

函数、变量命名规则

零、通则：

1. 所有命名都应使用标准的英文单词或缩写，不得使用拼音或拼音缩写，除非该名字描述的是中文特有的内容，如半角、全角，声母、韵母等。
2. 所有命名都应遵循达意原则，即名称应含义清晰、明确。
3. 所有命名都不宜过长，应控制在规定的最大长度以内。
4. 命名中，每个单词的第一个字母应该大写，单词与单词之间直接连接，用大写字母加以区别。
5. 所有命名都应尽量使用全称，如果使用缩写，则应该使用《通用缩写表》（见附录）中的缩写。原则上不推荐使用《通用缩写表》以外的缩写，如果使用，则必须对其进行注释和说明。
6. 命名的长度应当符合“min-length && max-information”原则。一般来说，长名字能够更好地表达含义。单字符的名字也是有用的，常见如 i、j、k、n、x、y、z 等，它们通常可用作函数内的局部变量。

一、类（class）的命名：

类的定义以大写“CD”开头，例如“CAceData”；

类的对象以大写“O”开头，例如“OAceData”；

类在作为函数参数传递时，以小写“c”开头，例如“CAceData& cAceData”；

全局类的对象全部大写，例如“ORNG”；

二、函数的命名：

函数的命名必须符合：动词 [+ 名词] 的原则，类的成员函数也可以只使用“动词”，被省略掉的名词就是对象本身。

三、变量的命名：

变量的命名应该遵循即：[限定词+ ‘_’ +] 类型前缀+意义名词。最终的变量名总长不得超过 32 个英文字符。

变量限定词

限定词	说明	例子
无	局部变量	
m_	类的私有成员变量	int m_Width
p_	类的公有成员变量	int p_Wid
s_	静态变量	static int s_InitValue
g_	外部全局变量	int g_HowManyPeople
sg_	静态全局变量	

变量类型前缀

前缀	说明
b	Bool
ch	Char
s	String
v	vector
n	int, _int16, _int32, _int64

u	unsigned
l	long
ll	long long
ul	unsigned long
d	double
f	float
p	pointer
fp	FILE*
e	enumeration
st	struct
set	set
uni	union
by	BYTE
w	WORD
dw	DWORD

3.1 全局变量的命名

全局变量的限定词为“g”，所以全局变量必须以小写字母“g_”开始，例如 int g_nImageNumber 中“g”表示全局变量，“n”表示此变量为 int 型，“ImageNumber”表示此变量的意义。

3.2 局部变量的命名

局部变量不必要加限定词，即：类型前缀+意义名词。例如 int nImageNumber，其中“n”表示此变量为 int 型，“ImageNumber”表示此变量的意义。

3.3 类中的成员变量的命名

类中的成员变量命名的限定词为字母“m”和“p”，所以类中的成员变量命名必须以小写字母“m_”或“p_”开始。例如：int m_nImageNumber 中，“m”表示类中私有变量，“n”表示此变量为 int 型，“ImageNumber”表示此变量的意义。

3.4 静态变量的命名

对于类中的成员静态变量命名必须以小写字母“ms_”开始。例如：int ms_nImageNumber 中，“ms”表示类中成员静态变量，“n”表示此变量为 int 型，“ImageNumber”表示此变量的意义。

对于局部的静态变量命名必须以小写字母“s_”开始。例如 int s_nImageNumber，其中“s”表示静态变量，“n”表示此变量为 int 型，“ImageNumber”表示此变量的意义。

3.5 指针变量的命名

所有的指针的类型缩写前都必须加前缀“p”，如对于类中私有成员指针变量必须以小写字母“m_p”开始。例如 int *m_pnImageNumber 中，“m”表示类中成员变量，“pn”表示变量为 int *型，“ImageNumber”表示此变量的意义。

3.6 常量的命名

常量必须全部用大写字母，不需要加前缀。当常量由多个单词组成时，各单词由“_”

加以分割，并且用 `const` 来定义常量。例如：

```
const int MAX_NUMBER = 100;  const float PI = 3.14159;
```

在程序中，需要对外公开的常量放在头文件中，不需要对外公开的常量放在定义文件的头部。

3.7 参数的命名

参数的命名和局部变量的命名相同，即：类型缩写+意义名词。例如：`long GetImageNumber(int &nImageNumber)`之中，“n”表示参数为 `int` 型，“ImageNumber”表示参数的意义。

3.8 其他

3.8.1 枚举（enum）

枚举类型的命名中，单词与单词之间直接连接，用大写字母加以区别，并且加“e”作为前缀。枚举列表之的命名必须遵循敞亮的命名规则，全部用大写字母，当有多个单词组成时，各单词由“_”加以分割，例如：

```
enum eClockDirecter{CLOCKWISE = 1, ANTICLOCKWISE = -1};
```

3.8.2 联合（union）

联合类型的命名中，单词与单词之间直接连接，用大写字母加以区别，并且加“uni”作为前缀。联合包含的类型数据的命名必须遵循局部变量的命名规则，例如：

```
union uniValue // Declare union type
{
    char cType; // Declare member types
    int nNumber;
    float fValue;
}
```

3.8.2 结构体（struct）

结构类型的命名中，单词与单词之间直接连接，用大写字母加以区别，并且加“st”作为前缀。结构包含的类型数据的命名必须遵循局部变量的命名规则，例如：

```
struct stPerson // Declare struct type
{
    int nAge; // Declare member types
    float fWeight;
}
```

