this关键字和索引器

参考：<http://blog.csdn.net/chensi16114/article/details/51894852>

一、this关键字

在C#中，this关键字有以下3种常见的用法：

　　1.用在类的属性、实例方法或实例构造方法中，区分成员名和本地变量（或参数）。下面的示例声明一个名为Myclass的类，类中包括一个实例字段myVal和一个实例构造函数，该构造函数带一个名为myVal的参数，在方法中，通过this可以在语义上区分成员名myVal和参数名myVal。[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

1 class MyClass

2 {

3 // 声明一个名为"myVal"的整型字段

4 public int myVal = 10;

5

6 // 声明一个构造函数，该函数带一个名为"myVal"的参数

7 public MyClass(int myVal)

8 {

9 // "this"表示MyClass类的当前实例

10 // 在这里通过this可以在语义上区分成员名myVal和参数名myVal

11 this.myVal += myVal;

12 }

13 }

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

　　2.this表示的是当前对象，可以把它作为实参传递到其它方法。例如下面声明一个类MyClass1,它包括一个实例成员方法Compute，而Compute带一个类型为MyClass的参数，该参数在方法中参加运算。

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

1 class MyClass1

2 {

3 // 声明一个计算的方法，参数类型为MyClass

4 public static int Compute(MyClass mc)

5 {

6 int resutl = 0;

7 if (mc != null)

8 {

9 resutl += mc.myVal;

10 }

11 return resutl;

12 }

13 }

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

　　下面在类MyClass的构造方法中添加了一个本地变量的声明，该变量初始化的值来自MyClass1的Compute方法的计算结果。而这时就可以将this作为实参传递给调用方法了：

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

1 class MyClass

2 {

3 // 声明一个名为"myVal"的整型字段

4 public int myVal = 10;

5

6 // 声明一个构造函数，该函数带一个名为"myVal"的参数

7 public MyClass(int myVal)

8 {

9 // "this"表示MyClass类的当前实例

10 // 在这里通过this可以在语义上区分成员名myVal和参数名myVal

11 this.myVal += myVal;

12

13 // 使用this作为当前对象的实参传递给Compute方法

14 int res = myVal + MyClass1.Compute(this);

15 }

16 }

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

[复制代码](http://blog.csdn.net/chensi16114/article/details/51894852)

　　3.声明索引器。接下来在第二部分当中分析索引器的声明和使用方法。

　　此外，应当注意的一点：由于静态成员函数存在于类一级，并且不是对象的一部分，没有 this 指针，因此不能在静态方法中使用this关键字。在静态方法中引用 this 是错误的。

二、索引器

　　1.通过点运算符访问实例成员

　　一般情况下，我们是通过点运算符（“实例名.成员名”）的方式来访问类的实例成员的。比如下面的示例，该例中其实是通过调用属性的set访问器为成员字段赋值，并且通过get访问器读取成员字段的值的（如果没有显示的添加get和set访问器则框架为我们自动添加）。

1     class Course  
2     {  
3 public float  Chinese { set; get; }  
4         public float Math { set; get; }  
5         public float Englisth { set; get; }   
6     }  
7   
8     class Program  
9     {  
10         static void Main(string[] args)  
11         {  
12 // 声明一个Course类的实例  
13             var course = new Course();  
14   
15             // 使用传统的“实例名.成员名”方式访问成员  
16             // 赋值  
17             course.Chinese = 95;  
18             course.Math = 100;  
19 course.Englisth = 80;  
20             // 取值  
21             Console.WriteLine("语文:{0}，数学:{1}，英语:{2}", course.Chinese, course.Math, course.Englisth);  
22             Console.ReadKey();  
23         }  
24     }

　　2.通过索引器访问实例成员

　　通过观察上面示例的代码，可以发现：该类的所有成员具有相同的类型float，它们其实可以像访问数组一样的方式访问，所以，我们可以为类提供另一种实例成员访问方式：索引器。下面的代码为上面的Course类声明加上了一个float类型的索引，并在Main方法中通过索引存取数据：

 1     class Course  
2     {  
3 public float  Chinese { set; get; }  
4         public float Math { set; get; }  
5         public float Englisth { set; get; }  
6   
7         // 声明一个公开的float类型的索引器  
8         public float this[int index]  
9         {  
10             // set访问器：赋值  
11             set  
12             {  
13                 switch (index)  
14                 {  
15                     case 0:  
16                         this.Chinese = value;  
17                         break;  
18                     case 1:  
19                         this.Math = value;  
20                         break;  
21                     case 2:  
22                         this.Englisth = value;  
23                         break;  
24                     default:  
25                         // 索引越界时抛出异常  
26                         throw new ArgumentOutOfRangeException();  
27                 }     
28             }  
29             // get访问器：取值  
30             get  
31             {  
32                 switch (index)  
33                 {  
34                     case 0:  
35                         return this.Chinese;  
36                     case 1:  
37                         return this.Math;  
38                     case 2:  
39                         return this.Englisth;  
40                     default:  
41                         throw new ArgumentOutOfRangeException();  
42                 }  
43             }  
44         }  
45     }  
46   
47     class Program  
48     {  
49         static void Main(string[] args)  
50         {  
51             // 声明一个Course类的实例  
52             var course = new Course();  
53   
54             // 使用索引器访问实例成员  
55             // 赋值  
56             course[0] = 95;  
57             course[1] = 100;  
58             course[2] = 80;  
59             // 取值  
60             Console.WriteLine("语文:{0}，数学:{1}，英语:{2}", course[0], course[1], course[2]);  
61             Console.ReadKey();  
62         }  
63     }

　　3.什么是索引？

 　　　　上面已经声明并且使用了索引，这里分析总结索引的概念和特征：

　　　　1).索引的概念

　　索引也是一种类成员，而且必须为实例成员，因为它就是为实例成员提供的一种访问方式，所以不能声明为static的。索引与属性类似，也有get访问器和set访问器。而且索引实际上就是一组get访问器和set访问器，而访问器的本质是方法，所以说，索引器的本质就是方法。索引器经常是在主要用于封装内部集合或数组的类型中声明的。

 　　　　2).索引声明

　　　　索引声明使用下面的语法：

　　　　[访问修饰符] 返回值类型 this [ 参数1，参数2...]

　　　　{

　　　　　　get{......}

　　　　　　set{......}

　　　　}

　　　　语法要点：索引名称永远为this；索引的参数在方括号中间；索引的参数列表中至少有一个参数。

　　　　3).set访问器

　　　　　　当索引被用于赋值时，它的set访问器被隐式调用， set访问器的特征如下:

　　　　　　第一，它的返回值类型为void。

　　　　　　第二，它的参数列表和索引中声明的参数列表的相同。

　　　　　　第三，它有一个名为value的隐式值参数，参数类型和索引的类型相同。

　　　　4).get访问器

　　　　　　当索引被用于取值时，get访问器被隐式调用，get访问器的特征如下:

　　　　　　第一，它的返回值类型与索引类型相同。

　　　　　　第二，它的参数列表和索引中声明的参数列表的相同。

　　　　5).索引的安全性

　　　　　在客户端代通过索引访问成员时，很可能发生索引越界等一些异常情况。有两种方式提高索引的安全性：

　　　　　　第一，为索引声明严格的访问级别控制，比如可以将set访问器的修饰符改为private。

1             // set访问器：赋值  
2             private set  
3             {  
4                 switch (index)  
5 { 6                     case 0:  
7                         this.Chinese = value;  
8                         break;  
9                     case 1:  
10                         this.Math = value;  
11                         break;  
12                     case 2:  
13                         this.Englisth = value;  
14                         break;  
15                     default:  
16                         // 索引越界时抛出异常  
17                         throw new ArgumentOutOfRangeException();  
18                 }  
19             }

　　　　　　第二，检查成员的个数，或抛出异常。

1 // 索引越界时抛出异常

2 throw new ArgumentOutOfRangeException();

　　　　6).索引重载

　　索引可以声明多个参数，也可以重载，索引重载只要参数列表不同就行了。下面为Course类声明几个索引的重载。

1     class Course  
2     {  
3 public float Chinese { set; get; }  
4         public float Math { set; get; }  
5         public float Englisth { set; get; }  
6   
7         // 声明一个公开的float类型的索引器  
8         public float this[int index]  
9         {  
10             // set访问器：赋值  
11             set  
12             {  
13                 switch (index)  
14                 {  
15                     case 0:  
16                         this.Chinese = value;  
17                         break;  
18                     case 1:  
19                         this.Math = value;  
20                         break;  
21                     case 2:  
22                         this.Englisth = value;  
23                         break;  
24                     default:  
25                         // 索引越界时抛出异常  
26                         throw new ArgumentOutOfRangeException();  
27                 }  
28             }  
29             // get访问器：取值  
30             get  
31             {  
32                 switch (index)  
33                 {  
34                     case 0:  
35                         return this.Chinese;  
36                     case 1:  
37                         return this.Math;  
38                     case 2:  
39                         return this.Englisth;  
40                     default:  
41                         throw new ArgumentOutOfRangeException();  
42                 }  
43             }  
44         }  
45   
46         // 索引重载  
47         public float this[string name, float val]  
48         {  
49             set  
50             {  
51                 switch (name)  
52                 {  
53                     case "Chinese":  
54                         this.Chinese = value + val;  
55                         break;  
56                     case "Math":  
57                         this.Math = value + val;  
58                         break;  
59                     case "English":  
60                         this.Englisth = value + val;  
61                         break;  
62                     default:  
63                         throw new ArgumentOutOfRangeException();  
64                 }  
65             }  
66             get  
67             {  
68                 switch (name)  
69                 {  
70                     case "Chinese":  
71                         return this.Chinese;  
72                     case "Math":  
73                         return this.Math;  
74                     case "English":  
75                         return this.Englisth;  
76                     default:  
77                         throw new ArgumentOutOfRangeException();  
78                 }  
79             }  
80         }  
81   
82         // 重载2：只读索引  
83         protected string this[int index, string name, bool flag]  
84         {  
85             set  
86             {  
87   
88             }  
89         }  
90   
91     }

三、索引器和属性的比较

　　1.相同点

　　　　1).索引和属性都不用分配内存位置来存储。

　　　　2).索引和属性都是为类的其它成员提供访问控制的。

　　　　3).索引和属性都有get访问器和set访问器，它们可以同时声明两个访问器，也可以只声明其中一个。

　　2.不同点

　　　　1).属性通常表示单独的数据成员，而索引表示多个数据成员。

　　　　2).属性既可以声明为实例属性，也可以声明为静态属性，而索引不能声明为静态的。

　　　　3).属性有简洁的自动实现属性，而索引必须声明完整。

　　　　4).get访问器：属性的 get 访问器没有参数，索引器的 get 访问器具有与索引器相同的形参表。

　　　　5).set访问器：属性的 set 访问器包含隐式 value 参数。除了值参数外，索引器的 set 访问器还具有与索引器相同的形参表。

参考：<http://msdn.microsoft.com/zh-cn/library/6x16t2tx(VS.100).aspx>