

课外阅读 1：匈牙利命名法

黄永峰，2011-10-10

匈牙利命名法是一种编程时的命名规范。基本原则是：变量名 = 属性 + 类型 + 对象描述，其中每一对象的名称都要求有明确含义，可以取对象名字全称或名字的一部分。命名要基于容易记忆容易理解的原则。保证名字的连贯性是非常重要的。举例来说，表单的名称为 form，那么在匈牙利命名法中可以简写为 frm，则当表单变量名称为 Switchboard 时，变量全称应该为 frmSwitchboard。这样可以很容易从变量名看出 Switchboard 是一个表单，同样，如果此变量类型为标签，那么就应命名成 lblSwitchboard。可以看出，匈牙利命名法非常便于记忆，而且使变量名非常清晰易懂，这样，增强了代码的可读性，方便各程序员之间相互交流代码。

这种命名技术是由一位能干的 Microsoft 程序员查尔斯·西蒙尼(Charles Simonyi) 提出的，他出生在匈牙利。在 Microsoft 公司中和他一起工作的人被教会使用这种约定。这对他们来说一切都很正常。但对那些 Simonyi 领导的项目组之外的人来说却感到很奇特，他们认为这是死板的表达方式，甚至说带有这样奇怪的外观是因为它是用匈牙利文写的。从此这种命名方式就被叫做匈牙利命名法。

据说这种命名法是一位叫 Charles Simonyi 的匈牙利程序员发明的，后来他在微软呆了几年，于是这种命名法就通过微软的各种产品和文档资料向世界传播开了。现在，大部分程序员不管自己使用什么软件进行开发，或多或少都使用了这种命名法。这种命名法的出发点是把量名变按：属性+类型 +对象 描述的顺序组合起来，以使程序员作变量时对变量的类型和其它属性有直观的了解，下面是 HN 变量命名规范，其中也有一些是我个人的偏向：

1) 属性部分

全局变量 g_

常量 c_

c++类成员变量 m_

静态变量 s_

类型部分

指针 p

函数 fn

无效 v

句柄 h

长整型 l

布尔 b

浮点型（有时也指文件） f

双字 dw

字符串 sz

短整型 n

双精度浮点 d

计数 c（通常用 cnt）

字符 ch（通常用 c）

整型 i（通常用 n）

字节 by

字 w

实型 r

无符号 u

3) 描述部分

最大 Max

最小 Min

初始化 Init

临时变量

T（或 Temp）

源对象 Src

目的对象 Dest

这里顺便写几个例子：

hwnd: h 是类型描述，表示句柄，wnd 是变量对象描述，表示窗口，所以 hwnd 表示窗口句柄；

`pfnEatApple` : `pfn` 是类型描述, 表示指向函数的指针, `EatApple` 是变量对象描述, 所以它表示指向 `EatApple` 函数的函数指针变量。

`g_cch` : `g_`是属性描述, 表示全局变量, `c` 和 `ch` 分别是计数类型和字符类型, 一起表示变量类型, 这里忽略了对象描述, 所以它表示一个对字符进行计数的全局变量。

上面就是 HN 命名法的一般规则。