

Collections : SortedList

- A sorted list is a combination of an **array** and a **hash table**.
- The SortedList class represents a collection of key-and-value pairs that are sorted by the keys and are accessible by key and by index.
- It contains a list of items that can be accessed using a key or an index.
- The collection of items is always sorted by the key value.
- If we access items using an index, it is an ArrayList, and if you access items using a key, it is a Hashtable.

Property of SortedList class

Capacity

Gets or sets the capacity of the SortedList.

Count

Gets the number of elements contained in the SortedList.

IsFixedSize

Gets a value indicating whether the SortedList has a fixed size.

IsReadOnly

Gets a value indicating whether the SortedList is read-only.

Item

Gets and sets the value associated with a specific key in the SortedList.

Keys

Gets the keys in the SortedList.

Values

Gets the values in the SortedList.

Methods of SortedList

public virtual void Add(object key, object value);

Adds an element with the specified key and value into the SortedList.

public virtual void Clear();

Removes all elements from the SortedList.

public virtual bool ContainsKey(object key);

Determines whether the SortedList contains a specific key.

public virtual bool ContainsValue(object value);

Determines whether the SortedList contains a specific value.

public virtual object GetByIndex(int index);

Gets the value at the specified index of the SortedList.

public virtual object GetKey(int index);

Gets the key at the specified index of the SortedList.

public virtual IList GetKeyList();

Gets the keys in the SortedList.

public virtual IList GetValueList();

Gets the values in the SortedList.

public virtual int IndexOfKey(object key);

Returns the zero-based index of the specified key in the SortedList.

public virtual int IndexOfValue(object value);

Returns the zero-based index of the first occurrence of the specified value in the SortedList.

public virtual void Remove(object key);

Removes the element with the specified key from the SortedList.

public virtual void RemoveAt(int index);

Removes the element at the specified index of SortedList.

public virtual void TrimToSize();

Sets the capacity to the actual number of elements in the SortedList.

Application Program which demonstrate use of SortedList methods and properties.

```
using System;
using System.Collections;

public class Marvellous
{
    public static void Main(string[] args)
    {
        SortedList sl = new SortedList();

        sl.Add("10","Pre-Placement Activity");
        sl.Add("11","Angular");
        sl.Add("12","Logic Building");
        sl.Add("13","Inndustrial Project Development");

        if (sl.ContainsValue("Linux System Programming"))
        {
            Console.WriteLine("This batch is already in the list");
        }
        else
        {
            sl.Add("15", "Linux System Programming");
        }

        ICollection keys = sl.Keys;

        foreach (String k in keys)
        {
            Console.WriteLine(sl[k]);
        }

        sl.Remove("11");

        Console.WriteLine("Elements of SortedList after removal");

        foreach (String k in keys)
        {
            Console.WriteLine(sl[k]);
        }
    }
}
```