**索引器（C# 编程指南）**

索引器允许类或结构的实例就像数组一样进行索引。索引器类似于[属性](https://msdn.microsoft.com/zh-cn/library/x9fsa0sw.aspx)，不同之处在于它们的取值函数采用参数。

在下面的示例中，定义了一个泛型类，并为其提供了简单的 [get](https://msdn.microsoft.com/zh-cn/library/ms228503.aspx) 和 [set](https://msdn.microsoft.com/zh-cn/library/ms228368.aspx) 取值函数方法（作为分配和检索值的方法）。 Program 类创建了此类的一个实例，用于存储字符串。

class SampleCollection<T>

{

// Declare an array to store the data elements.

private T[] arr = new T[100];

// Define the indexer, which will allow client code

// to use [] notation on the class instance itself.

// (See line 2 of code in Main below.)

public T this[int i]

{

get

{

// This indexer is very simple, and just returns or sets

// the corresponding element from the internal array.

return arr[i];

}

set

{

arr[i] = value;

}

}

}

// This class shows how client code uses the indexer.

class Program

{

static void Main(string[] args)

{

// Declare an instance of the SampleCollection type.

SampleCollection<string> stringCollection = new SampleCollection<string>();

// Use [] notation on the type.

stringCollection[0] = "Hello, World";

System.Console.WriteLine(stringCollection[0]);

}

}

// Output:

// Hello, World.

| **https://wizardforcel.gitbooks.io/msdn-csharp/content/img/note.jpg 注意** |
| --- |
| 有关更多示例，请参阅[相关章节](https://wizardforcel.gitbooks.io/msdn-csharp/content/guide/82.html#BKMK_RelatedSections)。 |

**表达式主体定义**

直接只返回表达式结果的索引器很常见。下面的语法快捷方式使用 **=>** 来定义这些索引器：

public Customer this[long id] => store.LookupCustomer(id);

索引器必须为只读，并且你不能使用 **get** 取值函数关键字。

**索引器概述**

* 使用索引器可以用类似于数组的方式为对象建立索引。
* **get** 取值函数返回值。 **set** 取值函数分配值。
* [this](https://msdn.microsoft.com/zh-cn/library/dk1507sz.aspx) 关键字用于定义索引器。
* [value](https://msdn.microsoft.com/zh-cn/library/a1khb4f8.aspx) 关键字用于定义由 **set** 索引器分配的值。
* 索引器不必根据整数值进行索引；由你决定如何定义特定的查找机制。
* 索引器可被重载。
* 索引器可以有多个形参，例如当访问二维数组时。

**相关章节**

* [使用索引器（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/2549tw02.aspx)
* [接口中的索引器（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/tkyhsw31.aspx)
* [属性和索引器之间的比较（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/4bsztef7.aspx)
* [限制访问器可访问性（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/75e8y5dd.aspx)

**C# 语言规范**

有关详细信息，请参阅 [C# 语言规范](https://msdn.microsoft.com/zh-cn/library/ms228593.aspx)。该语言规范是 C# 语法和用法的权威资料。

**请参阅**

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

[属性（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/x9fsa0sw.aspx)

# 使用索引器（C# 编程指南）

索引器在语法上方便您创建客户端应用程序可将其作为数组访问的[类](https://msdn.microsoft.com/zh-CN/library/0b0thckt.aspx)、[结构](https://msdn.microsoft.com/zh-CN/library/ah19swz4.aspx)或[接口](https://msdn.microsoft.com/zh-CN/library/87d83y5b.aspx)。索引器经常是在主要用于封装内部集合或数组的类型中实现的。例如，假定具有一个名为 TempRecord 的类，此类表示在 24 小时内的 10 个不同时间记录的华氏度。此类包含一个表示温度的 float 类型的名为“temps”的数组和表示记录温度的日期的 [DateTime](https://msdn.microsoft.com/zh-CN/library/system.datetime.aspx)。通过在此类中实现一个索引器，客户端可以通过 float temp = tr[4] 而不是 float temp = tr.temps[4] 语法访问 TempRecord 实例中的温度。索引器表示法不仅简化了客户端应用程序的语法，还使其他开发人员能够更加直观地理解类及其用途。

要声明类或结构上的索引器，请使用 [this](https://msdn.microsoft.com/zh-CN/library/dk1507sz.aspx) 关键字，如下例所示：

public int this[int index] // Indexer declaration

{

// get and set accessors

}

## 备注

索引器类型及其参数类型必须至少如同索引器本身一样是可访问的。有关可访问级别的更多信息，请参见[访问修饰符](https://msdn.microsoft.com/zh-CN/library/wxh6fsc7.aspx)。

有关如何对接口使用索引器的更多信息，请参见[接口索引器](https://msdn.microsoft.com/zh-CN/library/tkyhsw31.aspx)。

索引器的签名由其形参的数量和类型组成。它不包括索引器类型或形参名。如果在同一类中声明一个以上的索引器，则它们必须具有不同的签名。

索引器值不属于变量；因此，不能将索引器值作为 [ref](https://msdn.microsoft.com/zh-CN/library/14akc2c7.aspx) 或 [out](https://msdn.microsoft.com/zh-CN/library/t3c3bfhx.aspx) 参数进行传递。

要为索引器提供一个其他语言可以使用的名字，请使用声明中的 name 特性。例如：

[System.Runtime.CompilerServices.IndexerName("TheItem")]

public int this [int index] // Indexer declaration

{

}

此索引器将具有名称 TheItem。不提供名称特性将生成 Item 默认名称。

## 示例 1

### 说明

下面的示例说明如何声明私有数组字段、temps 和索引器。使用索引器可直接访问实例 tempRecord[i]。另一种使用索引器的方法是将数组声明为 [public](https://msdn.microsoft.com/zh-CN/library/yzh058ae.aspx) 成员并直接访问它的成员 tempRecord.temps[i]。

请注意，当计算索引器的访问时（例如，在 **Console.Write** 语句中），将调用 [get](https://msdn.microsoft.com/zh-CN/library/ms228503.aspx) 访问器。因此，如果 **get** 访问器不存在，将发生编译时错误。

### 代码

class TempRecord

{

// Array of temperature values

private float[] temps = new float[10] { 56.2F, 56.7F, 56.5F, 56.9F, 58.8F,

61.3F, 65.9F, 62.1F, 59.2F, 57.5F };

// To enable client code to validate input

// when accessing your indexer.

public int Length

{

get { return temps.Length; }

}

// Indexer declaration.

// If index is out of range, the temps array will throw the exception.

public float this[int index]

{

get

{

return temps[index];

}

set

{

temps[index] = value;

}

}

}

class MainClass

{

static void Main()

{

TempRecord tempRecord = new TempRecord();

// Use the indexer's set accessor

tempRecord[3] = 58.3F;

tempRecord[5] = 60.1F;

// Use the indexer's get accessor

for (int i = 0; i < 10; i++)

{

System.Console.WriteLine("Element #{0} = {1}", i, tempRecord[i]);

}

// Keep the console window open in debug mode.

System.Console.WriteLine("Press any key to exit.");

System.Console.ReadKey();

}

}

/\* Output:

Element #0 = 56.2

Element #1 = 56.7

Element #2 = 56.5

Element #3 = 58.3

Element #4 = 58.8

Element #5 = 60.1

Element #6 = 65.9

Element #7 = 62.1

Element #8 = 59.2

Element #9 = 57.5

\*/

## 使用其他值进行索引

C# 并不将索引类型限制为整数。例如，对索引器使用字符串可能是有用的。通过搜索集合内的字符串并返回相应的值，可以实现此类索引器。由于访问器可被重载，字符串和整数版本可以共存。

## 示例 2

### 说明

在此例中，声明了存储星期几的类。声明了一个 **get** 访问器，它接受字符串（天名称），并返回相应的整数。例如，星期日将返回 0，星期一将返回 1，等等。

### 代码

// Using a string as an indexer value

class DayCollection

{

string[] days = { "Sun", "Mon", "Tues", "Wed", "Thurs", "Fri", "Sat" };

// This method finds the day or returns -1

private int GetDay(string testDay)

{

for (int j = 0; j < days.Length; j++)

{

if (days[j] == testDay)

{

return j;

}

}

throw new System.ArgumentOutOfRangeException(testDay, "testDay must be in the form \"Sun\", \"Mon\", etc");

}

// The get accessor returns an integer for a given string

public int this[string day]

{

get

{

return (GetDay(day));

}

}

}

class Program

{

static void Main(string[] args)

{

DayCollection week = new DayCollection();

System.Console.WriteLine(week["Fri"]);

// Raises ArgumentOutOfRangeException

System.Console.WriteLine(week["Made-up Day"]);

// Keep the console window open in debug mode.

System.Console.WriteLine("Press any key to exit.");

System.Console.ReadKey();

}

}

// Output: 5

## 可靠编程

提高索引器的安全性和可靠性有两种主要的方法：

* 确保结合某一类型的错误处理策略，以处理万一客户端代码传入无效索引值的情况。在本主题前面的第一个示例中，TempRecord 类提供了 Length 属性，使客户端代码能够在将输入传递给索引器之前对其进行验证。也可以将错误处理代码放入索引器自身内部。确保为用户记录在索引器的访问器中引发的任何异常。
* 应当为 **get** 和 [set](https://msdn.microsoft.com/zh-CN/library/ms228368.aspx) 访问器的可访问性设置尽可能多的限制。这一点对 **set** 访问器尤为重要。有关更多信息，请参见 [限制访问器可访问性（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/75e8y5dd.aspx)。

## 请参阅

[C# 编程指南](https://msdn.microsoft.com/zh-CN/library/67ef8sbd.aspx)

[索引器（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/6x16t2tx.aspx)

[属性（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/x9fsa0sw.aspx)

# 属性和索引器之间的比较（C# 编程指南）

索引器与属性类似。除下表中显示的差别外，为属性访问器定义的所有规则同样适用于索引器访问器。

| **属性** | **索引器** |
| --- | --- |
| 允许像调用公共数据成员一样调用方法。 | 允许对一个对象本身使用数组表示法来访问该对象内部集合中的元素。 |
| 可通过简单的名称进行访问。 | 可通过索引器进行访问。 |
| 可以为静态成员或实例成员。 | 必须为实例成员。 |
| 属性的 [get](https://msdn.microsoft.com/zh-cn/library/ms228503.aspx) 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。 |
| 属性的 [set](https://msdn.microsoft.com/zh-cn/library/ms228368.aspx) 访问器包含隐式 **value** 参数。 | 除了[值](https://msdn.microsoft.com/zh-cn/library/a1khb4f8.aspx)参数外，索引器的 **set** 访问器还具有与索引器相同的形参表。 |
| 支持对[自动实现的属性（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/bb384054.aspx)使用短语法。 | 不支持短语法。 |

## 请参阅

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

[索引器（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/6x16t2tx.aspx)

[属性（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/x9fsa0sw.aspx)