



Sabitler

01.09.2006

Şimdi bir sinema bileti programı yazalım. Normal bilet ve öğrenci bileti için ayrı ayrı fiyat seçenekleri olsun. Program sinemaya gidince ödeyeceğimiz toplam tutarı bulsun.

```
using System;

class SinemaBileti
{
    static void Main()
    {
        int TamBilet = 10, OgrenciBileti = 7;

        Console.WriteLine("Kaç tane tam bilet: ");
        int TamBiletSayisi = Int32.Parse(Console.ReadLine());

        Console.WriteLine("Kaç tane öğrenci bileti: ");
        int OgrenciBiletiSayisi = Int32.Parse(Console.ReadLine());

        int ToplamTutar = TamBiletSayisi * TamBilet +
            OgrenciBiletiSayisi * OgrenciBileti;

        Console.WriteLine("Toplam tutar (YTL)" + ToplamTutar);
        Console.ReadLine();
    }
}
```

Yukarıdaki programda 5 tane değişken var. 2 tanesi ilk değer almış ve 2 tanesi kullanıcıdan isteniyor ve 1 tanesi de hesaplama sonucu oluşuyor.

İlk değer ataması yaptığımız 2 değişkene bir de başka açıdan bakalım.

Bu 2 değişkeni hiç kullanmasak ve programı aşağıdaki gibi değiştirsek yine aynı işlevi yerine getirir.

```
using System;

class SinemaBileti
{
    static void Main()
    {
        //int TamBilet = 10, OgrenciBileti = 7;

        Console.WriteLine("Kaç tane tam bilet: ");
        int TamBiletSayisi = Int32.Parse(Console.ReadLine());

        Console.WriteLine("Kaç tane öğrenci bileti: ");
        int OgrenciBiletiSayisi = Int32.Parse(Console.ReadLine());

        //int ToplamTutar = TamBiletSayisi * TamBilet +
        //    OgrenciBiletiSayisi * OgrenciBileti;

        int ToplamTutar = TamBiletSayisi * 10 + OgrenciBiletiSayisi * 7;
        Console.WriteLine("Toplam tutar (YTL)" + ToplamTutar);
        Console.ReadLine();
    }
}
```

```
}
```

Aralarında bir fark yok ama bu yeni şekilde 10 ve 7 değerleri birer sayı ve isimleri yok. Belki yorum satırları eklessek yardımcı olur. Eğer yorumlar olmazsa bir vakit sonra birisi sizin yazdığınız koda bakar ve "bu sayılar da ne?" der. Hatta bu kişi siz bile olabilirsiniz. Yazdığımız kodun okunurluğunu ve anlaşılabilirliğini kaybetmesi çok kötü bir durumdur.

Diğer yandan, kullanıcının bilet fiyatlarını kendisinin girmesine izin de vermek istemiyoruz. Aslında aşağıdaki gibi yapabirdik.

```
Console.WriteLine("Tam bilet fiyatını gir: ");  
  
int TamBilet = Int32.Parse(Console.ReadLine());
```

Bilet fiyatlarının programın kaynak kodunda bulunmasını istiyoruz. Programın kodlarının içerisine değerleri yazmak en iyi çözüm değil. Bilet fiyatları değiştiğinde kodu açıp değiştirip tekrar derlememiz lazım. Eğer sadece kendimiz kullanacaksak problem yok. Ama kodu değiştirip tekrar derledikten sonra yeni programı bütün kullanıcılara tekrar göndermek, yüklemek büyük problemler.

Kodu yeniden derlemektense, diskte bir dosyada tutmak ve gerektiğinde yeniden programı derlemeden sadece o dosyayı değiştirmek güzel bir çözüm. Bu işlerin nasıl yapılabileceğini başka yazılarda anlatacağız. Şimdilik bu değişkenleri kodun en üstüne koyalım, program açılır açılmaz değiştirilmesi kolay olsun.

Program kodlarının en üstünde aşağıdaki gibi iki ücret tanımlaması var.

```
int TamBilet = 10, OgrenciBileti = 7;
```

Ama programın sonlarına doğru aşağıdaki gibi bir kod bloğu gördüğünüz düşünün.

```
TamBilet = 12;  
OgrenciBileti = 9;
```

Bilet fiyatları değişmiş. Muhtemelen siz değiştirdiniz ama yorum cümlesi eklemeyi unuttunuz.

Bunun gibi bir programda TamBilet ve OgrenciBileti fiyatlarının değişken olmaması gerekiyor. Sadece bir tek değere sahip olmaları ve program boyunca da değiştirilememeleri gerekiyor. Diğer bir deyişle TamBilet ve OgrenciBileti değişken değil de sabit olmalıdır.

Sabitleri başlarına const yazarak tanımlarız.

```
const int TamBilet = 10;  
const int OgrenciBileti = 7;
```

Bir tanımlama ifadesinde const kullandıktan sonra artık TamBilet ve OgrenciBileti için değişken değil de sabit ifadesini kullanmak daha doğru olacaktır, TamBilet ve OgrenciBileti sabitleri.

Sabitlerin değişkenlerden ne farkı olduğunu inceleyelim. Tanımlama işlemini yaptıktan sonra değerini değiştirmeye çalıştığınız zaman derleme hatası ile karşılaşacaksınız.

```
TamBilet = 12;
```

Eğer programın bir yerlerinde yukarıdaki gibi bir ifade kullanırsanız C# derleyicisi size aşağıdaki hata mesajını verecektir.

The left-hand side of an assignment must be a variable, property, or indexer.

(Bir atama ifadesinin sol tarafında bir değişken, özellik ya da indeksleyici bulunmalıdır.)

Ve TamBilet yukarıda sayılanlardan hiç birisi değildir. Gerçekte bir değişken değil, sabittir. Böyle bir derleme hatası aldığınız durumda, kendinize sormalısınız: "Acaba programın ilerleyen bölümlerinde değiştirmek isteyeceğim bir şeyi sabit olarak mı tanımladım?" ya da bu örnekte olduğu gibi "Acaba bir sabit tanımladım ama programın ilerleyen bölümlerinde yanlışlıkla değerini mi değiştirmeye çalıştım?" diye.

Bir sabit tanımladığınız zaman ona bir değer ataması da yapmalısınız.

```
const int TamBilet; // Derleme hatası!
```

Eğer yapmazsanız aşağıdaki hata mesajını görürsünüz.

```
A const field requires a value to be provided.
```

(Sabit bir alan için bir değer tanımlamanız gerekiyor.)

Program boyunca değeri değişmeyecek olan bir değişken sabit olmaya adaydır. Sabit tanımlamak sadece programcının değer değiştirmek suretiyle hata yapmasını önlemez, aynı zamanda programın performansını da artırır. Derleyici sabitlerin sakladığı değerleri stack bölgesinde oluşturmak durumunda kalmaz. Eğer TamBilet ve OgrenciBileti değerlerini saklamak için sabit tanımlarsanız, derleyici ToplamTutar değerini ilk yaptığımız örnekte olduğu gibi değil de ikinci örnekteki gibi hesaplar.

Sabitleri tanımlarken sadece literaller değil ifadeler de kullanabilirsiniz.

```
const int TamBilet = 10;  
const int OgrenciBileti = 3 * TamBilet / 4;
```

OgrenciBileti, TamBilet değeri ile yapılan bir takım matematiksel hesaplamalardan sonra ortaya çıktı. TamBilet değerini değiştirerek OgrenciBileti değerini de değiştirebilirsiniz. Bu durumda TamBilet, OgrenciBileti'nden önce tanımlanmış olmalıdır. Yukarıdaki örnekte de olduğu gibi, bir sabite değer ataması yaparken başka bir sabitin değeri kullanılabilir ama değişken kullanılmaz.

Yorum Cümleleri adlı yazımızda geçen örneği hatırlarsak eğer, aşağıdaki gibi bir ifade vardı.

```
haftalikUcret = 80 * GunSayisi;
```

Bu örnekteki 80 sayısının ne için orada bulunduğunu kestirmek zor. Açıklayıcı bir isimle değiştirilse daha iyi olur. Yorum Cümleleri adlı yazımızda bir değişkenle değiştirmiştik. Fakat sabit kullanmak daha doğru bir karardır.

```
const int GunlukUcret = 80;  
haftalikUcret = GunlukUcret * GunSayisi;
```

Sabit kullanmak ifadeyi daha anlaşılır yaptı, ama daha da anlaşılır yapabiliriz.

Sabit ifadeyi değerini açık hesaplamalar şeklinde yazarak da tanımlayabiliriz.

```
const int GunlukUcret = 8 * 10; //Günde 8 saat çalışıldığını varsaydık.
```

Yukarıdaki çarma işlemi derleme esnasında yapılabacağı için programın çalışmasında da bir performans kaybı olmayacaktır.

Bir sabitin değeri program derlenirken belli olmak zorundadır. Meselâ,

```
const string Prompt = "Adınızı giriniz: ";
```

Prompt bir sabit olduğu için başka bir string ifade ataması yapılamaz. Fakat şu şekilde bir stringi bir sabite atayamazsınız.

```
const string Response = Console.ReadLine(); // Olmaz.
```

ReadLine çağrısı çalışma esnasında oluşur, derleme esnasında değil. Derleme esnasında değeri belirli olan bir şeyi ancak bir sabite atayabilirsiniz.