**编程规范学习**

—— 李宣廷

一、基本原则

1. 维持代码的易读、易维护；

2. 保持代码的简明清晰；

3. 所有的代码尽量与原始基线一致；

4. 尽可能复用代码。

二、缩进

1.与原始基线缩进保持一致，源文件采用几个空格，缩进就用几个空格；源文件用TAB缩进，就用TAB缩进。

2.新添加的源文件，与使用基线的主流风格一致。

三、花括号

花括号有两种风格，一种是Windows风格，一种是Linux风格。一个文件中用同一种风格。

1.Windows风格的花括号另起一行。

2.Linux风格的左花括号跟在语句后面，右花括号另起一行。若接下来的语句是相同语句块的一部分，则该语句紧跟前面右花括号写。

3.不需要花括号的语句尽量不用花括号。

四、空格

1.空格放在关键字周围。

2.函数名和括号之间无空格。

3.参数前后不加空格，参数与参数之间要加空格。

4.二元或三元操作符前后加空格。

5.一元操作符前后不加空格。

五、每行代码长度

代码的长度尽量限制在80个字符以内，如果超过分两行或多行进行编写，后面的行与第一行保持一个TAB倍数的缩进，尽量与括号对齐。

六、文件命名

新添加的文件一律以nwy开头。

七、函数命名

1.函数一律采用小写字母和下划线组合的方式命名。

2.函数名意思简洁准确。

3.新添加的函数，函数名以nwy\_开头。

八、变量命名

1.变量一律采用小写字母和下划线组合的方式命名。

2.变量名意思简洁准确。

3.变量名一律不以nwy\_开头。

九、结构体命名

一律采用小写字母和下划线组合的方式命名。

十、变量使用规范

1.尽可能少用全局变量。

2.作用域为同一文件的全局变量，应在变量前加上static。

3.变量在使用前，应该初始化。

4.数组在每次使用前，检查是否应清空。

十一、注释

1.原则上要求全部使用英文，若源文件使用中文，则可也可以使用中文。

2. 代码注释非常重要，但注释必须按照正确的方式进行。一般情况下，我们应该描述我们的代码要做什么和为什么要做，而不是具体通过什么方式实现的。怎么实现由代码本身展现。此外，注释不应用包含谁写了哪个函数、修改日期和其他那些琐碎而无实际意义的内容。这些信息应该集中在文件最开头的地方。  
十二、增删改代码

1.增加代码

格式：

/\*Begin: Add + Author + Reason + Date\*/

code

/\*End: Add + Author + Reason + Date\*/

2.删除代码

格式：

/\*Begin: Delete + Author + Reason + Date\*/

/\*code\*/

/\*End: Delete + Author + Reason + Date\*/

3.修改代码

格式：

/\*Begin: Modify + Author + Reason + Date\*/

code

/\*End: Modify + Author + Reason + Date\*/

注：1. 不要叠加注释，只需要保留最新的注释即可。  
 2. 如果宏控添加了注释，则宏控所包含的代码是可以不用加注释的。

十三、宏控

1.所有的宏名必须为大写加下划线。

2.宏控中不要出现项目名称。

3.去掉宏之后，编译正常。

4.修改基线中的源代码尽量使用宏控进行修改。

若不添加宏控直接修改代码要满足一下2个条件：

1.修改对平台上的所有项目有效或对其它项目无影响。

2.不是新添加的功能。

十四、#ifdef、#if defined 与#if

1.当判断单个宏是否有定义时，#ifdef等价于#if defined

2.当判断比较复杂的宏，使用#if defined

3.当判断一个宏的值是否满足要求，用#if

十五、宏控缩进

所有宏定义都没有缩进，直接顶格开始写。