# C#程序编写规范

## 1. 命名规范

　　表达清晰是命名规范的核心，常见的命名分格有如下两种：

### 1.1 Pascal风格

　　　　包含一到多个单词，每一个单词第一个字母大写，其余字母小写。例如：HelloWorld

### 1.2 Camel风格

　　　　包含一个到多个单词，第一个单词首字母小写，其余单词首字母大写。例如：name, userPhone。

注：除了参数与变量之外，所有命名空间，类，函数，接口，属性等名称的命名，使用Pascal风格参数与变量的命名使用Camel风格

## 2.通用命名约定

　　约定的是如何选择最适当的名称

### 2.1 命名方式

　　　　1、选择容易读的英文名称，可读性比详细描述重要。例如：ScreenX 比 ScreenHorizontally更具有可读性

　　　　2、除了下画线之外，不要使用连字符或者其他非字母数字字符

　　　　3、避免使用与常用编程语言的关键字冲突的标识符

　　　　4、变量和方法参数使用Camel风格

　　　　5、不要使用成员变量属性作为成员变量的前缀（其他变量命名也一样）。例如：不要像Users m\_users; 这样定义成员变量，应该是 Users userModel;

### 2.2 字母缩写

　　　　通常不要使用缩写，除非这种缩写已被广泛接受，或者团队当中大家都认可一种缩写。

### 2.3 namespace命名

　　　　命名空间的命名采用Pascal风格，取名的一般规则为Zhangsan.ProjectName(人名.项目名称)，

　　　　需要用复数时，使用复数；例如使用System.Collections而不是System.Collection

　　　　需要缩写时，不需要加复数；例如使用System.IO而不是System.IOs

### 2.4 类，结构和接口命名

　　　　2.4.1 按照Pascal大小写格式，使用名词或者名词短语为类，接口和值命名

　　　　2.4.2 接口命名以字母I为前缀，例如，Icomponent

　　　　2.4.3 派生类的末尾使用基类的名称。例如从Stream继承的Framework类型以Stream结尾

## 3 代码编码规范

### 3.1 适当使用空行

适当的使用空行，可以增加代码的可读性。

#### 3.1.1 应该添加一行空行的情况：

　　　　　　　　1.方法之间

　　　　　　　　2.局部变量和它后面的语气之间

　　　　　　　　3.方法内的功能逻辑部分之间

#### 3.1.2 应该添加两行空行的情况：

　　　　　　　　1.同一文件的不同部分之间

　　　　　　　　2.在类，接口及彼此之间

### 3.2 避免使用大文件。

如果一个文件里的代码超过300~400行，必须考虑将代码分开到不同的类中

### 3.3 方法行数规范

避免一个方法的有效代码行数超过50行，如果超过应该考虑将其分解成不同的方法。

### 3.4方法的命名

一般将其命名为动宾短语。一个方法只完成一个任务。不要把多个任务组合到一个方法中，即使那些任务非常小。

### 3.5 避免使用很多成员变量，声明局部变量，并传递给方法

在方法内部可以定义变量，被称为局部变量；在类内部可以定义变量，被称为成员变量

### 3.6 不要在方法间共享成员变量

如果在几个方法间共享一个成员变量，那就很难知道哪个方法什么时候修改它的值。

### 3.7 布尔型变量或者方法一般可以用is、can、has或者should做前缀

如，isFinished, canWork等

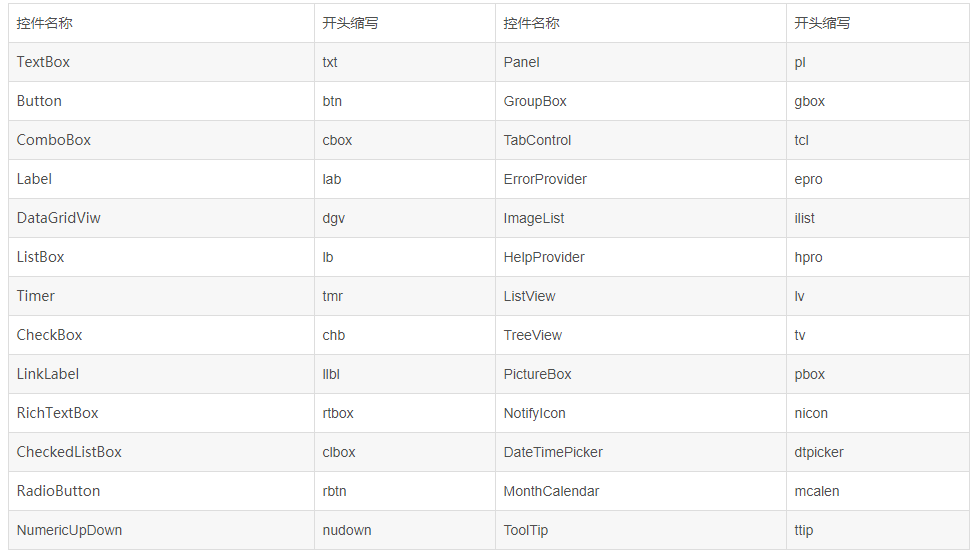
### 3.8 在类的顶部声明所有的成员变量，静态变量声明在最前面

## 4. 控件命名

　　c#控件命名往往采用首字母小写，其后的单词首字母大写的方式进行命名，而且往往以控件缩写作为前缀，

　　后面跟名字。举例:lblName(存放Name的lable), btnEdit(编辑button)

　　常见的控件缩写如下：



## 5. 注释

### 5.1 三斜杠(///)方式

　　　　一般用于类或者方法的前面

### 5.2 双斜杠(//)方式

　　　　一般是对临时变量，属性等的注释，当然也可以用在类或者方法前面，反正都是注释

### 5.3 块注释(/\*XXXX\*/)方式

　　　　一般用于一段连续的注释代码块