C++命名规则

如果想要有效的管理一个稍微复杂一点的体系，针对其中事物的一套统一、带层次结构、清晰明了的命名准则就是必不可少而且非常好用的工具。

活跃在生物学、化学、军队、监狱、黑社会、恐怖组织等各个领域内的大量有识先辈们都曾经无数次地以实际行动证明了以上公理的正确性。除了上帝（设它可以改变世间万物的秩序）以外，相信没人有实力对它不屑一顾。

在软件开发这一高度抽象而且十分复杂的活动中，命名规则的重要性更显得尤为突出。一套定义良好并且完整的、在整个项目中统一使用的命名规范将大大提升源代码的可读性和软件的可维护性。

在引入细节之前，先说明一下命名规范的整体原则：

同一性	在编写一个子模块或派生类的时候，要遵循其基类或整体模块的命名风格，保持命名风格在整个模块中的同一性。
标识符组成	标识符采用英文单词或其组合，应当直观且可以拼读，可望文知意，用词应当准确。
最小化长度 && 最大化信息量原则	在保持一个标识符意思明确的同时，应当尽量缩短其长度。
避免过于相似	不要出现仅靠大小写区分的相似的标识符，例如“i”与“I”，“function”与“Function”等等。
避免在不同级别的作用域中重名	程序中不要出现名字完全相同的局部变量和全局变量，尽管两者的作用域不同而不会发生语法错误，但容易使人误解。
正确命名具有互斥意义的标识符	用正确的反义词组命名具有互斥意义的标识符， 如："nMinValue"和"nMaxValue"，"GetName()" 和 "SetName()" ....
避免名字中出现数字编号	尽量避免名字中出现数字编号，如 Value1,Value2 等，除非逻辑上的确需要编号。这是为了防止程序员偷懒，不肯为命名动脑筋而导致产生无意义的名字（因为用数字编号最省事）。

类/结构

除了异常类等个别情况（不希望用户把该类看作一个普通的、正常的类之情况）外，C++类/结构的命名应该遵循以下准则：

C++类/结构的命名	类的名称都要以大写字母“C”开头，后跟一个或多个单词。为便于界定，每个单词的首字母要大写。
推荐的组成形式	类的命名推荐用"名词"或"形容词 + 名词"的形式，例如："CAnalyzer", "CFastVector" ....

不同于 C++类的概念，传统的 C 结构体只是一种将一组数据捆绑在一起的方式。传统 C 结构体的命名规则为：

传统 C 结构体的命名	传统 C 结构体的名称全部由大写字母组成，单词间使用下划线界定，例如："SERVICE_STATUS", "DRIVER_INFO" ....

函数

函数的命名	函数的名称由一个或多个单词组成。为便于界定，每个单词的首字母要大写。
推荐的组成形式	函数名应当使用"动词"或者"动词 + 名词"（动宾词组）的形式。例如："GetName()", "SetValue()", "Erase()", "Reserve()" ....
保护成员函数	保护成员函数的开头应当加上一个下划线“_”以示区别，例如："_SetState()" ....
私有成员函数	类似地，私有成员函数的开头应当加上两个下划线“__”，例如："__DestroyImp()" ....
虚函数	虚函数习惯以“Do”开头，如："DoRefresh()", "_DoEncryption()" ....
回调和事件处理函数	回调和事件处理函数习惯以单词“On”开头。例如："_OnTimer()", "OnExit()" ....

变量

变量应该是程序中使用最多的标识符了，变量的命名规范可能是一套 C++命名准则中最重要的部分：

变量的命名	变量名由作用域前缀 + 类型前缀 + 一个或多个单词组成。为便于界定，每个单词的首字母要大写。 对于某些用途简单明了的局部变量，也可以使用简化的方式，如：i, j, k, x, y, z ....
作用域前缀	作用域前缀标明一个变量的可见范围。作用域可以有如下几种： **前缀　　说明** *无*　　局部变量 m_　　类的成员变量（member） sm_　　类的静态成员变量（static member） s_　　静态变量（static） g_　　外部全局变量（global） sg_　　静态全局变量（static global） gg_　　进程间共享的共享数据段全局变量（global global） 除非不得已，否则应该尽可能少使用全局变量。
类型前缀	类型前缀标明一个变量的类型，可以有如下几种：

	前缀 **说明** n 整型和位域变量（number） e 枚举型变量（enumeration） c 字符型变量（char） b 布尔型变量（bool） f 浮点型变量（float） p 指针型变量和迭代子（pointer） pfn 特别针对指向函数的指针变量和函数对象指针（pointer of function） g 数组（grid） i 类的实例（instance） 对于经常用到的类，也可以定义一些专门的前缀，如：std::string 和 std::wstring 类的前缀可以定义为"st"，std::vector 类的前缀可以定义为"v"等等。 类型前缀可以组合使用，例如"gc"表示字符数组，"ppn"表示指向整型的指针的指针等等。
推荐的组成形式	变量的名字应当使用"名词"或者"形容词 + 名词"。例如："nCode", "m_nState"，"nMaxWidth" ....

常量

C++中引入了对常量的支持，常量的命名规则如下：

常量的命名	常量名由类型前缀 + 全大写字母组成，单词间通过下划线来界定，如：cDELIMITER, nMAX_BUFFER .... 类型前缀的定义与变量命名规则中的相同。

枚举、联合、typedef

枚举、联合及 typedef 语句都是定义新类型的简单手段，它们的命名规则为：

枚举、联合、typedef 的命名	枚举、联合、typedef 语句生成的类型名由全大写字母组成，单词间通过下划线来界定，如：FAR_PROC, ERROR_TYPE ....

宏、枚举值

宏、枚举值的命名	宏和枚举值由全大写字母组成，单词间通过下划线来界定，如：

	ERROR_UNKNOWN, OP_STOP ....