[**C#编码规范**](http://www.cnblogs.com/wulinfeng/archive/2012/08/31/2664720.html)

**1     代码注释**

**1.1     模块头部注释规范**

/\*

//////////////////////////////////////////////////////////////////////////////////

\* COPYRIGHT (c) 2016

\* HONEYWELL INTERNATIONAL INC.

\* ALL RIGHTS RESERVED

\* Legal rights of Honeywell International Inc. in this software is distinct

\* from ownership of any medium in which the software is embodied. Copyright

\* notices must be reproduced in any copies authorized by Honeywell International Inc.

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

如何将头部注释添加到模板中参考文档： Visual Studio 自动添加头部注释.doc

**1.2     方法注释规范**

1. C# 提供一种机制，使程序员可以使用含有XML 文本的特殊注释语法为他们的代码编写文档。示例如下：

/// <summary>

/// Get the weather condition

/// </summary>

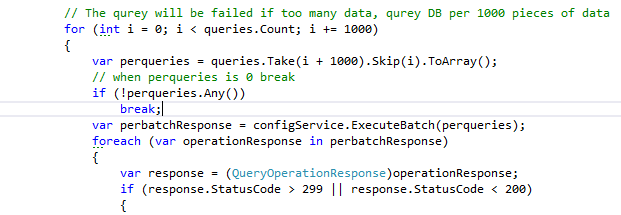
/// <param name="location">location</param>

/// <returns></returns>

1. 原则上方法注释不写，方法应是清晰的自我描述的，只有在public方法根据方法名无法理解应依据上面格式进行注释。（例如引用外部库）

**1.2     代码行注释规范**

1. 注释从新行增加，与代码开始处左对齐
2. 双斜线与注释之间以空格分开，示例图如下所示：



1. 原则上行注释不写，复杂逻辑应进行拆分

**2   命名规则**

**2.1     命名的基本约定**

1. 要使用可以准确说明变量/字段/类的完整的英文描述符，如firstName。对一些作用显而易见的变量可以采用简单的命名，如在循环里的递增（减）变量就可以被命名为 “i”。
2. 要尽量采用项目所涉及领域的术语。
3. 要采用大小写混合，提高名字的可读性。为区分一个标识符中的多个单词，把标识符中的每个单词的首字母大写。不采用下划线作分隔字符的写法。

　　　　有两种适合的书写方法，适应于不同类型的标识符：

**PasalCasing：**标识符的第一个单词的字母大写；

**camelCasing：**标识符的第一个单词的字母小写。

1. 下表描述了不同类型标识符的大小写规则：

|  |  |  |
| --- | --- | --- |
| **标识符** | **大小写** | **示例** |
| 命名空间 | Pascal | namespace Honeywell.Guomao.Platform.Dal.Weather.Heweather |
| 类型 | Pascal | public class DummyHeWeatherServiceProvider |
| 接口 | Pascal | public interface IWeatherServiceProvider |
| 方法 | Pascal | public void SetPoint(int siteId, PointValue presentValue) |
| 属性 | Pascal | public int SiteId { get; set; } |
| 事件 | Pascal | public event EventHandler Changed; |
| 私有字段 | Camel | private string \_alarmName; |
| 非私有字段 | Pascal | Public string FieldName; |
| 枚举值 | Pascal | FileMode { Append } |
| 参数 | Camel | public WeatherForecast GetWeatherCondition(Location location) |
| 局部变量 | Camel | string fieldName; |
| 类的常量命名 | Pascal | const string FieldName; |
| 静态只读 | Pascal | static readonly FieldName; |

1. 避免使用缩写，如果一定要使用，就谨慎使用。同时，参照标准缩写的列表，并且在使用时保持一致。
2. 对常见缩略词，两个字母的缩写要采用统一大小写的方式（示例：ioStream，   getIOStream）大小写应符合标准缩写列表；多字母缩写采用首字母大写，其他字母小写的方式（示例：getHtmlTag）；
3. 避免使用长名字（最好不超过 15 个字母）。
4. 避免使用相似或者仅在大小写上有区别的名字。

**2.2     各种标示符类型的命名约定**

1. **类和接口命名**

* 类的名字要用名词
* 一个文件一个类，类名和文件名相同
* 接口的名字要以字母I开头，接名词。保证对接口的标准实现名字只相差一个“I”前缀，例如对IComponent接口的标准实现为Component；
* 泛型类型参数的命名：命名要为T或者以T开头的描述性名字，例如：

　　　　public class List<T>

　　　　public class MyClass<Tsession>

* 对同一项目的不同命名空间中的类，命名避免重复。避免引用时的冲突和混淆；

1. **方法命名**

* 第一个单词一般是动词；
* 如果方法返回一个成员变量的值，方法名一般为Get+成员变量名，如若返回的值 是bool变量，一般以Is作为前缀。另外，如果必要，考虑用属性来替代方法；
* 如果方法修改一个成员变量的值，方法名一般为：Set + 成员变量名。同上，考虑 用属性来替代方法。

1. **变量命名**

* 按照使用范围来分，我们代码中的变量的基本上有以下几种类型，类的公有变量；类的私有变量（受保护同公有）；方法的参数变量；方法内部使用的局部变量。　　　　这些变量的命名规则基本相同，见标识符大小写对照表。区别如下：

　　　　a)   类的公有变量按通常的方式命名，无特殊要求；

　　　　c)   方法的参数变量采用camalString，例如workerName；

* 方法内部的局部变量采用camalString，例如workerName。
* 单字符的变量名一般只用于生命期非常短暂的变量：i,j,k,m,n一般用于integer；c,d,e 一般用于characters；s用于string
* 如果变量是集合，则变量名要用复数。例如表格的行数，命名应为：RowsCount；
* 命名组件要采用匈牙利命名法，所有前缀均应遵循同一个组件名称缩写列表缩写全部为小写。
* 基本类型定义应小写， 静态方法用大写:



1. **命名空间命名**

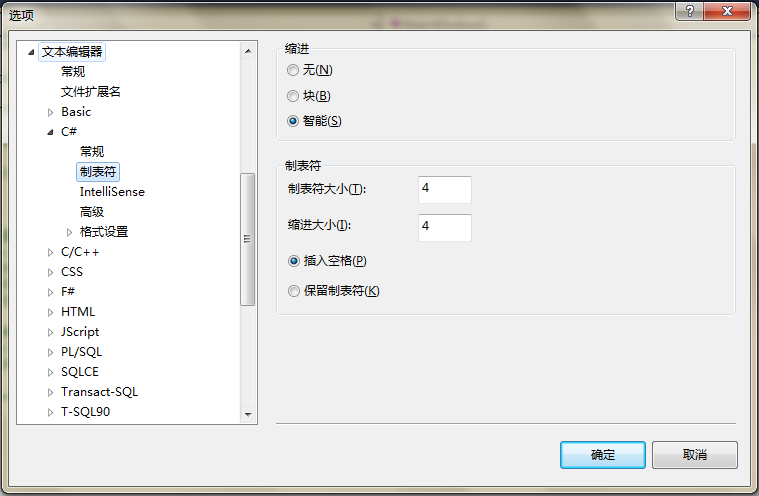
* 命名空间和目录结构相同

**3    其它规范**

**3.1      编程风格**

**代码缩进：**

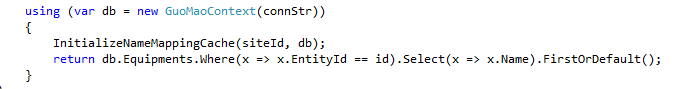
　　　使用VS默认设置，代码缩进使用Tab键实现，最好不要使用空格，为保证在不同机器上使代码缩进保持一致，特此规定C#的Tab键宽度为4个字符，设定界面如下(工具–选项)：



* 避免方法中有超过5个参数的情况，一般以2,3个为宜。如果超过了，则应使用struct来传递多个参数。
* 每行代码不要超过120个字符。
* 方法长度不宜大于50行
* 一个类中代码不超过500行
* 程序中不应使用goto语句。
* 在switch语句中总是要default子句来显示信息。
* 方法参数多于8个时采用结构体或类方式传递
* 操作符/运算符左右空一个半角空格
* 所有块的{}号分别放置一行，并嵌套对齐，不要放在同一行上

**3.2     资源释放**

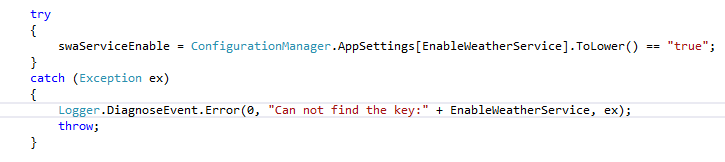
　　继承IDisposable接口的类必须使用using隐性释放。例如：



**3.3     错误处理**

1. 不要“捕捉了异常却什么也不做“，有异常应先log信息再抛出。
2. 只在WebApi实现层捕捉异常。
3. 只捕捉特定的异常，而不是一般的异常。

**正确做法：**



**3.4     其它**

1. 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。
2. 不要在程序中使用固定数值，用常量代替。
3. 避免使用很多成员变量。声明局部变量，并传递给方法。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，那就很难知道是哪个方法在什么时候修改了它的值。
4. 别把成员变量声明为 public 或 protected。都声明为 private 而使用 public/protected 的属性
5. 不在代码中使用具体的路径和驱动器名。 使用相对路径，并使路径可编程。
6. 应用程序启动时作些“自检”并确保所需文件和附件在指定的位置。必要时检查数据库连接。出现任何问题给用户一个友好的提示。
7. 如果需要的配置文件找不到，应用程序需能自己创建使用默认值的一份。
8. 如果在配置文件中发现错误值，应用程序要抛出错误，给出提示消息告诉用户正确值。
9. DataColumn取其列时要用字段名，不要用索引号。  
   　　　　例： 正确DataColumn[“Name”]  
       　　　　  不好 DataColumn[0]
10. 在一个类中，字段定义全部统一放在class的头部、所有方法或属性的前面。
11. 在一个类中，所有的字段定义，属性，构造函数，公有方法，私有方法全部定义在一个块中：



1. 在一个类中，属性快上下顺序：字段定义，属性，构造函数，公有方法，私有方法
2. Using 放在namespace 外面
3. 在定义类型并初始化时使用var

BPS标准缩写的列表

说明：

2、本缩写表中列出的都是通用性缩写，不提供标准缩写，如：Win9x、COM 等。

3、使用本缩写表里的缩写时，请对其进行必要的注释说明。

4、除少数情况以外，大部分缩写与大小写无关。

**BPS 缩写列表**

|  |  |  |
| --- | --- | --- |
| **简写** | **全名** | **标准命名举例** |
| ex | Exception | Exception ex |
| max | maximum | const int maxValue = 40; |
| min | minimum | const int minValue = 20; |
|  |  |  |

**ADO.NET命名规范**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **数据类型简写** | **标准命名举例** |
| Connection | con | conNorthwind |
| Command | cmd | cmdReturnProducts |
| Parameter | parm | parmProductID |
| DataAdapter | dad | dadProducts |
| DataReader | dtr | dtrProducts |
| DataSet | dst | dstNorthWind |
| DataTable | dtbl | dtblProduct |
| DataRow | drow | drowRow98 |
| DataColumn | dcol | dcolProductID |
| DataRelation | drel | drelMasterDetail |
| DataView | dvw | dvwFilteredProducts |

**WinForm Control 命名规范**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **数据类型简写** | **标准命名举例** |
| Label | lbl | lblMessage |
| LinkLabel | llbl | llblToday |
| Button | btn | btnSave |
| TextBox | txt | txtName |
| MainMenu | mmnu | mmnuFile |
| CheckBox | chk | chkStock |
| RadioButton | rbtn | rbtnSelected |
| GroupBox | gbx | gbxMain |
| PictureBox | pic | picImage |
| Panel | pnl | pnlBody |
| DataGrid | dgrd | dgrdView |
| ListBox | lst | lstProducts |
| CheckedListBox | clst | clstChecked |
| ComboBox | cbo | cboMenu |
| ListView | lvw | lvwBrowser |
| TreeView | tvw | tvwType |
| TabControl | tctl | tctlSelected |
| DateTimePicker | dtp | dtpStartDate |
| HscrollBar | hsb | hsbImage |
| VscrollBar | vsb | vsbImage |
| Timer | tmr | tmrCount |
| ImageList | ilst | ilstImage |
| ToolBar | tlb | tlbManage |
| StatusBar | stb | stbFootPrint |
| OpenFileDialog | odlg | odlgFile |
| SaveFileDialog | sdlg | sdlgSave |
| FoldBrowserDialog | fbdlg | fgdlgBrowser |
| FontDialog | fdlg | fdlgFoot |
| ColorDialog | cdlg | cdlgColor |
| PrintDialog | pdlg | pdlgPrint |

**WebControl 命名规范**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **数据类型简写** | **标准命名举例** |
| AdRotator | adrt | Example |
| Button | btn | btnSubmit |
| Calendar | cal | calMettingDates |
| CheckBox | chk | chkBlue |
| CheckBoxList | chkl | chklFavColors |
| CompareValidator | valc | valcValidAge |
| CustomValidator | valx | valxDBCheck |
| DataGrid | dgrd | dgrdTitles |
| DataList | dlst | dlstTitles |
| DropDownList | drop | dropCountries |
| HyperLink | lnk | lnkDetails |
| Image | img | imgAuntBetty |
| ImageButton | ibtn | ibtnSubmit |
| Label | lbl | lblResults |
| LinkButton | lbtn | lbtnSubmit |
| ListBox | lst | lstCountries |
| Panel | pnl | pnlForm2 |
| PlaceHolder | plh | plhFormContents |
| RadioButton | rad | radFemale |
| RadioButtonList | radl | radlGender |
| RangeValidator | valg | valgAge |
| RegularExpression | vale | valeEmail\_Validator |
| Repeater | rpt | rptQueryResults |
| RequiredFieldValidator | valr | valrFirstName |
| Table | tbl | tblCountryCodes |
| TableCell | tblc | tblcGermany |
| TableRow | tblr | tblrCountry |
| TextBox | txt | txtFirstName |
| ValidationSummary | vals | valsFormErrors |
| XML | xmlc | xmlcTransformResults |

**变量数据类型的前缀**

|  |  |  |
| --- | --- | --- |
| **C#数据类型** | **类库数据类型** | **标准命名举例** |
| Sbyte | System.sbyte | sbte |
| Short | System.Int16 | sht |
| Int | System.Int32 | int |
| Long | System.Int64 | lng |
| Byte | System.Byte | bte |
| Ushot | System.Uint16 | usht |
| Uint | System.Uint32 | uint |
| Ulong | System.Uint64 | ulng |
| Float | System.Single | flt |
| Double | System.Double | dbl |
| Decimal | System.Decimal | dcl |
| Bool | System.Boolean | bol |
| Char | System.Char | chr |
| Object | System.Object | obj |
| String | System.String | str |
|  | System.DateTime | dte |
| IntPtr | System.Intpre | intptr |