C#编码规范

# 命名规范

**Pascal法(大驼峰法) 所有单词的首字母大写**

**文件，代码统一采用Pacal命名法则**

## 文件

1.  文件名遵从Pascal命名法，扩展名小写。

2.  使用统一而又通用的文件扩展名： C# 类 .cs

## 类，接口,函数名,枚举,结构体

遵从Pascal命名法

|  |  |
| --- | --- |
| 类别 | 前缀 |
| class | 无 |
| interface | I |
| function | 无 |
| enum | 无 |
| 结构体 | st |
|  |  |

比如 要定义一个接口Resouece，则必须为 I+Resouece -> IResource

注：函数命名 一定要是 动词+xxx

如果是返回bool值判断类函数，则 可以采用 IsXXX, CanXXX, TryXXX,HasXXX

其余，则视函数功能而定

通用的动词有 Get Set Enter Exit Create Close Destroy Remove Add Clear Clean

## 变量

变量名称遵从Pascal命名法

类（结构体） 成员变量 m\_ + [类型缩写] + 变量名称

函数参数 [类型缩写] + 变量名称

[类型缩写] 常用类型才会类型缩写，自定义类型则可以不写

|  |  |
| --- | --- |
| 类型 | 缩写 |
| int | n |
| byte | by |
| short | s |
| long | l |
| float | f |
| double | d |
| string | str |
| char | c |
| list | lst |
| Dictionary | dic |
| Queue | que |
| Vector2 | vec |
| Vector3 | vec |
| Vector4 | vec |
| Quaternion | qua |
| Material | mat |
| Texture | tex |
| GameObject | go |
| Transform | ts |
|  |  |
|  |  |

静态变量： s\_+[类型缩写]+变量名称

const变量 使用全大写，单词间使用\_间隔, 比如private const int STUDENT\_COUNT=50;

函数局部变量: [类型缩写]+变量名称

For,foreach 循环内变量 可以采用通用缩写方式 比如 i,j,m,n等

以上所有名称的命名，必须保证通俗易懂，使用常用单词

## 常量

 所有单词大写，多个单词之间用 "\_" 隔开。 如

public const string PAGE\_TITLE = "Welcome";

## ****集合****

一组组合在一起的类似的类型化对象，如哈希表、查询、堆栈、字典和列表，集合的命名,建议用复数。

## ****属性**** ****(property)****

      以下规则概述属性的命名指南：   
            1 使用名词或名词短语命名属性。   
            2 使用 Pascal 大小写。   
 3 考虑用与属性的基础类型相同的名称创建属性。例如，如果声明名为 Color 的属性，则属性的类型同样应该是 Color。

## **事件**

以下规则概述事件的命名指南：

1. 对事件处理程序名称使用 EventHandler 后缀。
2. 指定两个名为 sender 和 e 的参数。sender 参数表示引发事件的对象。sender 参数 始终是object 类型的，即使在可以使用更为特定的类型时也如此。与事件相关联的状态封装在名为 e 的事件类的实例中。对 e 参数类型使用适当而特定的事件类。
3. 用 EventArgs 后缀命名事件参数类。
4. 考虑用动词命名事件。
5. 使用动名词（动词的“ing”形式）创建表示事件前的概念的事件名称，用过去式表示事件后。例如，可以取消的 Close 事件应当具有 Closing 事件和 Closed 事件。不要使用BeforeXxx/AfterXxx 命名模式。
6. 不要在类型的事件声明上使用前缀或者后缀。例如，使用 Close，而不要使用 OnClose。
7. 通常情况下，对于可以在派生类中重写的事件，应在类型上提供一个受保护的方法（称为 OnXXX）。此方法只应具有事件参数 e，因为发送方总是类型的实例。

# 注释

## 文件注释

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// 创建日期: 2015-7-4 12:05**

**// 文件名称: MainPanel.cs**

**// 创 建 人: Even**

**// 版权所有: 中青宝**

**// 说 明: 游戏主界面**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**在文件顶部，书写**

## **函数注释**

**/\*\***

**@brief 向颜色缓冲区写入数据**

**@param destPixels 目标缓冲区**

**@param srcPixels 源缓冲区**

**@paran nDestWidth nDestHeight 目标区域大小**

**@param nOffsetX nOffsetY 写入偏移**

**@param nSrcWidth nSrcHeight 写入大小**

**\*/**

**private bool WriteColorToTex(Color32[] destPixels, int nDestWidth, int nDestHeight, Color32[] srcPixels, int nOffsetX, int nOffsetY, int nSrcWidth, int nSrcHeight)**

**{**

**}**

**在函数顶部书写**

## **类\结构体注释**

**方法注释，同函数注释**

**成员变量,可以采用以下两种形式**

**// 光照贴图索引**

**private int m\_nLightMapIndex = 0;**

**public IRenderable m\_PartObj = null; // 挂接对象**

**类中的成员变量，都要写上注释，说明下变量用途**

## **枚举注释**

**// 实体类型**

**public enum EntityType**

**{**

**EntityType\_Null = 0, // 无效**

**EntityType\_Player = 1, // 玩家**

**EntityType\_Monster, // 怪物**

**EntityType\_NPC, // NPC**

**EntityType\_Item, // 物品**

**EntityType\_Pet, //宠物**

**EntityType\_Max,**

**}**

**如以上示例，要枚举的每一项进行说明**

## **代码注释**

**在函数体内，可以对单行代码或者代码进入注释说明**

# **代码排版**

## ****类\结构体****

**public class EntityConst**

**{**

**private int m\_nNum = 0; // 实体数量**

**/\*\***

**@brief 创建**

**\*/**

**public void Create()**

**{**

**}**

**}**

**花括号，必须换行开始**

**成员变量和方法，采用统一缩进**

**每个成员变量，要写注释说明用途**

## ****函数\类成员函数****

* **空行**

为了将逻辑上相关联的代码分块，以便提高代码的可阅读性

* **列宽**

**不要太长 以屏幕宽度为宜 现在大约200个字符**

**一行只允许存在一个语句**

* **换行**  当表达式超出或即将超出规定的列宽，遵循以下规则进行换行  
            1、在逗号后换行。  
            2、 在操作符前换行。  
            3、规则1优先于规则2。
* **缩进**     缩进应该是每行一个Tab(4个空格)，不要在代码中使用Tab字符。
* **空格** 在以下情况中要使用到空格

1. 关键字和左括符 “(” 应该用空格隔开。如while (true)

   注意在方法名和左括符 “(” 之间不要使用空格，这样有助于辨认代码中的方法 调用与关键字。

2、  多个参数用逗号隔开，每个逗号后都应加一个空格。  
3、  除了 . 之外，所有的二元操作符都应用空格与它们的操作数隔开。一元操作符、++ 及--与操作    数间不需要空格。如  
                 a += c + d;  
                 a = (a + b)   /   (c \* d);  
                while  (d++ = s++)  
                {  
                    n++;  
                }

4、 语句中的表达式之间用空格隔开。如  
       for (expr1; expr2; expr3)

* **括号 - ()**

   1、 左括号“(” 不要紧靠关键字，中间用一个空格隔开。  
      2、 左括号“(”  与方法名之间不要添加任何空格。  
      3、 没有必要的话不要在返回语句中使用()。如  
       if (condition)

{  
            Array.Remove(1)  
            return 1 ;

}

* **花括号 - {}**

1、 左花括号 “{” 放于关键字或方法名的下一行并与之对齐。如  
  
             if (condition)  
             {  
             }  
  
             public int Add(int x, int y)  
             {  
             }  
  
       2、 左花括号 “{” 要与相应的右花括号 “}”对齐。  
       3、 通常情况下左花括号 “{”单独成行，不与任何语句并列一行。  
       4、 if、while、do语句后一定要使用{}，即使{}号中为空或只有一条语句。如  
  
             if (somevalue == 1)  
             {  
                 somevalue = 2;  
             }  
  
5、 右花括号 “}” 后建议加一个注释以便于方便的找到与之相应的 {。如  
  
               while (1)  
               {  
                   if (valid)  
                   {  
                   } // if valid  
                   else  
                   {  
                   } // not valid  
                } // end forever  
  
6、所有的 if while switch 必须放在花括号里面 不允许出现 if() doThing();

* **位置**

 变量建议置于块的开始处，不要总是在第一次使用它们的地方做声明。如  
  
 void MyMethod()  
 {  
       int int1 = 0;         // beginning of method block

if (condition)  
 {  
                  int int2 = 0;     // beginning of "if" block  
                  ...  
       }  
 }

  不过也有一个例外  
          
                for (int i = 0; i < maxLoops; i++)  
                {  
                    ...  
                }

* **类和接口的声明**

1 在方法名与其后的左括号间没有任何空格。  
      2 左花括号 “{” 出现在声明的下行并与之对齐，单独成行。  
      3 方法间用//-------------------- 隔开

# **注意事项**

* **初始化**

在变量声明时就必须对其做初始化。

不论是 类中的成员变量还是函数中的局部变量，都必须初始化。     