

7.1- Operatörler ve Giriş-Çıkış İşlemleri

7- PROGRAMLAMAYA GİRİŞ

7.10- Paketleme (BOXING) : (Veriyi yeni tipe uygunlaştırma işlemi)

```
int  ort = (int) 3.14; // 3 olur
```

```
int  n= 5/ 2 ;      // 2 olur
```

```
double  x= 5/ 2 ;   // 2 olur
```

```
double  x= 5.0/ 2 ; // 2.5 olur
```

```
double  x= 5/ 2.0 ; // 2.5 olur
```

```
Char  c = (Char) 65 ; // c' nin değeri A olur
```

7- PROGRAMLAMAYA GİRİŞ

7.11- OPERATÖRLER:

Atama

= ata (x=2;)
+= topla ata
-=
*= çarp ata
/=
%=
>>=
<<=

Aritmetik

+
-
*
/
%

İlişkisel

>
>=
<
<=
==
!=

Mantıksal

&& and
|| OR
! Not

Bit

& ve
| veya
~ değil
^ xor
>> sağa kaydır
<< sola kaydır

Unary

++
--

7- PROGRAMLAMAYA GİRİŞ

```
int x = 5 ;  
x += 4 ; // x ' 4 ekle. Yani x = x + 4; tür.  
x %=2 ; // x ' deki değer mod 2 yapılır ve sonuç x ' e atanır. Yani x = x %2;  
x >>=1 ; // x ' deki değer in tüm bitleri bir kez sağa kaydırılıp sonuç x ' e atanır.
```

```
bool devam = true;  
string mail = "milkucar@gmail.com" ;  
char dogrumu = 'e' ;
```

```
if ( x>5 ) ..... ; // x 5 ' den büyük ise  
if ( x>5 && x<10 ) ..... ; // x 5 ' den VE 10 'dan küçük ise  
if ( x>5 || x<10 ) ..... ; // x 5 ' den VEYA 10 'dan küçük ise  
if ( !(x>5) ) ..... ; // x 5 ' den büyük DEĞİLSE  
if ( x != null ) ..... ; // x null (hiç) ' den FARKLI ise
```

```
int sayac = 0;  
sayac ++;  
++ sayac ;
```

```
int say = 0;  
say --;  
-- say ;
```

7- PROGRAMLAMAYA GİRİŞ

7.11- OPERATÖRLER:

```
int x= 5 + 8;
```

```
x += 6; // x = x + 6;
```

```
int y;
```

```
y++; ++y; // y = y + 1;
```

```
x++;
```

```
++x;
```

```
x = x + 1;
```

```
x += 1;
```

```
//Hepsi aynı işi yapar
```

```
int k=5;
```

```
int m=k++;
```

<u>k</u>	<u>m</u>
6	5

```
int k=5;
```

```
int m=++k;
```

<u>k</u>	<u>m</u>
6	6

7- PROGRAMLAMAYA GİRİŞ

OPERATÖRLER:

```
if( x> 5) Console.Write("A");
```

```
if( x> 5 && x<= 7) Console.Write("B");
```

```
if( x != 3) Console.Write("A");
```

```
if( harfnot=="DC" || harfnot=="DD")  
Console.Write("şartlı");
```

7- PROGRAMLAMAYA GİRİŞ

? : operatörü (if)

```
int x= (y>5 ) ? 4 : 7 ; // y> 5 ise x ' 4 aktarılır değilse x ' e 7 aktarılır.
```

```
if ( y>5 )  
    x = 4;  
else  
    x= 7;
```

```
int k = 4 ;  
(y>5 ) ? k=k+5 : k=k*5 ;
```

```
if ( y>5 )  
    k= k +5;  
else  
    k= k *5;
```

7- PROGRAMLAMAYA GİRİŞ

7.11- & (AND) bitsel operatörü doğruluk tablosu

AND (&)
Doğruluk tablosu

&	0	1
0	0	0
1	0	1

`int z = 5 & 3; // 1 olur`

5 -> 0 1 0 1
3 -> 0 0 1 1

0 0 0 1 -> (1)₁₀

Örn: x'in üçüncü bitinin 1 olup olmadığını test etmek

`if((x & 00100) > 0) ise x'in üçüncü biti 1 dir`

x -> b b b b b b
3 -> 0 0 0 1 0 0

y -> b 1 ise y 1 olur, b 0 ise y 0 olur

7- PROGRAMLAMAYA GİRİŞ

7.11- | (OR) bitsel operatörü doğruluk tablosu

OR (|) Doğruluk tablosu

	0	1
0	0	1
1	1	1

```
int x = 5 ;  
int y = 3 ;  
int z = 5 | 3; // 7 olur
```

```
5 -> 0 1 0 1  
3 -> 0 0 1 1  
-----  
      0 1 1 1 -> 7
```

7- PROGRAMLAMAYA GİRİŞ

7.11- ^ (XOR) bitsel operatörü doğruluk tablosu

XOR (^) Doğruluk tablosu

\wedge	0	1
0	0	1
1	1	0

```
int x = 5 ;  
int y = 3 ;  
int z = 5 ^ 3; // 6 olur
```

```
5 -> 0 1 0 1  
3 -> 0 0 1 1  
-----  
      0 1 1 0 -> 6
```

7- PROGRAMLAMAYA GİRİŞ

7.11- ~ (NOT) bitisel operatörü doğruluk tablosu

Değil (~) Doğruluk tablosu

~		
	~ 1	0
	~ 0	1

```
int x = 5 ;  
int y = ~ x ;
```

```
5   -> 0 1 0 1  
~ 5 -> 1 0 1 0 -> 10
```

7- PROGRAMLAMAYA GİRİŞ

7.11- << Sola shift (kaydırma) operatörleri

```
int x = 3 ;  
int y = x << 2 ; // x , iki kez sola kaydır
```

3 -> 0 0 1 1

0 1 1 0 -> değeri 6 , birinci sola kaydırma tüm bitler bir sola kayar

1 1 0 0 -> değeri 12 , ikinci sola kaydırma

Not : Her sola kaydırmada sayı 2 ile **ÇARPILIR**. Örn. 3 değeri bir kez sola kaydırılırsa 6 eder. İki kez sola kaydırılırsa 12 eder.

7- PROGRAMLAMAYA GİRİŞ

7.11- >> Sağa shift (kaydırma) operatörleri

```
int x = 8 ;  
int y = x >> 2 ; // x'i 2 kez sağa kaydır
```

8 -> 1 0 0 0

0 1 0 0 -> değeri 4 , birinci sağa kaydırma. Tüm bitler bir sağa kayar
0 0 1 0 -> değeri 2 , ikinci sağa kaydırma

Not : Her sağa kaydırmada sayı 2 ile **BÖLÜNÜR**. Örn. 8 değeri bir kez sağa kaydırılırsa 4 eder. İki kez sağa kaydırılırsa 2 eder.

7- PROGRAMLAMAYA GİRİŞ

7.11- Açıklama (Remark) Operatörleri (`//` , `/* ...*/`);

`//` : sonundaki satır derleyici tarafından komut olarak dikkate alınmaz

`/* ... */` : Arasında kalan komut satırları, derleyici tarafından komut olarak dikkate alınmaz

```
int x= 5 ; // x değişkenine 5 aktarılır
y +=1 ; // y++; ++y; y = y+1;
y <=<=1 ; // y' nin bitleri bir sola kaydırılır
```

```
int z = ~3;
/*
int x = 5 ;
int y = 5 + 8 ;
string s = "Hayat, biz gelecek için planlar yaparken başımızdan geçenlerdir. ";
*/
```

7- PROGRAMLAMAYA GİRİŞ

Örnek: Çıkarma işlemi kullanmadan x sayısından y sayısını çıkartınız?

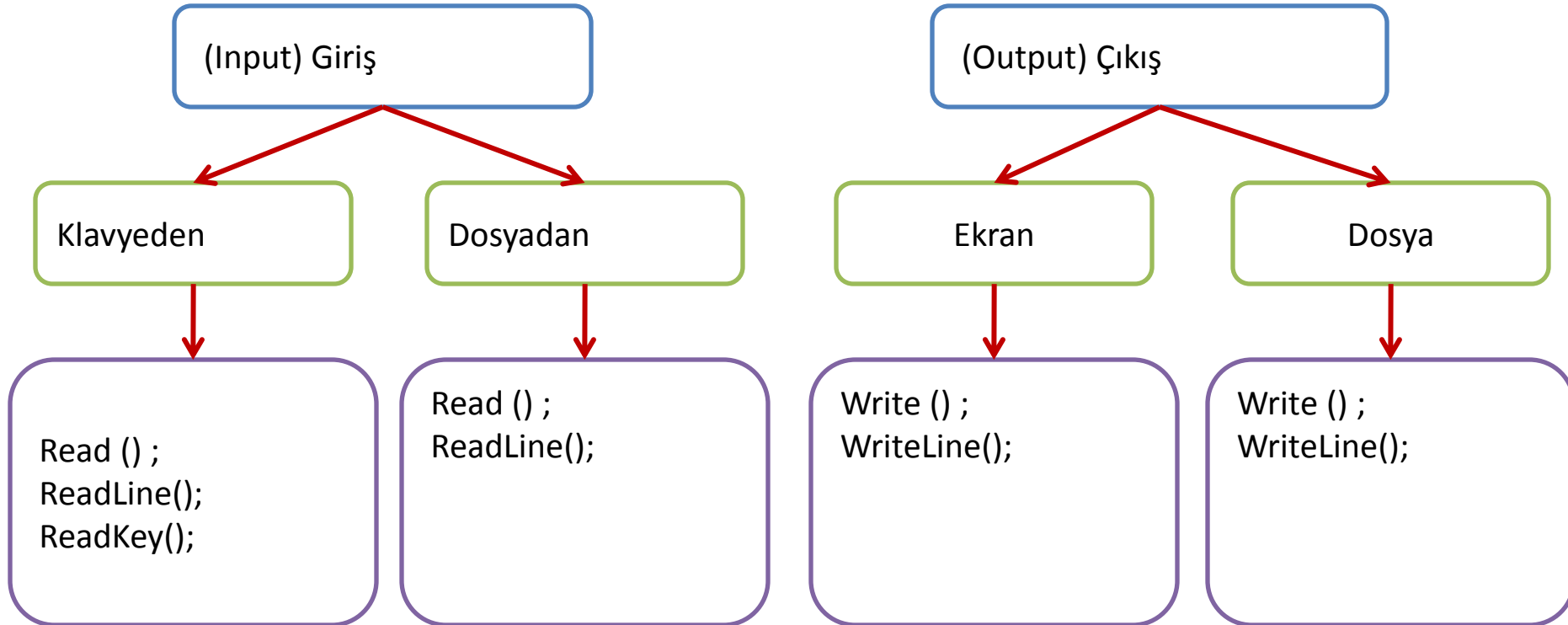
```
namespace ornek
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 5;
            int y = 3;
            y = ~y; // y' nin bitleri ters çevir (comploment)
            y += 1; // y 'ye 1 eklenir
            int z = x + y;
            Console.Write( z ); // 2 yazar
        }
    }
}
```

3-> 0 1 1
1 0 0 bitleri ter çevir
1 1 ekle
+ -----
1 0 1 (-) işaretli 3 sayısı

5-> 1 0 1
1 0 1
+ -----
(0 1 0)₂ = (2)₁₀

7- PROGRAMLAMAYA GİRİŞ

7.12- C# .NET I/O (Giriş / Çıkış Komutları- fonksiyonları) :



7- PROGRAMLAMAYA GİRİŞ

Program I/O (Giriş / Çıkış Komutları- fonksiyonları) :

```
int x = Console.Read( ) ; // basılan tuşun ASCII kodu okur  
int y = Console.Read( ) ; // bilgi girişi beklenir girip enter'e basınca y' ye ise Enter kodu(13) aktarılır
```

```
string adi = Console.ReadLine ( ) ;
```

```
Console.Write ("computer") ;
```

```
Console.Write ( 5 + 8 - 4/2-9) ;
```

```
Console.Write ( " Toplam = " + 5 + 8 ) ; // ekrana 58 yazar
```

7- PROGRAMLAMAYA GİRİŞ

7.13. Formatlı Yazdırma :

```
Console.Write( " Sonuç : {0} dir. " , 5 );
```

```
Console.Write( " {0} + {1} = {2} " , 5 , 8 , 13 );
```

```
string ad= Console.ReadLine(); // klavyeden Ali değeri girip Enter tuşuna basınız  
string soy= Console.ReadLine(); // klavyeden Can değeri girip Enter tuşuna basınız
```

```
int v= int.Parse (Console.ReadLine()); // klavyeden 70 değeri girip Enter tuşuna basınız
```

```
Console.Write( " Adı: {0}\n Soyadı: {1}\n Vize= {2} " , ad, soy, v ); // \n satır başı yapar
```

Ekran çıktısı

Adı : Ali
Soyadı : Can
Vize : 70

7- PROGRAMLAMAYA GİRİŞ

7.13. Formatlı Yazdırma :

```
Console.Write( " maaşı: {0,6} Yaşı:{1,4}" , maas, yas );
```

```
Maaşı :__1560 Yaşı : __ 28
```

```
Console.Write( " maaşı: {0,-6} Yaşı:{1,-4}" , maas, yas );
```

```
Maaşı :1560 __ Yaşı : 28__
```

```
Console.Write ( (21567.679).ToString("###,###.##") ); // 21.567,68
```

7- PROGRAMLAMAYA GİRİŞ

7.13. Formatlı Yazdırma :

```
static void Main(string[] args)
{
    int i = 654321;
    Console.WriteLine("{0:C}", i ); // Currency 654.321,00 YTL
    Console.WriteLine("{0:D}", i ); // Decimal 654321
    Console.WriteLine("{0:E}", i ); // Exponential 6.543210E+005
    Console.WriteLine("{0:N}", i ); // Numeric 654,321.00
    Console.WriteLine("{0:X}", i ); // hexadecimal 9FBF1
    Console.WriteLine("{0:x}", i ); // hexadecimal 9fbf1
}
```

Araştırma: Diğer format işlemlerini araştırınız (örn. tarih formatı gibi)

7- PROGRAMLAMAYA GİRİŞ

7.14. (\) White Space karakterleri- Kurtarma karakteri :

Dilde özel anlamı olan karakterleri özel anlamdan çıkartır.

Özel bir anlamı olmayan karakterlere de özel anlam kazandırır

r -> r karakteri **\r** -> Enter

**** -> kurtarma karakteri **** -> \ karakteri

KARAKTER	ANLAMI
\n	New line (Satır başı)
\r	Carridge Return (enter)
\t	Horizantal (yatay) tab
\v	Vertical(dikey) tab
\f	Form feed (yazıcıdan bir sayfa)
\a	Alert (buzzer dan beep sesi)
\"	"
\'	'
\\	\

7- PROGRAMLAMAYA GİRİŞ

WhiteSpace karakterleri- Kurtarma karakteri (\ , @) kullanımı :

```
string yol = "C:\windows\resim\newresim"; // Hatalıdır
```

```
string yol = "C:\\windows\\resim\\newresim";
```

C:\windows\resim\newresim

```
string yol = @ "C:\windows\resim\newresim";
```

C:\windows\resim\newresim

7- PROGRAMLAMAYA GİRİŞ

White Space karakterleri(Kurtarma karakteri) (\ , @) kullanımı :

```
string s = "MAKU \nMYO\nBILGISAYAR";
```

```
MAKU
MYO
BILGISAYAR
```

```
string k = @ "windows\nsystem\nwin32";
```

k' nin arındırılmış(gerçek) değeri nasıldır ?

```
windows\nsystem\nwin32
```



SORULAR

1-Aşağıdaki değişken tanımlamalarından hangisi geçerli bir tanımlamadır ?

A) ortalama not b) case c) 2vize d) @facebook e)_w

2- $x = x + 1;$ dengi olan ifadesine aşağıdakilerden hangisi olabilir?

I-) $x++;$ II-) $++x;$ III-) $x+=1;$

A) I-II b) I c) III d) I-II-III e) I-IV

3- `int x=4, y=3; int z=x & y;` ise z nin değeri ne olur?

a) 0 b) 1 c) 2 d) 3 e) 5

4- Aşağıdakilerden değişkenlerin hangisi macar yöntemine göre gösterilmiştir.?

a) vize b) Vize_int c) vize_1 d) vint e) int Vize

5- `string s=@ "MYO\nBIL";` komutuna göre s değişkenin değeri ekrana yazdırılırsa nasıl bir çıktı veririr?

a) MYO\nBIL b) MYO c) @MYO d) @MYO\nBIL e) @MYO
BIL @BIL BIL

SORULAR

6- `int x = (5 >=k) ? 8 : 9 ;` komutuna göre `k` 'nın 5 değeri için `x` 'in değeri nasıl olur ?

A) `x` b) `k` c) 8 d) 9 e) 5

7- `Console.Write (Convert.ToString(15,16)) ;` komutundan sonra ekranda ne yazar ?

A) f b) 15 c) 16 d) "15,16" e) Hiçbiri

8- C# . Net için aşağıdakilerden hangisi doğrudur ?

I- Oluşturulan programın çalışması için bilgisayarda framework kurulu olmalı

II- Nesne Tabanlı bir dildir

III- Web uygulamaları geliştirmeye müsait bir dildir

a) I b) I – II c) II- III d) I-II-III e) I-III

9- `string s = "\\\\";` komutuna göre `s` değişkenin değeri ekrana yazdırılırsa nasıl bir çıktı veririr ?

a) \ b) \\ c) \\\ d) "\\\" e) Hiçbir şey yazmaz

10- `string s = "Derinliği " + 15 + 22 + " m.";` komutuna göre `s` değişkenin değeri ekrana yazdırılırsa nasıl bir çıktı veririr ?

a) Derinliği 1522 m. b) Derinliği 37. c) Hata veririr d) Hiçbir şey yazmaz e) null yazar

Yazılım Geliştirme Modelleri

Yazılım Geliştirme Modelleri

Daha emniyetli yazılımların daha kısa sürede, daha az bütçeyle ve daha az hatayla geliştirilmesi için sürekli yeni teknolojiler ve modeller bulunmaya çalışılmaktadır. Bunlardan bazıları;

- I) Gelişigüzel Geliştirme
- II) Şelale (Waterfall) Modeli
- III) V- Modeli
- IV) Barok Modeli
- V) Helezonik (Spiral) Model
- VI) Arttırımsal (Incremental) Geliştirme Modeli
- VII) Döngüsel Model
- VIII) Çevik Yazılım Geliştirme Metodları

Yazılım Geliştirme Modelleri

I) Gelişigüzel Geliştirme

Gelişigüzel geliştirmede belirlenmiş bir model ya da yöntem bulunmaz.

Genellikle kişiye bağlı yazılım geliştirme şeklinde yapılır ve bu yüzden yazılımın izlenebilirliği, bakım yapılabilirliği oldukça zordur.

1960'lı yıllarda uygulanan bu yöntem, genellikle basit programlama içeren ve çoğunlukla tek bir kişinin üretim yaptığı yöntemidir.

II) Şelale (Waterfall) Modeli

Şelale modeli yakın zamanlara kadar en popüler yazılım geliştirme modeli olarak görülmüştür.

Geleneksel yazılım geliştirme modeli olarak da bilinir.

Şelale modelinde yazılım, aşamalar en az birer kez tekrarlanarak geliştirilir.

Çok iyi tanımlanmış ve üretimi az zaman gerektiren projeler için uygun bir model olmakla birlikte günümüzde kullanımı gittikçe azalmaktadır.

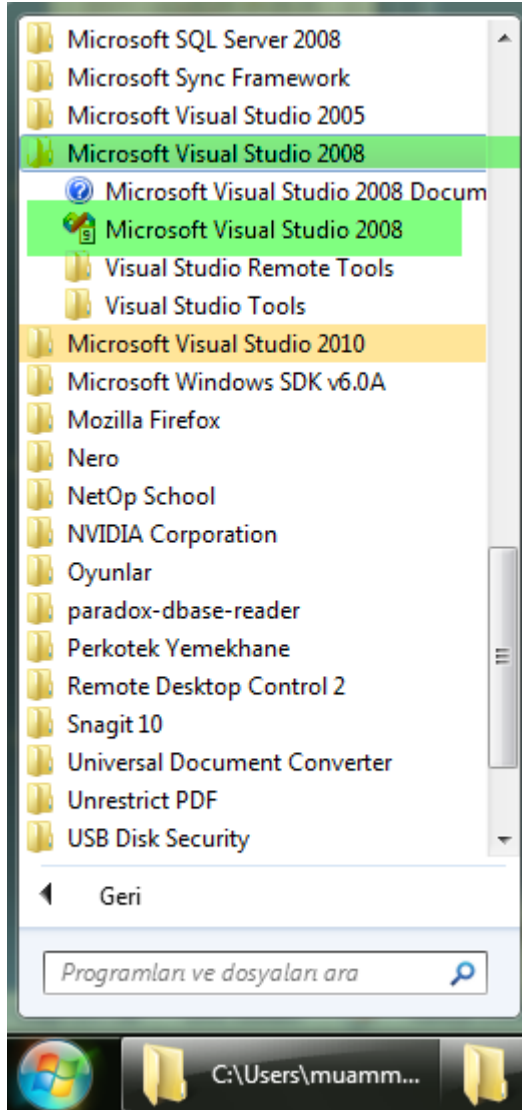
III) V -Modeli

Bu model popüler geliştirme modellerindendir. Sistemin iki bölümden oluşur ;tasarım ve kabul.

Kabul bölümünde isteklere göre tasarım aşamasının uygun adımına gidilerek programda düzeltmeler yapılabilir.

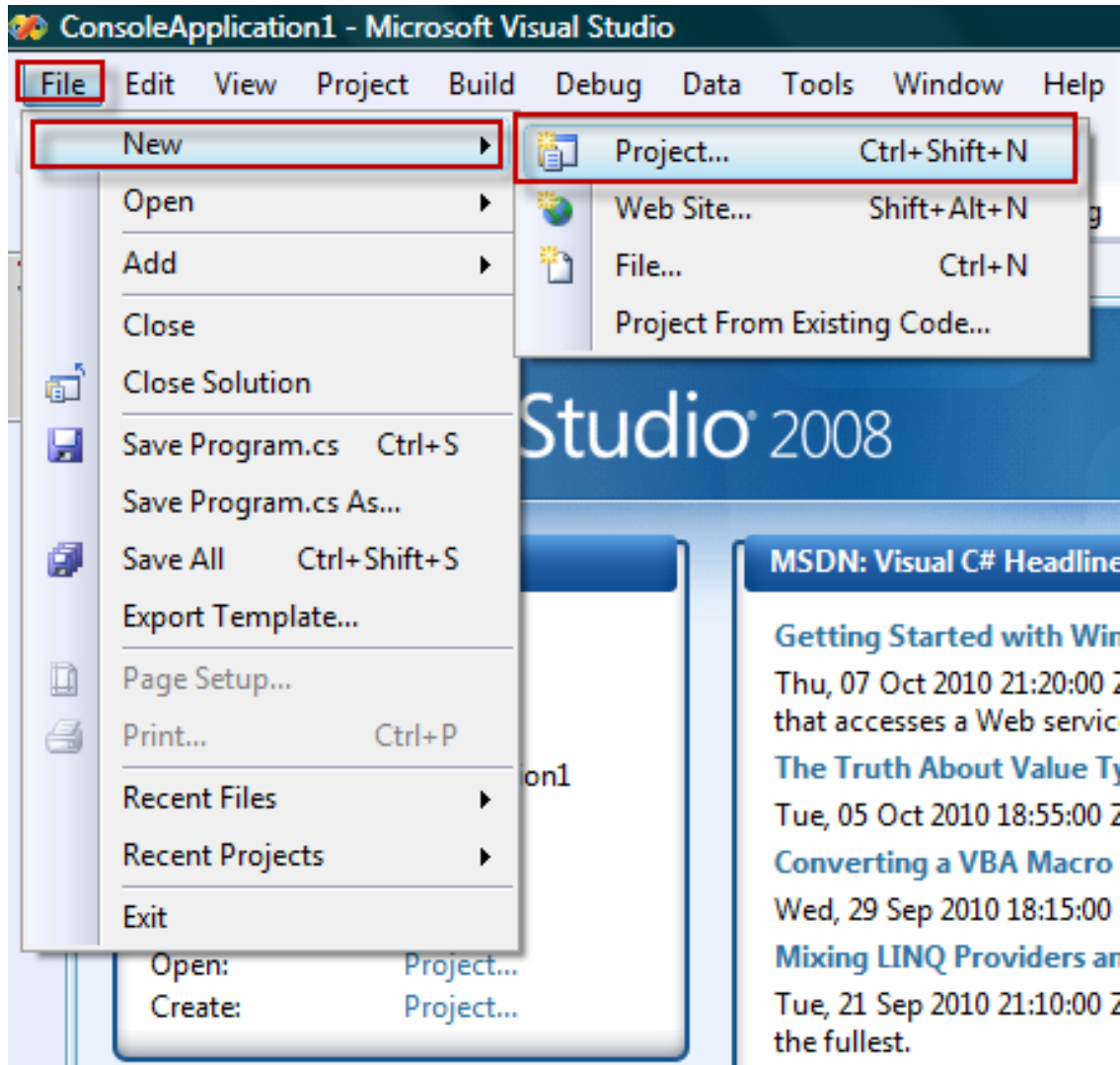
7- PROGRAMLAMAYA GİRİŞ

7.15- Visual Studio Editörün tanıtımı



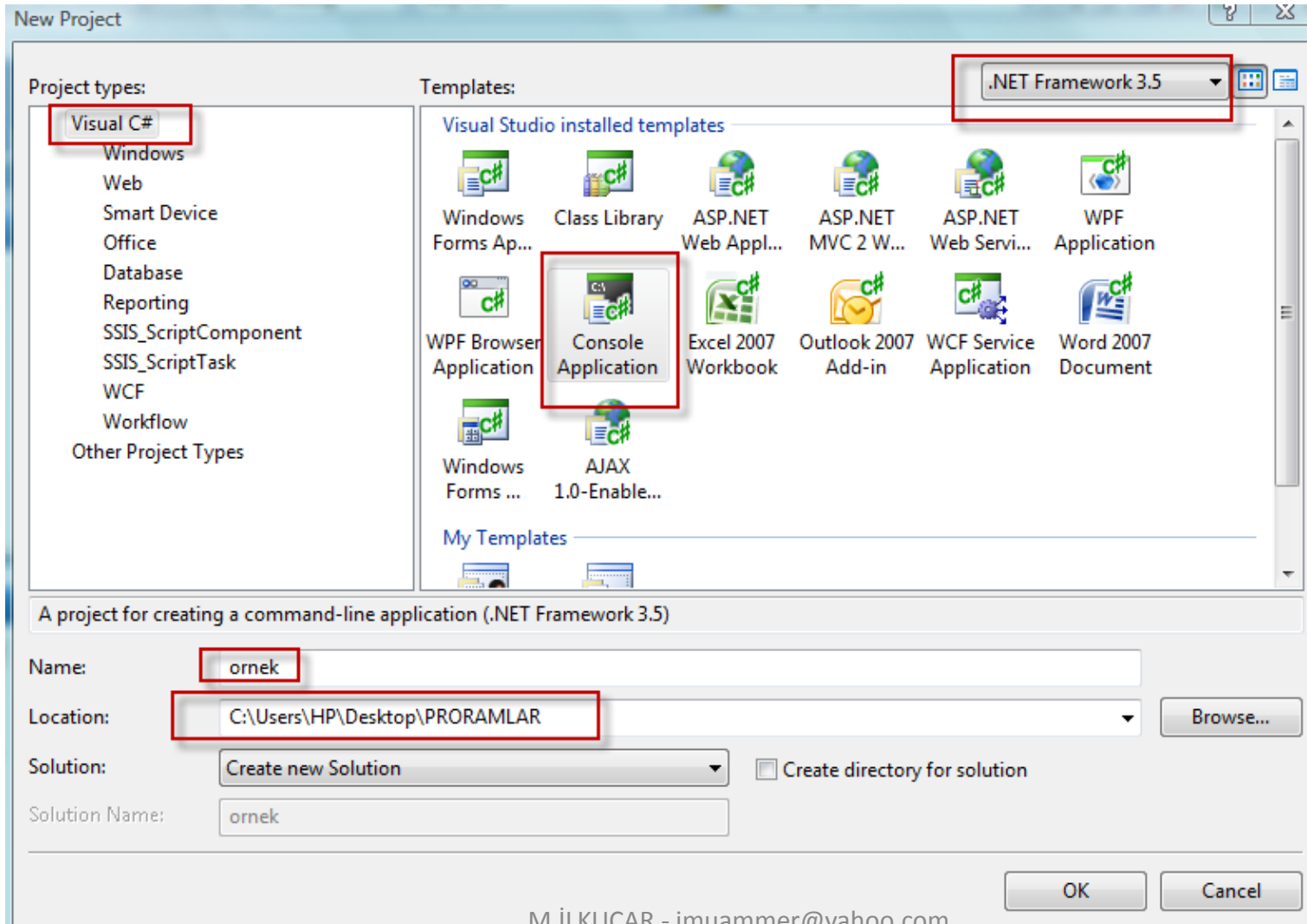
7- PROGRAMLAMAYA GİRİŞ

7.15- Visual Studio Editörün tanıtımı



7- PROGRAMLAMAYA GİRİŞ

Project types bölümünden dil seçilir(C#), *Templates* Bölümünden Proje türü seçilir (Console Application), *Name* bölümünden projenin adı verilir (ornek) ve *Location* bölümünden projenin kaydedileceği yer seçilir (c:\Users\HP\Desktop\PROGRAMLAR) ve OK tuşuna basılınca sonraki salyattaki gibi ekran gelir.

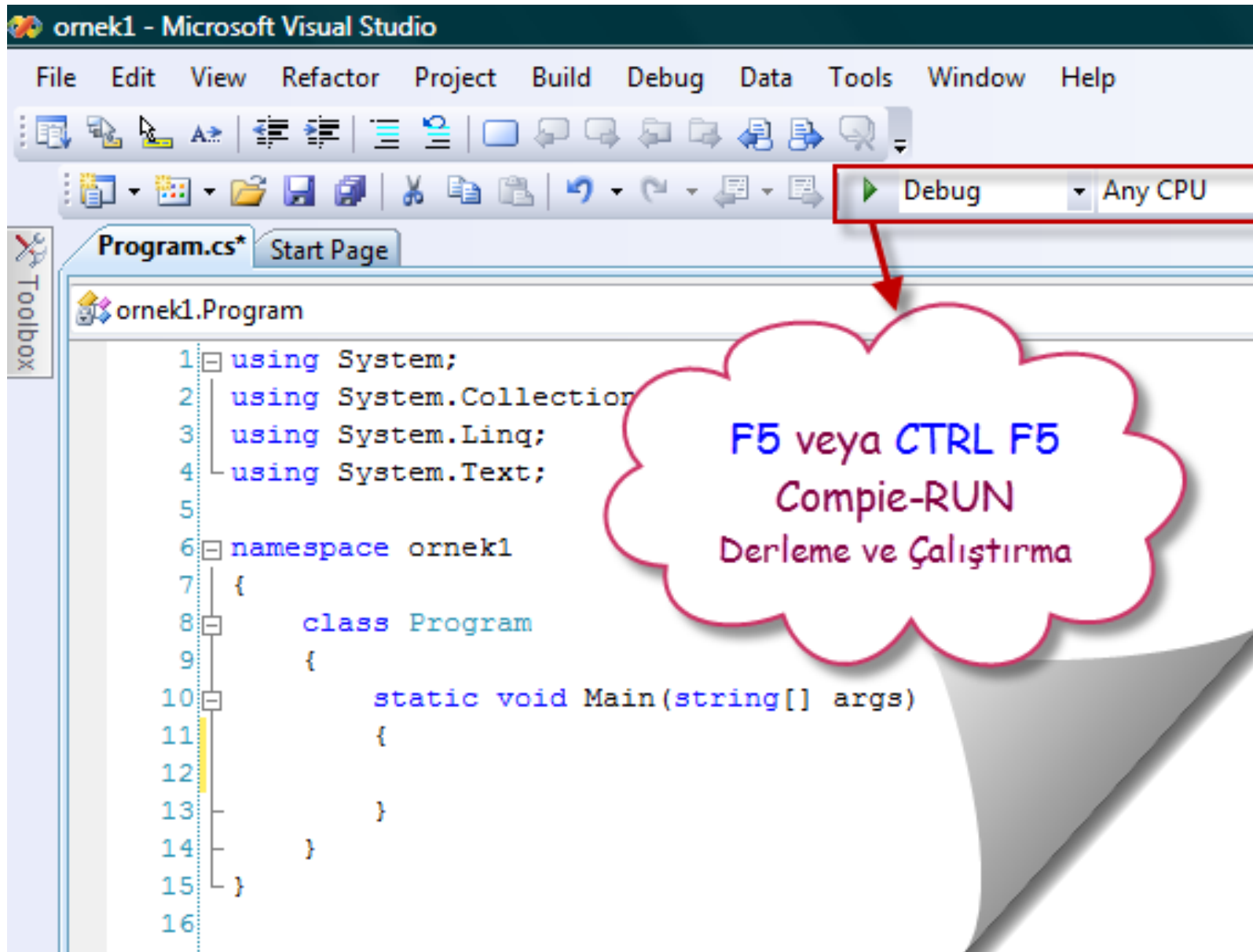


7- PROGRAMLAMAYA GİRİŞ

Program kodlarımızı temel fonksiyon olan *Main()* { ... } bloğu içine yazacağız. Ve F5 veya Ctrl+F5 ile Compile ve RUN işlemi yapıp programımızı çalıştıracağız. Yazım hatası varsa program çalışmayıp hata mesajı verir. Hata mesajı üzerine çift tıklayarak hatanın olduğu satıra gidilebilir.

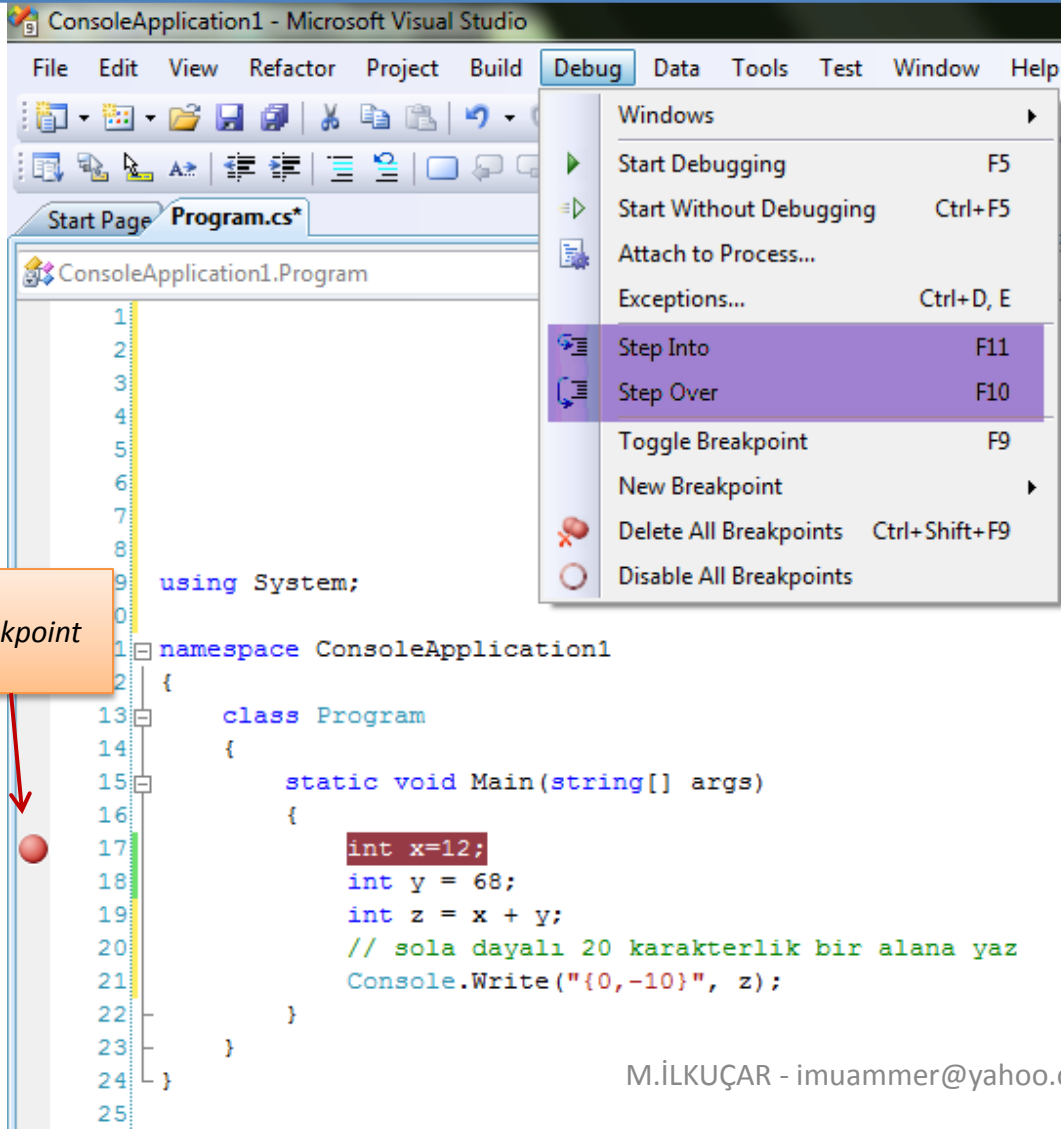
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace ornek1
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             // Kodlar bu alana yazılacak
13         }
14     }
15 }
16
```


7- PROGRAMLAMAYA GİRİŞ



7- PROGRAMLAMAYA GİRİŞ

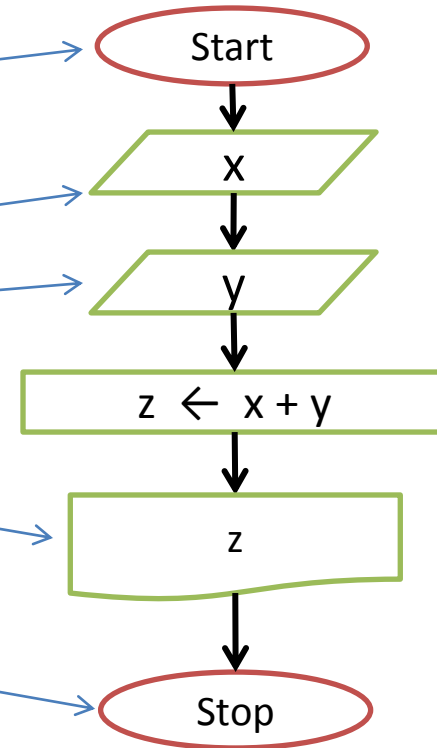
Program kodlarımız *DEBUG* (adım adım çalıştırarak) program adımlarını izleyebilir. Böylece hatalarımızı daha kolay düzeltebiliriz. Bunun için Şekilde görüldüğü gibi bir *Breakpoint* noktası koyarsak F5 ile çalıştırdığımız zaman bu noktaya kadar çalışır ve bu noktadan sonra F10 veya F11 ile adım adım çalıştırabiliriz



C# .net Programlama dilinin genel yapısı

```
using System;           // komutların bulunduğu sınıf

namespace ORNEK1       // projenin adı
{
    class Program
    {
        void Main ( )
        {
            int x = Console.Read() ;
            int y = Console.Read() ;
            int z = x + y;
            Console.Write (z);
        }
    }
}
```



7- PROGRAMLAMAYA GİRİŞ

```
using System;  
  
namespace ORNEK1  
{  
    class Program  
    {  
        void Main ( )  
        {  
            int x = Console.Read();  
            int y = Console.Read();  
            int z = x + y;  
            Console.Write (z);  
        }  
    }  
}
```

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.Write("Birinci Sayıyı giriniz :");
10             int x = Convert.ToInt32( Console.ReadLine() );
11
12             Console.Write("İkinci Sayıyı giriniz :");
13             int y = int.Parse( Console.ReadLine() );
14
15             int z = x + y;
16             Console.Write(z);
17             Console.ReadKey();
18         }
19     }
20 }
21
```

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.Write("Birinci Sayıyı giriniz :");
10             int x = Convert.ToInt32( Console.ReadLine() );
11
12             Console.Write("İkinci Sayıyı giriniz :");
13             int y = int.Parse( Console.ReadLine() );
14
15             int z = x + y;
16             Console.Write(" Toplam :" + z );
17             Console.ReadKey();
18         }
19     }
20 }
21
```

C# .net örnek

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.Write("Birinci Sayıyı giriniz :");
10             int x = Convert.ToInt32( Console.ReadLine() );
11
12             Console.Write("İkinci Sayıyı giriniz :");
13             int y = int.Parse( Console.ReadLine() );
14
15             int z = x + y;
16             Console.Write(" {0} + {1} = {2}" , x, y, z );
17             Console.ReadKey();
18         }
19     }
20 }
21
```

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int x = 0;
10             ETIKET1:
11             Console.Write(" {0} ", x);
12             ++x;
13             if (x < 10) goto ETIKET1;
14             Console.ReadKey();
15         }
16     }
17 }
18
```

0 1 2 3 4 5 6 7 8 9

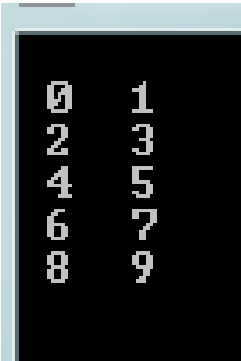

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int x = 0;
10             ETIKET1:
11             if(x%2 == 0) Console.Write(" {0} ", x);
12             ++x;
13             if (x < 10) goto ETIKET1;
14             Console.ReadKey();
15         }
16     }
17 }
18
```

0 2 4 6 8

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int x = 0;
10             ETIKET1:
11             if(x%2 != 0) Console.Write(" {0} ", x);
12             ++x;
13             if (x < 10) goto ETIKET1;
14             Console.ReadKey();
15         }
16     }
17 }
18
```

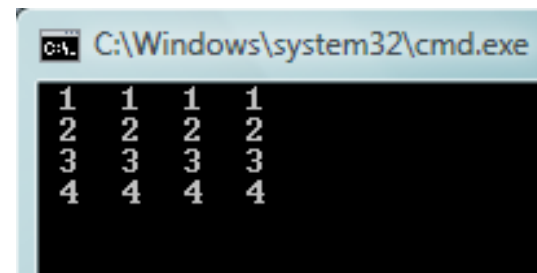
1 3 5 7 9

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int x = 0;
10             ETIKET1:
11             if (x % 2 == 0) Console.Write("\n");
12             Console.Write(" {0} ", x);
13             ++x;
14             if (x < 10) goto ETIKET1;
15             Console.ReadKey();
16         }
17     }
18 }
19
```



```
0 1
2 3
4 5
6 7
8 9
```

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int x = 1, k=5;
10             ETIKET_1:
11             int y = 1;
12             ETIKET_2:
13             Console.Write(" {0} ", x );
14             y++;
15             if (y < k) goto ETIKET_2;
16             Console.Write("\n");
17             ++x;
18             if (x < k) goto ETIKET_1;
19             Console.ReadKey();
20         }
21     }
22 }
23
```



```
C:\Windows\system32\cmd.exe
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

C# .net örnek

```
1  using System;
2
3  namespace ConsoleApplication2
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int x = 1, k=5;
10             ETIKET_1:
11             int y = 1;
12             ETIKET_2:
13             Console.Write(" {0} ", x );
14             y++;
15             if (y<=x) goto ETIKET_2;
16             Console.Write("\n");
17             ++x;
18             if (x < k) goto ETIKET_1;
19             Console.ReadKey();
20         }
21     }
22 }
23
```

C:\Windows\system32\cmd.exe

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

C:\ file:///C:/Users/t

1			
2	2		
3	3	3	
4	4	4	4