COL3代码编写规范

Author: 高帆

Date: 2017/11/2

编写规范的目的是通过标准化代码风格，提高代码一致性和可读性，从而提高团队效率，并且降低后续维护成本。

本份文档编写原则（优先级从前到后）：

1. 尽量简化规则，不定义一些没有实际用处的规范。
2. 使用目前已存在代码中使用量最多的写法。
3. 使用C#官方的规范。
4. 使用带来更好效果的规范。

代码编写规范没有好和坏，希望大家不要因为个人的习惯排斥某种写法，应当共同严格遵守这份规范。

目录

[COL3代码编写规范 1](#_Toc497414008)

[1工具 2](#_Toc497414009)

[2代码注释 2](#_Toc497414010)

[2.1 类注释 2](#_Toc497414011)

[2.2 字段及属性注释 2](#_Toc497414012)

[2.3 方法注释 3](#_Toc497414013)

[2.4方法体内变量及逻辑注释 3](#_Toc497414014)

[3命名规范 4](#_Toc497414015)

[3.1命名空间 4](#_Toc497414016)

[3.2类名 4](#_Toc497414017)

[3.3公共属性及字段 4](#_Toc497414018)

[3.4私有属性 5](#_Toc497414019)

[3.5私有字段 5](#_Toc497414020)

[3.6公共静态属性及字段 5](#_Toc497414021)

[3.7私有静态属性 5](#_Toc497414022)

[3.8私有静态字段 6](#_Toc497414023)

[3.9公共、私有、静态公共、静态私有方法 6](#_Toc497414024)

[3.10事件处理方法 6](#_Toc497414025)

[3.11方法内参数及方法内变量 6](#_Toc497414026)

[3.12枚举 7](#_Toc497414027)

[4其它 7](#_Toc497414028)

[4.1代码书写区域划分 7](#_Toc497414029)

# 1工具

1. 代码编写或修改完成，请使用Ctrl+K+D快捷键， 使用VS自带的格式化工具格式化代码，它将统一代码中缩进和括号风格。

如果装了ReSharper，可以使用Ctrl+Alt+F来进行代码格式化。

1. 以下所写的部分书写规则可以被ReSharper自动检测出来，安装ReSharper来自动化的检查你所写的代码是否有问题。

插件下载更新地址：<https://10.0.3.2/svn/Zero/Program_Resource/Tools/ReSharper>。

# 2代码注释

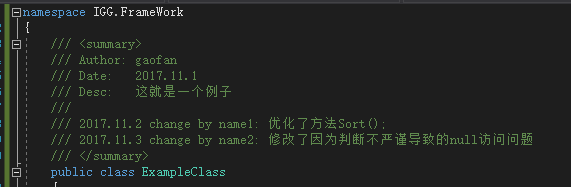
## 2.1 类注释

**规则：**

1. 注释写于类上方，而不是文件开头处。
2. 使用VS注释机制,输入”///”后生成的格式进行注释，注释中由上到下需要写明作者(Author)，创建时间(Date)，类的作用及使用方法(Desc)，如果后续有对类进行维护，跟据需要在最下方加入修改细节，修改细节包括时间，由谁修改，修改的内容。
3. 每个类都必需有这个注释，需要给外部调用的类必需有完整的注释。

某模块内部使用的类可以只有Author & Date 两个内容，具体格式见示例截图。

**示例：**

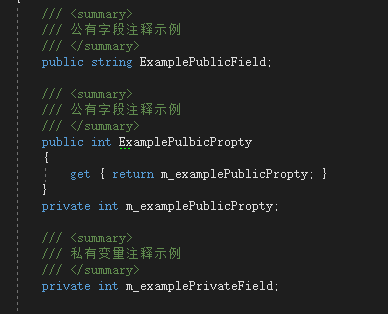


## 2.2 字段及属性注释

**规则：**

1. 使用VS注释机制,输入”///”后生成的格式进行注释.
2. 公共类、需要给外部调用的接口必需有注释，其它情况自行决定。

**示例：**

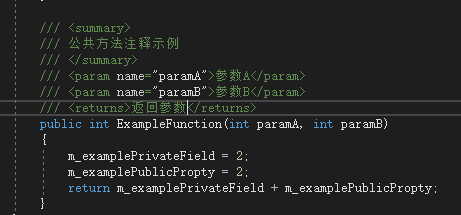


## 2.3 方法注释

**规则：**

1. 使用VS注释机制,输入”///”后生成的格式进行注释。
2. 需要写明方法的作用，参数的作用，返回值的作用。
3. 公共类、需要给外部调用的方法必需有注释，其它情况自行决定。

**示例：**



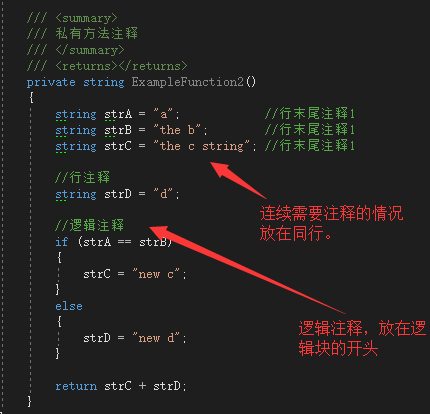
## 2.4方法体内变量及逻辑注释

**规则：**

1．方法体内的变量注释在有连续多行注释时可以放在变量声明语句的后面，这是唯一注释可以直接放在行后面的情况，与上下行变量声明的注释左对齐，注释与代码间以Tab隔开。如果只有单独的一个变量需要注释，则需单独一行，具体见示例。

2．逻辑注释从新行增加，与代码开始处左对齐

**示例：**



# 3命名规范

## 3.1命名空间

**规则：**以“IGG”开头，与Scripts下面的目录同命名空间

**示例：**

类“Assets/Scripts/FrameWork/Example.cs”的命名空间为”IGG.FrameWork”

类“Assets/Scripts/FrameWork/Test/Example.cs”的命名空间为”IGG.FrameWork.Test”

类”Assets/Scripts/Game/Test/Example.cs”的命名空间为”IGG.Game”

## 3.2类名

**规则：**首字母大写，使用帕斯卡命名法。

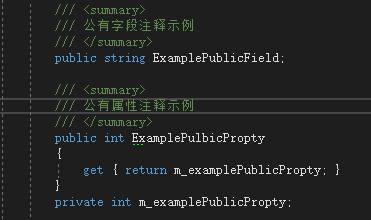
**示例：** public class ExampleClass

## 3.3公共属性及字段

**规则：**

1．首字母大写，使用帕斯卡命名法（同类名）。

**示例：**public string ExampleValue;



## 3.4私有属性

规则：同公共属性规则一样，不推荐使用，请用私有方法代替

## 3.5私有字段

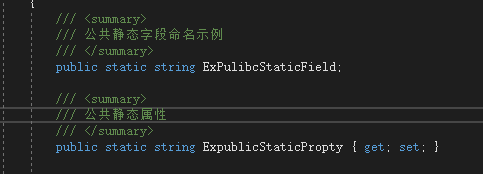
**规则：**m\_首字母小写，驼峰命名法

**示例：**private string m\_exampleValue;

## 3.6公共静态属性及字段

**规则：**首字母大写，使用帕斯卡命名法（同类名）。

**示例：**public static string ExampleValue;



## 3.7私有静态属性

**规则：**与公共静态属性相同，不推荐使用，请用私有静态方法代替

## 3.8私有静态字段

**规则：**g\_首字母小写，驼峰命名法

**示例：**private static string g\_exampleValue;

## 3.9公共、私有、静态公共、静态私有方法

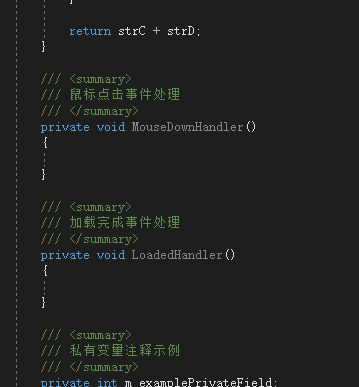
**规则：**首字母大写，使用帕斯卡命名法（同类名）。

**示例：**public string ExampleFunction()

## 3.10事件处理方法

**规则:** 事件处理的方法的命名方式为事件名+Handler, 如果事件名太长，可以适当缩写，以以表达出意义的最短命名方式为准。

**示例：**public void MouseDownHandler()



## 3.11方法内参数及方法内变量

**规则：**首字母小写，驼峰命名法

**例子：**

public string ExampleFunction(int param1, string param2)

{

string tempValue = “方面内变量”

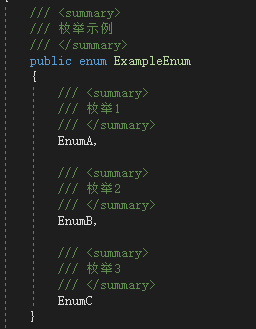
}

## 3.12枚举

**规则：**

1. 枚举名与类的命名规则一致。
2. 枚举成员变量名与类的命名规则一致。
3. 枚举的注释必需使用示例中的格式

**示例：**



# 4其它

## 4.1代码书写区域划分

|  |  |
| --- | --- |
| 区域名 | 示例 |
| 内部使用枚举定义  在最开始的地方 |  |
| 类定义 |  |
| 静态相关区域，从上到下有：  公共的静态字段  公共静态属性  公共静态方法。  私有静态字段  私有静态方法  这些夹在类开始，和类公共字段定义之间 |  |
| 公共字段  在静态相关定义，和构造函数之间 |  |
| 构造函数 |  |
| 公共属性 |  |
| 公共方法 |  |
| 私有方法 |  |
| 各种Handler |  |
| 私有属性 |  |
| 私有字段 |  |
| 类结束 |  |
| 额外类型定义，从上到下主要有  内部使用结构体定义  内部使用类定义 |  |

## 4.2花括号的使用

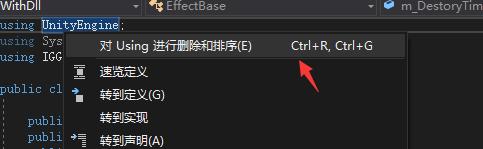
**规则：**所有可以使用{}的地方使全部都要使用{}，不能出现简写。

**示例：**

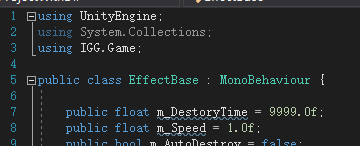
|  |  |
| --- | --- |
| 错误的写法 | 正确的写法 |
|  |  |

## 4.3整理导入

在编写或修改完代码后，请使用Ctrl+R, Ctrl +G对类的导入进行清理和排序, 但使用这个功能时，要注意有跟平台相关宏判断的代码。



在执行之前:



执行之后

