

Bölüm 25

Numaratör (enumerator)

Numaratör Nedir?

Numaratör Bildirimi

Numaratör Kullanımı

Numaratör Nedir?

Nesneleri numaralamak için kullanılır. C# var olan temel veri tiplerine ek olarak, programcı, gerekseme duyduğunda sınıf, struct, ve array veri tiplerini yaratabiliyordu. *Numaratör (enumerate)* de istediğinde programcının yarattığı bir veri tipidir. Tabii, bir veri tipi yaratırken, programcı belirli bir işi yapmak ister. Numaratör bir liste içindeki öğeleri numaralamaya yarar. Bunun nasıl olduğunu aşağıdaki örneklerle göreceğiz.

Enumerator01.cs

```
using System;

enum Aylar : byte
{
    Ocak, Şubat, Mart, Nisan, Mayıs, Haziran
}
class Enumerate01
{
    public static void Main()
    {
        byte a = (byte)Aylar.Ocak;
        byte b = (byte)Aylar.Şubat;
        byte c = (byte)Aylar.Mart;
        byte d = (byte)Aylar.Nisan;
        byte e = (byte)Aylar.Mayıs;
        byte f = (byte)Aylar.Haziran;
        Console.WriteLine("Ocak={0} , Nisan = {1}, Haziran={2}", a, d, f);
    }
}
```

Çıktı

Ocak=0 , Nisan = 3, Haziran=5

Çözümleme :

Numaratör (enumerate) daima

```
enum Aylar : byte
```

deyimine benzer bir başlıkla başlar.

enum numaratör için anahtar sözcüktür.
Aylar numaralanacak öğelere verilen addır.
: byte koleksiyonun numaralarının byte veri tipinden olacağını belirtir.

```
enum Aylar : byte  
{  
    Ocak, Şubat, Mart, Nisan, Mayıs, Haziran  
}
```

bloku içindeki {Ocak, Şubat, Mart, Nisan, Mayıs, Haziran } kümesi, numaralanacak öğeleri belirten listedir. enum bu listenin öğelerini sırayla numaralıyor. Bu blok Aylar 'ı bir veri tipi yapar. Numaralama, aksi istenmedikçe öndeğer (default) olarak 0 dan başlar birer artarak Aylar'a ait öğelere sırayla birer numara verir. Verilen numaralar, öğe adlarının yerine geçer, bir tür takma ad olurlar; yani öğenin asıl adı ile numarası birbirlerini kesinlikle belirlerler: Mart'ın numarası 2 dir ve 2 numaralı öğe Mart'tır.

```
(byte)Aylar.Ocak;
```

deyimi Aylar veri tipinin Ocak adlı öğesine verilen numaradır. Bu numarayı byte tipinden bir değişkene atayabiliriz:

```
byte a = (byte)Aylar.Ocak;
```

Artık bu numarayı, istersek konsola yazdırabiliriz. Yukarıdaki Main() metodu bu işi yapmaktadır.

Numaratör için sözdizimi şöyledir:

```
<belirteç> enum <enum_adı>  
{  
    // Numaralanacak öğelerin listesi  
}
```

Numaralama istenen bir sayıdan başlatılabilir.

```
using System;  
enum Hafta: long  
{  
    pzt = 1, sal, çar, per, cm, cmt, paz  
}  
class Uygulama  
{  
    public static void Main()  
    {  
        byte a = (byte)Hafta.pzt;  
        byte b = (byte)Hafta.sal;  
        byte c = (byte)Hafta.çar;  
    }  
}
```

```

        Console.WriteLine("Pazartesi = {0} , Salı = {1}, Çarşamba = {2}",
a, b, c);
    }
}

```

İstersek, numaralamayı kendimiz belirleyebiliriz. Yukarıdaki programı şöyle yazalım:

Enumerator03.cs

```

using System;
enum Meyve: long
{
    elma=3 , armut = 7, çilek = 17
}
class Uygulama
{
    public static void Main()
    {
        byte a = (byte)Meyve.elma;
        byte b = (byte)Meyve.armut;
        byte c = (byte)Meyve.çilek;

        Console.WriteLine("Armut = {0} , Elma = {1}, Çilek = {2}", a, b,
c);
    }
}

```

Çıktı

```

elma = 3 , armut = 7, çilek = 5

```

İstenirse bir numara birden çok öğeye de verilebilir:

Enumerator04.cs

```

using System;
enum Ağaç:int
{
    Çam = 3, KızılÇam = 3, Ladin =1,   Meşe = 7
}
class Uygulama
{
    public static void Main()
    {
        int x = (int)Ağaç.Çam;
        int y = (int)Ağaç.KızılÇam;
        int z = (int)Ağaç.Ladin;
        int w = (int)Ağaç.Meşe;

        Console.WriteLine("Çam = {0} , KızılÇam = {1}, Ladin = {2} , Meşe
= {3}", x, y, z, w);
    }
}

```

Numaratör (enum) vereceği numaralar için şu veri tiplerini kullanabilir: byte, sbyte, short, ushort, int, uint, long, or ulong.

Numaratörün taban öndeğer (default) tipi int veri tipindedir. Dolayısıyla, bildirim anında onu yazmayabiliriz:

Enumerator05.cs

```
using System;
enum Mevsim
{
    İlkBahar = 1, Yaz, SonBahar, Kış
}
class Uygulama
{
    public static void Main()
    {
        int a = (int)Mevsim.İlkBahar;
        int b = (int)Mevsim.Yaz;
        int c = (int)Mevsim.SonBahar;
        int d = (int)Mevsim.Kış;

        Console.WriteLine("İlkbahar = {0} , Yaz = {1}, Sonbahar = {2} ,
Kış = {3}", a, b, c, d);
    }
}
```

enum tipler üzerinde kısıtlar:

1. enum blokunda metot tanımlanamaz
2. Arayüz kullanamazlar (implement)
3. enum blokunda özgen (property) ve indeksçi tanımlanamaz.