验对我的平时项目实验中补充了知识上不足，但我也在将实验内容和实践应用相结合，比如试着做一个闹钟程序之类的，代码之路还很长，只有以认真态度才能有所突破。

实验课程名称： 可视化编程

实验项目名称	基于 MFC 编程（二）			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	(无)			实验日期	2018 年 5 月 24 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

1. 创建一个基于 MFC 的工程，利用 MFC 应用程序向导创建一个基本对话框；
2. 在对话框里添加能够实现多组数据输入、数据修改、数据查询的相关控件，并为相应控件建立关联的成员变量；
3. 创建相应的消息函数，并且添加消息响应代码，正确实现多组数据的输入、修改以及查询的功能。

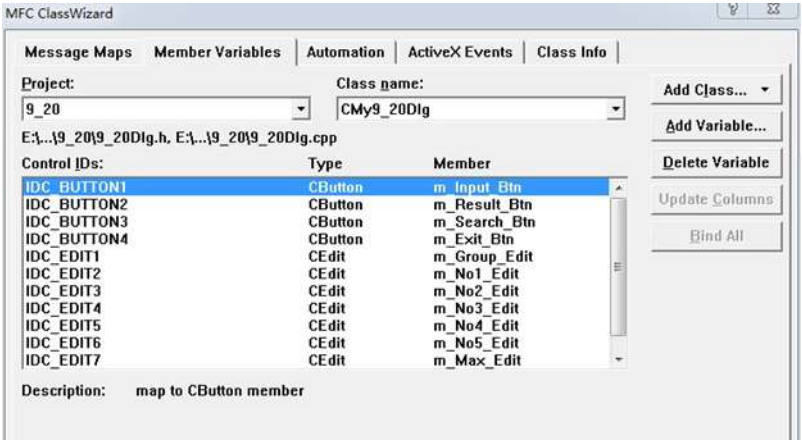
二、实验基本原理与设计

实验过程步骤：

1. 在 VC++ 6.0 环境下，选择“新建”，再选择“新建”下的“工程”中的“MFC Appwizard[exe]”，选择要创建的工程所在文件夹，输入要创建的工程文件名；
2. 在 MFC 应用程序向导第一步中选择“基本对话框”；
3. 依次按照 MFC 应用程序向导步骤创建一个包含对话框的 MFC 工程；
4. 在对话框资源中添加相应的静态文本框、编辑框、组框与按钮控件，如下图所示：



5. 同时按下 Ctrl+W 键，打开 MFC ClassWizard, 对相应控件对象添加相应的成员变量，如下图所示：



6. 给“输入数据”、“输出结果”、“检索数据”、“退出”四个按钮添加单击消息响应函数，并键入各个函数的代码，代码如下：

(1) “输入数据”按钮单击消息函数代码：

```
void CMy9_20Dlg::OnInputBtn()
{
    CString cstrGroup, cstrInputData[5];
    int iGroup;
    m_Group_Edit.GetWindowText(cstrGroup);
    m_No1_Edit.GetWindowText(cstrInputData[0]);
    m_No2_Edit.GetWindowText(cstrInputData[1]);
    m_No3_Edit.GetWindowText(cstrInputData[2]);
    m_No4_Edit.GetWindowText(cstrInputData[3]);
    m_No5_Edit.GetWindowText(cstrInputData[4]);
    iGroup=atoi((LPCTSTR)cstrGroup);
    for(int i=0;i<5;i++)
    {
        InputData[iGroup-1].dfData[i]=atof((LPCTSTR)cstrInputData[i]);
    }
}
```

(2) “输出结果”按钮单击消息函数代码：

```
void CMy9_20Dlg::OnResultBtn()
{
    CString cstrGroup;
    int iGroup;
    double dfMax=0, dfMin=0;
    char cBuffer[10];
    m_Group_Edit.GetWindowText(cstrGroup);
    iGroup=atoi((LPCTSTR)cstrGroup);
    dfMax=InputData[iGroup-1].dfData[0];
    dfMin=InputData[iGroup-1].dfData[0];
    for(int i=0;i<5;i++)
    {
        if(dfMax<InputData[iGroup-1].dfData[i])
        {
            dfMax=InputData[iGroup-1].dfData[i];
        }
        if(dfMin>InputData[iGroup-1].dfData[i])
        {
            dfMin=InputData[iGroup-1].dfData[i];
        }
    }
    _gcvt(dfMax, 10, cBuffer);
    m_Max_Edit.SetWindowText((LPCTSTR)cBuffer);
    _gcvt(dfMin, 10, cBuffer);
    m_Min_Edit.SetWindowText((LPCTSTR)cBuffer);
}
```

(3) “检索数据”按钮单击消息函数代码：

```

void CMy9_20Dlg::OnSearchBtn()
{
    CString cstrGroup;
    int iGroup;
    char cBuffer[10];
    int i;
    m_Group_Edit.GetWindowText(cstrGroup);
    iGroup=atoi((LPCTSTR)cstrGroup);
    _gcvt(InputData[iGroup-1].dfData[0], 10, cBuffer);
    m_No1_Edit.SetWindowText((LPCTSTR)cBuffer);
    _gcvt(InputData[iGroup-1].dfData[1], 10, cBuffer);
    m_No2_Edit.SetWindowText((LPCTSTR)cBuffer);
    _gcvt(InputData[iGroup-1].dfData[2], 10, cBuffer);
    m_No3_Edit.SetWindowText((LPCTSTR)cBuffer);
    _gcvt(InputData[iGroup-1].dfData[3], 10, cBuffer);
    m_No4_Edit.SetWindowText((LPCTSTR)cBuffer);
    _gcvt(InputData[iGroup-1].dfData[4], 10, cBuffer);
    m_No5_Edit.SetWindowText((LPCTSTR)cBuffer);
}

```

(4) “退出”按钮单击消息函数代码：

```

void CMy9_20Dlg::OnExitBtn()
{
    OnOK();
}

```

三、主要仪器设备及耗材

1. 一台 windows 7 的 PC 机；
2. 软件 Microsoft Visual Studio 2010；
3. 草稿纸和笔；
4. 参考资料：Visual C++面向对象与可视化程序设计。

第二部分：实验调试与结果分析（可加页）

一、调试过程

调试方法：

- (1) 按照实验步骤操作输入代码，然后先编译后，再点击“开始执行”；
- (2) 分别按照以下数据输入：
 - 组号：1 数据：10、50、20、30、40，点击“输入数据”按钮，再点击“输出结果”；
 - 组号：2 数据：60、90、70、100、80，点击“输入数据”按钮，再点击“输出结果”；
 - 组号：3 数据：10、50、100、20、70，点击“输入数据”按钮，再点击“输出结果”。
- (3) 然后按照实验结果记录数据，并对实验结果分析。

二、实验结果及分析

按照实验步骤，然后调试程序，记录数据如下：

实验结果截图：



图 1



图 2



图 3



图 4

结果分析：

图 1、图 2、图 3 为分别输入三组数据并按下“输入数据”和“输出结果”的截图；

图 4 为输入组号 3，在按下“检索数据”的截图；

以上结果符合实验预期结果，因此实验成功。

三、实验小结、建议及体会

通过本次实验，我掌握了 MFC 下如何对标准控件设置添加关联的成员变量，熟悉了 VC++ 集成开发环境，学会了 MFC 下如何添加消息响应函数。

并且通过实验的一次次出错，我深刻的意识到自己的动手能力远远不足。我可以通过实验弥补自己开发经验的不足，但实验中的工作量和知识量是远远不够的，每当找出问题并把它解决的时候都会觉得受益匪浅，努力和付出都没有浪费。以前自己使用的各种软件现在自己也可以通过所学的 API 和 MFC 来编程相应的简单软件来实现，体会到学以致用快乐，今后我会继续尝试去编写其他的软件，加深对所学知识的掌握和应用。这次实验对我的平时项目实验中补充了知识上不足，但我也在将实验内容和实践应用相结合学会学以致用才是真正的学习目的，不能不知变通的学习，否则很难学会深层知识，因此学习还要继续努力。