



Kırklareli Üniversitesi  
Lüleburgaz Meslek Yüksekokulu

## Görsel Programlama-I

### Hafta-9 : Diziler (Arrays)

Öğr. Gör. Özcan EKEN  
ozcan.eken@klu.edu.tr

## Diziler

Diziler aynı türdeki verilerin sıralanarak gruplandırılmasında kullanılan yapılardır. Aynı türdeki verileri saklamada kullanılan bir koleksiyondur. Diziler içerisinde veriler kaydedilirken 0 numaralı sıra değerinden itibaren kaydedilmeye başlanılır. Dizi elemanlarına bu index numarası ile erişim sağlanır. Dizi oluştururken eleman sayısı belirtilmelidir. Dizi şu şekilde tanımlanır.

**Değişken\_türü [ ] Dizi\_değişken\_adı = new Değişken\_türü [ Eleman\_sayısı ];**

Örneğin;

```
int [ ] numara = new int[10];
```

şeklinde bir tanımlamada Numara adında integer tipinde 10 elemanı bulunan bir dizi tanımlanmaktadır.

## Diziler

Dizi elemanına değer atama işlemi yapılırken, dizi elemanının index değeri köşeli parantez içerisinde yazılarak istenen dizi elemanına atama yapılabilir.

```
string[] adsoyad = new string[3];  
adsoyad[0] = "Özcan EKEN";  
adsoyad[1] = "Onur BULUT";  
adsoyad[2] = "Fusun YAVUZER ASLAN";
```

Örnekte elemanları string türünde olan 3 elemanlı adsoyad dizisi tanımlanarak, daha sonra dizinin elemanlarına atama yapılmıştır.

Dizinin elemanlarına index numarası ile erişim sağlanmıştır.

Dizilere erişimde dizi elemanlarının index numaralarını tek tek yazmak her zaman kullanışlı olmayacağından, 0 (Sıfır) 'dan başlayarak dizi eleman sayısının bir eksiğine kadar bir döngü kurarak dizi elemanlarına ulaşmak daha kullanışlı olacaktır. Ya da dizi elemanlarının tümünde işlem yapmak için foreach döngüsünde kullanılabilir.

## Örnek

```
static void Main(string[] args)  
{  
    int i,j=0;  
    int[] dizi = new int[51];  
    for(i=0;i<=100;i++)  
    {  
        if (i%2==0)  
        {  
            dizi[j] = i;  
            j++;  
        }  
    }  
    for(i=0;i<=50;i++)  
    {  
        Console.WriteLine(dizi[i]);  
    }  
    Console.ReadKey();  
}
```

```
static void Main(string[] args)  
{  
    int i,j=0;  
    int[] dizi = new int[51];  
    for(i=0;i<=100;i++)  
    {  
        if (i%2==0)  
        {  
            dizi[j] = i;  
            j++;  
        }  
    }  
    foreach(int k in dizi)  
    {  
        Console.WriteLine(k);  
    }  
    Console.ReadKey();  
}
```

## Dizi Değişkenlerinde Kullanılan Metodlar

### CREATEINSTANCE METODU

Yeni bir diziyi değişken tipini ve eleman sayısını belirleyerek tanımlamak için kullanılan methoddur.

**Kullanımı:** `Array.CreateInstance(Değişkentürü , ElemanSayısı);`

Örneğin tek boyutlu 3 elemanlı bir dizi oluşturarak, bir elemanına erişelim.

```
static void Main(string[] args)
{
    Array okulturu = Array.CreateInstance(typeof(string), 3);
    string[] okulturleri = (string[])okulturu;
    okulturleri[0] = "Endüstri Meslek";
    okulturleri[1] = "Teknik";
    okulturleri[2] = "Anadolu Teknik";
    Console.WriteLine(okulturleri[1]);
    Console.ReadKey();
}
```

1. satırda okulturu object olarak geriye döndüğünden dolayı 2. satırda okulturleri dizisine dönüştürüyoruz.

## Dizi Değişkenlerinde Kullanılan Metodlar

### ISARRAY METODU

Bir değişkenin dizi değişkeni olup olmadığını anlamak amacıyla kullanılan methoddur. Geriye True yada False olmak üzere boolean değer döndürür.

```
static void Main(string[] args)
{
    string[] okulturu = { "Anadolu", "Meslek", "Teknik" };
    string ad = "Volkan";
    Type tip;
    tip = okulturu.GetType();
    Console.WriteLine(tip.IsArray);
    tip = ad.GetType();
    Console.WriteLine(tip.IsArray);
    Console.ReadKey();
}
```

Kodumuzda GetType metodu ile okulturu ve ad isimli değişkenlerin tipini elde edip, tip isimli değişkene aktarıyoruz. Daha sonrada IsArray metodu ile dizi olup olmadığını kontrol ediyoruz.

Programı çalıştırdığımızda okulturu değişkeni bir dizi olduğundan dolayı True sonucu üretirken, ad değişkeni dizi olmadığından False sonucunu üretir.

## Dizi Değişkenlerinde Kullanılan Metodlar

### SETVALUE METODU ve GETVALUE METODU

SetValue, Dizi elemanlarına değer atamak için kullanılan methoddur. GetValue, Dizi içerisindeki elemanlara ulaşmak için kullanılır.

```
static void Main(string[] args)
{
    Array okulturu = Array.CreateInstance(typeof(string), 4);
    okulturu.SetValue("Genel Lise", 0);
    okulturu.SetValue("Fen Lisesi", 1);
    okulturu.SetValue("Endüstri Meslek", 2);
    okulturu.SetValue("Teknik", 3);
    Console.WriteLine(okulturu.GetValue(2));
    Console.ReadKey();
}

static void Main(string[] args)
{
    Array okulturleri = Array.CreateInstance(typeof(string), 4);
    string[] okulturu = (string[])okulturleri;
    okulturu[0] = "Genel Lise";
    okulturu[1] = "Fen Lisesi";
    okulturu[2] = "Endüstri Meslek";
    okulturu[3] = "Teknik";
    Console.WriteLine(okulturu[2]);
    Console.ReadKey();
}
```

## Dizi Değişkenlerinde Kullanılan Metodlar

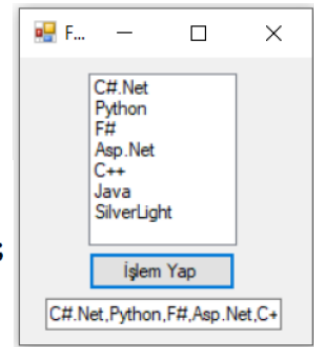
### JOIN METODU

Dizi içerisindeki değerlerin hepsini bir string değişkene belirtilen ayırıcı karakteri kullanarak aktaran methoddur.

**Kullanımı:** `String.Join(ayırıcı karakter, dizi değişkeni adı);`

Formumuza bir listBox ekleyerek items özelliğinden Şekildeki gibi değerleri girelim. Bir buton bir de textBox ekleyerek butona kodumuzu yazalım.

```
private void button1_Click(object sender, EventArgs e)
{
    string[] diller = new string[listBox1.Items.Count];
    byte i;
    for(i=0;i<=listBox1.Items.Count-1;i++)
    {
        diller[i] = listBox1.Items[i].ToString();
    }
    textBox1.Text = String.Join(",", diller);
}
```



## Dizi Değişkenlerinde Kullanılan Metodlar

### COPY METODU

Bir dizinin içeriğini başka bir diziye kopyalamak için kullanılan metoddur.

**Kullanımı:** `Array.Copy(Kaynak dizin adı, Kedef dizin adı, Eleman sayısı) ;`

```
static void Main(string[] args)
{
    string[] diller = new string [3];
    diller[0] = "VB.Net";
    diller[1] = "C#";
    diller[2] = "ASP.Net";
    string[] digerdiller = new string[3];
    digerdiller[0] = "Python";
    digerdiller[1] = "F#";
    digerdiller[2] = "C++";
    Array.Copy(digerdiller, diller, 3);
    Console.WriteLine(diller[0]);
    Console.WriteLine(diller[1]);
    Console.WriteLine(diller[2]);
    Console.ReadKey();
}
```

diller dizisinde VB.Net,C#,Asp.Net elemanları olduğu halde kodu çalıştırdığımızda diller dizisinin elemanlarını ekrana Python,F#,C++ olarak yazar.

Digerdiller dizisinin 3 elemanını, diller dizisinin başından itibaren kopyalar.

Kopyalanacak eleman sayısı 2 olsaydı  
Ekran çıktısı : Python, F#, Asp.Net olurdu.

## Diziler İçerisinde Arama İşlemleri

### CONTAINS METODU

Bir dizi içerisinde bir elemanın var olup olmadığını bulma amacı için kullanılır. Geriye TRUE yada FALSE olmak üzere boolean değer döndürür.

```
static void Main(string[] args)
{
    string aranan;
    string[] isimler = new string[4];
    isimler[0] = "Hasan";
    isimler[1] = "Murat";
    isimler[2] = "Talha";
    isimler[3] = "Onur";
    Console.Write("Aranan değeri giriniz:");
    aranan = Console.ReadLine();
    if (isimler.Contains(aranan) == false)
        Console.Write("Aranan değer bulunamadı");
    else
        Console.Write("Aranan eleman bulundu");
    Console.ReadKey();
}
```

## Diziler İçerisinde Arama İşlemleri

### INDEXOF METODU

Dizi içerisinde aranan elemanın index numarasını geriye döndüren metoddur.

**Array.IndexOf(Arama yapılacak dizi, aranacak değer, aramanın başlayacağı index) ;**

```
static void Main(string[] args)
{
    int indexno;
    string aranan;
    Array okulturu = Array.CreateInstance(typeof(string), 5);
    okulturu.SetValue("Genel Lise", 0);
    okulturu.SetValue("Fen Lisesi", 1);
    okulturu.SetValue("Endüstri Meslek", 2);
    okulturu.SetValue("Teknik", 3);
    okulturu.SetValue("Anadolu Teknik", 4);
    Console.WriteLine("Aranan değeri giriniz:");
    aranan = Console.ReadLine();
    indexno = Array.IndexOf(okulturu, aranan.ToString());
    if (indexno == -1)
        Console.WriteLine("Aranan değeri bulunamadı");
    else
        Console.WriteLine("Aranan eleman bulundu. Index değeri = " + indexno);
    Console.ReadKey();
}
```

IndexOf metodu geriye, eğer aranan değer bulunursa o değer index değerini, bulunamazsa -1 değerini döndürür.

## Diziler İçerisinde Arama İşlemleri

### LASTINDEXOF METODU

IndexOf metodunda olduğu gibi diziler içerisinde arama yapmak için kullanılır. Farklı olarak eğer aranan elemandan birden fazla mevcut ise bu elemanlardan sonuncusunun index değerini verir.

**Array.LastIndexOf(Arama yapılacak dizi, aranacak değer, aramanın başlayacağı index) ;**

```
static void Main(string[] args)
{
    int indexno;
    string aranan;
    Array okulturu = Array.CreateInstance(typeof(string), 5);
    okulturu.SetValue("Genel Lise", 0);
    okulturu.SetValue("Teknik", 1);
    okulturu.SetValue("Anadolu Teknik", 2);
    okulturu.SetValue("Teknik", 3);
    okulturu.SetValue("Anadolu Teknik", 4);
    Console.WriteLine("Aranan değeri giriniz:");
    aranan = Console.ReadLine();
    indexno = Array.LastIndexOf(okulturu, aranan.ToString());
    if (indexno == -1)
        Console.WriteLine("Aranan değeri bulunamadı");
    else
        Console.WriteLine("Aranan eleman bulundu. Index değeri = " + indexno);
    Console.ReadKey();
}
```

LastIndexOf metodu geriye, eğer aranan değer bulunursa o değer index değerini, bulunamazsa -1 değerini döndürür.

## Diziler İçerisinde Arama İşlemleri

### BINARYSEARCH METODU

Dizi içerisinde aranan elemanın index numarasını geriye döndüren metoddur. Bu metod ile arama yapabilmek için üzerinde arama yapılacak dizinin sıralı olması gerekir.

**Array.BinarySearch(Arama yapılacak dizi, aranacak değer, aramanın başlayacağı index) ;**

```
static void Main(string[] args)
```

```
{
```

```
    int indexno;
```

```
    string aranan;
```

```
    Array okulturu = Array.CreateInstance(typeof(string), 5);
```

```
    okulturu.SetValue("Anadolu Teknik", 0);
```

```
    okulturu.SetValue("Fen Lisesi", 1);
```

```
    okulturu.SetValue("Genel Lise", 2);
```

```
    okulturu.SetValue("Teknik", 3);
```

```
    Console.WriteLine("Aranan değeri giriniz:");
```

```
    aranan = Console.ReadLine();
```

```
    indexno = Array.BinarySearch(okulturu, aranan.ToString());
```

```
    if (indexno < 0)
```

```
        Console.WriteLine("Aranan değeri bulunamadı");
```

```
    else
```

```
        Console.WriteLine("Aranan eleman bulundu. Index değeri = " + indexno);
```

```
    Console.ReadKey();
```

```
}
```

BinarySearch metodu geriye, eğer aranan değer bulunursa o değer index değerini, bulunamazsa negatif bir değer döndürür.



Kırklareli Üniversitesi  
Lüleburgaz Meslek Yüksekokulu

## Görsel Programlama-I

**Hafta-9 : Diziler (Arrays)**

Öğr. Gör. Özcan EKEN  
ozcan.eken@klu.edu.tr