# C++规范

**1 注释**

1.1 源文件头部应进行注释，列出：文件名、作者、创建日期、修改日期、函数功能、迭代版本等

1.2 函数头部应进行注释，列出：函数的目的/功能、输出/输出参数、返回值等

1.3 注释应该和代码同时更新

1.4 注释的内容要清晰、明了，不能有二义性

1.5 避免在注释之中使用非常用的缩写或者术语

1.6 注释应注明为什么做，而不是注明正在做什么

1.7 注释的版式应和其描述代码一致，进行相同的缩排

1.8 有实体意义的变量和常量、数据结构声明等，若其命名还不构充分表达其作用，则必须加以注释

1.9 重要变量的定义应该有较为详细的注释，包括其功能、取值范围、存取注意事项等

1.10 分支语句（条件分支、循环语句等）应进行注释

1.11 注释应适宜，有效注释量控制在20%~30%之间

**2 程序版式**

2.1 程序块需采用缩进风格编写，缩进空格数为4，尽量不用TAB键

2.2 相对独立的程序块之间、变量说明之后必须加空格或用空行分开，函数之间用空行分开

2.3 较长的语句要分为多行书写

2.4 不能将多个短语句写在同一行之中，即一行只写一条语句

2.5 if、for、do、while、case、switch、default等语句各自占一行，切if、for、do、while等语句的执行语句部分无论多少都要加{}

2.6 代码行之内应留有适当的空格，关键字后必须加空格，函数名后不要留空格,  如 Func(x)。如果不是一行的结束符号， 其后也要留空格，二元操作符前后加空格，一元操作符前后不加空格

2.7 程序块的分界符应各占一行并且位于同一列

**3 标识符命名**

3.1 命名应尽量使用英文单词，避免使用引起误解或意思模糊的缩写

3.2 命名规范必须与使用的系统风格一致，并且同一项目应该统一格式

3.3 C语言变量命名使用TypePrefix+Name，C++用m\_xxxx表示类的成员变量，用g\_xxxx表示全局变量

3.4 常量、宏和模板名采用全大写字母，每个单词之间用下划线“\_”分隔

3.5 命名中若使用了特殊约定或缩写，应在源文件开始处注明

3.6 变量命名禁止取单个字符，但做局部循环变量可以

3.7 函数名以大写开头，采用动-名结构，反映函数执行的操作和返回类型

3.8 类、结构、联合、枚举的命名分别以C、S、U、E开头

**4 可读性**

4.1 用括号明确表达式的操作顺序，避免使用默认优先级

4.2 建议不要编写过于复杂、多用途的复合表达式

4.3 涉及物理状态或含有物理意义的常量，应用有意义的枚举或常量来代替

4.4 禁止使用难以理解，易产生歧义的语句

**5 变量、结构**

5.1 建议少用全局变量，尽量去掉无必要的公共变量

5.2 变量被创建后应及时初始化

5.3 尽量减少数据类型转换

**6 函数、过程**

6.1 调用函数要检查所有的可能的返回情况，不应该的返回情况使用ASSERT来确认

6.2 函数规模尽量限制在100行以内

6.3 建议一个函数仅完成一个功能

6.4 类的构造函数，拷贝构造函数、析构函数和赋值函数应尽量自己写明，而不是使用系统内缺省的

6.5 谨慎使用与系统环境有密切关系的系统函数

**7 程序效率及质量保证**

7.1 在保证软件系统的正确性、稳定性、可读性及可测性的前提下，提高代码效率

7.2 循环体内的工作量应该最小化

7.3 模块中函数的划分和组织方式应进行分析、优化，提高程序效率

7.4 避免循环体内含有判断语句

7.5 尽量使用标准库函数

7.6 只引用属于自己的命名空间（C++）

7.7 函数/过程中动态分配的资源，在函数/过程退出之前要释放

7.8 对指针的操作应小心翼翼

7.9 尽量少用goto语句