

前言

我们在实际的代码开发中，经常会遇到代码的命名以及规范问题。而不同的人有不同的风格，因此，这篇文章主要讲述本人 **C#代码的规范**，仅作个人以及愿意参与本人项目中人使用。

注意：大可不必争论，代码的命名是否合理，选择一套自己最喜欢的风格，并将其养成习惯，才是最重要的。

C#系列教程：

i

- 1 [C#基础语法](#) |  星空鸟  (yuilexi.cn)
- 2 [C#高级语法](#) |  星空鸟  (yuilexi.cn)
- 3 [C#代码规范](#) |  星空鸟  (yuilexi.cn)  当前的位置 °ᄇ'ᄇ'ᄇ'ᄇ'°

一、标识符规则

- **Pascal** 命名规范：标识符中，所有单词的首字母均大写，其余字母小写。
- **camel** 命名规范：标识符中，首字母小写，其余单词的首字母大写，其余字母小写。
- 标识符命名的字符选择：**[a-zA-Z0-9_]** 中的字符任意搭配，并且不能以数字作为开头

类型	命名规范	举例
类： class	Pascal	PlayerControl
非静态字段	camel ，并且必须以单下划线 _ 开头	_workerQueue
静态字段	camel ，并且必须以 s_ 开头	s_workerQueue
属性	Pascal ，不能以下划线或者 s_ 开头	WorkerQueue
方法	Pascal	PascalCasing()
局部变量	camel ，并且不能以下划线或者 s_ 开头	workerQueue
结构体： struct	Pascal	PascalCasing
接口 interface	Pascal ，并在开头加上 I	IWorkerQueue

C#语言命名空间

- 1 命名空间名称采用 **Pascal** 规则，且首字符大写。
- 2 命名空间名称尽量反映其内容所提供的整体功能。

对于引用的命名空间：

- 每一条命名空间的引用，独占一行
- 按照：**系统的命名空间** → **第三方库的命名空间** → **自定义的命名空间** 进行排序
- 每一大类的命名空间，按照字母排序

三、类

- 1 使用 **Pascal** 规则命名类名，即首字母要大写。
- 2 使用能够反映类功能的名词或名词短语命名类。
- 3 不要使用 “I”、“C”、“_” 等特定含义前缀。
- 4 自定义异常类应以 Exception 结尾。
- 5 文件名要能反映类的内容，最好是和类同名。

四、字段

- 1 用 **camel** 规则来命名类成员变量名称，即首单词（或单词缩写）小写。
- 2 类字段变量名前加 “_” 前缀，静态字段变量名前加 “s_” 前缀。
- 3 坚决禁止在普通变量前加“m_”（这是 VC 老命名规则）。

五、属性

- 1 使用名词定义属性
- 2 属性使用 **Pascal** 规则，首字符大写。
- 3 属性和相应字段名称要关联，可以使用“重构”菜单来生成属性。

六、方法

- 1 方法名采用 **Pascal** 规则，第一个字符要大写。
- 2 方法名应使用动词或动词短语。
- 3 类中访问修饰符或功能相同的方法应该放在一起
- 4 通常情况下，类中方法的排序为：实现接口的方法 -> 私有方法 -> 公共方法

七、参数（局部变量）

- 1 参数采用 **camel** 规则命名，且首字符小写。
- 2 使用描述性参数名称，参数名称应当具有足够的说明性。
- 3 不要给参数加匈牙利语类型表示法的前缀，即不能添加“_”、“s_”等前缀。
- 4 检查方法所有输入参数的有效性，即不能定义不使用的参数。

八、常量

- 1 只读常量使用 **Pascal** 命名规则，即首字母大写。
- 2 枚举名使用 **Pascal** 规则命名，枚举成员本质属于常量，命名规则同上。
- 3 枚举值从小到大顺序定义。
- 4 静态字段或属性采用 **Pascal** 规则，即首字符大写。

九、接口

- 1 接口定义使用 **Pascal** 规则，且必须以大写“T”开头。
- 2 接口名称要有意义，中间不要有下划线“_”等字符。
- 3 如果类实现了接口，名称尽量和接口相同，只是省掉“T”字符。

十、委托与事件

- 1 委托名称采用 **Pascal** 规则，即首字符大写。
- 2 定义事件的委托要使用 **EventHandler** 后缀，且包括 **sender** 和 **e** 两个参数。
- 3 事件用到的参数类，名称要带 **EventArgs** 后缀。
- 4 事件建议使用 **Event** 后缀

代码规范

- 1 代码块的大括号 `{ }` 独占一行，不要使用 `Java` 习惯
- 2 接口、类、结构体、枚举、字段、属性、方法等必须添加访问修饰符
- 3 不写注释是流氓，乱写注释是混蛋

README

✧ 更新日志

{% folding 更新日志 %}

{% timeline 更新日志,orange %}

- 1 在“前言”部分，增加对 ① ② 文章的引用链接
- 1 创建该文档，并规范了常用的代码

{% endtimeline %}

{% endfolding %}