C#筆記

1. 關鍵字out: 引數會以傳址方式進入副程式

int initializeInMethod;

OutArgExample(out initializeInMethod);

Console.WriteLine(initializeInMethod); // value is now 44

void OutArgExample(out int number)

{

number = 44;

}

1. 方法中的擴展方法

扩展方法使您能够向现有类型“添加”方法，而无需创建新的派生类型、重新编译或以其他方式修改原始类型。扩展方法是一种特殊的静态方法，但可以像扩展类型上的实例方法一样进行调用。对于用 C# 和 Visual Basic 编写的客户端代码，调用扩展方法与调用在类型中实际定义的方法之间没有明显的差异。

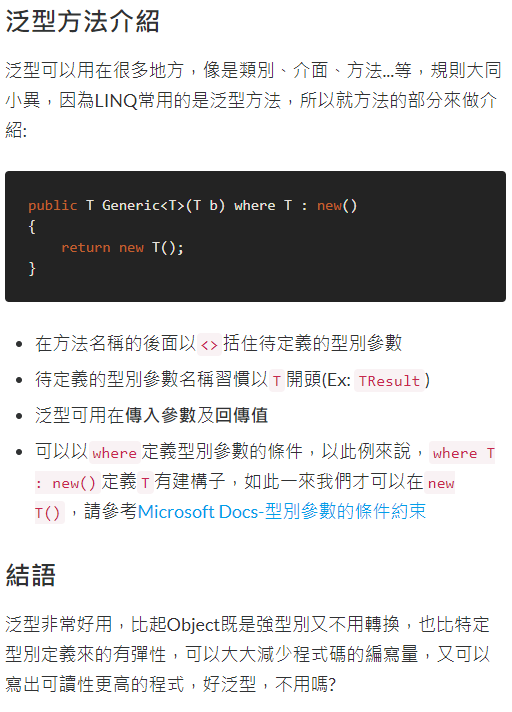
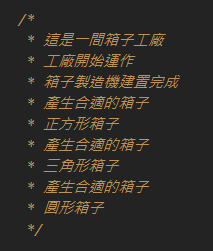
扩展方法被定义为静态方法，但它们是通过实例方法语法进行调用的。它们的第一个参数指定该方法作用于哪个类型，并且该参数以 this 修饰符为前缀。仅当您使用 using 指令将命名空间显式导入到源代码中之后，扩展方法才位于范围中。

下面的示例演示为 System.String 类定义的一个扩展方法。请注意，它是在非嵌套、非泛型静态类内部定义的：  
**namespace** **ExtensionMethods**  
**{**  
    **public** **static** **class** **MyExtensions**  
    **{**  
        **public** **static** int WordCount(**this** String str)  
        **{**  
            **return** str.Split(**new** char[] **{** ' ', '.', '?' **}**, StringSplitOptions.RemoveEmptyEntries).Length;  
        **}**  
    **}**     
**}**

　　可使用以下 using 指令将 WordCount 扩展方法放入范围中：  
　　using ExtensionMethods;  
　　而且，可以在应用程序中使用以下语法对该扩展方法进行调用：  
　　string s = "Hello Extension Methods";  
　　int i = s.WordCount();

1. 泛型用法範例 (<T>, return T())





1. Action、Fúnc、Delegate

委派的目的: 保留程式實作部分給外部程式。

1. Action: 封裝無回傳值的方法
2. Func: 封裝回傳一個值的方法
3. Func的最後一個T，永遠是返回的值，其餘的則是傳入的參數。範例: Func<T, TResult>
4. Lambda C#
5. 運算式: (input-parameters) => expression

Func<int, int> square = x => x \* x;

Console.WriteLine(square(5));

// Output:

// 25

int[] numbers = { 2, 3, 4, 5 };

var squaredNumbers = numbers.Select(x => x \* x);

Console.WriteLine(string.Join(" ", squaredNumbers));

1. 陳述式: (input-parameters) => { <sequence-of-statements> }

Action<string> greet = name =>

{

string greeting = $"Hello {name}!";

Console.WriteLine(greeting);

};

greet("World");

// Output:

// Hello World!

Action line = () => Console.WriteLine();

1. $: 字串插補

Console.WriteLine($"The low and high temperature on {weatherData.Date:MM-DD-YYYY}");

Console.WriteLine($" was {weatherData.LowTemp} and {weatherData.HighTemp}.");

// Output (similar to):

// The low and high temperature on 08-11-2020

// was 5 and 30.

1. virtual: 覆寫父類別中虛擬方法的方法需要利用 override 關鍵字，避免意外重複定義。
2. int? 表示該宣告變數可為null值
3. 匿名型別

var v = new { Amount = 108, Message = "Hello" };

// Rest the mouse pointer over v.Amount and v.Message in the following

// statement to verify that their inferred types are int and string.

Console.WriteLine(v.Amount + v.Message);

1. 只會傳回product的color與price屬性資料

var productQuery =

from prod in products

select new { prod.Color, prod.Price };

foreach (var v in productQuery)

{

Console.WriteLine("Color={0}, Price={1}", v.Color, v.Price);

}

1. 成員存取範圍

public：存取不受限制。

private：存取僅限於此類別。

protected：存取僅限於此類別或衍生自這個類別的類別。

internal：存取限於目前元件 (.exe 或 .dll) 。

protected internal：存取僅限於此類別、衍生自這個類別的類別，或相同元件中的類別。

private protected：存取權僅限於此類別或在相同元件中衍生自此類型的類別。