SortedDictionary Class in C#

geeksforgeeks.org/sorteddictionary-class-in-c-sharp

February 11, 2019

In C#, the SortedDictionary<TKey,TValue> class is used to represent the collection of key/value pairs. This pair is in sorted form and the sorting is done on the key. This class defined under *System.Collections.Generic* namespace. In SortedDictionary class, the keys are immutable, always unique, and cannot be null. You are allowed to use null in value if the type of value is of reference type. The SortedDictionary class provides the fastest insertion and removal operations for unsorted data. The key/value pair of the SortedDictionary class is retrieved by using the KeyValuePair structure.

Constructors

Constructor	Description
SortedDictionary<TKey,TValue> ()	Initializes a new instance of the SortedDictionary class that is empty and uses the default IComparer implementation for the key type.
SortedDictionary<TKey,TValue> (IComparer)	Initializes a new instance of the SortedDictionary class that is empty and uses the specified IComparer implementation to compare keys.
SortedDictionary<TKey,TValue> (IDictionary)	Initializes a new instance of the SortedDictionary class that contains elements copied from the specified IDictionary and uses the default IComparer implementation for the key type.
SortedDictionary<TKey,TValue> (IDictionary, IComparer)	Initializes a new instance of the SortedDictionary class that contains elements copied from the specified IDictionary and uses the specified IComparer implementation to compare keys.

Example:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections.Generic;
public class GFG {
static public void Main()
{
SortedDictionary< string , string > myDr =
new SortedDictionary< string , string >();
myDr.Add( "One" . "C" ):
myDr.Add( "Two" . "C++" );
myDr.Add( "Three" , "C#" );
foreach (KeyValuePair< string , string > pair in myDr)
{
Console.WriteLine( "Key: {0} and Value: {1}" ,
pair.Key, pair.Value);
}
}
}
```

Output:

```
Key: One and Value: C
Key: Three and Value: C#
Key: Two and Value: C++
```

Properties

Property	Description
Comparer	Gets the IComparer used to order the elements of the SortedDictionary.
Count	Gets the number of key/value pairs contained in the SortedDictionary.
Item[TKey]	Gets or sets the value associated with the specified key.
Keys	Gets a collection containing the keys in the SortedDictionary.
Values	Gets a collection containing the values in the SortedDictionary.

Example:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections.Generic;
public class GFG {
static public void Main()
{
SortedDictionary< int , string > myDr =
new SortedDictionary< int , string >();
myDr.Add(1, "Dog" );
myDr.Add(2, "Cat" );
myDr.Add(3, "Birds" );
myDr.Add(4, "Rabbits" );
myDr.Add(5, "Fish" );
myDr.Add(6, "Hamster" );
myDr.Add(7, "Turtle" );
Console.WriteLine( "Total number of pairs " +
"present in myDr : {0}" , myDr.Count);
}
}
```

Output:

Total number of pairs present in myDr : 7

Methods

Method	Description
Add(TKey, TValue)	Adds an element with the specified key and value into the SortedDictionary.
Clear()	Removes all elements from the SortedDictionary.
ContainsKey(TKey)	Determines whether the SortedDictionary contains an element with the specified key.
ContainsValue(TValue)	Determines whether the SortedDictionary contains an element with the specified value.
CopyTo(KeyValuePair<TKey,TValue>[], Int32)	Copies the elements of the SortedDictionary to the specified array of KeyValuePair structures, starting at the specified index.

Equals(Object)	Determines whether the specified object is equal to the current object.
GetEnumerator()	Returns an enumerator that iterates through the SortedDictionary.
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(TKey)	Removes the element with the specified key from the SortedDictionary.
ToString()	Returns a string that represents the current object.
TryGetValue(TKey, TValue)	Gets the value associated with the specified key.

Example:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections.Generic;
class GFG {
static public void Main()
{
SortedDictionary< int , string > myDr =
new SortedDictionary< int , string >();
myDr.Add(1, "Dog" );
myDr.Add(2, "Cat" );
myDr.Add(3, "Birds" );
myDr.Add(4, "Rabbits" );
myDr.Add(5, "Fish" );
myDr.Add(6, "Hamster" );
myDr.Add(7, "Turtle" );
Console.WriteLine( "Pet animals list:" );
foreach (KeyValuePair< int , string > pair in myDr)
{
Console.WriteLine( "Key:{0} and Value: {1}" ,
pair.Key, pair.Value);
}
}
}
```

Output:

```
Pet animals list:
Key:1 and Value: Dog
Key:2 and Value: Cat
Key:3 and Value: Birds
Key:4 and Value: Rabbits
Key:5 and Value: Fish
Key:6 and Value: Hamster
Key:7 and Value: Turtle
```

Reference:

https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sorteddictionary-2?view=netframework-4.7.2#definition

Recommended Posts:

ankita_saini

Check out this Author's contributed articles.

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.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.