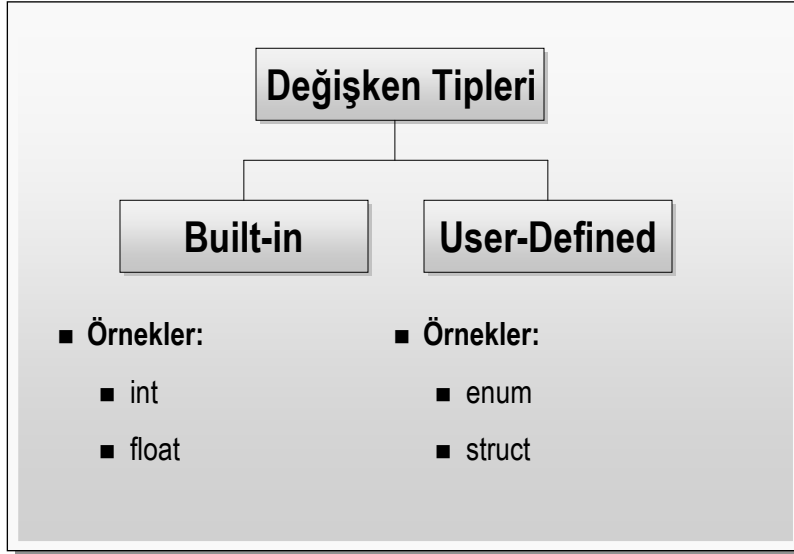


Tip Deęişkenleri (Type Variables)

Özet

- Deęişkenlere isim verme
- Built-in Data Tipleri
- User-Defined Data Tipleri
- Data Tipleri arasında dönüştürme

Built-in ve User-Defined Değişken Tipleri



Basit Veri Tipleri

- **Ayrılmış Kelimelerle Tanımlanır**
 - int // Reserved keyword
 - veya -
 - **System.Int32**

◆ Değişkenlere İsim Verme

- İsim Vermede Gerekli Kurallar ve Öneriler
- C# Anahtar Kelimeleri

Kurallar ve Öneriler

■ Rules

- İlk karakter harfler, altçizgi
- Daha sonraki karakterler rakamda olabilir

■ Öneriler

- Tümü büyük harf olmasın
- Başlangıç alt çizgi olmasın
- Birden fazla kelimedede her kelimenin baş harfi büyük olsun
- Kısaltmalardan kaçınınız

Cevap42 ✓
42Cevap ✗

fark ✓✓
Fark ✓✓

KOTUSONUC ✗
_KotuSonuc ✗
KotuSonuc ✓✓
kotuSonuc ✓✓

Msj ✗
Mesaj ✓✓

C# Anahtar Kelimeler

- Keyword'ler ayrılmış tanımlayıcılardır

```
abstract, base, bool, default, if, finally
```

- Keyword'ler deęişken adı olamazlar

- compile-time hatası üretir

- Keyword'ler tanımlandığı gibi kullanılmalıdır.
(Case-Sensitive)

```
int INT; // Kötü Kullanım
```

Soru: Geçerli olmayan veya önerilmeyen tanımlamaları bulunuz.

1

2

3

4

5

◆ Built-in Data Tiplerinin Kullanımı

- Değişken Tanımlama
- Değişkenlere Değer Atama
- Compound Atama
- Operatörler
- Değer Artırma ve Azaltma

Lokal Değişken Tanımlama

- Genellikle data tipi ve değişken adıyla tanımlanır.

```
int itemCount;
```

- Birden fazla değişken aynı tanımlamada belirtilebilir.

```
int itemCount, employeeNumber;
```

```
int itemCount,  
employeeNumber;
```

Değişkenlere Değer Atama

■ Tanımlanmış değişkene değer atama:

```
int employeeNumber;  
employeeNumber = 23;
```

■ Tanımlanırken başlangıç değeri atama:

```
int employeeNumber = 23;
```

■ Karakter tipindeki değişkene başlangıç değeri atama:

```
char middleInitial = 'J';
```

Compound Atama

■ Bir değişkene değer atama

```
itemCount = itemCount + 40;
```

■ Daha kısa yazımlar

```
itemCount += 40;
```

```
itemCount -= 24;
```

Genel Operatörler

Genel Operatörler	Örnek
• Eşitlik operatörleri	== !=
• İlişki operatörleri	< > <= >= is
• Şart operatörleri	&& ?:
• Artırma operatörü	++
• Azaltma operatörü	--
• Aritmetik operatörler	+ - * / %
• Atama operatörleri	= *= /= %= += -= <<= >>= &= ^= =

Artırma ve Azaltma

■ Birer birer artırma veya azaltma

```
itemCount += 1;  
itemCount -= 1;
```

■ Daha kısa yazım

```
itemCount++;  
itemCount--;
```

```
++itemCount;  
--itemCount;
```

◆ Aritmetik İşlemler

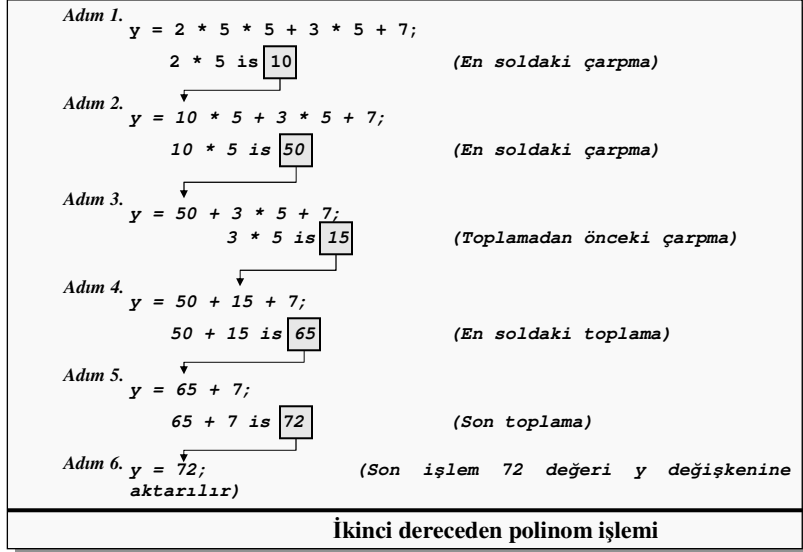
İşlem	Aritmetik operatör	Cebirsel ifade	C# ifade
Ekleme	+	$f + 7$	$f + 7$
Çıkarma	-	$p - c$	$p - c$
Çarpma	*	$b.m$	$b * m$
Bölme	/	x / y	x / y
Modlu bölme	%	$r \text{ mod } s$	$r \% s$

◆ İşlem Öncelikleri

■ Sıralama

- Parantez
- Bölme, Çarpma ve Modlu Bölme
- Toplama ve Çıkarma

◆ İşlem Öncelikleri (Örnek)



```
1 // Fig. 3.11: Addition.cs (Page 72)
2 // An addition program.
3
4 using System;
5
6 class Addition
7 {
8     static void Main( string[] args )
9     {
10         string firstNumber, // first string entered by user
11             secondNumber; // second string entered by user
12
13         int number1, // first number to add
14             number2, // second number to add
15             sum; // sum of number1 and number2
16
17         // prompt for and read first number from user as string
18         Console.Write( "Please enter the first integer: " );
19         firstNumber = Console.ReadLine();
20
21         // read second number from user as string
22         Console.Write( "\nPlease enter the second integer: " );
23         secondNumber = Console.ReadLine();
```

```
24
25     // convert numbers from type string to type int
26     number1 = Int32.Parse( firstNumber );
27     number2 = Int32.Parse( secondNumber );
28
29     // add numbers
30     sum = number1 + number2;
31
32     // display results
33     Console.WriteLine( "\nThe sum is {0}.", sum );
34
35 } // end method Main
36
37 } // end class Addition
```

◆ User-Defined Data Tipi Tanımlama

- Enumeration Tipleri
- Structure Tipleri

Enumeration Tipleri

■ Enumeration Tip Tanımlama

```
enum Color { KIRMIZI, YEŞİL, MAVİ }
```

■ Enumeration Tip Kullanımı

```
Color colorPalette = Color.KIRMIZI;
```

■ Enumeration Değişken Yazdırma

```
Console.WriteLine("{0}", colorPalette); //KIRMIZI yazar
```

Structure Tipleri

■ Structure Tip Tanımlama

```
public struct Person  
{  
    public string ad;  
    public int yas;  
}
```

■ Structure Tip Kullanımı

```
Person companyPerson;  
companyPerson.ad = "Mehmet";  
companyPerson.yas = 23;
```

◆ Data Tiplerini Dönüştürme

- Implicit Data Tipi Dönüştürme
- Explicit Data Tipi Dönüştürme

Implicit Data Tipi Dönüştürme

- int değeri long değere dönüştürme:

```
using System;
class Test
{
    static void Main( )
    {
        int intValue = 123;
        long longValue = intValue;
        Console.WriteLine("(long) {0} = {1}", intValue,
        ↪longValue);
    }
}
```

Explicit Data Tipi Dönüştürme

- cast expression kullanılır:

```
using System;
class Test
{
    static void Main( )
    {
        long longValue = Int64.MaxValue;
        int intValue = (int) longValue;
        Console.WriteLine("(int) {0} = {1}", longValue,
        ↪intValue);
    }
}
```