软件开发规范：编码规范

C#编码规范

目标：

1. 安全：代码完成所需的功能之余,不要产生负作用，即要稳定可靠。 2. 易读：类、实例、成员变量、成员函数的命名一目了然

3. 美观：尽量统一项目组内人员的编程风格。

第一部分：命名

1. 命名原则

1) 所有的函数（变量/类/文件名）应该代表其实际的作用，应该使用有意义的单词或

多个词组合，但不要使用人名、项目组名。

2) 所有的函数（变量/类名）一律使用英文。

3) 使用多个单词时不需要使用连线(如下划线)，但对于全部大写的宏需要使用连

线。

4) 多个词组合较长时，可以使用单词的缩写。

5) 不得使用非常相近的名字类表示几个不同含义的函数（变量/类）。

6) 命名时请考虑名字的唯一性和含义的准确性。

7) 使用项目组专用词汇来表达特定的含义(概念)，不得把专用词汇挪作他用。

2. 变量的命名

原则：使用匈牙利命名法命名变量

1）变量名一般由“类型修饰＋代表变量含意的英文单词或单词缩写”等部分组成。

类型修饰(小写字母)：

n: int ，l: LONG/long ， s: short ，u: UINT，f: float

b: bool ，by: BYTE，ch: char ， sz: char[],str: string

2）针对异常捕获过程中的 Exception 变量命名，在没有冲突的情况下，统一命名为

e；如果有冲突的情况下，可以重复 e ，比如：ee。 3. 函数的命名

1）使用动宾词组表达函数实际所作的事。

2）同名的函数(重载函数)在功能上应该完全相同，在参数上的差别也应一目了然。

3）不得出现名字非常相近但功能不同的函数. 如 CreatePage1(), CreatePage2() 等。

4. 类命名

1）名字应该能够标识事物的特性。

2）名字尽量不使用缩写，除非它是众所周知的。

3）名字可以有两个或三个单词组成，但通常不应多于三个。

4）在名字中，所有单词第一个字母大写，缩写都要大写。 5.控件命名规则 5）不要使用下划线字符（ _ ）。

1）控件命名=Web控件缩写前缀 + “_” +变量名

控件 Label TextBox Button ListBox DropDownList 等等

缩写 lb_XXX tb_XXX Btn_XXX Lb_XXX Drd_XXX XXXXX

6. 文件命名

1）文件起名要有实际意义。

2）源文件应尽量使用 8.3 格式，文件名只能包含字母、数字和下划线，不得使用其他的

字母。超长的文件名应使用缩写方式减少文件名的长度。建议使用如下的缩写的规

则（部分情况可以有例外，视具体情况而定）： ，缩写一般可以去掉元音字母以及不发音字母

，单词的首字母一般应该保留。

，当一个单词必须缩成一个字母时，应该选用最有代表性的字母，或首字母。

，多音节的单词可以去掉后面的音节而只保留前面的第一、二音节。较短的二音节单词

一般不缩写，如果必须缩写，可以只保留一个字母 ，多个单词组成文件名时，应该保留较重要的有意义的单词（或多留几个字母），其他

单词可以使用简写或只用首字母，去掉不必要的无意义的单词 ，可以使用一些谐音表示一个单词，如 2 表示 to，4 表示 for 等等，当单词数量少，字

母少的情况下，不需要缩写

，较短的单词一般不缩写，或缩写为一到二个字母

第二部分：代码格式书写规范

1. 基本格式

1）所有的缩进 TAB 键为 4 个空格，每个单词的首字符大写，其余小写。

2）在代码中垂直对齐左括号和右括号。例：

```
if(x==0)
{
                用户编号必须输入!
}
```

不允许以下情况：

if(x==0) {

用户编号必须输入！

或者：

用户编号必须输入！ 在大多数运算符之前和之后使用空格，这样做时不会改变代码的意图却可以使代码容易阅读。 例：int j = i + k; 而不应写为： int j=i+k; 4) 编写 SQL 语句时，对于关键字使用全部大写，对于数据库元素（如表、列和视图）使用大小写混合。

2. 注释的写法 5) 将每个主要的 SQL 子句放在不同的行上，这样更容易阅读和编辑语句。 1) 在你劳神的地方请加上详细的注释说明。除了最简单的存取成员变量的 Set_/Get_

成员函数之外，其余大部分的函数写上注释是良好的习惯。尽量使你的程序让别人

很容易看懂

2) 太多的注释会使程序很难看，但一些复杂的算法和数据结构处还是要加上注释的，

这样别人就容易看懂。否则时间长了，你自己都未必看明白了. 3) 如果是对某一段程序(算法/结构)的注释，在程序头直接用//再空一格来进行说明，

一行不要超过 80 字符

4) 为了防止在阅读代码时不得不左右滚动源代码编辑器，每行代码或注释在不得超过

一显示屏。

5) 使用连续的多个//表示注释行(不要超过 80 字符)

6）文件头部应有注释，简单描述文件的内容

7）对于程序中的比较关键的算法和函数，必须加注释

3．cs 文件的书写

1）各个部分应使用注释行和空行分割，并在必要的地方写上注释 2）函数之间用注释行和空行分割

3）重要的函数在函数头部加上注释

4）{ 和 }分别单独占用一行，且上下对齐，中间的内容缩进一个 TAB

第三部分：其他

1. 变量

1) float 和 bool 禁止用 判断。bool 应该用逻辑运算关系符，而 float 应该用差值区

间来判断“相等”

2）类型转换一律用显式类型转换

3）类型的长度一律用 sizeof() 获得

4）当声明一个变量时，务必要自己初始化一下变量

2. 函数

1）功能要单一，函数名要名符其实

2）要易懂，实现时不要过分追求技巧，优化放到后面去做 3）长度一般禁止超过 200 行

4）要检查输入值是否合法。实现(成员)函数时务必要求输入参数是在要求范围之内，

尤其你定义的(成员)函数给别人调用时，要判断其合法性。 5）调用函数时要严格按照接口规范调用，调用后要判断执行情况，并做适当的错误处

理（稍后会给出错误和异常处理规范）。

6）尽量避免整块复制代码段，如果出现这样的情况要分析原因，如果这段代码完成独

立的功能，应考虑使用函数，否则，应考虑使用宏定义。否则因为修改引起的不一

致往往是错误的根源。

7）除极其简单的函数外，其他的函数在入口处必须加上 FMAT_TRACE()，参见错误和异

常处理规范。

8）函数的出口尽量唯一，最好在出口处加上 FMAT_TRACE()

9）写代码时，尽量减少堆的分配次数，能使用 Stack 的尽量使用 Stack 10) 函数编写必须精炼，消除冗余的代码，删除不用的变量 11) if/while 等语句中的条件表达式的运算结果必须为显示的 BOOL量 12）禁止用 goto 语句