

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_KEYDOWN)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。