# 一、构建一个 Windows 应用程序——API 编程方式

【 参 考 代 码 】 #include <windows.h> #include <stdio.h>

LRESULT CALLBACK WinSunProc(

HWND hwnd, // handle to window UINT uMsg, // message identifier

WPARAM wParam, // first message parameter LPARAM lParam // second message parameter

);

int WINAPI WinMain(

HINSTANCE hInstance, // handle to current instance HINSTANCE hPrevInstance, // handle to previous instance LPSTR lpCmdLine, // command line

int nCmdShow // show state

)

{

WNDCLASS wndcls; wndcls.style=CS\_HREDRAW | CS\_VREDRAW; wndcls.lpfnWndProc=WinSunProc; wndcls.cbClsExtra=0;

wndcls.cbWndExtra=0; wndcls.hInstance=hInstance; wndcls.hIcon=LoadIcon(NULL,IDI\_ERROR); wndcls.hCursor=LoadCursor(NULL,IDC\_CROSS);

wndcls.hbrBackground=(HBRUSH)GetStockObject(WHITE\_BRUSH); wndcls.lpszClassName="Weixin2003"; wndcls.lpszMenuName=NULL;

RegisterClass(&wndcls);

HWND hwnd;

hwnd=CreateWindow("Weixin2003"," 北 京 维 新 科 学 技 术 培 训 中 心

",WS\_OVERLAPPEDWINDOW,0,0,600,400,NULL,NULL,hInstance,NULL);

ShowWindow(hwnd,SW\_SHOWNORMAL); UpdateWindow(hwnd);

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

{

TranslateMessage(&msg); //将虚拟键消息(如 WM\_KEYDONW)转换为字符消息

WM\_CHAR

DispatchMessage(&msg); // 将消息回传 OS ，由 OS 调用窗口过程函数

WinSunProc 完成消息处理

}

return 0;

}

LRESULT CALLBACK WinSunProc(

HWND hwnd, // handle to window UINT uMsg, // message identifier

WPARAM wParam, // first message parameter LPARAM lParam // second message parameter

)

{

switch(uMsg)

{

case WM\_CHAR: char szChar[20];

sprintf(szChar,"char is d",wParam); MessageBox(hwnd,szChar,"weixin",0); break;

case WM\_LBUTTONDOWN:

MessageBox(hwnd,"mouse clicked","weixin",0); HDC hdc;

hdc=GetDC(hwnd);

TextOut(hdc,0,50,"计算机编程语言培训",strlen("计算机编程语言培训")); ReleaseDC(hwnd,hdc);

break;

case WM\_RBUTTONDOWN:

MessageBox(hwnd, "right clicked","weixin", 0); break;

case WM\_PAINT: HDC hDC;

PAINTSTRUCT ps;

hDC=BeginPaint(hwnd,&ps);

TextOut(hDC,0,0,"维新培训",strlen("维新培训")); EndPaint(hwnd,&ps);

break;

case WM\_CLOSE:

if(IDYES==MessageBox(hwnd,"是否真的结束？","weixin",MB\_YESNO))

{

DestroyWindow(hwnd);

}

break;

case WM\_DESTROY: PostQuitMessage(0); break;

default:

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

}

return 0;

}

# 二、基于 API 构建一个 MFC 程序

重点：理解 MFC 程序的消息映射机制以及在 VC++6.0 开发环境中的操作方法。提示：注意在程序编译时设置工程，以共享方式使用 MFC 基础类库

【参考代码】

#include<afxwin.h> // MFC 头文件

class CHelloApp:public CWinApp // 声明应用程序类

{

public:

virtual BOOL InitInstance();

};

CHelloApp theApp; // 建立应用程序类实例

class CMainFrame:public CFrameWnd // 声明主窗口类

{

public:

CMainFrame()

{

//创建主窗口

Create(NULL, "我的窗口", WS\_OVERLAPPEDWINDOW, CRect(0,0, 600, 400));

}

protected:

afx\_msg void OnLButtonDown(UINT nFlag, CPoint point); afx\_msg void OnRButtonDown(UINT nFlag, CPoint point); DECLARE\_MESSAGE\_MAP();

};

//消息映射入口BEGIN\_MESSAGE\_MAP(CMainFrame, CFrameWnd)

ON\_WM\_LBUTTONDOWN() // 单击鼠标左键消息映射宏

ON\_WM\_RBUTTONDOWN() END\_MESSAGE\_MAP()

//定义消息映射函数

void CMainFrame::OnLButtonDown(UINT nFlags, CPoint point)

{

MessageBox("你好，我的 Visual C++世界!", "问候", 0); CFrameWnd::OnLButtonDown(nFlags, point);

}

void CMainFrame::OnRButtonDown(UINT nFlags, CPoint point)

{

MessageBox(" 右 键 被 单 击 "," 提 示 ",0); CFrameWnd::OnRButtonDown(nFlags, point);

}

//应用程序首次执行时都要调用的初始化函数BOOL CHelloApp::InitInstance()

{

m\_pMainWnd = new CMainFrame(); m\_pMainWnd->ShowWindow(m\_nCmdShow); m\_pMainWnd->UpdateWindow();

return TRUE;

}

# 三、基于向导构建一个 MFC 程序

要求：学会使用类向导添加消息映射，添加对鼠标右键单击事件、左键双击事件以及键盘消息的响应。