



CONSOLE EKRANI-2

CONSOLE EKRANI

- ✓ *Değişken Veri Tipleri*
- ✓ *Sayısal İfadeler*
- ✓ *Algoritma yapısı*

KAZANIMLAR

- ✓ *Tür dönüşümlerinin nasıl yapıldığını bilir.*
- ✓ *Değişken veri tiplerini öğrenir.*
- ✓ *Değişkenleri türlerine uygun tanımlar.*
- ✓ *Operatörlerin kullanımını öğrenir.*



CONSOLE ÖRNEKLERİ

Veri Tipleri

Hafıza Alanı 1
Hafıza Alanı 2
Hafıza Alanı 3
Hafıza Alanı 4
Hafıza Alanı 5

Ali	Ad
Barış	Soyad
Gül	AnneAd
Veli	BabaAd
Konya	DogumYeri

Bilgisayarda tutulacak değerlerin, verilerin özellikleri farklı olduğundan veriler veri tiplerine (türlerine) ayrılmıştır. Örneğin, sayısal değerleri tutmak için ayrı, metin bilgilerini tutmak için ayrı veri tipleri kullanılır. Tam sayılar için **integer** veri tipinin kullanılması uygunken, “Millî Eğitim Bakanlığı” ifadesi için **string** (dize, dizgi) veri tipinin, günün tarihini tutmak için **Date** veri tipinin kullanılması gerekir. Değişken tanımlama işlemi sırasında değişkene bir isim verilirken, değişkenin hangi tipte veri tutacağı da belirtilir. Benzer veri tipleri arasında işlemler yapılabilir. Örneğin, “**integer**” veri tipindeki 2 değişken içindeki değerler toplanabilir, fakat içeriği “**Ahmet**” olan “**string**” veri tipli bir değişkenle, içeriği 20 olan “**integer**” veri tipli bir değişken toplanamaz.

Değişken türleri	Boyut (Byte)	Değer Aralığı
byte	1	0-255
short	2	-32768 - 32767
int	4	-2.147.483.648 - 2.147.483.647
long	8	- 9.223.372.036.854.775.808 - 9.223.372.036.854.775.807
sbyte	1	-128 - 127
ushort	2	0 - 65535
uint	4	0 - 4.294.967.295
ulong	8	0 - 18.446.744.073.709.551.615
float	4	$\pm 1.5 \times 10^{-45}$ - $\pm 3.4 \times 10^{38}$
double	8	$\pm 5.0 \times 10^{-324}$ - $\pm 1.7 \times 10^{308}$
object		
char	2	Bir Unicode karakter
string		Karakterlerin tümü
decimal	16	$\pm 1.0 \times 10^{-28}$ to $\pm 7.9 \times 10^{28}$
bool	2	True veya False

1. Tam Sayı Veri Tipleri

Tam sayılar ondalık kısım içermeyen sayılardır. Örneğin, 10, 2450, -45, 245678 sayıları tam sayılardır. Gündelik hayattan tam sayı değerler içeren durumlar aşağıda sıralanmıştır:



Sınıf mevcudu, okuldaki öğrenci sayısı, bir sınıftaki kız öğrenci sayısı, boy uzunluğu (cm olarak, örneğin 175 cm).

2. Ondalık Sayı Veri Tipleri

6.8, 3.56, 6.9876 gibi sayılar ondalık sayıları oluşturmaktadır. Yaşantımızda birçok ondalık sayı içeren durumlar vardır. Örneğin; Boy uzunluğu (m olarak, örneğin 1.75 m), maaş miktarı (930.25 YTL)

Ali Ad

Barış Soyad

Gül AnneAd

Veli BabaAd

Konya DogumYeri

3. String Veri Tipi

String, bir karakter dizisini, grubunu ifade eder. Her türlü karakter grubu bir string oluşturabilir. Örneğin, "Seda", "Ev Adresiniz", "235" ifadeleri.

"235" ifadesi tam sayı tipinde mi yoksa bir string tipinde mi diye düşünülebilir? Eğer bir sayı üzerinde matematiksel işlemler yapılmayacaksa o string tipinde tanımlanabilir.

Örneğin, telefon numaraları tamamen sayısal ifadelerden oluşur fakat telefon numaraları üzerinde matematiksel işlemler yapmayız. Hiçbir zaman iki telefon numarasının toplanması, çıkarılması ihtiyacı duyulmamıştır. Okul öğrenci numaraları da birer sayı olmalarına rağmen, bu numaralar üzerinde de aritmetiksel işlemler yapılmaz. Bu nedenle telefon numarasını ve öğrenci numarasını tutacak değişken string tipinde tanımlanabilir. Hangi tipte tanımlandığında hafızada az yer kaplıyorsa o tip seçilmelidir.

Ayrıca sayılardan oluşan bir string tipindeki değişken ile tam sayı tipindeki bir değişken arasında aritmetik bir işlem yapılamaz. Çünkü değişkenlerin veri tipleri birbirinden farklıdır.

4. Char Veri Tipi

Char (Character) veri tipi, bir karakterlik veri tutulmasını sağlar. Bilgisayardaki her harf, rakam, sembol (a, C, d, 1, 7, ^, \$, [, ?) karakter olarak isimlendirilir.

Bilgisayar dünyasında kelimeler arasındaki boşluklar da karakter olarak değerlendirilir. Boşluk karakteri de diğer karakterler gibi bilgisayarın hafızasında yer kaplar. Char veri tipindeki bir değişkenin içeriğinde ara (boşluk) tuşuna basılarak üretilen boşluk karakteri tutulabilir.



5. Boolean Veri Tipi

Gündelik yaşamda doğru-yanlış, açık-kapalı, evet-hayır, 1-0 gibi birbirinin karşıtı olan durumlar vardır. Boolean veri tipi bu tür durumlarda tercih edilir. Boolean veri tipi sadece True (Doğru) ve False (Yanlış) değerlerinden birini alabilir. Örneğin, "onay" isimli bir değişkenin değeri ya "True"dur, ya da "False"dur. Bu değişken iki değere aynı anda sahip olamaz, örneğe göre düşünersek bir işlem ya onaylanmıştır ya onaylanmamıştır.

Örnek1: C# kesirli sayıları, aksi söylenmediği zaman double tipinden sayar. Bu öndeğer (default) durum istenmiyorsa, kesirli sayının sonuna *f* yada *F* yazılır. Aşağıdaki örnek, float veri tipinin kullanımını göstermektedir.

using System;

namespace ConsoleApp1

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 3;
            float y = 4.5f;
            short z = 5;
            Console.WriteLine("{0}*{1}/{2}={3}", x, y, z, x * y / z);
        }
    }
}
```

Örnek2: Kullanıcıdan ad soyad isteyen programlama:

string ad, soyad;

```
Console.WriteLine("Adınız nedir?");
ad = Console.ReadLine();
Console.WriteLine("Soyadınız nedir?");
soyad = Console.ReadLine();
Console.WriteLine("Merhaba " + ad + " " + soyad);
```



Tür Dönüşümleri

Program içerisinde değişkenlerle ilgili tür dönüşümleri yapmak durumunda kalabilirsiniz. Örneğin, sayısal bir veriyi string değişken türüne, string bir veriyi sayısal değişken türüne dönüştürmeniz gerekebilir.

Convert

Convert metoduyla dönüştürme işleminde derleyici tarafından izin verilen tüm türlere dönüştürme işlemi yapılabilir. Dönüştürme işlemi yapmadan önce dönüştürülecek bilginin hangi türlere dönüştürülebileceğine dikkat edilmelidir. Convert metodunu kullanırken; Convert yazıp nokta karakteri yazıldığında açılan menüden istenilen dönüşüm türü seçilerek dönüştürülecek bilgi parantez içinde yazılmalıdır.

Örnek:

Int x;

X= Convert.ToInt32(textBox5.Text);

Örnek3:Kullanıcıdan girilen iki sayının toplamını veren program satırları

int a, b,c=0;

```
Console.WriteLine("1.sayıyı giriniz");  
a = Convert.ToInt32(Console.ReadLine());  
Console.WriteLine("2.sayıyı giriniz");  
b = Convert.ToInt32(Console.ReadLine());  
c = a + b;  
Console.WriteLine("toplam:" + c);
```

Kod Açıklamaları

a = Convert.ToInt32(Console.ReadLine()); → a isminde sayısal türdeki değişkenimi direk direk a=Console.ReadLine(); yazamadım.Console ekranında okuduğu satırı int türüne çevirip değişkenimin içine atmasını istiyorum.Eğer değişkenim string türünde olsaydı,tabiki tanımlayabilirdim.

c=a+b; → Matematikteki toplama işlemini bilgisayara tanımlıyoruz değişkeninin içerisine a ve b değerlerinin toplamını ata demek istiyoruz.



`Console.WriteLine("toplam:" + c);` → Bu kod satırı çift tırnak içerisine ne yazarsam karakter, yazı hepsini console ekranına yazdırır.” +c “ derken c değişkenimi de ekrana yazdırmasını istiyorum. Yani çift tırnak içerisine yazdığım metinle c değişkenini birleştirmek için + kullanıyorum.

Aritmetiksel Operatörler

Bu operatörler aritmetik işlemlerinde, sayılarla veya sayı tutan ifadelerle kullanılır.

✓ Çarpma `int sayi = 100;`

`sayi = 200 * 2;`

✓ Bölme `double bolum;`

`bolum = sayi / 23;`

✓ Çıkarma `int sonuc = bolum - 100;`

✓ Toplama `int toplam;`

`toplam += sonuc;` // Bu ifade, “`toplam = toplam + sonuc`” ile aynı anlama gelir



İpucu

Aritmetik operatörleri, eşittir ifadesi ile beraber kullanılırsa, işlem değişkenin kendisi ile yapılır.

✓ Mod alma `int kalan = toplam % 42;` // Sonuç, toplam değişkenindeki değerin 42 ile bölümünden kalan sayıdır.

Örnek4: Kullanıcının girmiş olduğu değerlere göre dikdörtgenin alan ve çevresini hesaplama

```
Console.WriteLine("-----Dikdörtgen Alanı Hesaplama-----");
int a, b, cevre, alan;
Console.WriteLine("Kısa Kenar degerini giriniz");
a = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Uzun Kenar degerini giriniz");
b = Convert.ToInt32(Console.ReadLine());
alan = a * b;
cevre = 2*(a+b);
Console.WriteLine("Dikdörtgenin Cevresi: " + cevre);
Console.WriteLine("Dikdörtgenin Alanı: " + alan);
Console.ReadKey();
```



Mantıksal Operatörleri

Bu operatörler veri tiplerini birbirleriyle karşılaştırmak için kullanılır. Bu operatörler ile yapılan işlemlerin

sonucunda true ya da false değeri döner. Karşılaştırma operatörleri yalnızca sayı tipleri üzerinde yapılmaz.

✓ Küçük double sayı = 1.5;

float sayı2 = 1.3;

sayı2 < sayı // Sonuç: True

- ✓ Küçük Eşit sayı2 <= sayı // Sonuç: True
- ✓ Büyük sayı2 > sayı // Sonuç: False
- ✓ Büyük Eşit sayı2 >= sayı // Sonuç: False
- ✓ Eşit sayı2 == sayı // Sonuç: False
- ✓ Eşit Değil sayı2 != sayı // Sonuç: True

Örnek6 :Görmüş olduğunuz kod bloklarına göre ekrana ne çıkar?

```
using System;
```

```
namespace ConsoleApp1
```

```
{
```

```
    class Program
```

```
    {
```

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

```
        {
```

```
            bool a = 4 < 6;
```

```
            bool b = 6 > 5;
```

```
            bool c = 7 <= 7;
```

```
            bool d = 9 >= 12;
```

```
            bool e = 10 == 12;
```

```
            bool f = 1 != 8;
```

```
            Console.WriteLine("{0}\n{1}\n{2}\n{3}\n{4}\n{5}", a, b, c, d, e, f);
```

```
        }
```

```
    }
```

```
}
```



Koşul Operatörleri

Veri tipleri ve değişkenler üzerinde kontrol yapılırken birden fazla koşula ihtiyaç duyulabilir. Bu durumda, koşulları birbirleriyle karşılaştıracak operatörler kullanılır. Bu kontrollerden dönen değerler Boolean tipinde olduğu için, koşul operatörleri de bu değerler üzerinde işlem yaparlar.

&& (And)

Bu ifade, verilen koşulların kesişimini alır. Eğer tüm koşulların değeri true ise sonuç da true olur. En az bir tane false değerli koşul varsa, sonuç false olur.

|| (Or)

Bu ifade, verilen koşulların birleşimini alır. Eğer tüm koşulların değeri false ise sonuç false olur. En az bir tane true değeri varsa sonuç true olur.

! (Değil)

Bir koşulun değerini tersine çevirir. Koşul false ise true, true ise false olur.

Örnek7 :Görmüş olduğunuz kod bloklarına göre ekrana ne çıkar?

```
using System;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            bool b1 = 35 > 10 && 10 == 50;
            bool b2 = 35 > 10 && 10 != 50;
            bool b3 = 5 is int || 12 * 3 == 200;
            bool b4 = 5 < 4 || 45 / 5 == 9;
            bool b5 = !(5 < 4);
            Console.Write(b1 + " " + b2 + " " + b3 + " " + b4 + " " + b5);
        }
    }
}
```


