

Indexer in C

Indexer in C # helps index objects, such as an array. When you define an indexer for a class, this class operates similarly to a virtual array. Then you can access the instance of this class with the array access operator in C # ([]).

The **ndexer** in C # helps index objects, such as an array. When you define an indexer for a class, this class operates similarly to a **virtual array** . Then you can access the instance of this class with the array access operator in C # ([]).

Indexer syntax

In C #, one-way Indexer has the following syntax:

```
ki ? u_ph ? n_t ? this [ int index ] { get { // tr? v? giá tr? ???c xác
??nh b?i index } set { // thi?t l?p giá tr? ???c xác ??nh b?i index } }
```

Use Indexer in C

Indexer behavior declaration is similar to attribute declaration. You use **get** and **set** to define an Indexer. However, the properties return or set a specific data member, but, the Indexer returns or sets a specific value from the instance of the object. In other words, it divides instance data into smaller parts and indexes each section, grabs or sets each section.

The difference is, you need to provide the attribute name when declaring the attribute, and the Indexer needs the name, you only need the **this** keyword to refer to the instance of the object. The following example illustrates this:

```
using System ; namespace QTMCSsharp { class IndexedNames { private string [] da
??nh" ; names [ 6 ] = "V??
ng" ; for ( int i = 0 ; i IndexedNames . size ; i ++ ) { Console . WriteLi
```

Compiling and running the C # program on your side will have the following results:

```
Brother
Hoang
flower
Detective
jar
Dinh
```

Wang
NA
NA
NA

Load Indexer stack in C

Indexer can be overloaded in C #. Indexer can also be declared with multiple parameters and each parameter can be a different type. Indexers are not necessarily integers. C # allows indexes to be other value types, such as strings.

The following example will illustrate how to load Indexer in C #:

```
using System ; namespace IndexerApplication { class IndexedNames { private string  
??nh" ; names [ 6 ] = "V??ng" ; //s? d?ng indexer ??u tiên v?i tham s?  
int for ( int i = 0 ; i < IndexedNames . size ; i ++ ) { Console . WriteLine  
? d?ng indexer th? 2 v?i tham s? string Console . WriteLine ( names [ "V??  
ng" ] ); Console . ReadKey ( ); } } }
```

Running the code above we will get the following result:

```
Brother  
Hoang  
flower  
Detective  
jar  
Dinh  
Wang  
NA  
NA  
NA  
6
```

According to Tutorialspoint

Previous article: Reflection in C #

Next article: Delegate in C #

You finished reading the article "**Indexer in C #**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.