# 第18次实验

# Windows编程基础训练

**1、实验学时：**2个课时。

**2、实验类型：**设计性实验。

**3、实验目的：**

（1）了解Windows编程的消息句柄和事件触发等概念，理解消息句柄的捕获和事件的响应等Windows编程机制；

（2）熟悉在Visual C++ 6.0中编辑、编译、调试和运行一个实际窗口式应用程序的步骤和方法，能进行简单的Windows绘图程序设计；

（3）掌握Windows消息传递、事件驱动以及Windows API封装和调用的方法。

**4、实验内容：**

**实验题目1：圆心沿正弦曲线轨迹移动的实心圆程序**

（1）程序功能简介

设计一个程序，实现“圆心沿正弦曲线轨迹移动的实心圆”能够循环往复地不断运动。

（2）程序设计说明

① 打开Visual C++ 6.0，选择“文件”—>“新建”—>“win32 Application”命令，输入工程名，弹出“创建程序”对话框，选择建立一个空文档，进入工作区“Source file”，点击右键添加文件。打开添加的文件进行编译。

② 在代码编辑窗口内，添加如下代码：

#include <windows.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#define Pi 3.1415926

long WINAPI WndProc(HWND hWnd,UINT iMessage,UINT wParam, LONG lParam);

double dfRange=100.0; //正弦曲线的角度变量

long i=0,j=0;

long lCentreX=0,lCentreY=0,lRadious; //定义圆心坐标和圆半径

POINT lpSin[100]; //定义正弦曲线的点坐标

int WINAPI WinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPSTR lpCmdLine,

int nCmdShow) //主函数

{

MSG Message; //定义消息变量

HWND hWnd; //定义窗口句柄

WNDCLASS WndClass; //定义窗口类

WndClass.cbClsExtra=0; //无窗口类扩展

WndClass.cbWndExtra=0; //无窗口实例扩展

WndClass.hbrBackground=(HBRUSH)(GetStockObject(WHITE\_BRUSH));

WndClass.hCursor=LoadCursor(NULL,IDC\_ARROW);//窗口用箭头光标

WndClass.hIcon=LoadIcon(NULL,IDI\_APPLICATION); //用缺省图标

WndClass.hInstance=hInstance; //当前应用程序句柄

WndClass.lpfnWndProc=WndProc; //窗口处理函数

WndClass.lpszClassName="SIN"; //窗口类名称

WndClass.lpszMenuName=NULL; //无窗口菜单

WndClass.style=0; //窗口类型为缺省类型

if(!RegisterClass(&WndClass)) //注册窗口

{

MessageBeep(0);

return FALSE;

}

hWnd=CreateWindow("SIN", //窗口类名

"ex4\_3沿正弦曲线移动的实心圆", //标题名

WS\_OVERLAPPEDWINDOW,//带标题栏、最大和最小按钮的窗口

CW\_USEDEFAULT, //窗口左上角坐标

0,

CW\_USEDEFAULT, //缺省的宽度高度

0,

NULL, //无父窗口

NULL, //无主菜单

hInstance, //当前实例句柄

NULL); //不要此参数

ShowWindow(hWnd,nCmdShow); //显示窗口

UpdateWindow(hWnd); //更新并绘制用户区

for(int j=0;j<100;j++) //生成正弦曲线的点坐标

{

lpSin[j].x=(long)(j\*2\*Pi/100\*60);

lpSin[j].y=(long)(dfRange\*sin(j\*2\*Pi/100));

}

while(GetMessage(&Message,0,0,0))

{

TranslateMessage(&Message); //消息循环

DispatchMessage(&Message);

}

return Message.wParam;

}

long WINAPI WndProc(HWND hWnd,

UINT iMessage,

UINT wParam,

LONG lParam) //消息处理函数

{

HDC hDC; //定义设备环境句柄

HBRUSH hBrush; //定义画刷句柄

HPEN hPen; //定义画笔句柄

PAINTSTRUCT PtStr; //定义包含绘图信息的结构体变量

switch(iMessage)

{

case WM\_PAINT: //处理绘图消息

hDC=BeginPaint(hWnd,&PtStr); //获得设备环境指针

SetWindowOrgEx(hDC,-200,-200,NULL);

//设置原点坐标为(-200,-200)

hPen=CreatePen(PS\_SOLID,1,RGB(255,0,0)); //创新画笔

SelectObject(hDC,hPen); //选入画笔

Polyline(hDC,lpSin,100); //绘制正弦曲线

if(i%100<=25) //第一个1/4周期

{

hPen=CreatePen(PS\_SOLID,1,RGB(255,0,0)); //创新红笔

hBrush=CreateSolidBrush(RGB(255,0,0)); //创新红画刷

}

else if(i%100<=50) //第二个1/4周期

{

hPen=CreatePen(PS\_DOT,1,RGB(0,255,0)); //创新绿笔

hBrush=CreateSolidBrush(RGB(0,255,0)); //创新绿刷

}

else if(i%100<=75) //第三个周期

{

hPen=CreatePen(PS\_DASH,1,RGB(0,0,255)); //创新蓝笔

hBrush=CreateSolidBrush(RGB(0,0,255)); //创新蓝刷

}

else //第四个周期

{

hPen=CreatePen(PS\_DASHDOT,1,RGB(250,250,0)); //创新黄笔

hBrush=CreateSolidBrush(RGB(255,255,0)); //创新黄刷

}

lRadious=(long)(dfRange\*(0.2+i%25\*0.4/25)); //计算半径

SelectObject(hDC,hBrush); //选入画刷

SelectObject(hDC,hPen); //选入画笔

lCentreX=lpSin[i%100].x; //圆心x坐标

lCentreY=lpSin[i%100].y; //圆心y坐标

Ellipse(hDC,lCentreX-lRadious,

lCentreY-lRadious,

lCentreX+lRadious,

lCentreY+lRadious); //画圆

i++;

DeleteObject(hPen); //删除画笔

DeleteObject(hBrush); //删除画刷

EndPaint(hWnd,&PtStr); //删除设备环境指针

Sleep(100); //停0.1秒

if(i%100<100) InvalidateRect(hWnd,NULL,1);

//刷新用户区,并发送消息WM\_PAINT

return 0;

case WM\_DESTROY: //关闭窗口

PostQuitMessage(0);

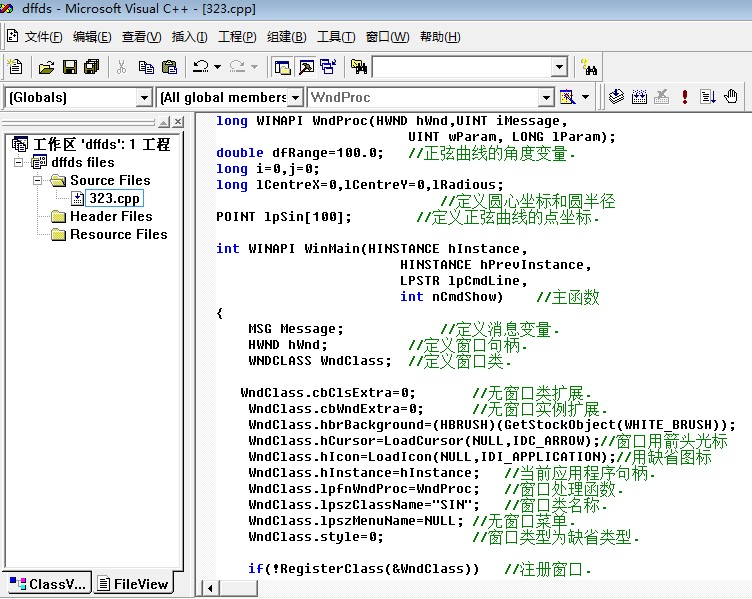
return 0;

default:

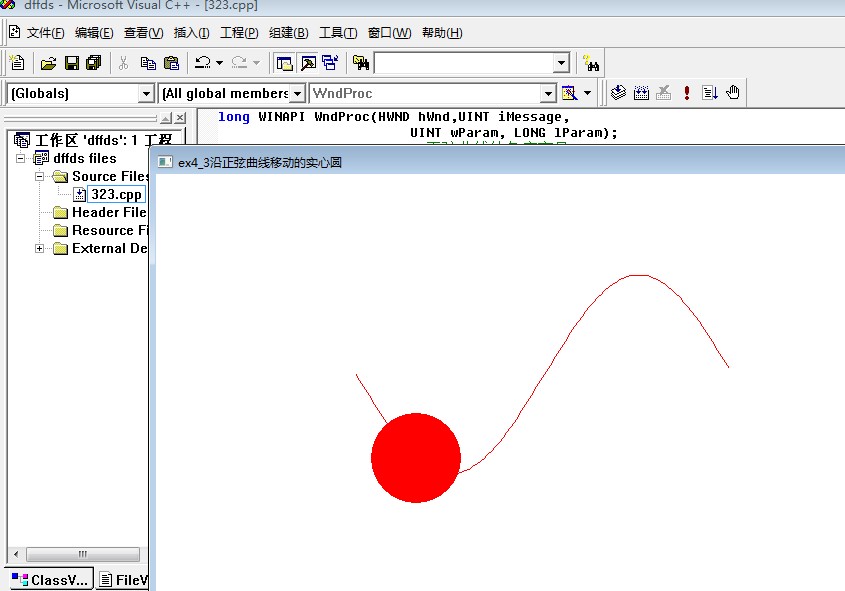
return(DefWindowProc(hWnd,iMessage,wParam,lParam));

}

}



③ 选择单击工具栏上的编译按钮，进行调试，未出现错误则单击执行按钮运行程序，查看运行结果。



（3）程序调试运行

运行程序查看结果，并进行源代码调试和优化。

**参考代码如下：**

略。

**执行结果如下：**

略。

**实验题目2：定时器程序**

（1）程序功能简介

设计程序在窗口中使用定时器，每隔1秒交替地用红、绿、蓝色画刷来填充整个窗口客户区。

（2）程序设计说明

① 打开Visual C++ 6.0，选择“文件”—>“新建”—>“win32 Application”命令，输入工程名，弹出“创建程序”对话框，选择建立一个空文档，进入工作区“Source file”,点击右键添加文件。打开添加的文件进行编译；

② 在代码编辑窗口内，添加如下代码：

#include "windows.h"

#include "string.h" //加载相应的头文件

#include "stdlib.h"

#include "stdio.h"

LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);

//窗口函数说明

int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInst, LPSTR lpszCmdLine,int nCmdShow)

{

HWND hwnd;

MSG Msg;

WNDCLASS wndclass;

char lpszClassName[]="窗口"; //窗口类名

char lpszTitle[]="定时刷新窗口"; //窗口标题

wndclass.style=CS\_HREDRAW|CS\_VREDRAW|CS\_DBLCLKS; //窗口类型

wndclass.lpfnWndProc=WndProc; //窗口处理函数为WndProc

wndclass.cbClsExtra=0; //窗口类无扩展

wndclass.cbWndExtra=0; //窗口实例无扩展

wndclass.hInstance=hInstance; //当前实例句柄

wndclass.hIcon=LoadIcon(NULL,IDI\_APPLICATION);

//窗口最小化图标为默认图标

wndclass.hCursor=LoadCursor(NULL,IDC\_ARROW); //窗口采用箭头光标

wndclass.hbrBackground=(HBRUSH)GetStockObject(WHITE\_BRUSH);

//窗口背景为白色

wndclass.lpszMenuName=NULL; //窗口中无菜单

wndclass.lpszClassName=lpszClassName;//窗口类名

if(!RegisterClass(&wndclass)) //如果窗口注册失败发出警告声音

{

MessageBeep(0);

return FALSE;

}

hwnd=CreateWindow(

lpszClassName, //窗口类名

lpszTitle, //窗口实例句柄的标题名

WS\_OVERLAPPEDWINDOW|WS\_HSCROLL|WS\_VSCROLL, //窗口风格

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT, //窗口左上角坐标为默认值

CW\_USEDEFAULT, //窗口的高和宽为默认值

NULL, //此窗口无父窗口

NULL, //此窗口无主菜单

hInstance, //创建此窗口的应用程序当前句柄

NULL //不使用该值

);

ShowWindow(hwnd,nCmdShow); //显示窗口

UpdateWindow(hwnd); //绘制用户区

while(GetMessage(&Msg,NULL,0,0)) //消息循环

{

TranslateMessage(&Msg);

DispatchMessage(&Msg);

}

return Msg.wParam; //程序终止时将信息返回系统

}

LRESULT CALLBACK WndProc(HWND hwnd,UINT message,WPARAM wParam,LPARAM lParam) //窗口函数

{

HDC hdc; //定义指向设备上下文的句柄

HBRUSH hB1,hB2,hB3; //定义指定的画刷

PAINTSTRUCT ps; //定义指向绘图信息的结构体变量

switch(message) //处理消息

{

case WM\_PAINT: //处理绘图消息

hdc=BeginPaint(hwnd,&ps); //取得环境设备句柄

SetMapMode(hdc,MM\_TEXT); //设置映射模式

hB1=CreateSolidBrush(RGB(255,0,0)); //创建红色画刷

hB2=CreateSolidBrush(RGB(0,255,0)); //创建绿色画刷

hB3=CreateSolidBrush(RGB(0,0,255)); //创建蓝色画刷

SelectObject(hdc,hB1); //选择红色画刷

Rectangle(hdc,0,0,800,800); //填充整个用户区

Sleep(1000); //设置定时器，间隔1秒

SelectObject(hdc,hB2); //选择绿色画刷

Rectangle(hdc,0,0,800,800); //填充整个用户区

Sleep(1000); //设置定时器，间隔1秒

SelectObject(hdc,hB3); //选择蓝色画刷

Rectangle(hdc,0,0,800,800); //填充整个用户区

InvalidateRect(hwnd,NULL,1); //刷新窗口，重新绘图

EndPaint(hwnd,&ps); //释放设备环境句柄

DeleteObject(hB1);

DeleteObject(hB2); //删除画刷句柄

DeleteObject(hB3);

break;

case WM\_DESTROY:

PostQuitMessage(0);

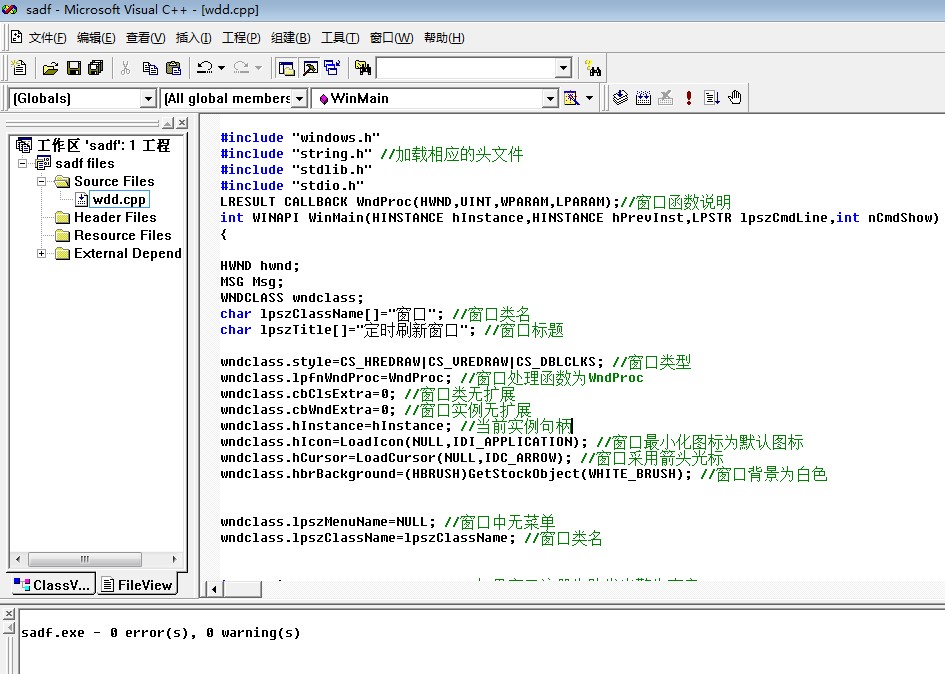
default:

return DefWindowProc(hwnd,message,wParam,lParam);

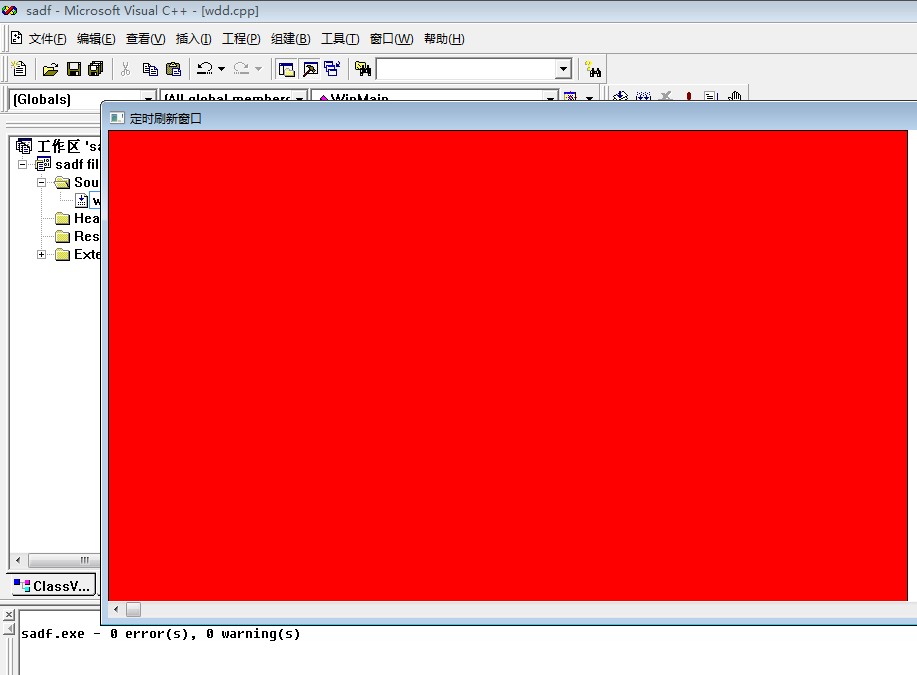
}

return(0);

}



③ 选择单击工具栏上的编译按钮，进行调试，未出现错误则单击执行按钮运行程序，查看运行结果。



（3）程序调试运行

运行程序查看结果，并进行源代码调试和优化。

**参考代码如下：**

略。

**执行结果如下：**

略。

**5、实验结果：**

参考实验报告模板，撰写本次实验的实验报告，并提交实验报告和实验题目的源代码。

告和实验题目的源代码。