



YALOVA ÜNİVERSİTESİ  
university of yalova

# Bilgisayar Programcılığı

Ögr. Gör. Cansu AYVAZ GÜVEN

# DİZİLER

---

- ▶ Kullanım Şekli
- ▶ Dizi Elemanlarını Yeniden Boyutlandırma
- ▶ Dizi Elemanlarına Ulaşım
- ▶ Çok Boyutlu Diziler
- ▶ Array Sınıfı Metodları
- ▶ Array List

# Diziler

---

- ▶ Diziler, aynı tipte verilerin bir arada tutularak kullanılmasını sağlayan yapılardır. Tek bir dizi tanımlamasıyla, aynı tipteki birçok veriyi tekrar tekrar değişken tanımlaması yapmadan kullanabiliriz.
- ▶ Dizileri kullanırken dikkat etmemiz gereken bazı hususlar vardır. Bunların başında, diziye bir isim vermek gelir. Bu isim sayesinde içindeki verilere teker teker isim vermekle uğraşmayız. Numaralandırma yöntemiyle, yani indis ile bütün elemanlara ulaşabiliriz.

# Diziler

---

- ▶ Dizilerde numaralandırma işlemi, yani elemanları diziye yerleştirme işlemi sıfırdan başlar. 1. elemanımız dizide 0. indistedir. Diziler böyle bir yapıya sahip olduklarından, dizide hem verilerimizin tutulduğu bir yer hem de indislerin tutulduğu ayrı bir yer vardır. Verilerimizin tutulduğu yere *heap* bölgesi denirken, indekslemede kullanılan değişkenler *stack* bölgesinde yer alır.

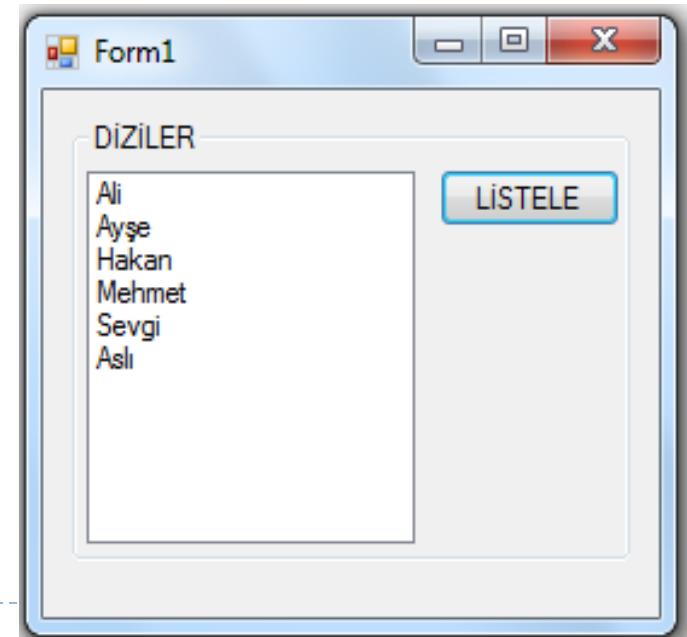
# Kullanım Şekli

---

- ▶ `veriTipi[] diziAdi = new veriTipi[elemanSayisi];`  
`int [] sayilar = new int [3];`
- ▶ `veriTipi[] diziAdi = { eleman1, eleman2, eleman3, ... };`
- ▶ `string [] sehirler = { "İstanbul" , "Ankara" , "İzmir" , "Eskişehir" };`

```
public Form1()
{
    InitializeComponent();
}

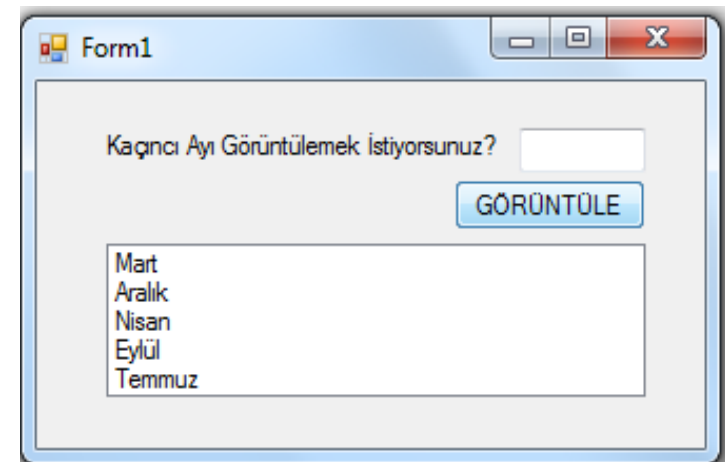
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    string[] isimler = {"Ali", "Ayşe", "Hakan", "Mehmet", "Sevgi", "Aslı" };
    foreach (string isim in isimler)
    {
        listBox1.Items.Add(isim);
    }
}
```



```
private void button1_Click(object sender, EventArgs e)
{
    string [] aylar={"Ocak","Şubat","Mart","Nisan","Mayıs","Haziran",
"Temmuz","Ağustos","Eylül","Ekim","Kasım","Aralık"};

    int i = 0;
    i = Convert.ToInt32(textBox1.Text);

    if (i>0 && i<=12)
    {
        listBox1.Items.Add(aylar[i-1]);
        textBox1.Clear();
    }
    else
    {
        MessageBox.Show("Lütfen 1 ile 12 arasında bir değer giriniz.");
        textBox1.Text = "";
    }
}
```



# Dizi Elemanlarını Yeniden Boyutlandırma

---

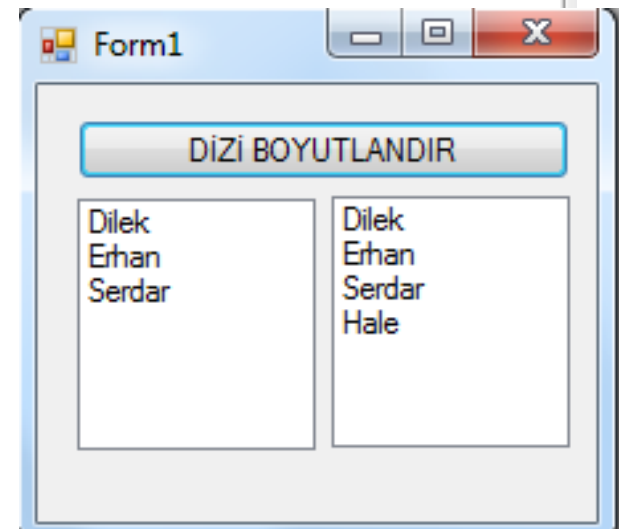
- ▶ Dizileri yeniden boyutlandırmak için *Array.Resize()* metodu kullanılır.
- ▶ Diziler yeniden boyutlandırıldığında dizi içeriği değişmez, sadece boyut yenilemesi yapılmış olur.

## !!! Kullanım Şekli:

```
Array.Resize (ref dizi_adi, yeni_boyut);
```



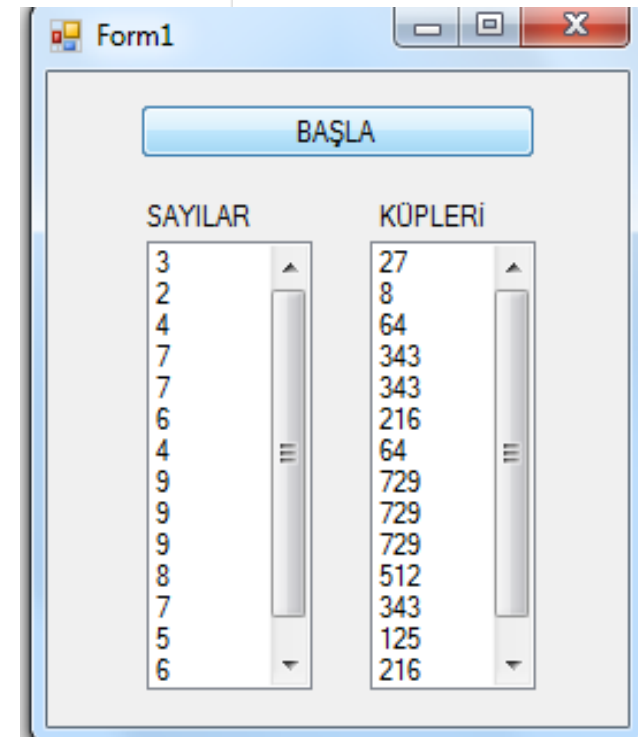
```
}  
  
private void button1_Click(object sender, EventArgs e)  
{  
    string[] dizi= new string [3];  
    dizi[0]="Dilek";  
    dizi[1] = "Erhan";  
    dizi[2] = "Serdar";  
  
    foreach (string isim in dizi)  
    {  
        listBox1.Items.Add(isim);  
    }  
    Array.Resize(ref dizi, 4);  
    dizi[3] = "Hale";  
  
    foreach (string isim in dizi)  
    {  
        listBox2.Items.Add(isim);  
    }  
}
```



# Dizi Elemanlarına Ulaşım

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    listBox2.Items.Clear();

    int [] sayi= new int [15];
    int [] kup= new int [15];
    Random rnd = new Random();
    for (int i = 0; i < sayi.Length; i++)
    {
        sayi[i] = rnd.Next(1,10);
        kup[i] = sayi[i] * sayi[i] * sayi[i];
    }
    foreach (int sayilar in sayi)
    {
        listBox1.Items.Add(sayilar);
    }
    foreach (int kupleri in kup)
    {
        listBox2.Items.Add(kupleri);
    }
}
```



# Çok Boyutlu Diziler

---

Kullanım Şekli:

- ▶ `veriTipi [satır, sütun] diziAdi = new veriTipi [satır, sütun];`

`int [ , ] sayi = new int [3,3];`

`int [0,0]= 5;`

`int [0,1]= 6;`

`int [0,2]= 2;`

`int [1,0]= 7;`

`int [1,1]= 9;`

`int [1,2]= 3;`

`int [2,0]= 1;`

`int [2,1]= 8;`

`int [2,2]= 4;`

5	6	2
7	9	3
1	8	4

}

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    int [,] dizi = new int [2,3];
```

```
    dizi[0, 0] = 21;
```

```
    dizi[0, 1] = 42;
```

```
    dizi[0, 2] = 2;
```

```
    dizi[1, 0] = 9;
```

```
    dizi[1, 1] = 13;
```

```
    dizi[1, 2] = 7;
```

```
    for (int i = 0; i < dizi.GetLength(0); i++)
```

```
    {
```

```
        for (int j = 0; j < dizi.GetLength(1); j++)
```

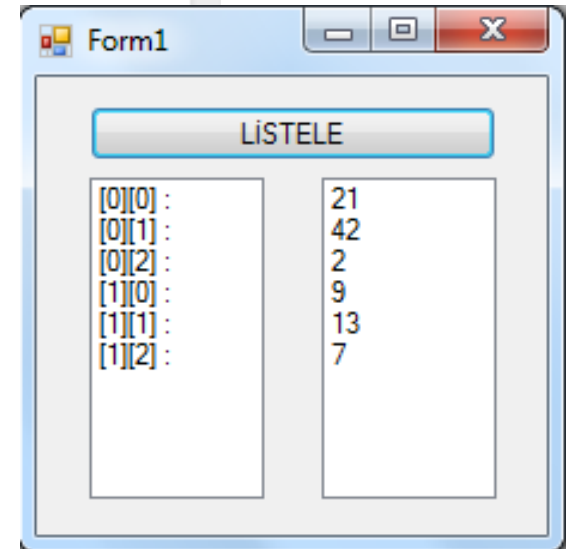
```
        {
```

```
            listBox1.Items.Add("[ "+i+" ]"+"["+j+" ] : ");
```

```
            listBox2.Items.Add(dizi [i,j]);
```

```
        }
```

```
    }
```



# Array Sınıfı Metodları

Metot Adı	Açıklama	Kullanım Şekli
Copy	Diziler arası kopyalama işlemlerinde kullanılır.	Copy(dizi1, dizi1_indis, dizi2, dizi2_indis, uzunluk)
CreateInstance	Bir dizi tanımlamak için kullanılır. Eleman sayısı ve boyutu kod bloğunda belirlenir.	Array.CreateInstance(typeof(türü), elemanAdedi, boyut);
SetValue	Yeri belirtilen elemana istenilen bilgiyi yazıp değiştirmek için kullanılır.	SetValue("değiştirilecekKarakter", adres);
GetValue	Yeri belirtilen karakteri geri döndürür.	GetValue(adres);

# Array List

---

Farklı tipte verileri tutmamıza yarayan bir dizidir. Bu diziler bütün verileri aynı yerde tuttuğu için koleksiyon adı verilir ve *System* isim alanı altında yer alır(*System.Collections*).

Array List oluştururken eleman sayısı vermemiz gerekmez, eleman ekledikçe dizi boyutu artar. Yani dizinin boyutu her zaman dinamik olarak belirlenir.

!!! Array List kullanacaksak *using* ile projeye *System.Collections* isim alanı dahil edilmelidir.

<b>Metod Adı</b>	<b>Açıklama</b>	<b>Kullanım Şekli</b>
Add	Dizi içerisinde eleman eklemek için kullanılır.	Dizi.Add(deger);
Insert	Dizide istenilen yere eleman eklemeye yarar.	Dizi.Insert(eklenecekYer, eklenecekDeğer);
Remove	Dizide bulunan elemanları silmeye yarar.	Dizi.Remove(deger);
Clear	Dizi içindeki bütün elemanları silmeye yarar	Dizi.Clear();
Sort	Diziyi sıralamak için kullanılır.	Dizi.Sort();
Reverse	Diziyi tersine çevirmek için kullanılır.	Dizi.Reverse();
AddRange	Diziyeye başka bir dizi eklemek için kullanılır.	ÜzerineEkleme YapılanDizi. AddRange(eklenenDizi);
InsertRange	Diziyeye belirtilen yerden itibaren başka bir dizi eklemek için kullanılır.	ÜzerineEkleme YapılanDizi. InsertRange(index, eklenenDizi);
RemoveRange	Belirttiğimiz yerden, belirttiğimiz sayıda elemanı diziden silmeye yarar.	Dizi.RemoveRange(index, sayi);
CopyTo	Dizinin içeriğini başka bir diziyeye kopyalamak için kullanılır.	KaynakDizi.CopyTo(Kopyalanacak_ Dizi)
Count	Dizinin eleman sayısını bulmak için kullanılır.	Dizi.Count
Capacity	Dizi için ayrılmış eleman kapasitesini verir.	Dizi.Capacity
Clone	Farklı türden verilere sahip dizileri kopyalamak için object türünden kopyalamadır	KopyaladığımızYer=KaynakDizi. Clone();

```
private void button1_Click(object sender, EventArgs e)
{
    ArrayList isim1 = new ArrayList();
    ArrayList isim2 = new ArrayList();
    ArrayList isim3 = new ArrayList();
    ArrayList isim4 = new ArrayList();

    isim1.Add("Ali");
    isim1.Add("Ahmet");
    isim1.Add("Mehmet");

    isim2.Add("Defne");
    isim2.Add("Sedef");
    isim2.Add("Burcu");

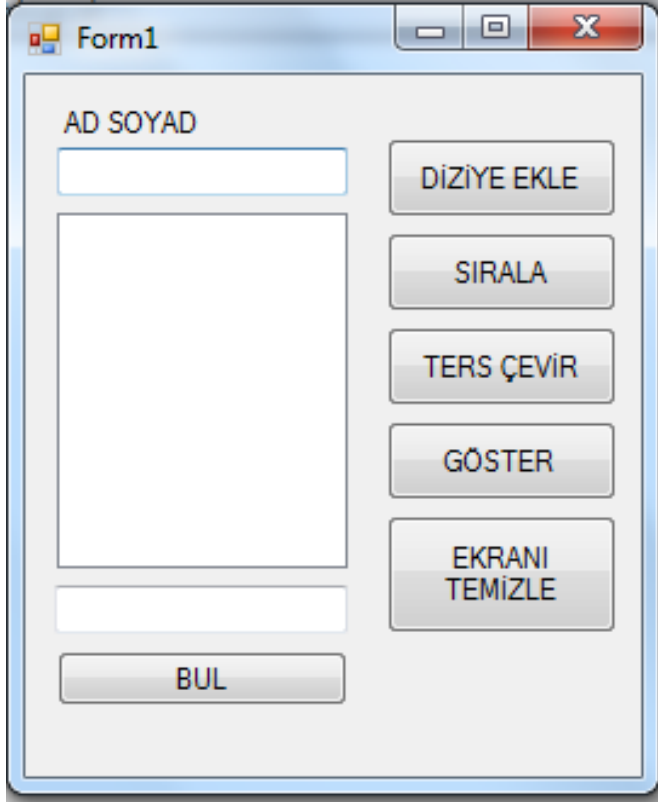
    isim3.AddRange(isim2);
    // isim2 dizisinin isim3 dizisine eklenmesi
    isim1.InsertRange(2, isim2);
    // 2. indisten itibaren isim1 dizisine isim2 dizisini ekle
    isim4 = (ArrayList)isim2.Clone();
    // isim2 dizisindeki tüm elemanların isim4 dizisine atanması

    MessageBox.Show(isim1.Count.ToString());
    //isim1 adlı dizinin eleman sayısı
    MessageBox.Show(isim2.Capacity.ToString());
    // isim2 adlı dizinin kapasitesi
    MessageBox.Show(isim3[0].ToString() + isim3[1].ToString() + isim3[2].ToString());
    // yeni isim3 adlı dizinin elemanlarının gösterilmesi
    MessageBox.Show(isim1[0].ToString() + isim1[1].ToString() + isim1[2].ToString());
    // yeni isim1 adlı dizinin elemanlarının gösterilmesi
    MessageBox.Show(isim4[0].ToString() + isim4[1].ToString() + isim4[2].ToString());
}
```





# Uygulama:



The image shows a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- A text input field labeled "AD SOYAD" at the top left.
- A search button labeled "BUL" at the bottom left.
- A vertical stack of five buttons on the right side: "DİZİYE EKLE", "SIRALA", "TERS ÇEVİR", "GÖSTER", and "EKRANI TEMİZLE".
- A large empty rectangular area in the center, likely intended for displaying the results of the search or list operations.

- ▶ Girmiş olduğumuz ad ve soyadın diziye eklenmesini, dizinin bütün elemanlarının alfabetik sıraya göre sıralanmasını, bütün dizinin tersine çevrilmesini ve aradığımız kelimenin dizide varsa yerinin gösterilmesini istiyoruz. Ayrıca ekran temizleme butonunun olmasını istiyoruz.

```
public Form1()
{
    InitializeComponent();
}
string [] bilgi = new string [5];
int index = 0;

private void btnEkle_Click(object sender, EventArgs e)
{
    if (index < 5)
    {
        bilgi[index] = textBox1.Text;
        index++;
    }
    else
    {
        MessageBox.Show("Dizi Dolu!!!");
    }
    textBox1.Text = "";
}
```

```
private void btnSiralama_Click(object sender, EventArgs e)
{
    Array.Sort(bilgi);
}

private void btnCevir_Click(object sender, EventArgs e)
{
    Array.Reverse(bilgi);
}

private void btnGoster_Click(object sender, EventArgs e)
{
    for (int i = 0; i < bilgi.Length; i++)
    {
        listBox1.Items.Add(bilgi[i]);
    }
}
```

```
private void btnTemizle_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
}
```

```
private void btnBul_Click(object sender, EventArgs e)
{
    string aranan = textBox2.Text;
    int yer = Array.IndexOf(bilgi, aranan);
    while (yer >= 0)
    {
```

```
        MessageBox.Show("Bulundu. " + aranan + " Listede " + (yer+1) + ". sırada");
```

```
        yer = Array.IndexOf(bilgi, aranan, yer+1 );
```

```
    }
```

```
}
```

```
}
```

```
}
```