

# Windows应用编程

## WinMain 函数

- Windows 应用程序的入口点（替代控制台的 `main`）。
- 参数：
  - `hInstance`：当前实例句柄。
  - `hPrevInstance`：已废弃（始终为 `NULL`）。
  - `lpCmdLine`：命令行参数。
  - `nCmdShow`：窗口显示方式（如最大化、最小化）。

## MFC

## 成员函数

## (1) 消息映射函数

- **作用：**处理 Windows 消息（如鼠标、键盘、窗口事件）。
- **典型函数：**
  - `OnPaint()` → 处理 `WM_PAINT` 消息。
  - `OnLButtonDown()` → 处理鼠标左键按下消息。
  - `OnCommand()` → 处理菜单、按钮点击等命令消息。
- **扩展方法：**
  - 在类中添加消息映射宏（如 `ON_WM_LBUTTONDOWN()`）。
  - 重写基类虚函数（如 `CWnd::OnLButtonDown`）。

## (2) 虚函数（多态行为）

- **作用：**允许派生类覆盖基类行为，实现定制化逻辑。
- **典型函数：**
  - `PreCreateWindow(CREATESTRUCT& cs)` → 在窗口创建前修改属性。
  - `OnDraw(CDC* pDC)` → 自定义绘图逻辑（覆盖 `CView::OnDraw`）。
  - `DoDataExchange(CDataExchange* pDX)` → 对话框数据交换（DDX/DDV）。
- **扩展方法：**
  - 直接继承并重写虚函数（如 `class CMyView : public CView { ... }`）。

### (3) 框架控制函数

- **作用：**管理应用程序生命周期、文档/视图结构等。
- **典型函数：**
  - InitInstance() → MFC 应用程序初始化入口。
  - OnFileOpen() → 处理文件打开命令（通常由 CWinApp 派生类实现）。
  - UpdateAllViews(CDocument\* pSender, LPARAM lHint, CObject\* pHint) → 通知所有视图更新。
- **扩展方法：**
  - 在 CWinApp、CFrameWnd 等派生类中覆盖这些函数。

### (4) 工具类函数

- **作用：**提供通用功能（如字符串处理、文件操作）。
- **典型函数：**
  - AfxMessageBox() → 显示消息框（全局函数，非成员函数）。
  - CString::Format() → 格式化字符串（CString 类成员函数）。
  - CFile::Open() → 文件操作（CFile 类成员函数）。
- **扩展方法：**
  - 封装自定义工具类（继承或组合 MFC 类）。

CView::GetDocument函数返回的是指向用户（文档）类的指针。

**问题1：在文档（CDocument）/视图（CView）程序结构中，当视图窗口需要重画时，视图类函数（ ）被调用。**

**答案：**

`*OnDraw(CDC pDC)`

## Edit控件

**MFC应用程序设计中，CEdit是（ ）控件。答案：**

**CEdit 是 MFC 提供的标准 单行或多行文本编辑控件（Edit Control）**

用于在对话框或视图中实现文本输入、编辑功能。

## 常见 Windows 消息

### 3. 常见 Windows 消息

消息	描述
WM_CREATE	窗口创建时触发。
WM_PAINT	窗口需要重绘时触发。
WM_SIZE	窗口大小改变时触发。
WM_CLOSE	用户关闭窗口时触发。
WM_COMMAND	菜单或控件事件（如按钮点击）。

## GDI（图形设备接口）

- 用于绘制图形和文本的 API：
  - CDC：设备上下文类（如 `CPaintDC` 用于 `WM_PAINT`）。
  - 常用函数：`Rectangle()`，`TextOut()`，`MoveTo()`，`LineTo()`。

## WINAPI标识符的作用

### 定义

`WINAPI` 是 Windows 头文件（如 `<windows.h>`）中定义的宏，通常展开为 `__stdcall`（在大多数现代 Windows 平台上）。

- **作用：**  
指定函数的**调用约定**（Calling Convention），即参数如何传递、谁负责清理栈等规则。

在Windows应用程序中，WinMain函数名前的WINAPI标识符是一种（调用约定）宏。

- `WINAPI` 是 Windows API 中常用的调用约定宏，通常定义为 `__stdcall`。

## Windows 消息

### 定义

- 消息本质是一个 **32位整数**（`UINT`），分为：
  - **标准消息**（如 `WM_PAINT`、`WM_CLOSE`）。
  - **命令消息**（如菜单、按钮点击，`WM_COMMAND`）。
  - **自定义消息**（用户定义，`WM_USER` 及以上值）。

# 鼠标和键盘

## (1) 鼠标消息

消息	触发条件	相关参数
<u><code>WM_LBUTTONDOWN</code></u>	<u>鼠标左键按下</u>	<code>wParam</code> : 按键状态 (如 <code>MK_SHIFT</code> )   <code>lParam</code> : 低16位为X坐标, 高16位为Y坐标
<u><code>WM_LBUTTONUP</code></u>	<u>鼠标左键释放</u>	同上
<code>WM_LBUTTONDBLCLK</code>	鼠标左键双击	同上
<code>WM_MOUSEMOVE</code>	鼠标移动	<code>wParam</code> : 按键状态  <code>lParam</code> : 当前坐标
<code>WM_RBUTTONDOWN</code> / <code>WM_RBUTTONUP</code>	右键按下/释放	类似左键消息

## (2) 键盘消息

消息	触发条件	相关参数
<code>WM_KEYDOWN</code>	键盘按键按下	<code>wParam</code> : 虚拟键码 (如 <code>VK_RETURN</code> )   <code>lParam</code> : 重复次数、扫描码等
<code>WM_KEYUP</code>	键盘按键释放	同上
<code>WM_CHAR</code>	字符输入 (已转换的ASCII码)	<code>wParam</code> : 字符的ASCII码

# 窗口和自定义

(3) 窗口管理消息

消息	触发条件	用途
WM_CREATE	窗口创建时	初始化资源
WM_SIZE	窗口大小改变	调整控件布局
WM_PAINT	窗口需要重绘	绘制界面（通过 BeginPaint/EndPaint）
WM_CLOSE	用户点击关闭按钮	可在此处询问是否保存数据
WM_DESTROY	窗口销毁时	释放资源，通常调用 PostQuitMessage(0) 退出程序

(4) 用户自定义消息

- 用户可通过 WM\_USER 或 WM\_APP 定义自己的消息：

cpp 复制

```
#define WM_MY_MESSAGE (WM_USER + 1) // 自定义消息ID
```

- 发送自定义消息：

cpp 复制

```
SendMessage(hWnd, WM_MY_MESSAGE, wParam, lParam);
```

# 相关API函数

## 4. 消息相关 API 函数

函数	功能
GetMessage	从消息队列获取消息（阻塞式）。
PeekMessage	检查消息队列（非阻塞式，可用于游戏循环）。
TranslateMessage	转换虚拟键消息为字符消息（如 WM_KEYDOWN → WM_CHAR）。
DispatchMessage	将消息分发到窗口过程。
SendMessage	直接发送消息到窗口过程（同步，等待处理完成）。
PostMessage	将消息放入消息队列（异步，不等待处理）。
SetTimer	设置定时器，触发 WM_TIMER 消息。
KillTimer	销毁定时器。