C# | ListDictionary Class

geeksforgeeks.org/c-sharp-listdictionary-class

October 25, 2018

ListDictionary is a specialized collection. It comes under the ***System.Collections.Specialized*** namespace. This type represents a non-generic dictionary type. It is implemented with a **linked list**. This class is a simple implementation of a dictionary collection (System.Collections.IDictionary) for small lists. It implements the **IDictionary** methods and properties, and is suggested for use with a small number of elements (less than 10).

Characteristics of ListDictionary Class:

- ListDictionary is a simple implementation of IDictionary using a singly linked list.
- It is smaller and faster than a Hashtable if the number of elements is 10 or less.
- ListDictionary should not be used if performance is important for large numbers of elements.
- Items in a ListDictionary are not in any guaranteed order.
- A key cannot be **null**, but a value can.

Constructors

Constructor	Description
ListDictionary()	Creates an empty ListDictionary using the default comparer.
ListDictionary(IComparer)	Creates an empty ListDictionary using the specified comparer.

Example:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections;
using System.Collections.Specialized;
class GFG {
public static void Main()
{
ListDictionary myDict = new ListDictionary();
myDict.Add("Australia", "Canberra");
myDict.Add("Belgium", "Brussels");
myDict.Add("Netherlands", "Amsterdam");
myDict.Add("China", "Beijing");
myDict.Add("Russia", "Moscow");
myDict.Add("India", "New Delhi");
Console.WriteLine("Total key/value pairs in myDict are : " + myDict.Count);
Console.WriteLine("The key/value pairs in myDict are : ");
foreach (DictionaryEntry de in myDict)
{
Console.WriteLine(de.Key + " " + de.Value);
}
}
}
```

Output:

```
Total key/value pairs in myDict are : 6
The key/value pairs in myDict are :
Australia Canberra
Belgium Brussels
Netherlands Amsterdam
China Beijing
Russia Moscow
India New Delhi
```

Properties

Property	Description
Count	Gets the number of key/value pairs contained in the ListDictionary.
IsFixedSize	Gets a value indicating whether the ListDictionary has a fixed size.
IsReadOnly	Gets a value indicating whether the ListDictionary is read-only.
IsSynchronized	Gets a value indicating whether the ListDictionary is synchronized (thread safe).
Item[Object]	Gets or sets the value associated with the specified key.

Keys	Gets an ICollection containing the keys in the ListDictionary.
SyncRoot	Gets an object that can be used to synchronize access to the ListDictionary.
Values	Gets an ICollection containing the values in the ListDictionary.

Example 1:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections;
using System.Collections.Specialized;
class GFG {
public static void Main()
{
ListDictionary myDict = new ListDictionary();
myDict.Add("Australia", "Canberra");
myDict.Add("Belgium", "Brussels");
myDict.Add("Netherlands", "Amsterdam");
myDict.Add("China", "Beijing");
myDict.Add("Russia", "Moscow");
myDict.Add("India", "New Delhi");
Console.WriteLine(myDict.Count);
}
}
```

Output:

```
6
```

Example 2:

filter_none
edit

play_arrow

brightness_4

```
using  System;
using  System.Collections;
using  System.Collections.Specialized;
class  GFG {
public  static  void  Main()
{
ListDictionary myDict =  new  ListDictionary();
myDict.Add(  "Australia"  ,  "Canberra"  );
myDict.Add(  "Belgium"  ,  "Brussels"  );
myDict.Add(  "Netherlands"  ,  "Amsterdam"  );
myDict.Add(  "China"  ,  "Beijing"  );
myDict.Add(  "Russia"  ,  "Moscow"  );
myDict.Add(  "India"  ,  "New Delhi"  );
Console.WriteLine(myDict.IsReadOnly);
}
}
```

Output:

```
False
```

Methods

Method	Description
Add(Object, Object)	Adds an entry with the specified key and value into the ListDictionary.
Clear()	Removes all entries from the ListDictionary.
Contains(Object)	Determines whether the ListDictionary contains a specific key.
CopyTo(Array, Int32)	Copies the ListDictionary entries to a one-dimensional Array instance at the specified index.
Equals(Object)	Determines whether the specified object is equal to the current object.
GetEnumerator()	Returns an IDictionaryEnumerator that iterates through the ListDictionary.
GetHashCode()	Serves as the default hash function.
GetType()	Gets the Type of the current instance.
MemberwiseClone()	Creates a shallow copy of the current Object.

Remove(Object)	Removes the entry with the specified key from the ListDictionary.
ToString()	Returns a string that represents the current object.

Example 1:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections;
using System.Collections.Specialized;
class GFG {
public static void Main()
{
ListDictionary myDict = new ListDictionary();
myDict.Add( "Australia" , "Canberra" );
myDict.Add( "Belgium" , "Brussels" );
myDict.Add( "Netherlands" , "Amsterdam" );
myDict.Add( "China" , "Beijing" );
myDict.Add( "Russia" , "Moscow" );
myDict.Add( "India" , "New Delhi" );
Console.WriteLine( "Total number of elements in myDict are :
"
+ myDict.Count);
foreach (DictionaryEntry de in myDict)
{
Console.WriteLine(de.Key + " " + de.Value);
}
}
}
```

Output:

```
Total number of elements in myDict are : 6
Australia Canberra
Belgium Brussels
Netherlands Amsterdam
China Beijing
Russia Moscow
India New Delhi
```

Example 2:

filter_none

edit

play_arrow

brightness_4

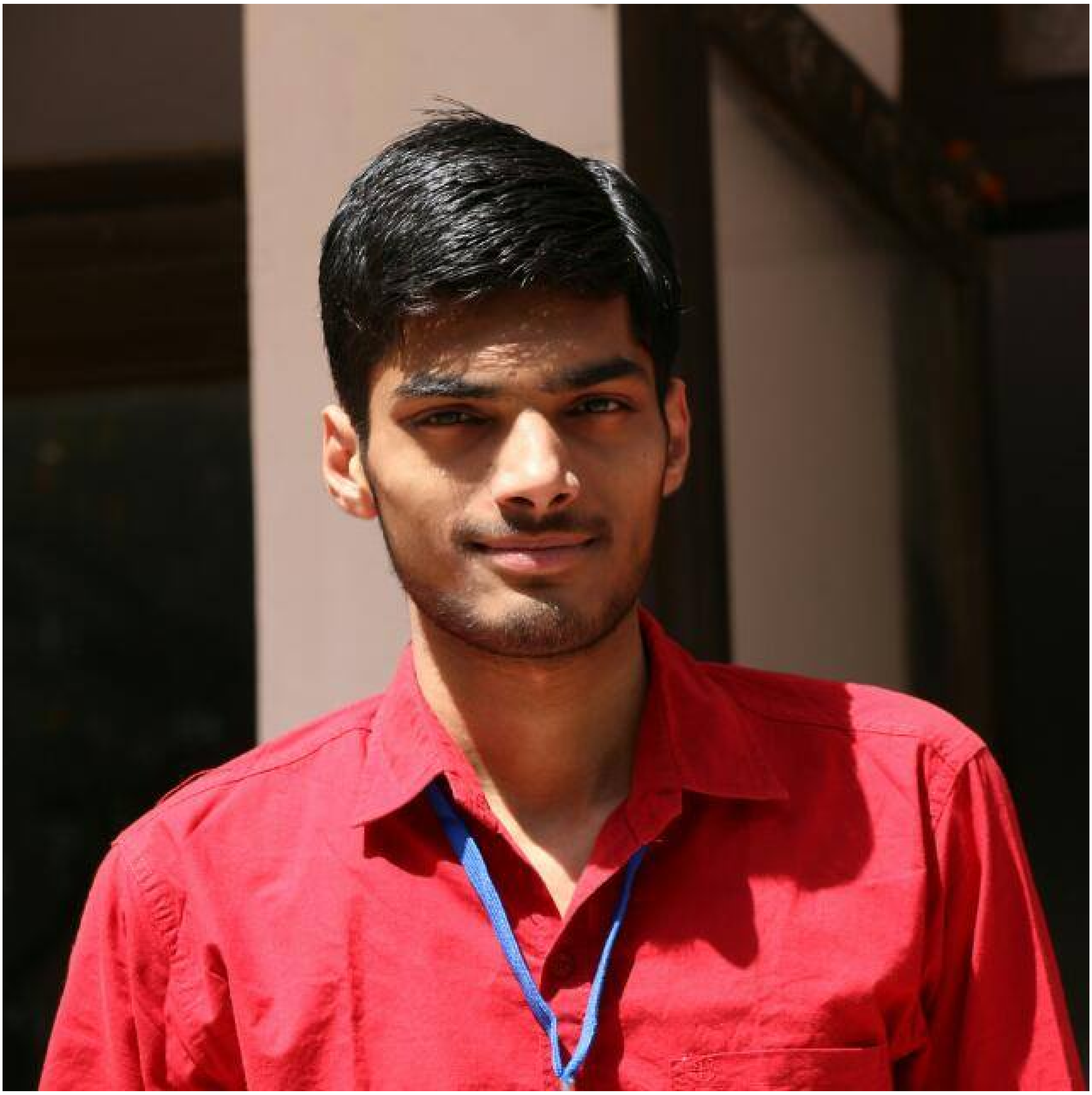
```
using System;
using System.Collections;
using System.Collections.Specialized;
class GFG {
public static void Main()
{
ListDictionary myDict = new ListDictionary();
myDict.Add("I" , "first" );
myDict.Add("II" , "second" );
myDict.Add("III" , "third" );
myDict.Add("IV" , "fourth" );
myDict.Add("V" , "fifth" );
Console.WriteLine( "Total key/value pairs in myDict are : "
+ myDict.Count);
Console.WriteLine( "The key/value pairs in myDict are : " );
foreach (DictionaryEntry de in myDict)
{
Console.WriteLine(de.Key + " " + de.Value);
}
myDict.Clear();
Console.WriteLine( "Total key/value pairs in myDict are : "
+ myDict.Count);
Console.WriteLine( "The key/value pairs in myDict are : " );
foreach (DictionaryEntry de in myDict)
{
Console.WriteLine(de.Key + " " + de.Value);
}
}
}
```

Output:

```
Total key/value pairs in myDict are : 5
The key/value pairs in myDict are :
I first
II second
III third
IV fourth
V fifth
Total key/value pairs in myDict are : 0
The key/value pairs in myDict are :
```

Reference:

https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.listdictionary?view=netframework-4.7.2



Sahil_Bansall
In love with a semicolon because sometimes i miss it so badly)

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.