文章數(363) 回應數(1233) 引用數(0)

[C#]Code Convention Sample

2011/12/21 21:33 | 閱讀數 : 4746 | 1 人收藏 我要推薦 | 4 Comments | 文章分類 : Tips 專案開發 軟體工程 | 訂閱

讚 0

前言

通常在團隊開發時，我們會定義出coding standard與coding style，其中也包括了naming的原則等等，這一篇文章，就把這些統稱為Code Convention（規範）。

在這一篇文章，以C#這個程式語言為例，來訂出一個比較common的一些rule，供大家當參考。在應用到自己的團隊時，可以增修某些rule，來更適應自己團隊的習慣。但要先說明的是，會定義出這些rule，就代表著有一定的目的。

Naming (命名規範)

1. 只用Camel Case或Pascal Case，請參考這邊
2. 避免使用全大寫或全小寫的wording，除了單一個word可以全小寫
3. 範例
   1. Camel: gradeOfStudent
   2. Pascal: Customer
4. 好的命名方式請參考：[讀書心得]Clean Code - Meaningful Names

Coding Style（格式）

1. 一個檔案只包含一個namespace，只包含一個class，class名稱與檔名符合。
2. Project名稱與dll名稱符合。
3. 大括號的{與}各自獨立新的一行，判斷式都一定要加上{}。
4. 宣告變數時，每個變數都獨立一行。
5. Using namespace時，原生的namespace放最上面，最好都經過IDE的排序功能。
6. Visibility越高的，放越前面。外面看的到的，通常命名也都是Pascal。通常public放最上面，protected次之，internal次之，private放下面。建構式放上面，屬性放上面。
7. Interface的實作使用region區分。
8. Namespace的階層與folder的階層一致。
9. 每一個code block，有{}括起來的需按照block層級縮排。
10. 使用tab, 斷行等對code進行排版。
11. 宣告Attribute時，每個Attribute獨立一行。

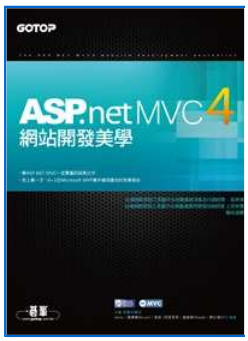
最高指導原則：

推薦這個blog：
95

Award

(ASP.NET 2010～2013年)

協作出版作品



1. *一致*
2. *清楚*

註解

1. 註解使用//或///，不用/*...*/，除非是版權宣告。
2. 禁止使用"flowbox"，範例如下：

```
// ************************************
// Comment block
// ************************************
```

3. inline comment只用來表示：假設、issues、演算法提示。
4. 好的程式碼應該從程式碼本身就能表現語意，而不是透過註解來說明程式碼的用途。
5. 善用TODO, UNDONE, HACK等工作清單的關鍵字，請參考這邊。
6. Public, protected, internal的部分，都應該使用///，來加上document註解。可透過工具來產生API文件。產生的部分，一定要包括<summary>，若有參數與回傳值，則要包括 <param>, <return>。工具的使用，請參考Sandcastle介紹。

Language Usage（變數與型別）

1. 宣告時要加上access modifier的宣告，而不使用默認的值。例如void MethodName()就是不好的，改用private void MethodName()。
2. 宣告access modifier時要謹慎，原則為：外面要使用此物件，能看到的東西最少，但又缺一不可。可見度範圍越小越好。
3. 宣告值的範圍，越小越好。可以用int就不用long，可以用double，就不用decimal。
4. 浮點數計算會有誤差，所謂『算錢用浮點，遲早被人扁』，需高精準度或算金額請用decimal。
5. 盡量不去修改Enum的預設型別(int)，除非不夠大，需要用到long。
6. 宣告成constants的應該是簡單的型別，複雜的型別應使用readonly或static readonly。
7. 用as轉型並檢查是否為null防呆，而避免使用強轉型。『用as轉型+判斷null，再使用轉型後的物件』，這樣的效率，比『用is判斷型別，再強轉型，再使用轉型後的物件』好。
8. 盡量使用強型別。
9. 避免boxing與unboxing的發生，會損耗內存記憶體。
10. 字串前使用@，來讓字串本身更容易瞭解，而不用脫逸字元。也可以讓字串可透過斷行來排版。
11. 善用String.Format來呈現字串的pattern。
12. 善用StringBuilder來連結動態的字串。
13. 判斷字串是否為空字串，避免用==string.Empty或==""，而使用.Length==0，或是string.IsNullOrEmpty。
14. 判斷字串是否相等，盡量透過string.Compare是否為0，而不是透過toUpper()與toLower()轉換後比較。

流程控制

1. 避免遞迴，而改用loop。（遞迴只應天上有，凡人應當用迴圈）
2. 在foreach中，不要去異動集合範圍。例如在loop中去remove enumerated items。
3. 避免在判斷式中，直接判斷某一個方法回傳值。
4. 只在很單純的情況下，使用條件運算子(三元運算子)，例如int result = isValid ? 9 : 4;
5. 不在判斷式裡面assign變數值，例如if((i=2)==2) {...}
6. 判斷式若為判斷某一個bool變數，則不需要再用==true或==false，因為bool值就代表該判斷式的意義。例如將if (isValid == true) {...}，改寫成if (isValid) {...}

其他資源

- 91's wiki
- 91's MSDN MVP 專欄
- 91's Curated Answers
- 91's slides
- .NET 分享會 on youtube
- 

文章標籤

AJAX AJAX control toolkit ASP.NET BizSpark forTest Gridview javascript MSDN SVN T-SQL VS2008 工作心得 自訂控制項 專案開發 實用小技巧

more tags...

每月文章

- 2013年9月 (5)
- 2013年8月 (1)
- 2013年5月 (3)
- 2013年3月 (1)
- 2013年2月 (2)
- 2013年1月 (21)
- 2012年12月 (9)
- 2012年11月 (7)
- 2012年10月 (1)
- 2012年9月 (4)
- 2012年7月 (9)
- 2012年6月 (7)
- 2012年5月 (9)
- 2012年4月 (8)
- 2012年3月 (2)
- 2012年1月 (5)
- 2011年12月 (20)
- 2011年11月 (7)
- 2011年10月 (1)
- 2011年9月 (5)
- 2011年8月 (2)
- 2011年7月 (4)
- 2011年6月 (4)
- 2011年5月 (6)
- 2011年4月 (1)
- 2011年3月 (5)
- 2011年1月 (4)
- 2010年12月 (7)
- 2010年11月 (7)
- 2010年10月 (24)
- 2010年9月 (9)
- 2010年8月 (6)

7. 判斷式組合太複雜時，請用多個bool來表示，請參考[ASP.NET]重構之路系列v7 –簡化判斷式
8. Switch case就只適合針對單一的變數進行比較，比較複雜的還是用if/else，但進階的設計方式，應該使用多型來取代switch case或多個判斷式，請參考[ASP.NET]重構之路系列v9 –使用介面+迴圈取代不穩定的判斷式

例外管理

1. 避免使用try/catch來當if/else用，try/catch應該要能明確捕捉特定的exception。嚴禁try/catch捕捉到錯誤後，完全沒處理。
2. 嚴禁在try/catch的catch block中，再使用try/catch
3. 避免re-throw exception
   錯誤的例子：

```
catch(Exception ex)
{
   Log(ex);
   throw ex;
}
```

   正確的例子：

```
catch(Exception ex)
{
   Log(ex);
   throw;
}
```

4. 可以用判斷式來決定流程，就不用try/catch。

Magic string / Magic number

1. 使用constant或Enum來避免magic string與magic number，可參考[ASP.NET]重構之路系列v8 –合併重複的條件片段
2. 使用Resources, Constants, Configuration Files, Registry 或其他資料來源，避免hard-code某一個字串。

結論

如同這個系列文的主題描述一樣，程式不是能動就好。誰都可以寫出讓機器懂的程式碼，但不是人人都能寫出人可以看懂的程式碼。訂出Code Convention，可以讓團隊降低開發成本、維護成本。透過工具來檢查程式碼，將可以更有效率的知道，系統是否有符合我們所定義的coding style。

關連文章

[測試]單元測試工具的選擇

[Tool]程式碼風格分析－StyleCop

[Security]Cross-Site Scripting(XSS)的簡介與預防

2010年7月 (4)
- 2010年6月 (5)
- 2010年5月 (3)
- 2010年4月 (6)
- 2010年3月 (3)
- 2010年2月 (3)
- 2010年1月 (8)
- 2009年12月 (17)
- 2009年11月 (5)
- 2009年10月 (15)
- 2009年9月 (8)
- 2009年8月 (12)
- 2009年7月 (8)
- 2009年6月 (5)
- 2009年5月 (1)
- 2009年4月 (8)
- 2009年3月 (9)
- 2009年2月 (8)
- 2009年1月 (16)
- 2008年12月 (10)
- 2008年11月 (1)
- 2008年10月 (2)

文章分類

- 30天快速上手TDD (33)
- AJAX (25)
- AJAX control toolkit (5)
- AOP (4)
- ASP.NET由淺入深系列 (30)
- BDD (17)
- Design pattern (25)
- GridView (16)
- Java Script (39)
- jQuery與plug-in (31)
- Linq (17)
- PowerShell (3)
- Scheme (1)
- Security (4)
- SignalR (1)
- Spring.Net (12)
- TDD (24)
- Tips (161)
- T-SQL (2)
- UML and SA (10)
- 專案開發 (60)
- 工作心得 (40)
- 工具介紹 (50)
- 快快樂樂學LINQ系列 (11)
- 測試 (37)
- 物件導向 training 系列 (4)
- 獨自murmur (23)
- 自訂控制項 (12)
- 讀書心得 (17)
- 軟體工程 (38)
- 重構之路系列 (31)

國內技術專家blog

- Bill叔
- demo小舖
- gipi
- The Will Will Web
- 小朱的技術空間
- 董大偉老師BLOG
- 黑暗執行緒

[Security]SQL injection的簡介與預防

回應

- # re: [C#]Code Convention Sample by sam

  好棒，受教了。

  2012/9/14 上午 11:58 | 回覆

- # re: [C#]Code Convention Sample by 小黑

  謝謝版主分享，這次又有收穫了；但是檢視你文中所述參考文章 [讀書心得]Clean Code - Meaningful Names

  發現裏頭似乎版面有跑掉，造成閱讀有些許不甚方便，回報給版主知道，謝謝

  2013/1/29 下午 09:28 | 回覆

- # re: [C#]Code Convention Sample by 91

  to 小黑：

  恩，這我知道。因為我在2012年調整過blog的css layout，太舊的文章，就無力回天了。

  有機會再來整理囉，感謝告知。

  2013/1/29 下午 10:11 | 回覆

- # re: [C#]Code Convention Sample by 小黑

  謝謝版主

  2013/6/25 下午 01:54 | 回覆

re: [C#]Code Convention Sample 標題 *

名稱 *

Email(將不會被顯示)

Url

記住我? 登入後使用進階評論

7QKM 請輸入

國外技術專家blog

- DotNetCurry
- jeffreypalermo.com
- Joel on Software
- John Resig
- Sara Ford
- Scott Gu
- Scott Hanselman
- Stephen Walther
- Visual Web Developer Team Blog

技術論壇

- CodePlex教學(中文)
- infoQ
- IT邦幫忙
- stackoverflow

簡體中文blog

- Jeffrey Zhou

最新回應

- **re: [ASP.NET]重構之路系列v6 –抽象來看程式是否符合DRY原則**
  to 91 : 瞭解，感謝 91 大的說明:)
- by Arvin Hsieh
- **re: [ASP.NET]重構之路系列v6 –抽象來看程式是否符合DRY原則**
  to Arvin Hsieh : 應用場景端或所謂的使用情境，簡體翻成上下文。其實就....
- by 91
- **re: [ASP.NET]重構之路系列v6 –抽象來看程式是否符合DRY原則**
  to 91 : 痾~另外問一下請問 Context 端指的是?
- by Arvin Hsieh
- **re: [ASP.NET]重構之路系列v6 –抽象來看程式是否符合DRY原則**
  to 91 : 這就是無招勝有招的境界了嗎，哈哈XD 感謝 91 大的提點，讓我有....
- by Arvin Hsieh
- **re: [.NET]快快樂樂學LINQ系列 - Where() 簡介**
  to Blackie1019 : 您客氣了，我覺得您的文章品質跟版面完全不比我差啊！期待您整個....
- by 91
- **re: [ASP.NET]重構之路系列v6 –抽象來看程式是否符合DRY原則**
  to Arvin Hsieh : 千萬不要把這些招式記起來啊，這跟太極拳一樣。要了解....
- by 91
- **re: [.NET]快快樂樂學LINQ系列 - Where() 簡介**
  你文章的內容都很棒～！！而且版面區塊好乾淨配置得很好 可以讓

驗證碼:

送出

我效法嗎～
- by Blackie1019
- **re: [ASP.NET]重構之路系列v6 –抽象來看程式是否符合DRY原則**
  請問 91 大，重構-改善既有程式的設計 我正在看，寫的也很好 但是有個疑問是，重構的手法這....
- by Arvin Hsieh
- **re: [.NET]快快樂樂學LINQ系列前哨戰 - 延遲執行 (Deferred Execution)**
  to 小雨 : 過獎了，只希望我整理出來的東西，可以幫助大家理解背後運作的機制。 感謝....
- by 91
- **re: [.NET]快快樂樂學LINQ系列前哨戰 - 延遲執行 (Deferred Execution)**
  對你的景仰不能更多了!
- by 小雨