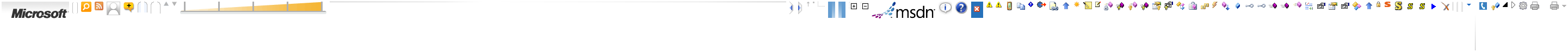
**Использование индексаторов (руководство по программированию в C#)**

**Visual Studio 2008**

[Другие версии](javascript:;)



Обновлен: Ноябрь 2007

Индексаторы являются синтаксическим удобством, позволяющим создавать [класс](http://msdn.microsoft.com/ru-ru/library/0b0thckt(v=vs.90).aspx), [структуру](http://msdn.microsoft.com/ru-ru/library/ah19swz4(v=vs.90).aspx) или [интерфейс](http://msdn.microsoft.com/ru-ru/library/87d83y5b(v=vs.90).aspx), доступ к которому клиентские приложения получают, как к массиву. Чаще всего индексаторы реализуются в типах, главная цель которых — инкапсуляция внутренней коллекции или массива. Например, предположим, что имеется класс с именем "TempRecord", представляющий набор температур по шкале Фаренгейта, полученных в 10 различных моментов в течение 24 часов. Класс содержит массив с именем "temps" типа "float", представляющий температуры, и [DateTime](http://msdn.microsoft.com/ru-ru/library/system.datetime(v=vs.90).aspx), представляющий дату регистрации температур. Путем внедрения в этот класс индексатора клиенты получат доступ к температурам в экземпляре TempRecord с помощью float temp = tr[4], а не float temp = tr.temps[4]. Использование индексатора не только упрощает синтаксис для клиентских приложений, но и делает класс и его назначение интуитивно понятными для других разработчиков.

Чтобы объявить индексатор для класса или структуры, используйте ключевое слово [this](http://msdn.microsoft.com/ru-ru/library/dk1507sz(v=vs.90).aspx) как показано в следующем примере:

**other**

public int this[int index] // Indexer declaration

{

// get and set accessors

}

[[http://i.msdn.microsoft.com/Hash/030c41d9079671d09a62d8e2c1db6973.gif](javascript:void(0))**Заметки**](javascript:void(0))

Тип индексатора и типы его параметров должны иметь по крайней мере такой же уровень доступности, как и сам индексатор. Дополнительные сведения об уровнях доступности см. в разделе [Модификаторы доступа](http://msdn.microsoft.com/ru-ru/library/wxh6fsc7(v=vs.90).aspx).

Дополнительные сведения об использовании индексаторов с интерфейсом см. в разделе [Индексаторы интерфейсов](http://msdn.microsoft.com/ru-ru/library/tkyhsw31(v=vs.90).aspx).

Подпись индексатора состоит из количества и типов его формальных параметров. В подпись не включается тип индексатора или имена формальных параметров. Если в одном классе объявлено несколько индексаторов, у них должны быть различные подписи.

Значение индексатора не классифицируется как переменная, поэтому не допускается передача значения индексатора как параметра [ref](http://msdn.microsoft.com/ru-ru/library/14akc2c7(v=vs.90).aspx) или [out](http://msdn.microsoft.com/ru-ru/library/t3c3bfhx(v=vs.90).aspx).

Чтобы предоставить индексатору имя, которое можно использовать в других языках, используйте в объявлении атрибут name. Пример.

**other**

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

public int this [int index] // Indexer declaration

{

}

Этот индексатор будет иметь имя TheItem. Если атрибут имени не предоставлен, используется имя по умолчанию Item.

[[http://i.msdn.microsoft.com/Hash/030c41d9079671d09a62d8e2c1db6973.gif](javascript:void(0))**Пример 1**](javascript:void(0))

**Описание**

В следующем примере показано, как объявить закрытое поле массива temps и индексатор. Индексатор обеспечивает прямой доступ к экземпляру tempRecord[i]. В качестве альтернативы применению индексатора можно объявить массив как член типа [public](http://msdn.microsoft.com/ru-ru/library/yzh058ae(v=vs.90).aspx) осуществлять прямой доступ к его членам tempRecord.temps[i].

Обратите внимание, что при вычислении доступа индексатора, например, в инструкции Console.Write вызывается метод доступа [get](http://msdn.microsoft.com/ru-ru/library/ms228503(v=vs.90).aspx). Таким образом, если не существует метода доступа get, происходит ошибка времени компиляции.

**Код**

**C#**

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();

}

}

[[http://i.msdn.microsoft.com/Hash/030c41d9079671d09a62d8e2c1db6973.gif](javascript:void(0))**Индексирование с использованием других значений**](javascript:void(0))

C# не ограничивает тип индексатора типом "integer". Например, может оказаться полезным использовании в индексаторе строки. Такой индексатор можно реализовать, выполнив поиск строки в коллекции и возвратив соответствующее значением. Методы доступа можно перегружать, версии типа "string" и "integer" могут сосуществовать.

[[http://i.msdn.microsoft.com/Hash/030c41d9079671d09a62d8e2c1db6973.gif](javascript:void(0))**Пример 2**](javascript:void(0))

**Описание**

В этом примере объявляется класс, в котором хранятся дни недели. Объявляется метод доступа get, который принимает строку (название дня недели) и возвращает соответствующее целое число. Например, воскресенье возвращает 0, понедельник возвращает 1 и т. д.

**Код**

**C#**

// 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 - 1; 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

[[http://i.msdn.microsoft.com/Hash/030c41d9079671d09a62d8e2c1db6973.gif](javascript:void(0))**Надежное программирование**](javascript:void(0))

Существуют два основных способа повышения надежности и безопасности индексаторов.

* Внедрите стратегию обработки ошибок на тот случай, если код клиента передаст недопустимое значение индекса. В первом примере, приведенном ранее в этом разделе, класс "TempRecord" предоставляет свойство "Length", позволяющее коду клиента проверить введенные данные перед тем, как передать их индексатору. Также можно поместить код обработки ошибки в индексатор. Задокументируйте для пользователей любые исключения, которые были созданы внутри метода доступа индексатора. Дополнительные сведения см. в разделе [Правила разработки исключений](http://msdn.microsoft.com/ru-ru/library/ms229014(v=vs.90).aspx).
* Установите максимальное обоснованное ограничение доступности для методов доступа get и [set](http://msdn.microsoft.com/ru-ru/library/ms228368(v=vs.90).aspx). Это особенно важно для метода доступа set. Дополнительные сведения см. в разделе [Асимметричные методы доступа (Руководство по программированию в C#)](http://msdn.microsoft.com/ru-ru/library/75e8y5dd(v=vs.90).aspx).

[[http://i.msdn.microsoft.com/Hash/030c41d9079671d09a62d8e2c1db6973.gif](javascript:void(0))**См. также**](javascript:void(0))

**Задачи**

[Пример индексаторов](http://msdn.microsoft.com/ru-ru/library/b75y0xk8(v=vs.90).aspx)

**Основные понятия**

[Руководство по программированию в C#](http://msdn.microsoft.com/ru-ru/library/67ef8sbd(v=vs.90).aspx)

**Ссылки**

[Индексаторы (руководство по программированию в C#)](http://msdn.microsoft.com/ru-ru/library/6x16t2tx(v=vs.90).aspx)

[Свойства (руководство по программированию в C#)](http://msdn.microsoft.com/ru-ru/library/x9fsa0sw(v=vs.90).aspx)