匈牙利命名法是微软推广的一种关于变量、函数、对象、前缀、宏定义等各种类型的符号的命名规范。匈牙利命名法的主要思想是：在变量和函数名中加入前缀以增进人们对程序的理解。它是由微软内部的一个匈牙利人发起使用的，结果它在微软内部逐渐流行起来，并且推广给了全世界的Windows开发人员。下面将介绍匈牙利命名法，后面的例子里也会尽量遵守它和上面的代码风格。还是那句话，并不是要求所有的读者都要去遵守，但是希望读者作为一个现代的软件开发人员都去遵守它。

有关项目的全局变量用g\_开始，类成员变量用m\_，局部变量若函数较大则可考虑用l\_用以显示说明其是局部变量。

前缀       类型       例子

g\_    全局变量       g\_Servers

C     类或者结构体       CDocument，CPrintInfo

m\_   成员变量       m\_pDoc，m\_nCustomers

VC常用前缀列表：

前缀       类型       描述       例子

ch    char 8位字符    chGrade

ch    TCHAR       16位UNICODE类型字符       chName

b     BOOL       布尔变量       bEnabled

n     int    整型（其大小由操作系统决定）       nLength

n     UINT       无符号整型（其大小由操作系统决定）       nLength

w    WORD       16位无符号整型    wPos

l      LONG       32位有符号整型    lOffset

dw   DWORD       32位无符号整型       dwRange

p     \*       Ambient memory model pointer 内存模块指针，指针变量    pDoc

lp     FAR\*       长指针       lpDoc

lpsz  LPSTR       32位字符串指针       lpszName

lpsz  LPCSTR       32位常量字符串指针       lpszName

lpsz  LPCTSTR       32位UNICODE类型常量指针       lpszName

h     handle       Windows对象句柄       hWnd

lpfn  (\*fn)()       回调函数指针 Callback Far pointer to CALLBACK function       lpfnAbort

一般前缀命名规范

|  |  |  |
| --- | --- | --- |
| 前缀 | 类型 | 实例 |
| C | 类或结构 | CDocument，CPrintInfo |
| m\_ | 成员变量 | m\_pDoc，m\_nCustomers |

变量命名规范

|  |  |  |  |
| --- | --- | --- | --- |
| 前缀 | 类型 | 描述 | 实例 |
| ch | char | 8位字符 | chGrade |
| ch | TCHAR | 如果\_UNICODE定义，则为16位字符 | chName |
| b | BOOL | 布尔值 | bEnable |
| n | int | 整型（其大小依赖于操作系统） | nLength |
| n | UINT | 无符号值（其大小依赖于操作系统） | nHeight |
| w | WORD | 16位无符号值 | wPos |
| l | LONG | 32位有符号整型 | lOffset |
| dw | DWORD | 32位无符号整型 | dwRange |
| p | \* | 指针 | pDoc |
| lp | FAR\* | 远指针 | lpszName |
| lpsz | LPSTR | 32位字符串指针 | lpszName |
| lpsz | LPCSTR | 32位常量字符串指针 | lpszName |
| lpsz | LPCTSTR | 如果\_UNICODE定义，则为32位常量字符串指针 | lpszName |
| h | handle | Windows对象句柄 | hWnd |
| lpfn | callback | 指向CALLBACK函数的远指针 | ? |