C#コーディング規約

# 目次

## [コメント規約](#_コメント規約)

1. [クラス](#_クラス)
2. [メンバ関数](#_メンバ関数)
3. [メンバ変数](#_メンバ変数)

## [命名規則](#_命名規則)

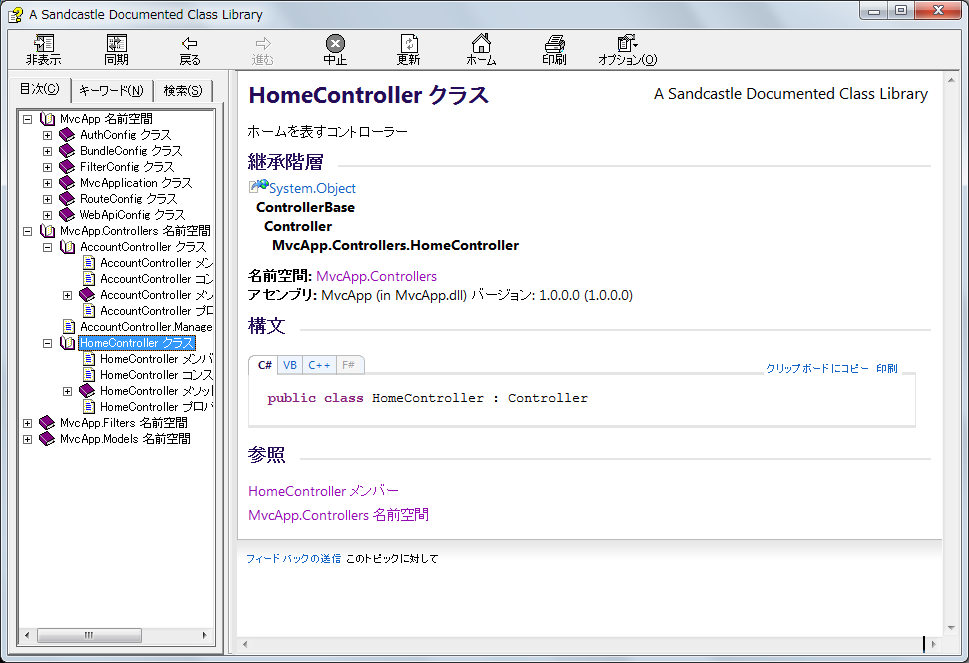
# コメント規約

<https://msdn.microsoft.com/ja-jp/library/5ast78ax.aspx>に準拠しています。

Visual Studioで///を入力すると自動でコメントのテンプレートを追加してくれます。

[Sandcastle](http://blog.nextscape.net/research/dotnetenv/sandcastle)をインストールすると自動的にドキュメント化してくれます。

例



## クラス

/// <summary>

/// 一行説明

/// </summary>

/// <remark>

/// 詳細の説明

/// <para>

/// 詳細の説明内で段落わけが必要ならば

/// </para>

/// </remark>

public class Hoge

{

public Hoge(){}

private bool\_isPiyo(){return true;}

private m\_isPiyo;

};

## メンバ関数

/// <summary>

/// 一行説明 (Doxygen の@brief )

/// </summary>

/// <param name="type">引数 type の説明 </param>

/// <param name="name">引数 name の説明 </param>

/// <returns>

/// 戻り値の説明

/// 複数の場合は

/// true : 成功

/// false : 失敗

/// といったように書く

/// </returns>

/// <remark>

/// 詳細の説明

/// <para>

/// 詳細の説明内で段落わけが必要ならば

/// </para>

/// </remark>

public bool DoSomething( int type, string name ){...}

## メンバ変数

/// <summary>

/// 一行説明

/// </summary>

/// <remark>

/// 詳細の説明

/// <para>

/// 詳細の説明内で段落わけが必要ならば

/// </para>

/// </remark>

private bool m\_isHoge

プロパティがある場合は続けて記述する。

private bool isHoge;

/// <summary>

/// 一行説明

/// </summary>

/// <remark>

/// 詳細の説明 (Doxygenの @par )

/// <para>

/// 詳細の説明内で段落わけが必要ならば

/// </para>

/// </remark>

public bool IsHoge { get {return isHoge;} }

# 命名規則

要素名に略称は使用しないでください。

例:Mgr→Managerなど

|  |  |  |
| --- | --- | --- |
| 要素 | 例 | 補足 |
| 定数 | ConstVariable |  |
| enum型 | PlayerCharacterState |  |
| enum要素 | Jump |  |
| クラス | PlayerCharacter |  |
| publicメソッド | DoMove() |  |
| protectedメソッド | DoJump() |  |
| privateメソッド | \_DoPunch | 先頭に”\_”を付ける |
| publicメンバ変数 | isJump | 原則publicメンバ変数は使わずにプロパティもしくはGetter/Setterを使用してください |
| protectedメンバ変数 | m\_movebule | 先頭に”m\_”を付ける |
| privateメンバ変数 | m\_jumpbule | 先頭に”m\_”を付ける |
| プロパティ | moveBule | 対応させるメンバ変数名と一致させる |
|  |  |  |