

Bölüm 14

Özgenler (properties)

Özgen (property) nedir?

Get /set metotları

Yalnız-okunur (read-only) özgen

Yalnız-yazılır özgen

Erişimciler (accessors)

C# sınıf değişkenlerini *veri alanı* (*alan*, *field*) ve *özgen* (*özellik*, *property*) diye ikiye ayırır. Alan (*field*) ana bellekte kendisine bir yer ayrılan normal bir değişkendir. Özgen (*property*) ise alan ile metot arasında bir yeredir. Alan özelliklerini taşır, onunla aynı rolü oynar; yani veri tutan bir bellek adresidir. Bu yönüyle alan sayılır. Öte yandan, metodun bazı özelliklerine sahiptir; tutacağı verinin giriş ve çıkışını bir metot gibi yapar. Bu yönüyle bakınca bir metottur. Bu nedenle, bazı kaynaklar, özgene akıllı alan (*smart field*) derler. Özgen' nin ne olduğunu anlamanın en iyi yolu onu kullanmaktır. Aşağıda o işi yapacağız.

Skorlar.cs

```
using System;
class Skorlar
{
    private int skor;
    public int Skor
    {
        get
        {
            return skor;
        }
        set
        {
            skor = value;
        }
    }
}
```

```
    }  
    }  
}
```

Skorlar sınıfının iki ögesi (class member) var. `private` olan birincisine dışarıdan erişilemez ve alışkın olduğumuz bir yöntemle bildirilmiş:

```
private int skor;
```

Ama `public` nitelemeli ikinci bildirim

```
public int Skor  
{  
    get  
    {  
        return skor;  
    }  
    set  
    {  
        skor = value;  
    }  
}
```

kodlarıyla bildirilmiş. Ancak bu kodlar şimdiye kadar öğrendiklerimizle hemen çözümlenemiyor. Bu bildirim metot bildirimine benziyor, ama metot değil; çünkü `Skor` 'un sonunda `()` metot parantezi yok. Kurucu olmadığı da açık. `Skor` blokunun içinde `get` ve `set` blokları sanki birer metot gibi, ama başlıkları (imzaları) alışılmış metot tanımına uymuyor. `Get` bloku, `'return skor;'` deyimi ile sınıfın `private` nitelemeli değişkenini okuyor. `Set` bloku ise `'skor = value;'` deyimi ile sınıfın `private` nitelemeli değişkenine `'value'` değerini atıyor, ancak `'value'` sınıf içinde tanımlı değil.

Çok karmaşık görünen bu bildirim, adına özgen (property) denilen özel bir tür (*akıllı alan – smart field*) bildiriminden başka bir şey değildir. `private` olan `skor` değişkenine değer atama ve okuma eylemlerini `public` olan `Skor` değişkeni yapıyor. Böylece sınıf dışından erişilemeyen `private` nitelemeli değişkene dışarıdan erişme olanağı yaratılıyor. `set` blokundaki `'value'` C# dilinde bir anahtar sözcüktür. İşlevi burada olduğu gibi dışarıdan erişilemeyen değişkene değer atamaktır. Atayacağı değer, `Skorlar` sınıfına ait bir *nesne* yaratılınca `public` nitelemeli `Skor` değişkenine atanacak değerdir.

Bu bildirimdeki `get` ve `set` bloklarına *getter/setter* metotları ya da erişimciler (accessors) denilir. Türkçe'de *getir/götür* veya *ver/al* diyebileceğimiz özel iki metottur. Anahtar sözcükler gibi kullanıldığı için, biz *getter/setter* sözcüklerini kullanacağız.

Artık `private` değişkenimize değer vermeyi deneyebiliriz.

```
class Uygulama  
{  
    static public void Main()  
    {  
        Skorlar yeniSkor = new Skorlar(); // nesneyi yarat  
        yeniSkor.Skor = 73; // Skor'a setter metodu ile değer ata  
        int ySkor = yeniSkor.Skor; //getter metodu ile Skor'u oku  
  
        Console.WriteLine(" Skor : {0}" , ySkor);  
    }  
}
```

Aşağıdaki program bir sınıfın iki nesnesini yaratıyor ve onların private değişkenlerine erişimcilerle değer atıyor.

Kişi.cs

```
using System;
class Kişi
{
    private int yaş;
    private string doğumYeri;
    public int Yaş
    {
        get
        {
            return yaş;
        }
        set
        {
            if (value <= 65 && value >= 18)
            {
                yaş = value;
            }
            else
                yaş = 18;
        }
    }
    public string DoğumYeri
    {
        get
        {
            return doğumYeri;
        }
        set
        {
            doğumYeri = value;
        }
    }
}

class Uygulama
{
    static void Main(string[] args)
    {
        Kişi Mehmet = new Kişi();
        Kişi Dilek = new Kişi();

        Mehmet.Yaş = 21;
        Mehmet.DoğumYeri = "Malatya";
        Dilek.Yaş = 18;
        Dilek.DoğumYeri = "İzmir";

        Console.WriteLine("Mehmet'in yaşı = {0}, doğum yeri = {1}",
Mehmet.Yaş, Mehmet.DoğumYeri);
        Console.WriteLine("Dilek'in yaşı = {0}, doğum yeri = {1}",
Dilek.Yaş , Dilek.DoğumYeri );
    }
}
```

Çıktı

Mehmet'in yaşı = 21, doğum yeri = Malatya

Dilek'in yaşı = 18, doğum yeri = İzmir

Veri Kapsülleme ve Veri Saklama

Nesne Yönelimli Programlamada veri kapsülleme ve veri saklama çok önemli iki kavramdır. Veri kapsüllemeyi sınıf (class) içinde veya yapı (struct) içinde yaparız. Erişim belirteçleri denilen private, public, protected, internal gibi nitelermelerle, dışarıdan veriye erişimi izne bağlarız. Özgenlerin asıl işlevi, bir sınıfta başkaları tarafından görünmesini istemediğimiz verileri tutmaktır. Bu yönüyle, özgen, güvenliği sağlanmış bir veri alanıdır. (Tabii, her veri alanının bir değişken olduğunu unutmuyoruz.) Sözkonusu güvenlik çok basit bir yöntemle sağlanır. C# bir veri alanındaki veriyi korumaya almak istiyorsa, o veri alanına veri girişini ve girilen veriyi okumayı özgen (property) ile yapar. Yukarıdaki örnekler Veri Kapsülleme (field encapsulation) ve veri saklamaya örneklerdir. Bu konu, ileride yeri geldikçe tekrar ele alınacaktır.