



Liste tarzı collection sınıflar

Sözlük tarzı collection sınıflar

interface IEnumerable

```
{  
    IEnumerator GetEnumerator();  
}
```

Üç Elemanı vardır.

MoveNext

Current

Reset

bool MoveNext()

```
using System;  
using System.Collections;  
  
namespace BM  
{  
    class App  
    {
```

```

public static void Main()
{
    ArrayList al = new ArrayList();

    for (int i = 0; i < 10; ++i)
        al.Add(i * 10);

    IEnumerator ie = al.GetEnumerator();
    while (ie.MoveNext())
    {
        int val = (int)ie.Current;
        Console.Write("{0} ", val);
    }
    Console.WriteLine();

    ie.Reset();
    while (ie.MoveNext())
    {
        int val = (int)ie.Current;
        Console.Write("{0} ", val);
    }
    Console.WriteLine();
}
}
}
/*
- Liste Tarzı Collection Sınıflar
- Sözlük Tarzı Collection sınıflar
- Atipik Collection Sınıflar
*/
using System;
using System.Collections;

namespace BM
{
    class App
    {
        public static void Main()
        {
            ArrayList al = new ArrayList();

            for (int i = 0; i < 10; ++i)
                al.Add(i * 10);

            Sample.Walk(al);
        }
    }
    class Sample
    {
        public static void Walk(IEnumerable ie)
        {
            IEnumerator ien = ie.GetEnumerator();

            while(ien.MoveNext())
                Console.Write("{0} ", (int)ien.Current);
        }
    }
}

```

```

        Console.WriteLine();
    }
}

using System;
using System.Collections;

namespace BM
{
    class App
    {
        public static void Main()
        {
            double[] d = new double[] { 1, 2, 3, 4, 5 };
            int[] i = new int[] { 10, 20, 30, 40, 50 };
            string[] s = new string[] { "Hülya", "Ahmet", "Harun", "Murat", "Fatma" };

            ArrayList al = new ArrayList();
            al.Add("Bursa");
            al.Add("İzmir");
            al.Add("Adana");

            Sample.Walk(d);
            Sample.Walk(i);
            Sample.Walk(s);
            Sample.Walk(al);
        }
    }

    class Sample
    {
        public static void Walk(IEnumerable ie)
        {
            IEnumerator ien = ie.GetEnumerator();

            while(ien.MoveNext())
                Console.Write("{0} ", ien.Current.ToString());

            Console.WriteLine();
        }
    }
}

//////////
//////////

/*
  1 2 3 4 5
10 20 30 40 50
Hülya Ahmet Harun Murat Fatma
Bursa İzmir Adana
*/

using System;
using System.Collections;

namespace BM
{
    class App

```

```

{
    public static void Main()
    {
        MyCollection mc = new MyCollection(10, 20);
        IEnumerator ien = mc.GetEnumerator();

        while (ien.MoveNext())
        {
            int val = (int)ien.Current;
            Console.Write("{0} ", val);
        }
        Console.WriteLine();

        foreach(int x in mc)
            Console.Write("{0} ", x);
        Console.WriteLine();
    }
}

class MyCollection : IEnumerable
{
    private int m_low;
    private int m_high;
    public MyCollection(int low, int high)
    {
        m_low = low;
        m_high = high;
    }
    public IEnumerator GetEnumerator()
    {
        return new MyEnumerator(this);
    }
    private class MyEnumerator : IEnumerator
    {
        private MyCollection m_mc;
        private int m_curVal;

        public MyEnumerator(MyCollection mc)
        {
            m_mc = mc;
            m_curVal = mc.m_low-1;
        }
        public bool MoveNext()
        {
            if (m_curVal == m_mc.m_high)
                return false;
            ++m_curVal;
            return true;
        }
        public object Current
        {
            get { return m_curVal; }
        }
        public void Reset()
        {
            m_curVal = m_mc.m_low - 1;
        }
    }
}

```

```

}
////////////////////
////////////////////

/*
10 11 12 13 14 15 16 17 18 19 20
10 11 12 13 14 15 16 17 18 19 20
*/

using System;
using System.Collections;

namespace BM
{
    class App
    {
        public static void Main()
        {
            ArrayList al = new ArrayList();

            for (int i = 0; i < 10; ++i)
                al.Add(i);

            Foo(al);
        }
        public static void Foo(ICollection ic)
        {
            int[] a = new int[ic.Count];
            ic.CopyTo(a, 0);

            foreach(int x in a)
                Console.WriteLine("{0} ", x);
        }
    }
}
////////////////////
////////////////////

/*
ICollection
- CopyTo
- Count
- IsSynchronized
- SyncRoot
*/

using System;
using System.Collections;

namespace BM
{
    class App
    {
        public static void Main()
        {
            Hashtable ht = new Hashtable();

```

```

        ht.Add("Koray Aki", 123);
        ht.Add("Hülya Aydın", 512);
        ht.Add("Mert Ergin", 512);

        int val = (int)ht["Koray Aki"];
        Console.WriteLine(val);
        ht["Hakan Düzgün"] = 654;
        ht["Hakan Düzgün"] = 657;

        val = (int)ht["Hakan Düzgün"];
        Console.WriteLine(val);

        foreach(string key in ht.Keys)
            Console.Write("{0} ", key);

        Console.WriteLine();

        foreach (int v in ht.Values)
            Console.Write("{0} ", v);
        Console.WriteLine();
    }
}

///////////////////////////////////////////////////
///////////////////////////////////////////////////

/*
IList
IDictionary
Hashtable >>> Eleman Sayısı 20'den fazla olduğu durumlarda kullanılır
SortedList >>> Eleman sayısı azsa kullanılır

123
657
Koray Aki Hülya Aydın Mert Ergin Hakan Düzgün
123 512 512 657
*/

using System;
using System.Collections;

namespace BM
{
    class App
    {
        public static void Main()
        {
            SortedList sl = new SortedList();
            sl.Add("Koray Aki", 123);
            sl.Add("Hülya Aydın", 512);
            sl.Add("Mert Ergin", 512);

            int val = (int)sl["Koray Aki"];
            Console.WriteLine(val);
            sl["Hakan Düzgün"] = 654;
            sl["Hakan Düzgün"] = 657;

            val = (int)sl["Hakan Düzgün"];

```

```

        Console.WriteLine(val);

        foreach(string key in sl.Keys)
            Console.Write("{0} ", key);

        Console.WriteLine();

        foreach (int v in sl.Values)
            Console.Write("{0} ", v);
        Console.WriteLine();
    }
}

////////////////////
////////////////////

/*
IList
IDictionary
Hashtable >>> Eleman Sayısı 20'den fazla olduğu durumlarda kullanılır
SortedList >>> Eleman sayısı azsa kullanılır

123
657
Koray Aki Hülya Aydın Mert Ergin Hakan Düzgün
123 512 512 657
*/

```