

Bu bölümde çok fazla kullanılmayan ancak bilinmesinde faydası anahtar kelimeleri inceleyeceğiz. Özellikle ref ve out anahtar kelimelerini iyi öğrenmenizi tavsiye ederim. İyi bir yazılımcı olma yolunda ilerlediğiniz sürece belki de bu anahtar kelimelere ihtiyaç duyacaksınız.

Metodlar 3. Bölüm

Ref, out, in ve params anahtar kelimeleri

Oğuzhan Karagüzel

İçindekiler

ŞEKİLLER DİZİNİ.....	2
GİRİŞ	3
METODLAR 3. BÖLÜM	3
1. REF ANAHTAR KELİMESİ.....	3
1.1. REF NEDİR?	3
1.2. REF KULLANIM KURALLARI	4
1.3. REF İLE ÖRNEK KULLANIM	4
2. OUT ANAHTAR KELİMESİ	5
2.1. OUT NEDİR?	5
2.2. OUT KULLANIM KURALLARI	5
2.3. OUT İLE ÖRNEK KULLANIM	5
3. IN ANAHTAR KELİMESİ	6
3.1. IN NEDİR?	6
3.2. IN KULLANIM KURALLARI	6
3.3. IN İLE ÖRNEK KULLANIM	6
4. PARAMS ANAHTAR KELİMESİ.....	7
4.1. PARAMS NEDİR?	7
4.2. PARAMS KULLANIM KURALLARI	7
4.3. PARAMS İLE ÖRNEK KULLANIM	8

ŞEKİLLER DİZİNİ

Şekil 1 ref kullanımı	4
Şekil 2 out kullanımı	5
Şekil 3 in kullanımı	6
Şekil 4 in anahtar kelimesi kullanımında alınan hata.....	7
Şekil 5 Params kullanımı	8

Öğuzhan KARAGÜZEL

GİRİŞ

Metodların bu bölümünde önceki konularımızdan bir tanesini tekrar etmenizi istiyorum. Değişkenler ve operatörler dersinde detaylıca incelemiştik. O bölümde değişkene verdiğimiz adın bir adresi temsil ettiğiniz söylemiştik. Eğer o dersi hatırlamıyorsanız lütfen tekrar edin. Bu bölümde “ref ve out” anahtar kelimelerini daha iyi anlayabilmek için faydalı olacaktır.

METODLAR 3. BÖLÜM

C# programlama dilinde metotlar, belirli bir işlevi yerine getiren kod bloklarıdır. Metotlar, programın tekrar eden işlemlerini düzenli bir şekilde yazmamıza yardımcı olur. Ancak, metotlar yalnızca basit giriş ve çıkış işlemleriyle sınırlı değildir. C# dilinde, parametreleri farklı şekillerde yönetmek için kullanılan özel anahtar kelimeler bulunmaktadır. Bu dökümanda ref, out, in ve params anahtar kelimelerini derinlemesine inceleyeceğiz.

Bir önceki bölümde argüman alan ve geriye değer döndüren metotları incelemiştik. Yine aynı biçimde argüman alacak ve değer geriye döndüreceğiz. Bu daha etkin yapabilmek için “ref, out, params ve in” anahtar kelimelerini inceleyeceğiz. Nesne tabanlı programlamanın faydaları bu anahtar kelimelerin kullanımı giderek azalmıştır. İhtiyaç azalmıştır.

Ancak programınızı ultra optimize etmek istediğinizde ya da eskiden yazılmış bir programa dahil olduğunuzda bu anahtar kelimeler ile karşılaşacaksınız ya da kullanmaya başlayacaksınız.

1. REF ANAHTAR KELİMESİ

1.1. REF NEDİR?

ref anahtar kelimesi, bir değişkenin metot içinde değiştirilmesine ve değişikliğin çağırılan metoda geri yansımaya olanak tanır. ref kullanıldığında, değişkenin bellekteki adresi metoda referans olarak iletilir.

Kısaca şu şekilde; bir değişken oluşturduğunuzda, örnek olarak;

“int sayi = 5;” bu değişkeni bir metoda gönderdiğinizde şu gerçekleşir. “sayi” adlı değişkenin adresi 0xa96c gibi bir değer ise. metoda yolladığınız zaman bu adresteki değer metotta ki değişkene kopyalanır.

```
public int KaresiniAl(int sayi);
```

Bu metotta ki “sayi” değişkeninin adresi ise 0xb548 gibi bambaşka bir değer. Ancak bu adresin içine 0xa96c adresindeki değer kopyalanır. Her iki adresin değeri o an aynı olur. Karesini al metodunun içerisinde ki işleme göre

Örnek ;

```
public int KaresiniAl(int sayi);
```

```

{
    sayi *= sayi;
    return sayi;
}

```

0xb548 adresinde ki değer 25 olacaktır ve bu adresteki değer geriye döndürülecektir. Ancak 0xa96c adresindeki değerde bir değişim olmayacaktır.

Ancak ref anahtar kelimesini kullanırsak işte o durumda metoda sadece 0xa96c adresindeki değeri değil doğrudan adresi göndermiş oluruz. Yani hem metot dışındaki “sayi” değişkenimiz hem de metot içindeki “sayi” değişkenimiz tamamı ile aynı adrestir. Her iki değişkende aslında aynı değişkendir. Adları farklı olabilir. Birisi “sayi” diğer “kok” gibi olabilir. Ancak her iki değişkende aynı adresi gösterir. Diğer bir adrese değer kopyalama gerçekleşmez.

Bunu nesne tabanlı programlamada daha iyi anlayacaksınız.

1.2. REF KULLANIM KURALLARI

- ref ile kullanılan değişkenin, metot çağrılmadan önce başlatılmış olması gerekir.
- Değişken, metot içinde değiştirilirse, bu değişiklik değişkene yansır.

1.3. REF İLE ÖRNEK KULLANIM

```

static void Main()
{
    int sayi = 10;
    Console.WriteLine("Metot çağrılmadan önce: " + sayi);

    //metoda referans yollanacak argüman ref anahtar kelimesi ile yollanır
    RefKullanimi(ref sayi);

    Console.WriteLine("Metot çağrıldıktan sonra: " + sayi);
}

/// <summary>
/// metotta referansı istenen parametre ref anahtar kelimesi ile belirtilir
/// </summary>
/// <param name="deger"></param>
static void RefKullanimi(ref int deger)
{
    deger *= 2;
}

```

Şekil 1 ref kullanımı

Çıktı:

Metot çağrılmadan önce: 10

Metot çağrıldıktan sonra: 20

Yukarıdaki kodda, sayi değişkeni ref olarak metoda iletilmiş ve metot içinde değiştirildiğinde çağırılan metodun da değeri güncellenmiştir.

Şekil 1’de de anlatıldığı üzere ref anahtar kelimesinin kullanımı oldukça basittir. Metodunuzun parametrelerini belirlerken referansını yani adresini istediğiniz parametreyi ref anahtar kelimesi ile işaretlersiniz.

Metoda argüman yollarken referansını yollayacağınız argümanı ref ile işaretlersiniz.

Bu sayede ram’de sadece bir adres ve bir değer olur. Eğer ref anahtar kelimesi kullanmazsanız 2 adres ve 2 ayrı değer olur.

Özellikle nesne tabanlı programlama dillerinde genellikle Garbage Collector (çöp toplayıcı) bulunur. Bu sayede ram yönetimini düşünmenize gerek kalmaz. Ref anahtar kelimesi ile ram’de daha az alan kaplamayı sağlarız. Ancak bu gerekmedikçe uğraşmaya değmeyecek bir performans optimizasyonundan ibarettir.

2. OUT ANAHTAR KELİMESİ

2.1. OUT NEDİR?

out anahtar kelimesi, metot içinde bir değerın ayarlanmasını ve çağırın metoda döndürölmesini sağlar. ref ile benzerlik gösterse de önemli bir farkı vardır:

- ref kullanırken değışkenin önceden başlatılması gerekirken,
- out ile kullanılan değışken, metot içinde başlatılmalıdır.

2.2. OUT KULLANIM KURALLARI

- out parametresi, metot içinde mutlaka bir değeri almalıdır.
- out parametresi çağırın metotta önceden başlatılmış olmak zorunda değildir.
- Metot, birden fazla değeri out parametreleri ile döndürebilir.

2.3. OUT İLE ÖRNEK KULLANIM

```
static void Main()
{
    int sonuc;
    Topla(5, 10, out sonuc);
    Console.WriteLine("Toplam: " + sonuc);
}

static void Topla(int a, int b, out int toplam)
{
    toplam = a + b;
}
```

Şekil 2 out kullanımı

Çıktı:

Toplam: 15

Burada, sonuc değişkeni başlangıçta bir değer almamış olmasına rağmen, out kullanıldığı için metot içinde atanmış ve çağırın metotta güncellenmiştir.

Görüldüğü üzere ref anahtar kelimesi ile kullanımı neredeyse aynıdır. out anahtar kelimesi genellikle birden fazla geriye değer döndürmek için kullanılır. Ancak nesne tabanlı programlamanın getirdiği özellikler ile yine bu anahtar kelimenin kullanımına gerek kalmamıştır. Ayrıca C# programlama dilinin yeni bir özelliği olan tuple'lar (demetler) ile neredeyse bitti diyebiliriz. Ancak hala daha performans optimizasyonu için kullanılabilir.

“int” yapısının metodlarını inceleyin. Değişkenler ve operatörler dersinde “.” Operatörü ile yapıların içerisine ulaşabilirsiniz. Sayı dönüşümü için “Convert” kullanımını hatırlayın. int yapısının TryParse metodunu araştırın. Kullanımını öğrenin Bu metod iki değer geriye döndürmektedir. Birisi bool bir değer ve verdiğiniz değer dönüştürülüp dönüştürülemediğini belirtir. Out parametresini kullanarak dönüşüm gerçekleşirse dönüşmüş sayıyı elde edersiniz.

Bu size verilmiş bir araştırma ödevidir. Bu metodu kullanmaya çalışın.

3. IN ANAHTAR KELİMESİ

3.1. IN NEDİR?

in anahtar kelimesi, bir parametrenin yalnızca okunabilir olmasını sağlar. in ile iletilen değişken metot içinde değiştirilemez.

İlerleyen aşamalarda “const – readonly” gibi anahtar kelimeleri öğreneceksiniz. “in” anahtar kelimesini iyi anladığınızda bu anahtar kelimeleri anlamakta hiç zorluk çekmeyeceksiniz.

Asıl özelliği şudur. Bir metotta “in” anahtar kelimesi ile işaretlenmiş bir parametrenin değeri metot içerisinde değiştirilemez. Peki değiştirilememesi ne işimize yarar? Şu an çok fazla bir işinize yarayacak değil. Aslında iş hayatım boyunca bu anahtar kelimenin kullanıldığını hiç görmedim. Ancak bilmeniz faydanıza olacaktır.

3.2. IN KULLANIM KURALLARI

- in parametresi yalnızca okunabilir, değiştirilmesi engellenmiştir.
- Performans açısından özellikle büyük veri yapıları (struct) için avantajlıdır.

3.3. IN İLE ÖRNEK KULLANIM

```
static void Main()
{
    int sayi = 10;
    Yazdir(in sayi);
}

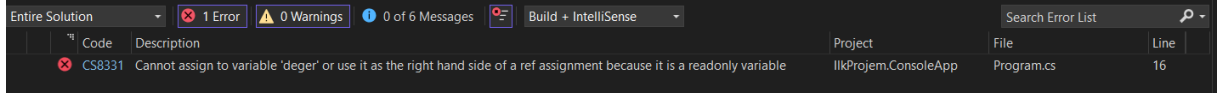
static void Yazdir(in int deger)
{
    Console.WriteLine("Sayı: " + deger);
    deger = 20; // HATA! 'in' ile değiştirilemez.
}
```

Şekil 3 in kullanımı

Çıktı:

Sayı: 10

Burada in parametresi ile gelen deger deđiřtirilemez ve yalnızca okunabilir. Bu metodu olduđu gibi yazdığınızda hatadan bu durumu anlayabilirsiniz.



Şekil 4 in anahtar kelimesi kullanımında alınan hata.

4. PARAMS ANAHTAR KELİMESİ

4.1. PARAMS NEDİR?

params anahtar kelimesi, deđişken sayıda parametre alan metotlar yazmamızı sağlar. Tek bir dizi (array) olarak ele alınır.

Çok kullanılan bir anahtar kelime deđildir. İlk başta kafa karıştııcı gibi görünür. Birkaç kez kullanımdan sonra hemen öğrenilir. metodun parametrelerinin en sonuncusu olabilir, bir kere kullanılabilir. Metoda argüman gönderirken array'i önceden tanımlamak yerine metodun içerisine "," kullanarak array tanımlamadan argümanları gönderebilirsiniz. Anlamsız gelebilir. Haksız sayılmazsınız. Ancak bilmenizde fayda var.

4.2. PARAMS KULLANIM KURALLARI

- params yalnızca bir metot başına bir kez kullanılabilir.
- params parametresi her zaman son parametre olmalıdır.
- params yerine normal bir dizi (array) de kullanılabilir ancak params sayesinde metodun çağırılma şekli daha esnektir.

4.3. PARAMS İLE ÖRNEK KULLANIM

```
static void Main()
{
    ToplamYazdir(1, 2, 3, 4, 5);
    ToplamYazdir(10, 20);
}

static void ToplamYazdir(params int[] sayilar)
{
    int toplam = 0;
    foreach (int sayi in sayilar)
    {
        toplam += sayi;
    }
    Console.WriteLine("Toplam: " + toplam);
}
```

Şekil 5 Params kullanımı

Çıktı:

Toplam: 15

Toplam: 30

Burada params sayesinde metodu çağırırken istediğimiz kadar parametre verebiliriz.

Bu dökümanda ref, out, in ve params anahtar kelimelerini detaylı olarak inceledik. Bu anahtar kelimeler, metodun çalışma şeklini özelleştirmek için büyük avantajlar sağlar. Özellikle büyük projelerde daha esnek ve performanslı kod yazmanıza yardımcı olur.

Bu anahtar kelimeler C#'ın yeni versiyonları ile