## 编程命名规范

2015-06-15

**一、匈牙利命名法说明:**

  匈牙利命名法关键是：标识符的名字以一个或者多个小写字母开头作为前缀；前缀之后的是首字母大写的一个单词或多个单词组合，该单词要指明变量的用途。

匈牙利命名法中常用的小写字母的前缀：

前　缀             类　　型  
a                数组 (Array)    
b               布尔值 (Boolean)    
by               字节 (Byte)    
c                有符号字符 (Char)    
cb               无符号字符 (Char Byte，没有多少用)    
cr               颜色参考值 (ColorRef)    
cx,cy             坐标差（长度 ShortInt）    
dw               Double Word    
fn               函数    
h                Handle（句柄）    
i                整型    
l                长整型 (Long Int)    
lp               Long Pointer    
m\_               类的成员    
n                短整型 (Short Int)    
np               Near Pointer    
p                Pointer    
s                字符串型

s\_ static类型

sz               以0结尾字符串 (String with Zero End)    
w                 Word

S 结构体

U 联合体

**二、骆驼命名法说明:**

  骆驼式命令法，正如它的名称所表示的那样，是指混合使用大小写字母来构成变量和函数的名字。

  在驼峰命名法中，一般比较流行的是变量用小驼峰法，函数、类、结构体、联合体等用大驼峰命名法，详见以下示例：

1. 小驼峰法：变量采用这种命名法，意思是除第一个单词外，其他单词首字母都大写，例

如：int myStudentCount;

Double\* pBuf;

1. 大驼峰法：类名、函数名、命名空间名采用这种命名法，意思是从第一个单词开始，首

字母都要大写，例如：public class DataBaseUser;

void SetAttribute(char\* pAttr);

**三、归总**

从上述对匈牙利命名法和驼峰命名法的重点看，主要区别在于：

1. 匈牙利命名法对变量或者函数的命名前缀以及单词组成都有对命名含义上的要求；
2. 驼峰命名法主要是对变量或者函数的开头的单词大小写做了严格要求；
3. 匈牙利命名法强调函数名以fn开头，而驼峰法习惯上函数名首字母大写。

结合helloX的代码看，变量基本都是以有意义的小写子母开头，往后没遇到一个单词则首字母大写，helloX函数命名则大多都已首单词首字母大写开头，这个和驼峰命名是一致的，整体看，helloX目前的现有的代码已经包含了这两种命名规范。

**因此这里在现有的基础上做一个统一的命名规范：**

1）、变量命名以匈牙利命名法为准：

变量（常量除外）以匈牙利命名法规定的前缀开头，其后每个有意义的单词首字母大写，结构体以大写S打头，结构体内变量遵循匈牙利变量命名法；联合体以大写U打头，联合体内部变量遵循匈牙利命名法。（这里边其实包含了小驼峰命名法，以小写字母开头，其后每到一个单词的开头首字母大写）。

示例：int iNum; char\* pBuf; struct SstudentMsg{short int nAge;};

Union UMoney{double dwMoney;inti Money;};

2）、函数命名以驼峰命名法为准：

函数名首字母大写，其后每个单词的开头首字母也都大写（这起哄包含了匈牙利命名法中对命名中每个单词应该有意义的要求）。

示例：void DeleteFile(string sPath);

3）、重载函数命名规范：

因为c语言不支持函数重载（也就是函数名不能相同），针对有些函数的功能扩展，建议在已经有的参照函数名的后边加上”Ex”后缀，如果第三次扩展则在原始函数名基础上后边加上”Exx”,根据实际情况，一个函数的功能扩展不会有太多次，这种方式后边的x应该不会重复几次，顾应该可以接受。

特别说明，对于内核开发和外部提供的API函数命名，可能会存在命名冲突的情况，而实际情况是，API函数应该通俗易懂，不易过多用下划线或者其它符号来区别内核命名，因此，建议在以后的内核函数命名中，按照传统unix/linux下c的命名方式，内核函数前默认可以以”\_”打头。

示例：void SetName(string sName);

void SetNameEx(string sName,string sEngName);

4）、宏定义、常量命名规范：

宏定义和常量都以有意义的单词的大写方式命名，其中宏定义所代表的表达式可以按照匈牙利变量命名方式命名，也可以全大写命名，这里怎么容易理解怎么给表达式命名即可。

示例：#define PI 3.14 #define ADD(a,b) (a)+(b)

const int IAGE = 10; const char\* PBUF = “helloX”;

5）、头文件条件编译命名规范：

沿用helloX现有命名规范

示例：#include “math.h”

则math.h头文件内格式：

#ifndef \_\_MATH\_H\_\_

#define \_\_MATH\_\_H\_\_

…

#endif

6）、文件名命名规范：

根据目前helloX文件名命名看，个别用了纯小写名称命名，大部分都是用的纯大写纯英文命名方式，为了统一，以后如果新增加文件，则统一以纯大写英文命名（数字命名除外）。