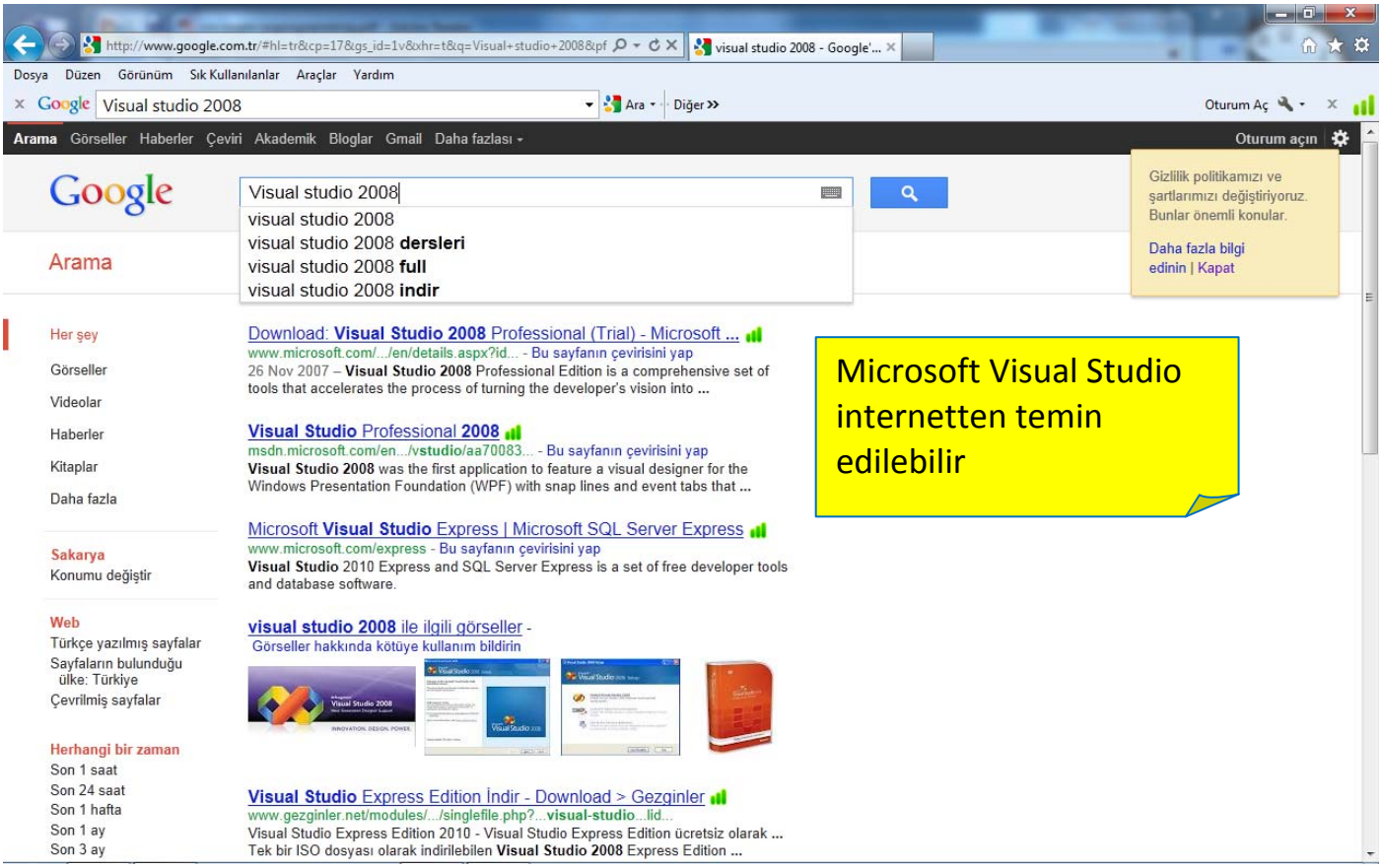
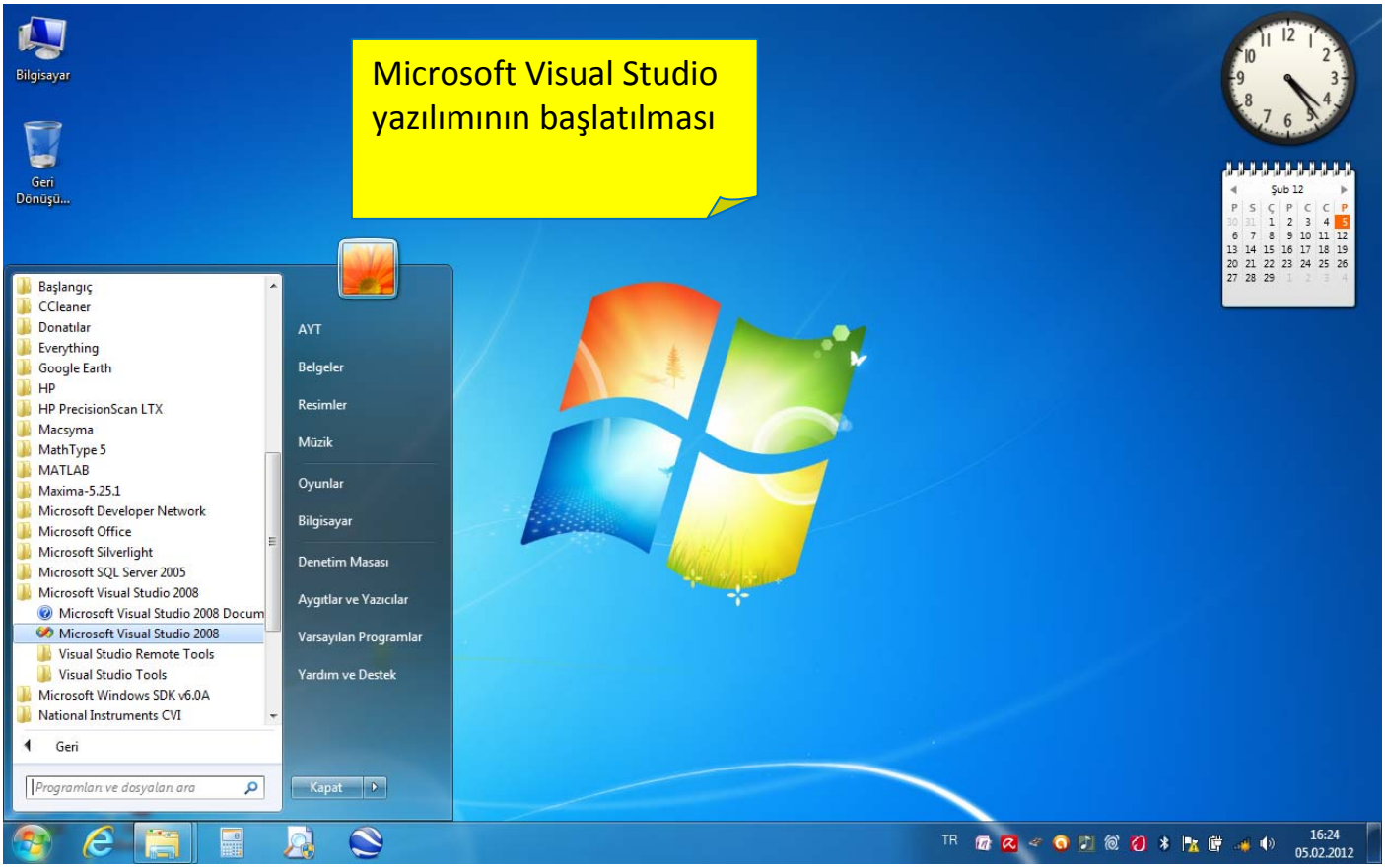
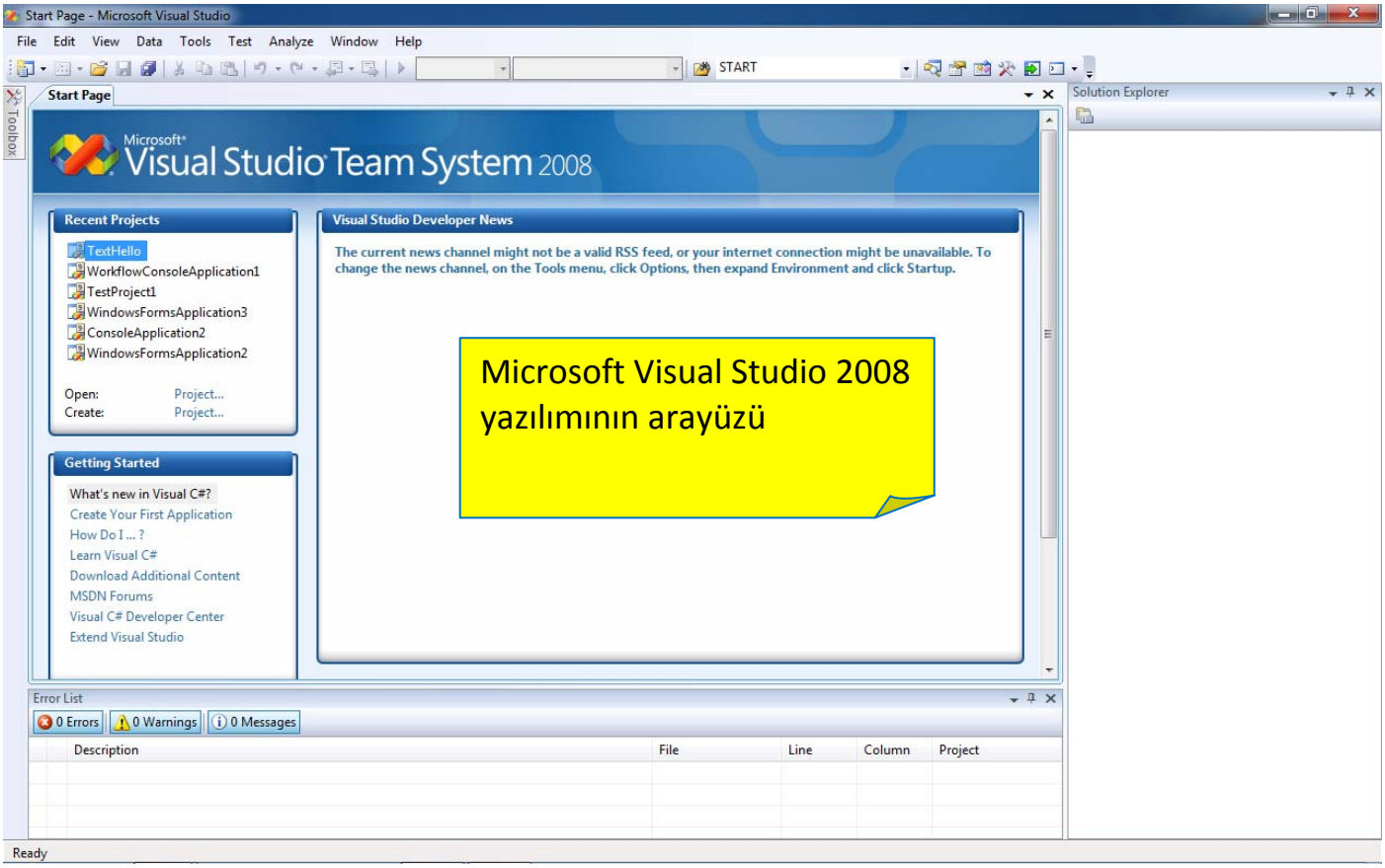
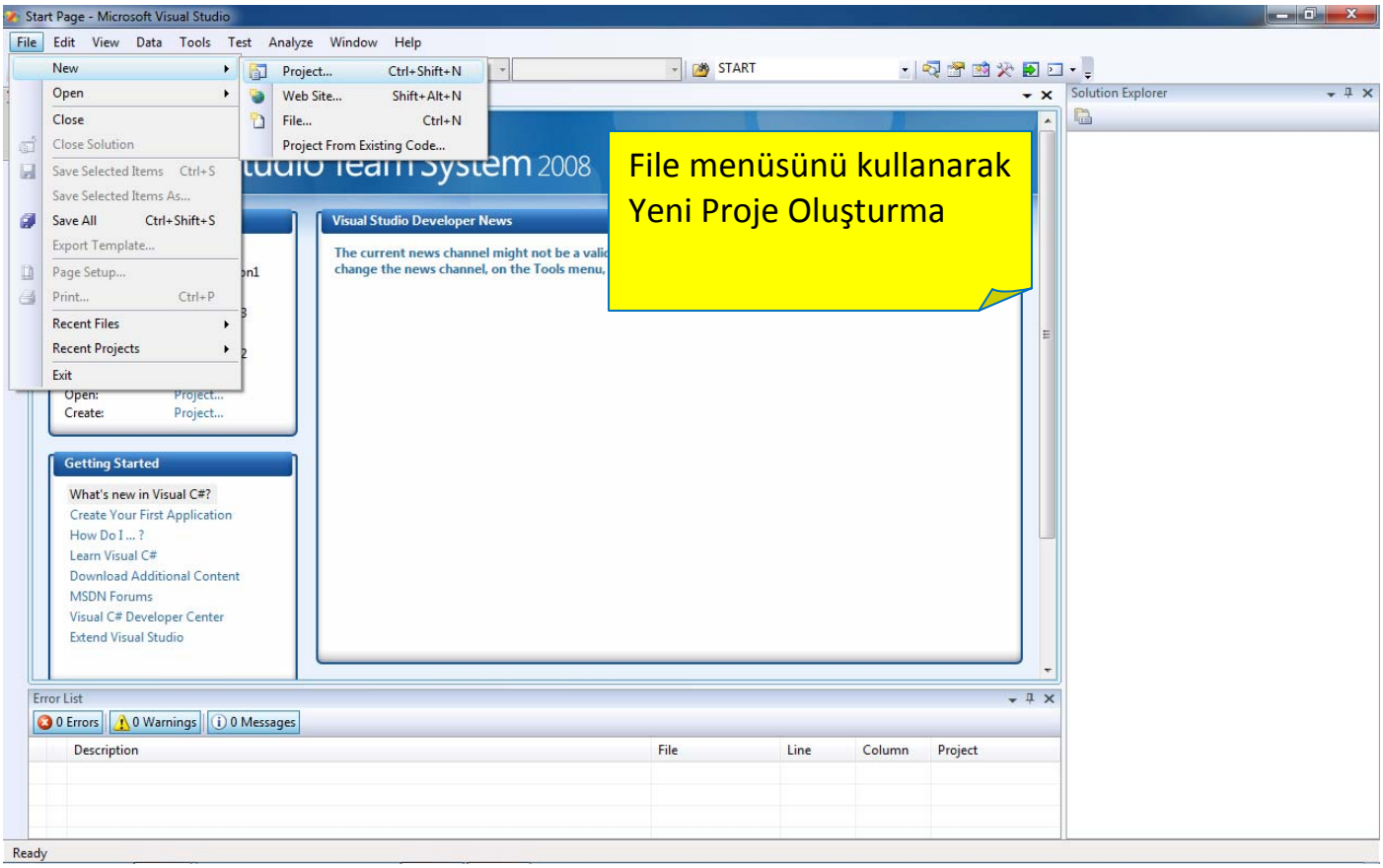
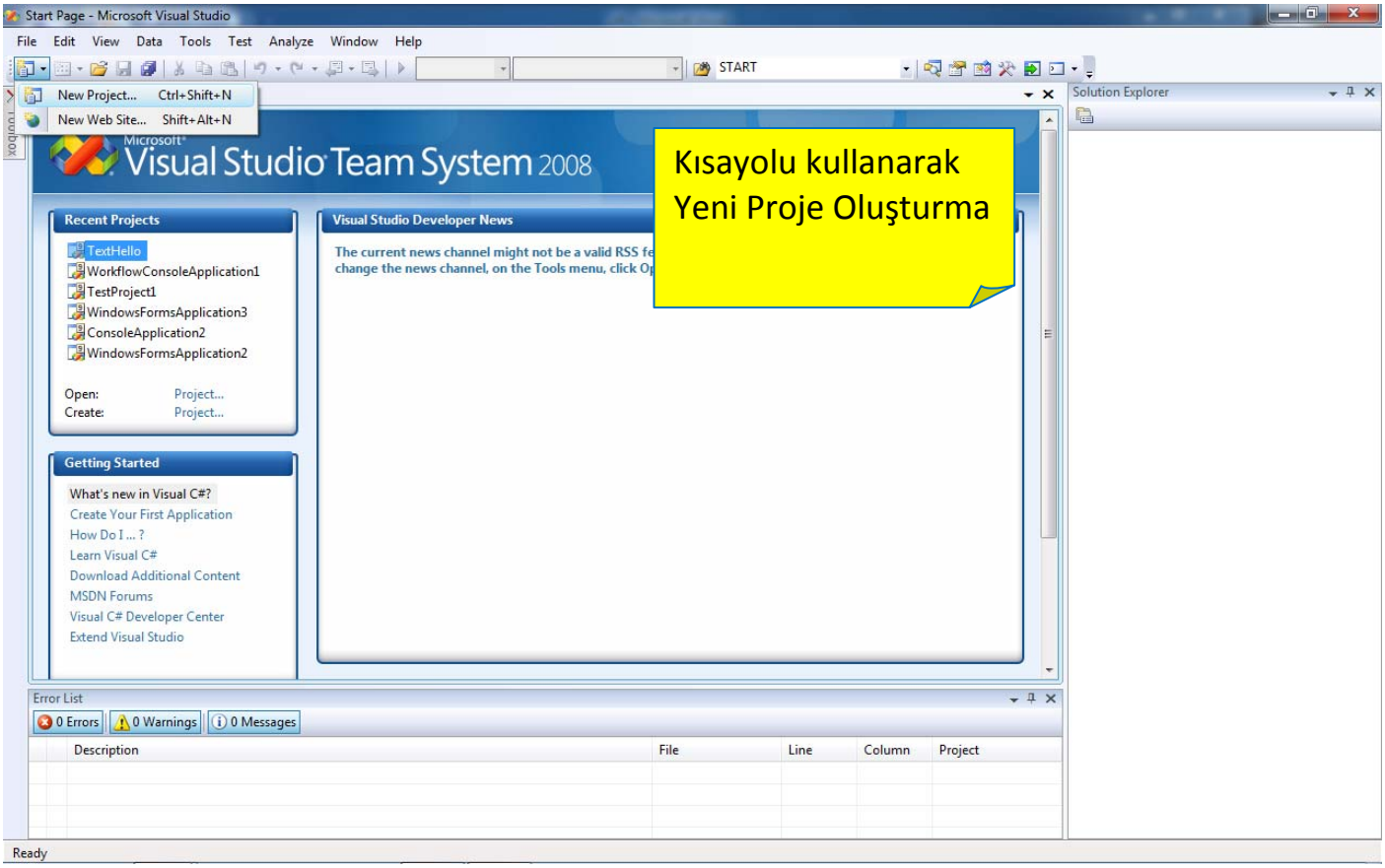
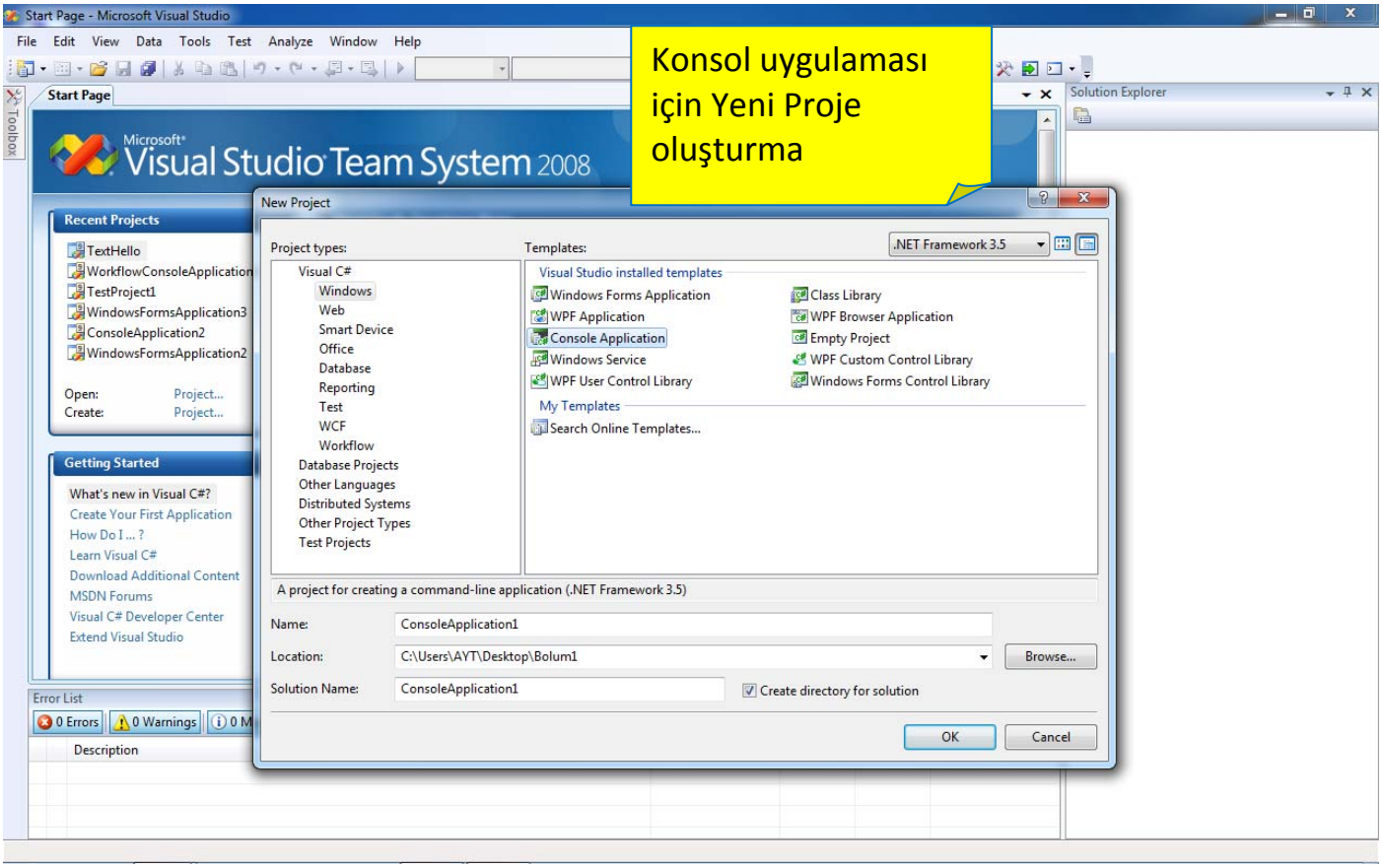

Visual studio 2008
visual studio 2008
visual studio 2008 dersleri
visual studio 2008 full
visual studio 2008 indir
Microsoft Visual Studio internetten temin edilebilir

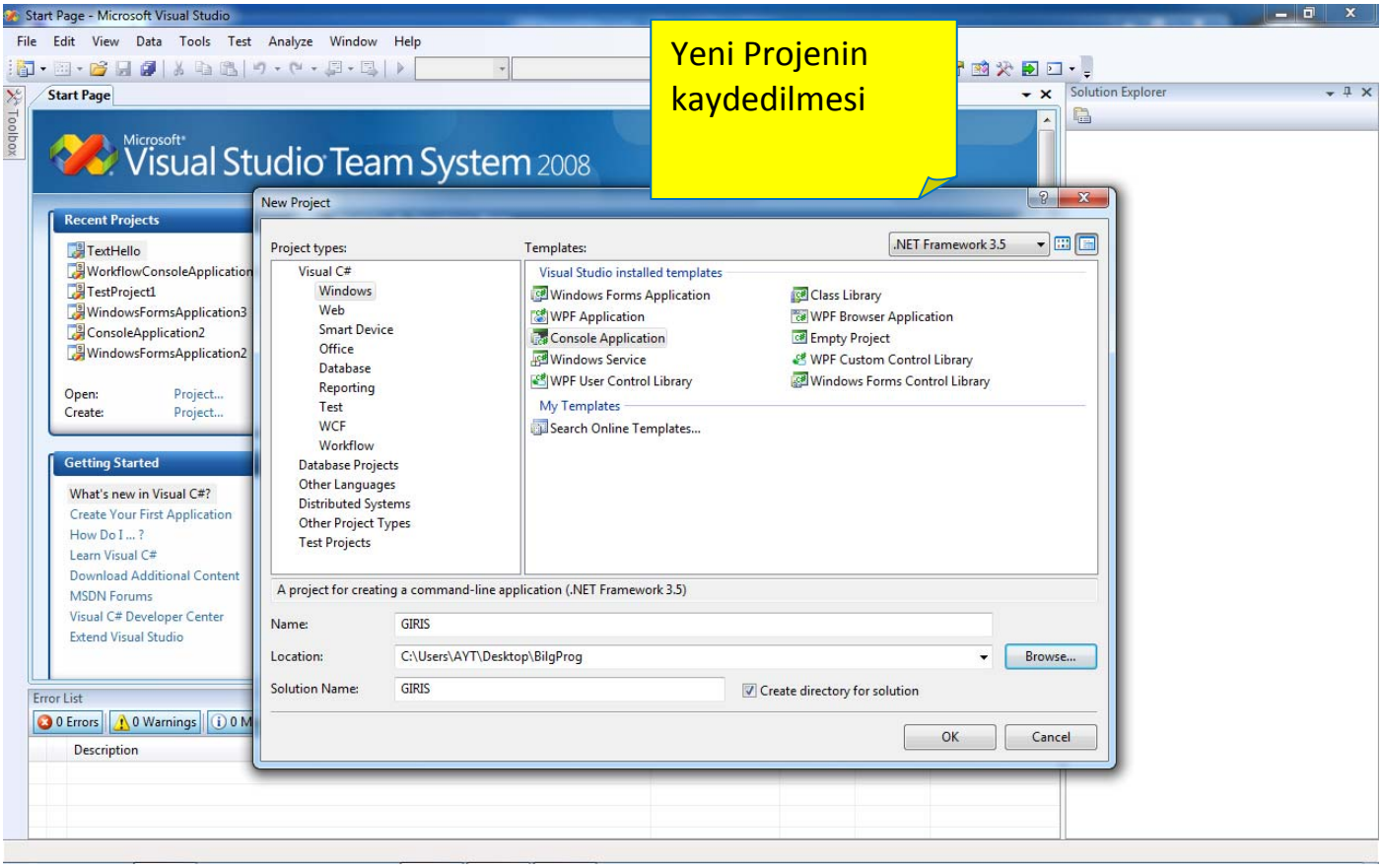

Microsoft Visual Studio yazılımının başlatılması
Bilgisayar
Geri Dönüşü...
Başlangıç
CCleaner
Donatılar
Everything
Google Earth
HP
HP PrecisionScan LTX
Macsyma
MathType 5
MATLAB
Maxima-5.25.1
Microsoft Developer Network
Microsoft Office
Microsoft Silverlight
Microsoft SQL Server 2005
Microsoft Visual Studio 2008
Microsoft Visual Studio 2008 Docum
Microsoft Visual Studio 2008
Visual Studio Remote Tools
Visual Studio Tools
Microsoft Windows SDK v6.0A
National Instruments CVI
Geri
Programları ve dosyaları ara
AYT
Belgeler
Resimler
Müzik
Oyunlar
Bilgisayar
Denetim Masası
Aygıtlar ve Yazıcılar
Varsayılan Programlar
Yardım ve Destek
Kapat
TR
16:24
05.02.2012

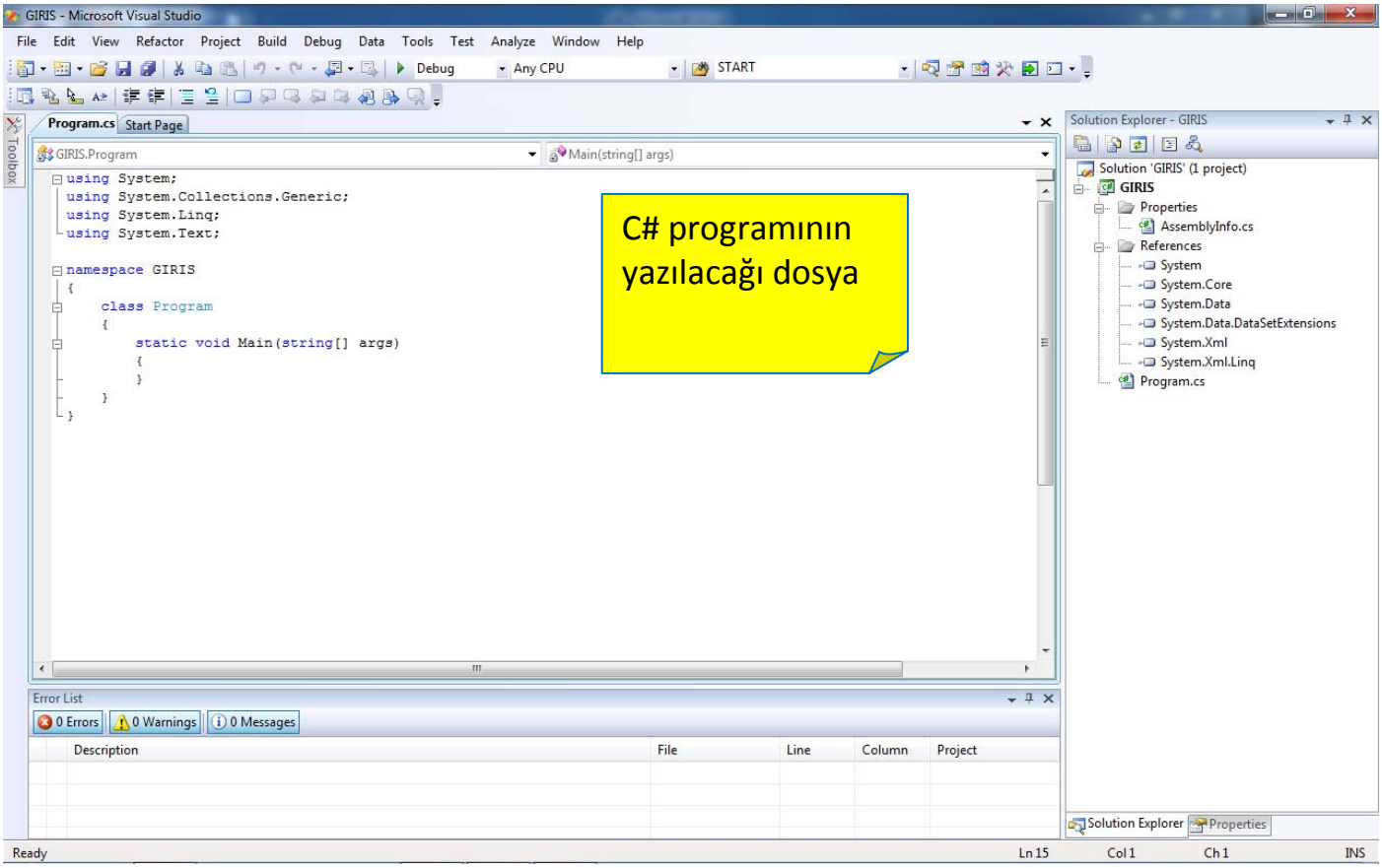

Start Page - Microsoft Visual Studio
File Edit View Data Tools Test Analyze Window Help
START
Start Page
Solution Explorer
Toolbox
Microsoft Visual Studio Team System 2008
Recent Projects
TextHello
WorkflowConsoleApplication1
TestProject1
WindowsFormsApplication3
ConsoleApplication2
WindowsFormsApplication2
Open: Project...
Create: Project...
Visual Studio Developer News
The current news channel might not be a valid RSS feed, or your internet connection might be unavailable. To change the news channel, on the Tools menu, click Options, then expand Environment and click Startup.
Getting Started
What's new in Visual C#?
Create Your First Application
How Do I ... ?
Learn Visual C#
Download Additional Content
MSDN Forums
Visual C# Developer Center
Extend Visual Studio
Microsoft Visual Studio 2008 yazılımının arayüzü
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Column
Project
Ready

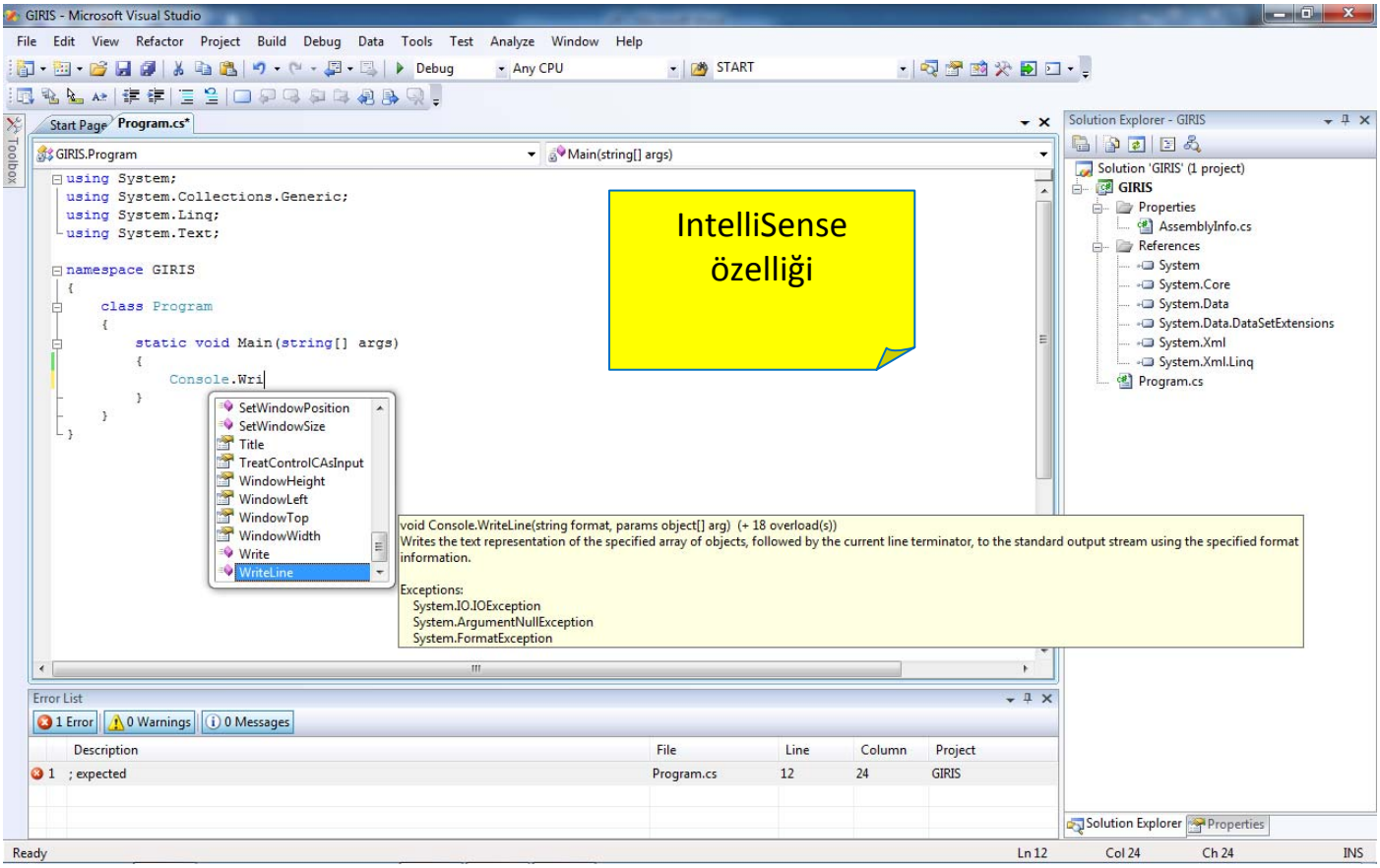

Start Page - Microsoft Visual Studio
File Edit View Data Tools Test Analyze Window Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects
Exit
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
File... Ctrl+N
Project From Existing Code...
START
Solution Explorer
Visual Studio Developer News
Getting Started
What's new in Visual C#?
Create Your First Application
How Do I ... ?
Learn Visual C#
Download Additional Content
MSDN Forums
Visual C# Developer Center
Extend Visual Studio
File menüsünü kullanarak Yeni Proje Oluşturma
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Column
Project
Ready

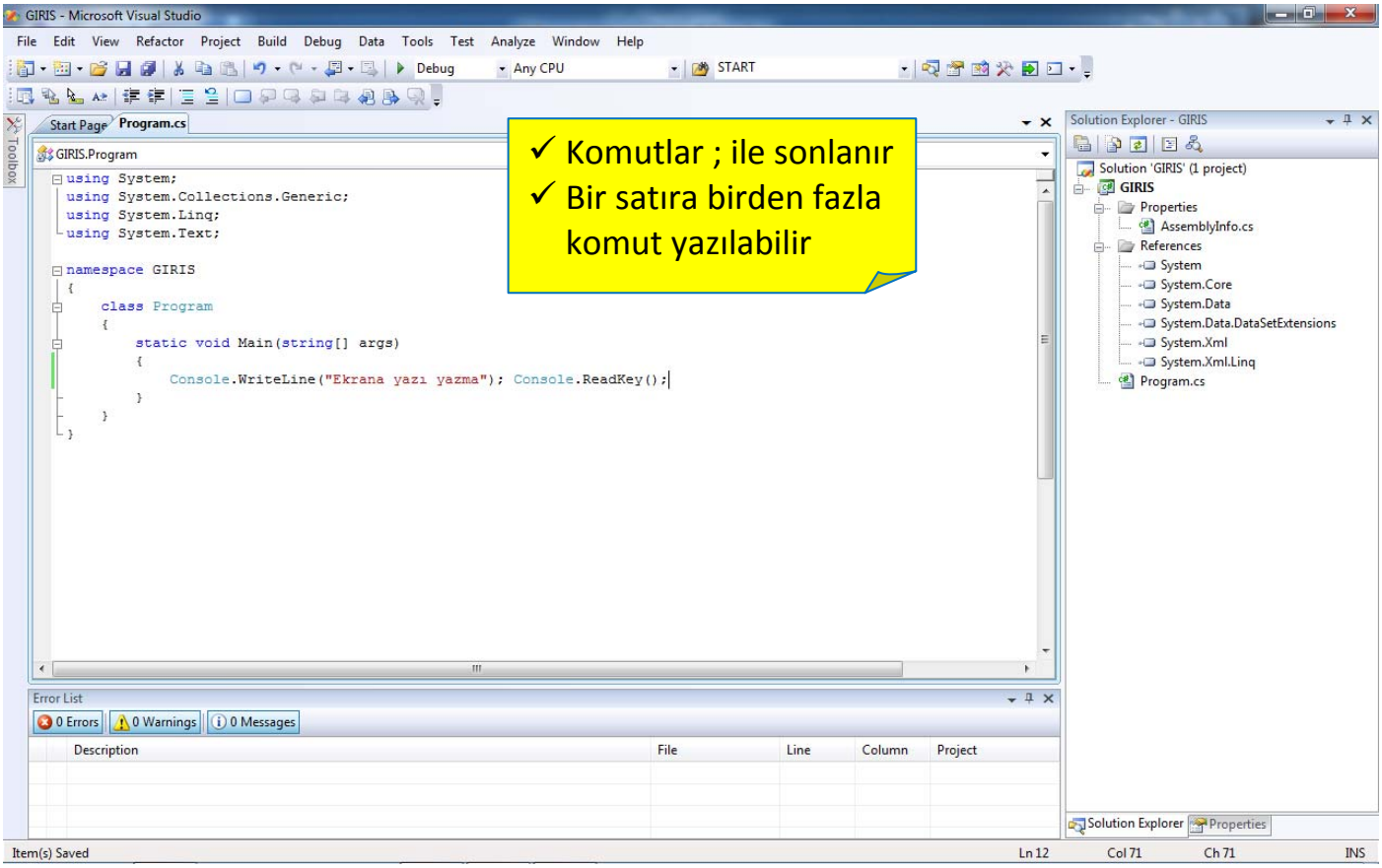

Start Page - Microsoft Visual Studio
File Edit View Data Tools Test Analyze Window Help
START
New Project... Ctrl+Shift+N
New Web Site... Shift+Alt+N
Visual Studio Team System 2008
Recent Projects
TextHello
WorkflowConsoleApplication1
TestProject1
WindowsFormsApplication3
ConsoleApplication2
WindowsFormsApplication2
Open: Project...
Create: Project...
Visual Studio Developer News
Getting Started
What's new in Visual C#?
Create Your First Application
How Do I ... ?
Learn Visual C#
Download Additional Content
MSDN Forums
Visual C# Developer Center
Extend Visual Studio
Solution Explorer
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Column
Project
Ready
Kısayolu kullanarak Yeni Proje Oluşturma

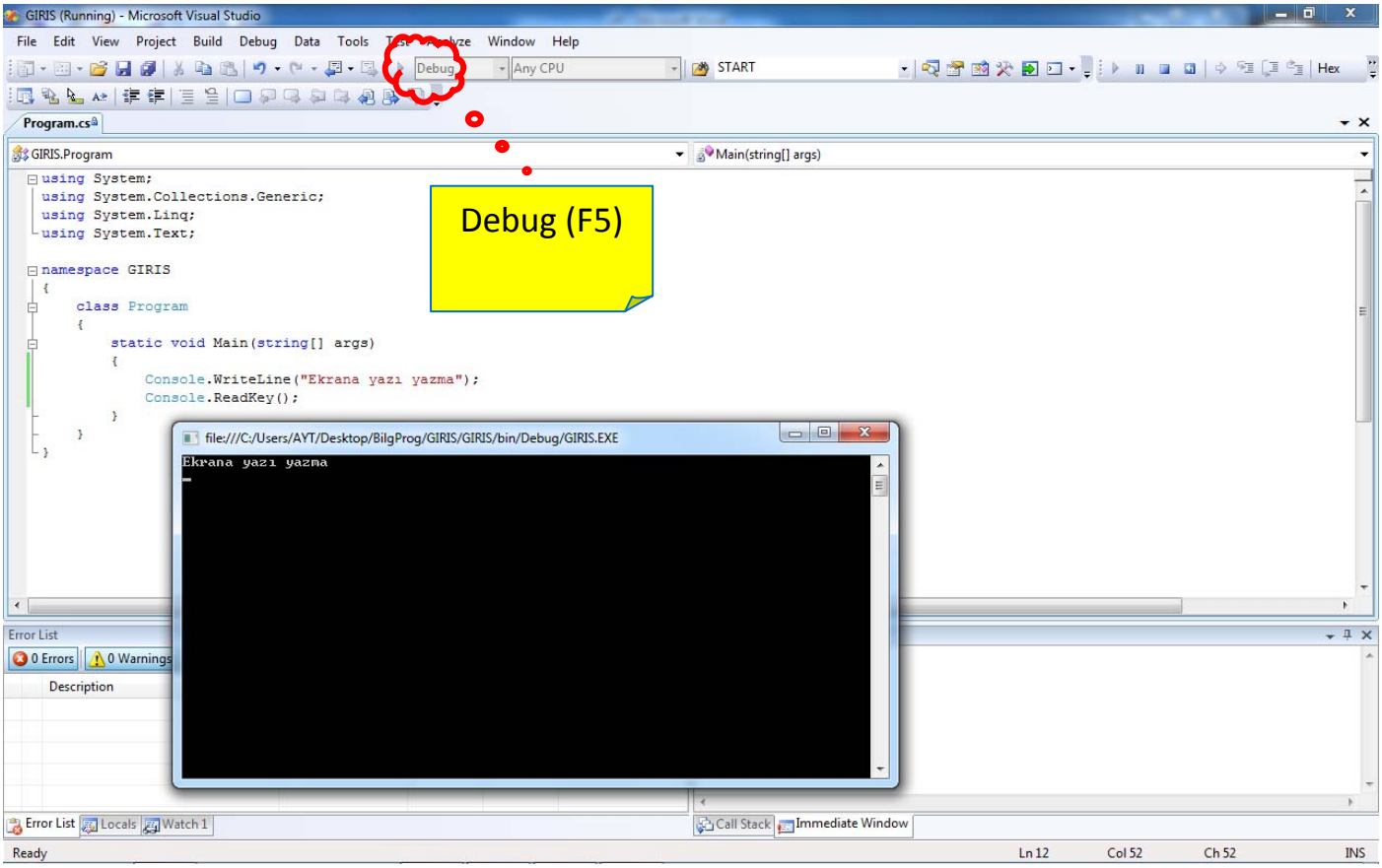

Start Page - Microsoft Visual Studio
File Edit View Data Tools Test Analyze Window Help
Start Page
Visual Studio Team System 2008
Recent Projects
TextHello
WorkflowConsoleApplication
TestProject1
WindowsFormsApplication3
ConsoleApplication2
WindowsFormsApplication2
Open: Project...
Create: Project...
Getting Started
What's new in Visual C#?
Create Your First Application
How Do I ... ?
Learn Visual C#
Download Additional Content
MSDN Forums
Visual C# Developer Center
Extend Visual Studio
Solution Explorer
Error List
0 Errors
0 Warnings
Description
New Project
Project types:
Visual C#
Windows
Web
Smart Device
Office
Database
Reporting
Test
WCF
Workflow
Database Projects
Other Languages
Distributed Systems
Other Project Types
Test Projects
Templates:
.NET Framework 3.5
Visual Studio installed templates
Windows Forms Application
WPF Application
Console Application
Windows Service
WPF User Control Library
Class Library
WPF Browser Application
Empty Project
WPF Custom Control Library
Windows Forms Control Library
My Templates
Search Online Templates...
A project for creating a command-line application (.NET Framework 3.5)
Name: ConsoleApplication1
Location: C:\Users\AYT\Desktop\Bolum1
Browse...
Solution Name: ConsoleApplication1
Create directory for solution
OK
Cancel
Konsol uygulaması için Yeni Proje oluşturma

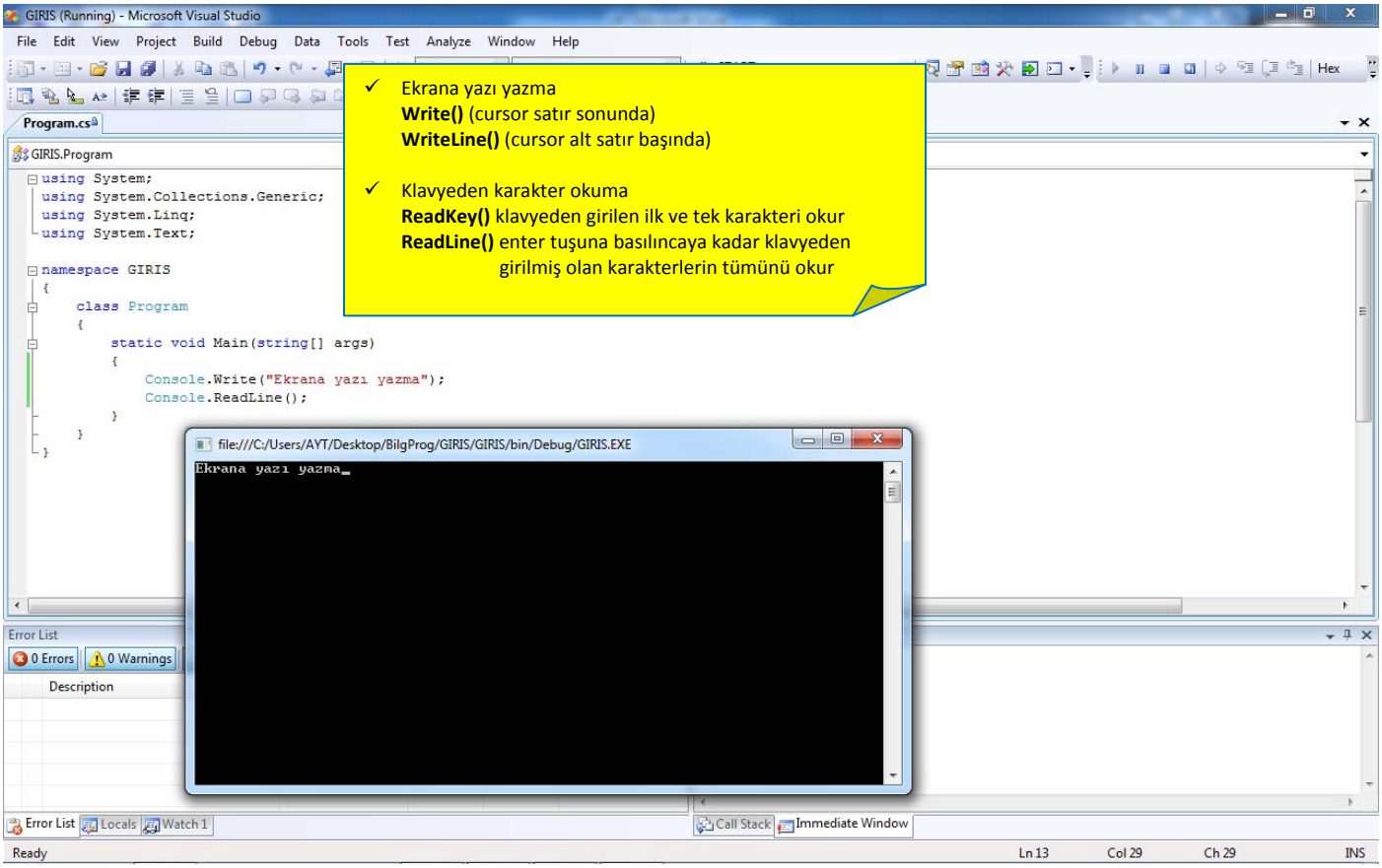

Yeni Projenin kaydedilmesi
Start Page - Microsoft Visual Studio
File Edit View Data Tools Test Analyze Window Help
Start Page
Microsoft Visual Studio Team System 2008
Recent Projects
TextHello
WorkflowConsoleApplication
TestProject1
WindowsFormsApplication3
ConsoleApplication2
WindowsFormsApplication2
Open: Project...
Create: Project...
Getting Started
What's new in Visual C#?
Create Your First Application
How Do I ... ?
Learn Visual C#
Download Additional Content
MSDN Forums
Visual C# Developer Center
Extend Visual Studio
New Project
Project types:
Visual C#
Windows
Web
Smart Device
Office
Database
Reporting
Test
WCF
Workflow
Database Projects
Other Languages
Distributed Systems
Other Project Types
Test Projects
Templates:
.NET Framework 3.5
Visual Studio installed templates
Windows Forms Application
WPF Application
Console Application
Windows Service
WPF User Control Library
Class Library
WPF Browser Application
Empty Project
WPF Custom Control Library
Windows Forms Control Library
My Templates
Search Online Templates...
A project for creating a command-line application (.NET Framework 3.5)
Name: GIRIS
Location: C:\Users\AYT\Desktop\BilgProg
Browse...
Solution Name: GIRIS
Create directory for solution
OK
Cancel
Solution Explorer
Error List
0 Errors
0 Warnings
Description


GIRIS - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
START
Program.cs
Start Page
GIRIS.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace GIRIS
{
class Program
{
static void Main(string[] args)
{
}
}
}
C# programının yazılacağı dosya
Solution Explorer - GIRIS
Solution 'GIRIS' (1 project)
GIRIS
Properties
AssemblyInfo.cs
References
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Xml
System.Xml.Linq
Program.cs
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Column
Project
Solution Explorer
Properties
Ready
Ln 15
Col 1
Ch 1
INS


GIRIS - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
START
Start Page
Program.cs*
GIRIS.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace GIRIS
{
class Program
{
static void Main(string[] args)
{
Console.Wri
}
}
}
IntelliSense özelliği
SetWindowPosition
SetWindowSize
Title
TreatControlCAsInput
WindowHeight
WindowLeft
WindowTop
WindowWidth
Write
WriteLine
void Console.WriteLine(string format, params object[] arg) (+ 18 overload(s))
Writes the text representation of the specified array of objects, followed by the current line terminator, to the standard output stream using the specified format information.
Exceptions:
System.IO.IOException
System.ArgumentNullException
System.FormatException
Solution Explorer - GIRIS
Solution 'GIRIS' (1 project)
GIRIS
Properties
AssemblyInfo.cs
References
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Xml
System.Xml.Linq
Program.cs
Error List
1 Error
0 Warnings
0 Messages
Description
File
Line
Column
Project
1 ; expected
Program.cs
12
24
GIRIS
Solution Explorer
Properties
Ready
Ln 12
Col 24
Ch 24
INS


GIRIS - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
START
Start Page
Program.cs
GIRIS.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace GIRIS
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Ekrana yazı yazma"); Console.ReadKey();
}
}
}
✓ Komutlar ; ile sonlanır
✓ Bir satıra birden fazla komut yazılabilir
Solution Explorer - GIRIS
Solution 'GIRIS' (1 project)
GIRIS
Properties
AssemblyInfo.cs
References
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Xml
System.Xml.Linq
Program.cs
Error List
0 Errors
0 Warnings
0 Messages
Description
File
Line
Column
Project
Solution Explorer
Properties
Item(s) Saved
Ln 12
Col 71
Ch 71
INS


GIRIS (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
START
Program.cs
GIRIS.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace GIRIS
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Ekrana yazı yazma");
Console.ReadKey();
}
}
}
Debug (F5)
file:///C:/Users/AYT/Desktop/BilgProg/GIRIS/GIRIS/bin/Debug/GIRIS.EXE
Ekrana yazı yazma
Error List
0 Errors
0 Warnings
Description
Error List
Locals
Watch 1
Call Stack
Immediate Window
Ready
Ln 12
Col 52
Ch 52
INS


GIRIS (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Program.cs
GIRIS.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace GIRIS
{
class Program
{
static void Main(string[] args)
{
Console.Write("Ekrana yazı yazma");
Console.ReadLine();
}
}
}
✓ Ekrana yazı yazma
Write() (cursor satır sonunda)
WriteLine() (cursor alt satır başında)
✓ Klavyeden karakter okuma
ReadKey() klavyeden girilen ilk ve tek karakteri okur
ReadLine() enter tuşuna basılıncaya kadar klavyeden girilmiş olan karakterlerin tümünü okur
file:///C:/Users/AYT/Desktop/BilgProg/GIRIS/GIRIS/bin/Debug/GIRIS.EXE
Ekrana yazı yazma
Error List
0 Errors
0 Warnings
Description
Error List
Locals
Watch 1
Call Stack
Immediate Window
Ready
Ln 13
Col 29
Ch 29
INS

VERİ TÜRLERİ VE DEĞİŞKENLER

- ✓ Bir ***ifade (statement)***, bir eylemi gerçekleştiren komuttur. İfadeler ***yöntemlerin (method)*** içinde bulunurlar. Bir yöntem, bir ***sınıf (class)*** içindeki adlandırılmış ve sıralanmış ifadeler olarak düşünülebilir. *Main* yöntem için bir örnektir (*Main* yöntemi özeldir programın başlangıç noktasını belirler).

  Program.cs dosyası, *Main* adlı bir yöntemi çağıran *Program* adındaki sınıfı tanımlar. Bütün yöntemler bir sınıfın içinde tanımlanmalıdır.

  System.Console.ReadKey();

System	Ad Alanı (namespace)
Console	Sınıf (class)
ReadKey	Yöntem (method)

  Console sınıfı *System* ad alanında yer alır. Bunun anlamı *Console*'un tam adının *System.Console* olduğudur. *Using* yönergesi kullanılırsa tam ad yerine kısa ad yazılabilir.

  using System;

  using yönergeleri bir kaynak dosyanın en üstüne veya bir ad alanında ilk ifade olarak yazılabilir. Sadece *using* yönergesini içeren kaynak dosyada veya ad alanında sınıf adları kısa yazılabilir.

  Console.ReadKey();

- ✓ C#'daki ifadeler iyi tanımlanmış bir kurallar kümesine uymalıdırlar. Bu kurallar topluca ***sözdizimi (syntax)*** olarak bilinir.

  İfadelerin topluca adı ***dil yapısı***dır.

  C# sözdizimi kurallarından biri her ifadenin bir **;** ile bitmesi gereğidir.

- ✓ ***Tanımlayıcılar (identifiers)***, öğeleri programlarınızda tanımlamak için kullandığınız adlardır. Tanımlayıcılar seçilirken aşağıdaki sözdizimi kurallarına uyulmalı:
  1) Yalnızca harf (büyük veya küçük) ve rakam kullanılmalı
  2) Harfle başlamalı (alt çizgi harf kabul edilir)

sonuc, _skor, fortyTwo, plan9	geçerli
sonuc%, fortyTwo$, 9plan	geçersiz

  C# büyük küçük harf duyarlı bir dildir. fortyTwo ve FortyTwo farklı tanımlayıcılar.

- ✓ C# dili, 76 adet tanımlayıcıyı kendi kullanımı için ayırmıştır. Bu tanımlayıcılara ***anahtar sözcükler*** denir ve her birinin özel bir anlamı vardır. c*lass, namespace, using* gibi… (program yazılırken anahtar sözcükler varsayılan ayar olarak mavi görüntülenir)

- ✓ ***Değişken (variable)***, bir değeri tutan depolama alanıdır. Her programda her değişkene benzersiz ve tek bir ad verilmelidir. Değişkenin adı taşıdığı değere başvurmak için kullanılır.

  Değişken adlandırması için öneriler (Microsoft.NET Framework):
  1) Alt çizgi kullanmayın.
  2) Yalnızca büyük küçük harf ayrımına bağlı tanımlayıcı seçmeyin. Aynı anda kullanmak için myVariable ve MyVariable adlı iki değişken adı seçilmemeli.
  3) Ad küçük harfle başlasın
  4) Birden çok sözcükten oluşan tanımlayıcılarda ikinci ve daha sonraki sözcükleri büyük harfle başlatın. (Bu camelCase gösterim biçimi olarak bilinir)

  Not: Microsoft Visual Basic.NET gibi diğer dillerle birlikte çalışacak programlar yazılacaksa ilk iki öneri zorunluluktur.

  score, fortyTwo geçerli ve önerilen biçimde
  _score, FortyTwo geçerli

- ✓ ***Değişkenlerin bildirilmesi (Türünün belirlenmesi):*** C# çok farklı türde değeri işleyebilir (tamsayılar, kayan noktalı sayılar, karakter dizeleri gibi). Bir değişken bildirildiğinde içinde ne tür bir veri tutacağı da belirtilmelidir.

  Bir değişkenin **adı** ve **türü** bir tanımlama ifadesi içinde bildirilir. Böylece değişken tanımlanmış olur.

  Örnek: *age* adında ve *int* değerler taşıyan bir değişkenin bildirilmesi

  ```
  int age;
  ```

  Değişken bildirildikten sonra ona bir değer atanabilir:

  ```
  age=42;
  System.Console.WriteLine(age);          (konsola 42 yazar)
  ```

  Not: Fare işaretçisi bir değişkenin üzerinde tutulursa, değişkenin türünü gösteren ekran ipucu belirir.

  Bir değişkeni bildirdikten sonra, kullanmadan önce ona bir değer atanması zorunludur. Aksi takdirde program derlenmez. Bu gerekliliğin adı Definite Assignment Rule (Kesin Atama Kuralı)'dır.

✓ ***Temel Veri Türleri:*** C#'ın temel veri türleri olarak adlandırılan birkaç tane yerleşik veri türü vardır.

Veri Türü	Açıklama	Aralık	Örnek
short	Tamsayı	-2^{15} ile $2^{15}-1$	short a; a=7;
int	Tamsayı	-2^{31} ile $2^{31}-1$	int count; count=42;
long	Tamsayı	-2^{63} ile $2^{63}-1$	long wait; wait=42**L**;
float	Noktalı Sayı	$1.5x10^{-45}$ ile $3.4x10^{38}$	float away; away=0.42**F**;
double	Noktalı Sayı	$5.0x10^{-324}$ ile $1.7x10^{308}$	double trouble; trouble=0.42;
decimal	Noktalı Sayı	$1.0x10^{-28}$ ile $7.9x10^{28}$	decimal coin; coin=0.42**M**; coin=4; coin=4**M**
char	Tek Karakter	0 ile $2^{16}-1$	char grill; grill='4';
string	Karakter Sıraları	-	string vest; vest="42";
bool	Boolean	doğru ya da yanlış	bool teeth; teeth=false;

2^{15}=32768, 2^{16}=65536, 2^{31}=2147483648, 2^{63}=9223372036854775808

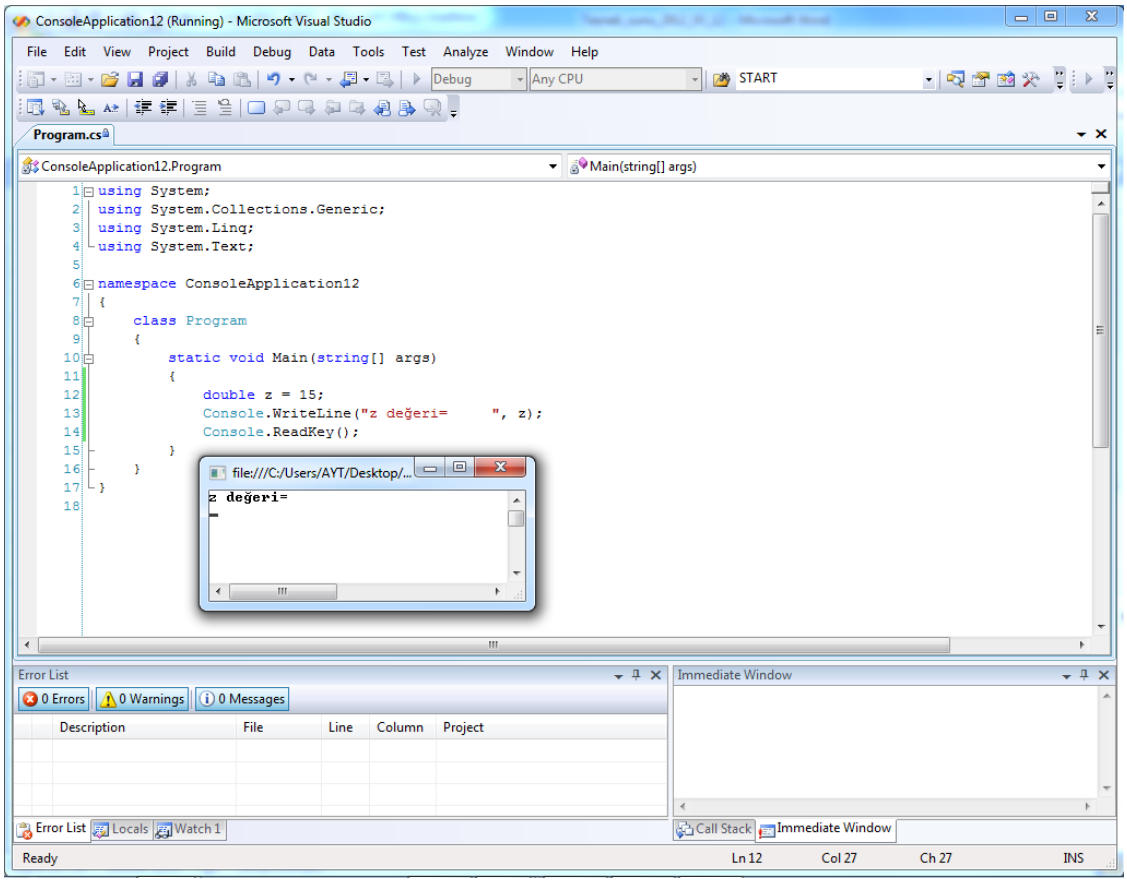
Örnekler:

```
int age;
age=25;

int age=25;

int xSize=4, ySize=5;

int xSize, ySize=5;
```

✓ ***Ekrana Yazı Yazma***

Console.WriteLine("Ad: {0} Soyad: {1} No: {2}", adı, soyadı, ogrNo);

Console.Write("Doğum Yeri: {0} Doğum Tarihi: {1}", dYeri, dTarihi);



✓ ***Program içinde açıklamalar veya notlar yazılması***

// Tek satırda bitirilecek açıklamalar için

/* Tek satırda bitirilecek veya
alt satırlara devam edecek açıklamalar için */

✓ ***Aritmetik İşlemler***

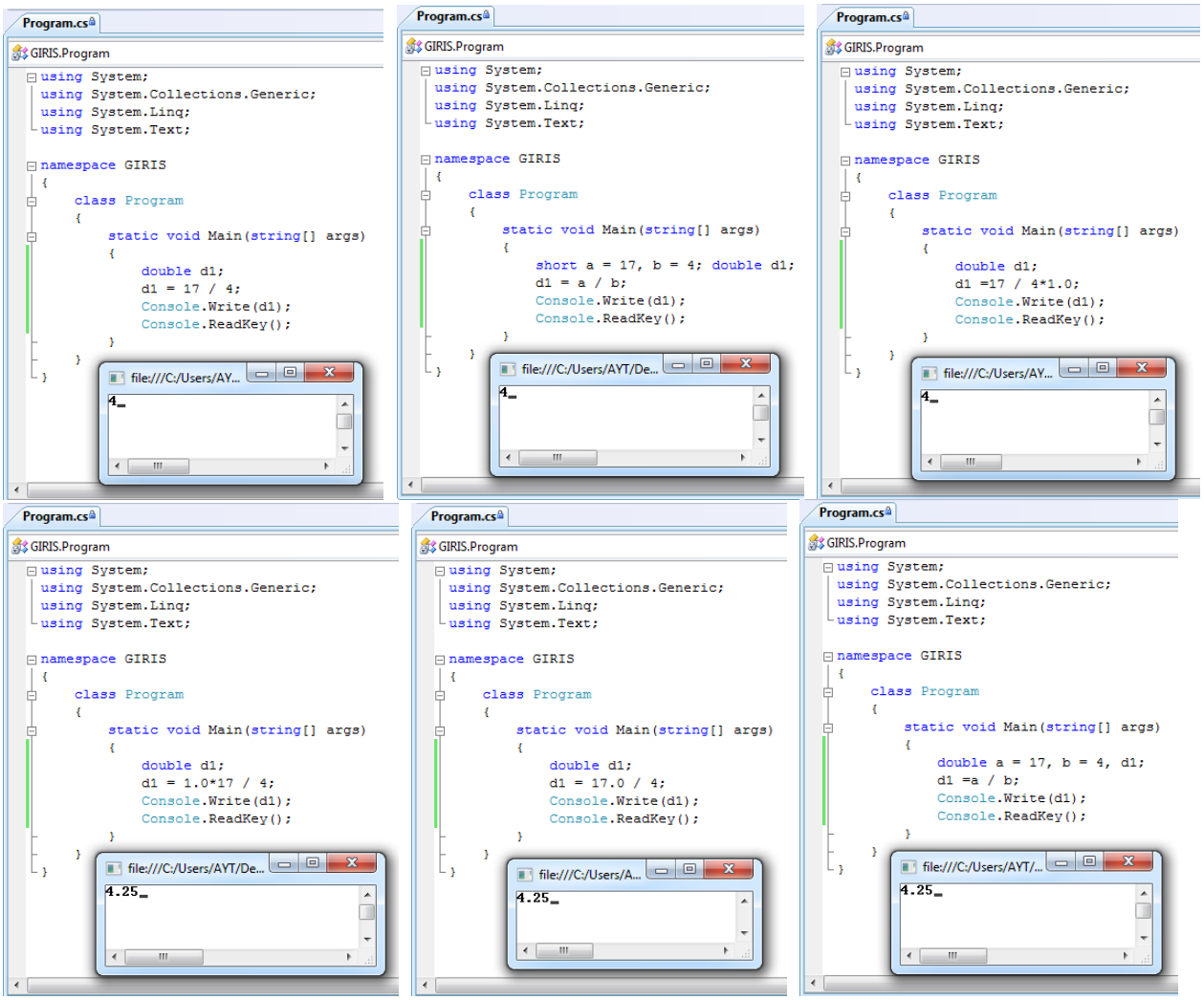
İşlemci (Operator)	Açıklama	Örnek
+	Toplama	d1=27+3; // =30
-	Çıkarma	d1=27-2; /* =25 */
*	Çarpma	d1=27+3*2; /* =33 (60 değil) işlemlerin öncelik sırası vardır.*/ d1=(27+3)*2; /* =60 (parantezler öncelik sırasını değiştirir) */

/	Bölme	double d1; d1=17/4;
		/* =4 hatalı!, iki tamsayı bölünürse kalan ihmal edilir. Sonucun türünü eşitliğin sağ tarafındaki işlem sonucunun türü belirler ve d1 değişkeninin double olarak bildirilmiş olması bu durumu değiştirmez.*/
		double d2; d2=1.0*17/4; /* =4.25 doğru sonuç */
		decimal d3; d3=17/4; // =4 hatalı!
		decimal d4; d4=1.0M*17/4; /* =4.25 doğru sonuç */

%	Modülüs (bölme işleminde kalan)	d1=17%4; // =1 (d1 eşittir 17 mod 4)
-	Negatif (Unary minus)	int i=5, j; j=-i; // j=-5 olur
+	Pozitif (Unary plus)	int i=-5, j; j=+i; // j=-5 olur

Not: İşlemciler **string** ve **bool** dışındaki veri türleriyle kullanılabilir. **string** için bir istisna vardır ve + işlemcisi karakter dizelerini birleştirmek için kullanılabilmektedir:

Console.WriteLine("43"+"1"); // ekrana 431 yazar (44 değil!)



✓ ***Değişkenleri Artırmak ve Azaltmak***

Syntax	Açıklama	Örnek

++i	önek artırma (pre-increment)	int i=5, j, k;
		j=++i; // j=6
		k=i; // k=6

i++	sonek artırma (post-increment)
	i=i+1 ile aynı
		int i=5, j, k;
		j=i++; // j=5
		k=i; // k=6

--i	önek eksiltme (pre-decrement)	int i=5, j, k;
		j=--i; // j=4
		k=i; // k=4

i--	sonek eksiltme (post-decrement)
	i=i-1 ile aynı
		int i=5, j, k;
		j=i--; // j=5
		k=i; // k=4

Örnek:

```
int x=42;
Console.WriteLine(x++);  // ekrana 42 yazar ve x'in değeri 43 olur
x=42;
Console.WriteLine(++x);  // x'in değeri 43 olur ve ekrana 43 yazar
```

Örnek:

```
int d1, d2=5, d3=6, d4, d5;
d1=d2++*--d3;     // d1=5*5=25 olur.
d4=d2;     // d4=6 olur.
d5=d3;     // d5=5 olur.
```

Diğer kısaltmalar ve işlemlerin öncelik sıraları ileride verilecek!

✓ ***Atama İşlemcileri***

İşlemci	Örnek ifade	Eşdeğeri
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

✓ ***Veri Türü Dönüşümleri***

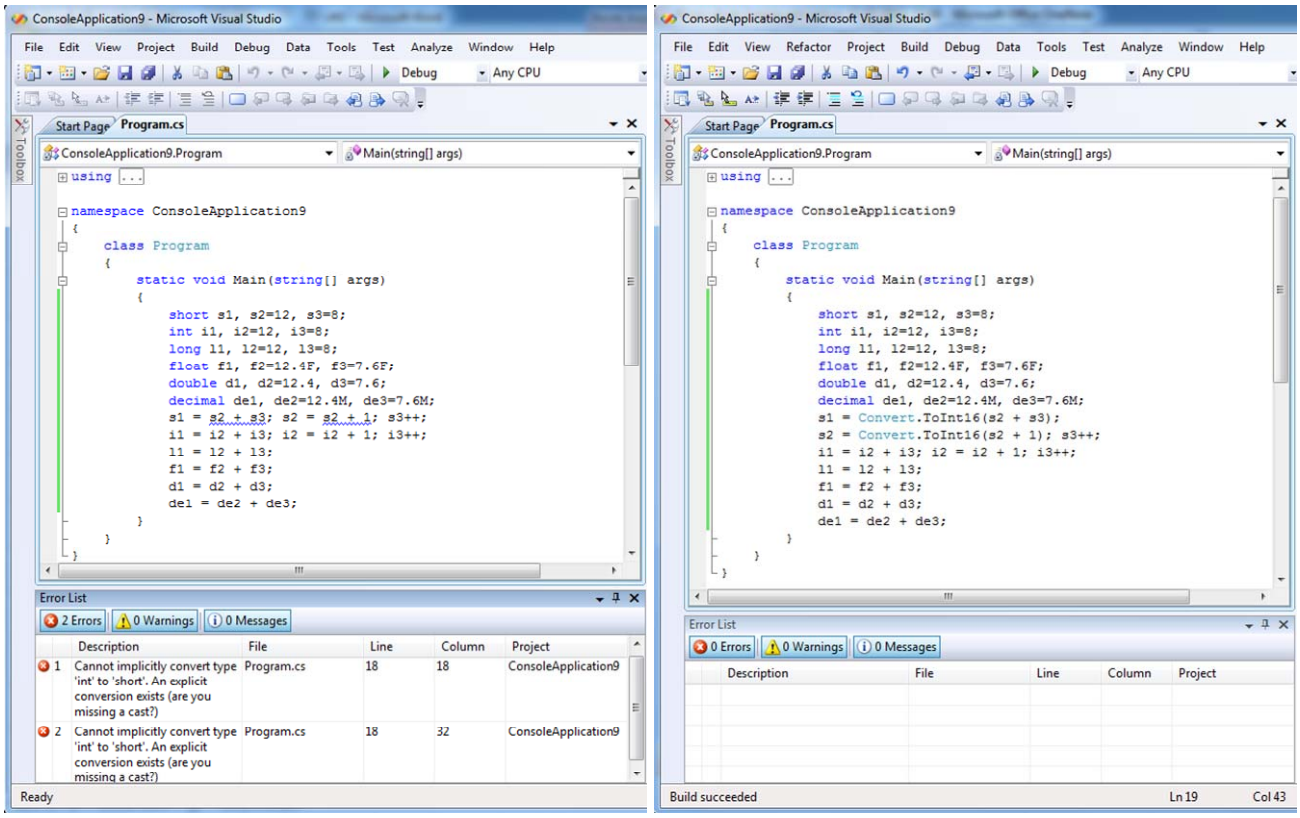
System.Convert (*Convert* sınıfı *System* ad alanında yer alır.)

Komut	Sonuç
Convert.ToInt16(*val*)	*val* **short**'a dönüşür
Convert.ToInt32(*val*)	*val* **int**'e dönüşür
Convert.ToInt64(*val*)	*val* **long**'a dönüşür
Convert.ToSingle(*val*)	*val* **float**'a dönüşür
Convert.ToDouble(*val*)	*val* **double**'a dönüşür
Convert.ToDecimal(*val*)	*val* **decimal**'a dönüşür
Convert.ToChar(*val*)	*val* **char**'a dönüşür
Convert.ToString(*val*)	*val* **string**'e dönüşür
Convert.ToBoolean(*val*)	*val* **bool**'a dönüşür

Not: **Console.ReadLine()** komutu ile klavyeden girilen veriler sadece **string** türünden değişkenler üzerine yazılabilir. **string**'den farklı türdeki bir değişken üzerine bu komut aracılığı ile değer atanabilmesi için yukarıdaki dönüşüm komutlarından uygun olanı kullanılmalıdır.

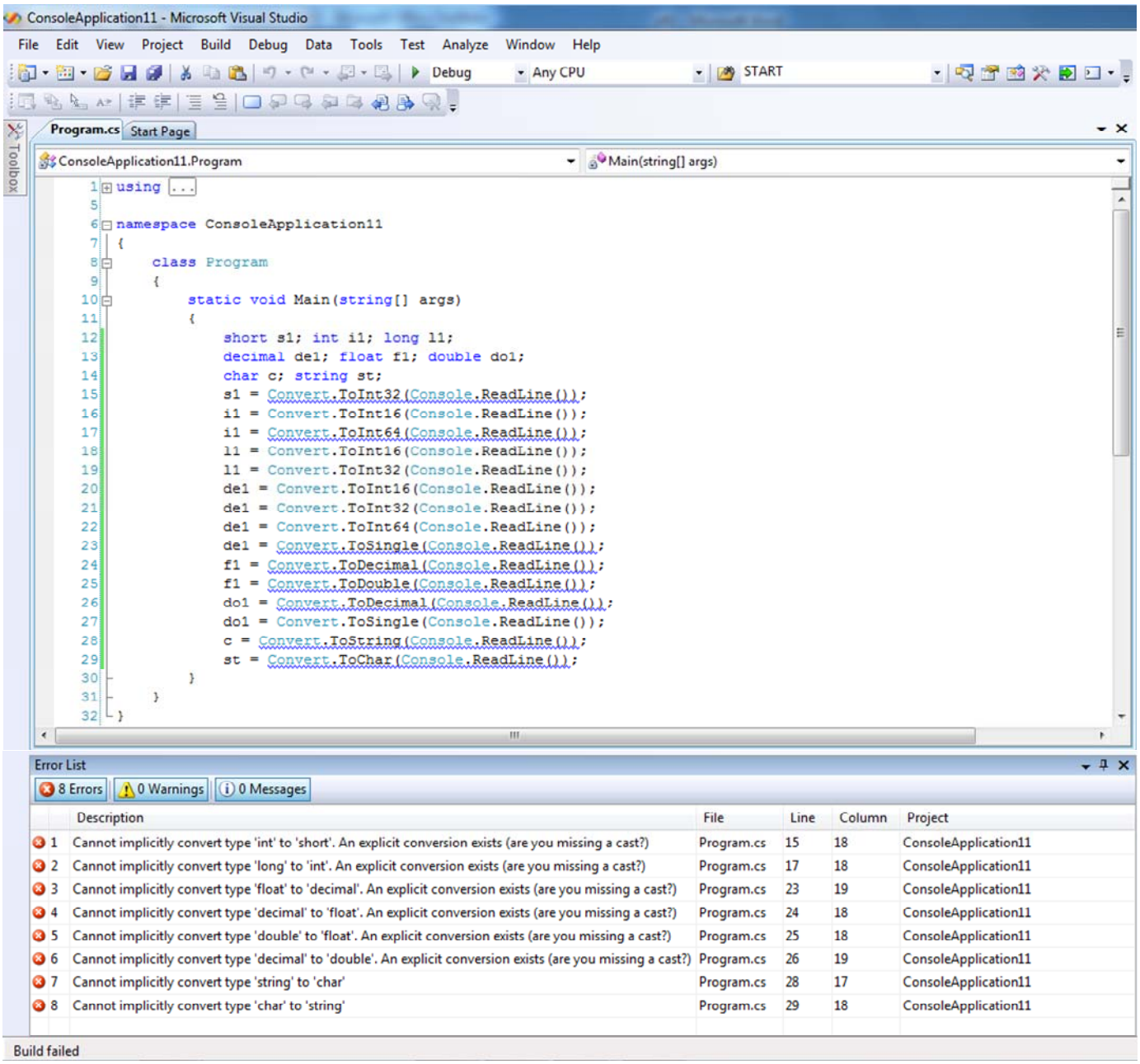
```
int a;
a=Convert.ToInt32(Console.ReadLine());

int a;
a=Convert.ToInt32("42");          // a'ya tamsayı olarak 42 atanır.
```

Not:



Short türündeki iki değişken arasındaki aritmetik işlemin sonucu short türünden bir değişken üzerine atanmak istenirse program derlenmez. Convert.ToInt16() kullanıldığında program derlenir ve aritmetik işlemin sonucu short veri türünün sınırlarını aşmadığı sürece program çalışır. Short türü değişkenler önek veya sonek ile kullanılırsa ve diğer veri türleri için bu kısıtlama yoktur.

Not:


ConsoleApplication11 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
START
Program.cs
Start Page
Toolbox
ConsoleApplication11.Program
Main(string[] args)
using ...
namespace ConsoleApplication11
{
class Program
{
static void Main(string[] args)
{
short s1; int i1; long l1;
decimal de1; float f1; double do1;
char c; string st;
s1 = Convert.ToInt32(Console.ReadLine());
i1 = Convert.ToInt16(Console.ReadLine());
i1 = Convert.ToInt64(Console.ReadLine());
l1 = Convert.ToInt16(Console.ReadLine());
l1 = Convert.ToInt32(Console.ReadLine());
de1 = Convert.ToInt16(Console.ReadLine());
de1 = Convert.ToInt32(Console.ReadLine());
de1 = Convert.ToInt64(Console.ReadLine());
de1 = Convert.ToSingle(Console.ReadLine());
f1 = Convert.ToDecimal(Console.ReadLine());
f1 = Convert.ToDouble(Console.ReadLine());
do1 = Convert.ToDecimal(Console.ReadLine());
do1 = Convert.ToSingle(Console.ReadLine());
c = Convert.ToString(Console.ReadLine());
st = Convert.ToChar(Console.ReadLine());
}
}
}
Error List
8 Errors
0 Warnings
0 Messages
Description File Line Column Project
1 Cannot implicitly convert type 'int' to 'short'. An explicit conversion exists (are you missing a cast?) Program.cs 15 18 ConsoleApplication11
2 Cannot implicitly convert type 'long' to 'int'. An explicit conversion exists (are you missing a cast?) Program.cs 17 18 ConsoleApplication11
3 Cannot implicitly convert type 'float' to 'decimal'. An explicit conversion exists (are you missing a cast?) Program.cs 23 19 ConsoleApplication11
4 Cannot implicitly convert type 'decimal' to 'float'. An explicit conversion exists (are you missing a cast?) Program.cs 24 18 ConsoleApplication11
5 Cannot implicitly convert type 'double' to 'float'. An explicit conversion exists (are you missing a cast?) Program.cs 25 18 ConsoleApplication11
6 Cannot implicitly convert type 'decimal' to 'double'. An explicit conversion exists (are you missing a cast?) Program.cs 26 19 ConsoleApplication11
7 Cannot implicitly convert type 'string' to 'char' Program.cs 28 17 ConsoleApplication11
8 Cannot implicitly convert type 'char' to 'string' Program.cs 29 18 ConsoleApplication11
Build failed

✓ ***Matematik Fonksiyonları***

System.Math (*Math* sınıfı *System* ad alanında yer alır.)

Fonksiyon	Açıklama
Math.E	e sayısını verir (double olarak) e≈2.71
Math.PI	π sayısını verir (double olarak) π≈3.14
Math.Sin(double a)	Sinüs (a radyan)
Math.Cos(double a)	Kosinüs (a radyan)
Math.Tan(double a)	Tanjant (a radyan)
Math.Sinh(double a)	Sinüs Hiperbolik (a radyan)
Math.Asin(double a)	Sinüsün Tersi (-1≤a≤1)
Math.Sqrt(double a)	Karekök
Math.Abs(a)	Mutlak değer
Math.Exp(double x)	e^x
Math.Pow(double x, double y)	Üslü işlemler (x^y)
Math.Log(double a)	Doğal logaritma (lna)
Math.Log10(double a)	Logaritma ($\log_{10}a$)
Math.Log(double a, double b)	a'nın b tabanında logaritması ($\log_b a$)
Math.Min(double a, double b)	a ve b'den küçük olanı verir
Math.Max(double a, double b)	a ve b'den büyük olanı verir
Math.Ceiling(double a)	a bir üst sayıya yuvarlanır (6.2, 7'ye)
Math.Floor(double a)	a bir alt sayıya yuvarlanır (6.2, 6'ye)
Math.Round(double a)	a yakın olan tamsayıya yuvarlanır (6.2, 6'ye – 6.5, 6'ya – 6.6, 7'ye)
Math.Round(double a, int b)	a, virgülden sonra b kadar basamağı olacak şekilde yuvarlanır (a=6.54321 ve b=2 ise sonuç 6.54)

Notlar:

1. Math.Abs() hariç hepsi double türünde sonuç üretir. Math.Abs()'nin sonucu argümanının türündedir.
2. Derece Radyan dönüşümü: 30°=(30*π/180) rad≈0.52 rad
   Radyan Derece dönüşümü: 1.05 rad=(1.05*180/ π) derece≈60°

✓ ***İlişkisel ve Mantıksal İşlemciler (Boolean İşlemcileri)***

İlişkisel İşlemciler

İşlemci	Anlamı
==	Eşittir
!=	Eşit değildir
<	Küçüktür
<=	Küçüktür veya eşittir
>	Büyüktür
>=	Büyüktür veya eşittir

Mantıksal İşlemciler

İşlemci	Yapılan İşlem
&&	AND (VE)
||	OR (VEYA)
!	NOT (DEĞİL)

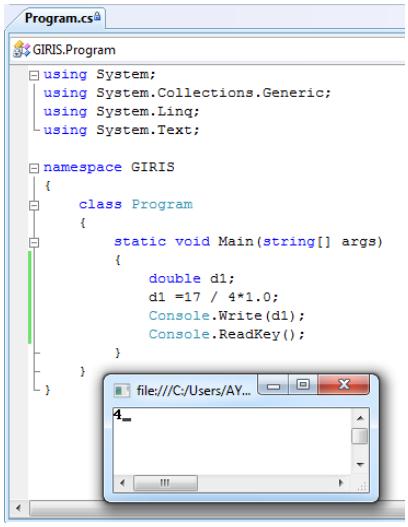
İlişkisel işlemciler iki değeri karşılaştırarak *doğru (true)* veya *yanlış (false)* şeklinde bir sonuç üretirler.
İlişkisel ve mantıksal işlemciler *doğru* sonuç için daima 1 *yanlış* sonuç için ise 0 değerini verirler.
Mantıksal işlemciler *doğru* ve/veya *yanlış* sonuçları birleştirip tek bir sonuç verirler. 0 ve 1 değerleri ile işlem yaparlar.

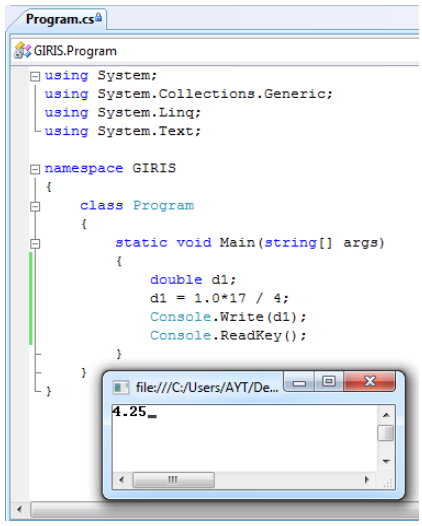
Mantıksal İşlemci Tablosu

x	y	x&&y	x||y	!x
0	0	0	0	1
0	1	0	1	1
1	1	1	1	0
1	0	0	1	0

✓ ***İşlemcilerin Öncelik Sırası***

Öncelik	İşlemciler
En Yüksek	++ (önek), -- (önek), (), + (pozitif), - (negatif), !
	*, /, %
	+ (toplama), - (çıkarma)
	<, >, <=, >=
	==, !=
	&&
	||
	=, *=, /=, %=, +=, -=
En Düşük	++ (sonek), -- (sonek)





✓ ***Karar İfadeleri (Koşula Bağlı Çalışma)***

"if"
"if" – "else"
"if" – "else if" – "else"

... **if (*şart*)** **{Komutlar;}** ...	... **if (*şart*)** **{Komutlar1;}** **else {Komutlar2;}** ...	... **if (*şart1*)** **{Komutlar1;}** **else if (*şart2*)** **{Komutlar2;}** **else if (*şart3*)** **{Komutlar3;}** **else {Komutlar4;}** ...
şart doğruysa Komutlar işlenir. *şart* yanlışsa Komutlar işlenmez.	*şart* doğruysa sadece Komutlar1 işlenir. *şart* yanlışsa sadece Komutlar 2 işlenir. Not: Aşağıdaki kod yukarıdaki kodla aynı değil! ... **if (*şart*)** **{Komutlar1;}** **Komutlar2;** ...	*şart1* doğruysa sadece Komutlar1 işlenir. *şart1* yanlışsa *şart2* kontrol edilir eğer doğruysa sadece Komutlar2 işlenir. *şart1* ve *şart2* yanlışsa *şart3* kontrol edilir eğer doğruysa sadece Komutlar3 işlenir. *şart1*, *şart2* ve *şart3* yanlışsa sadece Komutlar4 işlenir.

Eğer *şart*tan sonra tek komut yazılacaksa, komutun **kod bloğu {...}** içinde olması zorunlu değildir.

✓ ***Karar İfadeleri (Koşula Bağlı Çalışma)***

"switch" kalıbı

… **switch (*değişken*)** **{** **case *sabit1*:** **Komutlar1;** **break;** **case *sabit2*:** **Komutlar2;** **break;** **default:** **Komutlar3;** **break;** **}** …	≠	… **switch (*değişken*)** **{** **case *sabit1*:** **Komutlar1;** **break;** **case *sabit2*:** **Komutlar2;** **break;** **}** **Komutlar3;** …

değişken == *sabit1* ise sadece Komutlar1 işlenir ve break deyimi görülünce switch kalıbının dışına çıkılır.

değişken == *sabit2* ise sadece Komutlar2 işlenir ve break deyimi görülünce switch kalıbının dışına çıkılır.

Değişken ile aynı değeri taşıyan sabit yoksa Komutlar3 işlenir ve break deyimi görülünce switch kalıbının dışına çıkılır. (default bölümünün tanımlanması isteğe bağlıdır, bu bölüm tanımlanmasa da switch kalıbı çalışır. Sağ taraftaki örnekte olduğu gibi default bölümü yoksa önce switch kalıbından çıkılır sonra Komutlar3 işlenir, switch kalıbı içinde herhangi bir işlem yapılmamış olur.)

case önündeki sabitler (sabit1, sabit2, …) birbirinden farklı olmalı.
switch kalıpları birbiri içinde kullanılabilir.

✓ ***Karar İfadeleri (Koşula Bağlı Çalışma)***

"if-else if-else" ve "switch" kalıbı

…
if (*day==0*)
{
dayName=*"Pazar"*;
}
else if (*day==1*)
dayName=*"Pazartesi"*;
else if (*day==2*)
dayName=*"Salı"*;
else if (*day==3*)
dayName=*"Çarşamba"*;
…
…
…
else
dayName=*" Bilinmiyor"*;
…
		…
		switch (*day*)
		{
		case *0*:
		dayName=*"Pazar"*;
		break;
		case *1*:
		dayName=*"Pazartesi"*;
		break;
		case 2:
		dayName=*"Salı"*;
		break;
		case 3:
		dayName=*"Çarşamba"*;
		break;
		…
		…
		…
		default:
		dayName=*"Bilinmiyor"*;
		break;
		}
		…

if deyiminde şart ifadesinde tüm ilişkisel ve mantıksal işlemciler kullanılabilir.		switch kalıbında sadece eşitlik (==) için kontrol yapılır.

✓ ***Döngüler***

... **while (*şart*)** **{Komutlar;}** ... *şart* yanlış sonuç verene kadar Komutlar işlenir. *şart* doğru olduğu sürece döngü çalışmasına devam eder. *şart* döngünün başında kontrol edildiğinden *şart* yanlışsa döngü bir defa bile çalışmaz.	... **do** **{Komutlar;}** **while (*şart*);** ... Program döngüye geldiğinde şarta bağlı olmaksızın döngüye giriş yapılır. Komutlar işlendikten sonra *şart* kontrol edilir. *şart* yanlış sonuç verene kadar döngü çalışır. *şart* kontrolü döngü sonunda yapıldığından döngü en az bir defa çalışır.

...
for (*ilk_değer_atama ; şart ; artırma*)
{Komutlar;}
...

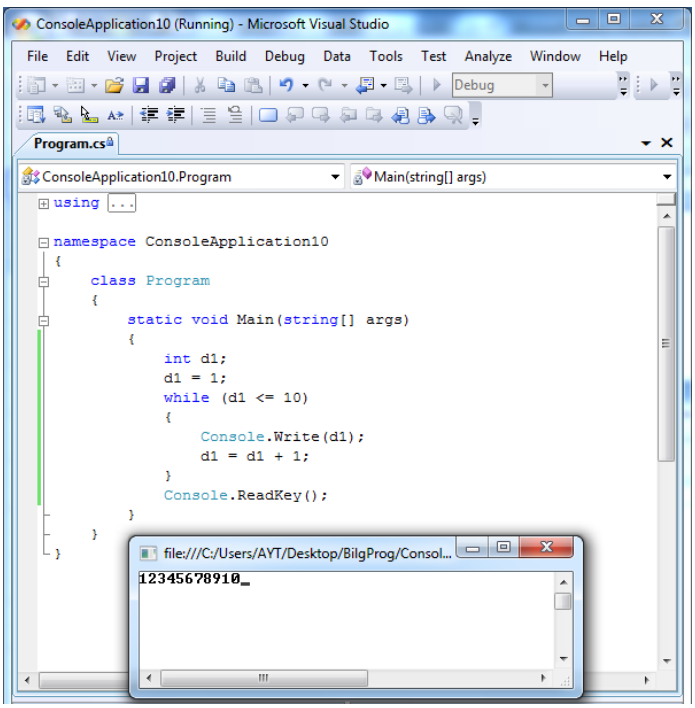
ilk_değer_atama: döngüyü kontrol eden değişkene bir ilk değer verilir. Bu bölüm bir defaya mahsus olmak üzere sadece döngünün başında çalıştırılır.

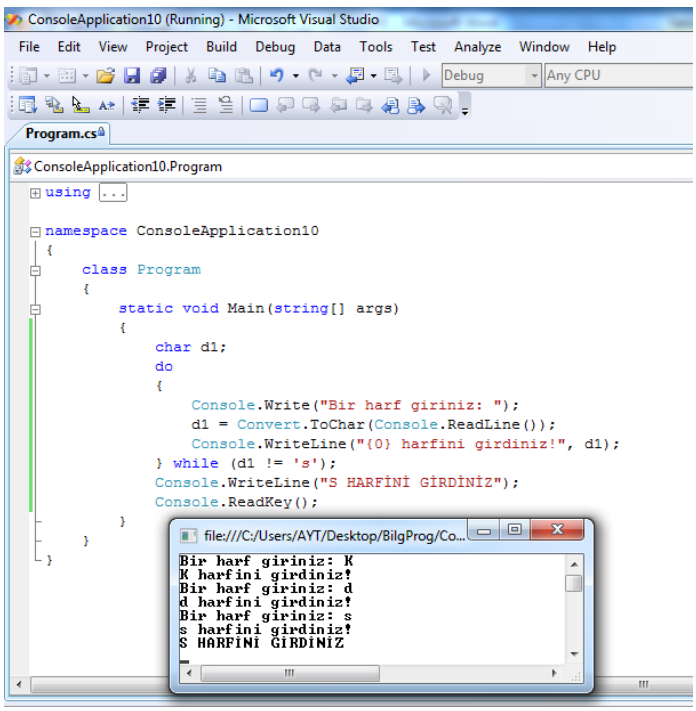
şart: döngünün devam edip etmeyeceğini belirler. *şart* yanlış sonuç verene kadar döngü devam eder. *şart* döngünün her tekrarında döngünün başlangıcında kontrol edilir.

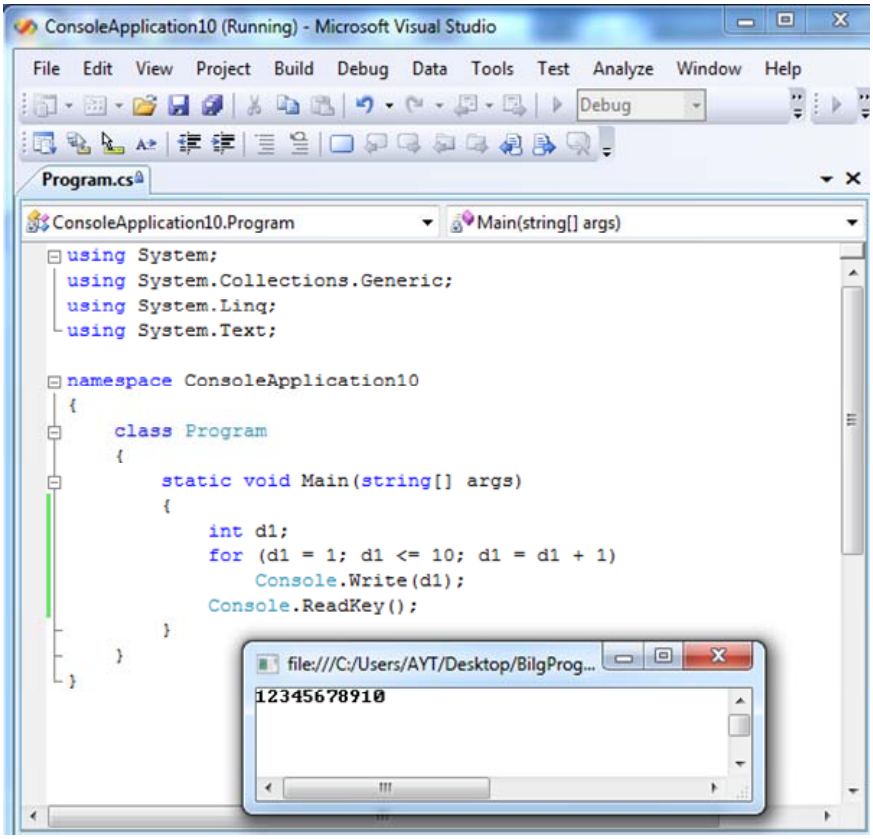
artırma: bu bölümde döngü kontrol değişkeni döngünün her tekrarında belirli bir oranda artırılır veya azaltılır. Artırma bölümü Komutlar'dan sonra *şart* bölümünden önce çalıştırılır.

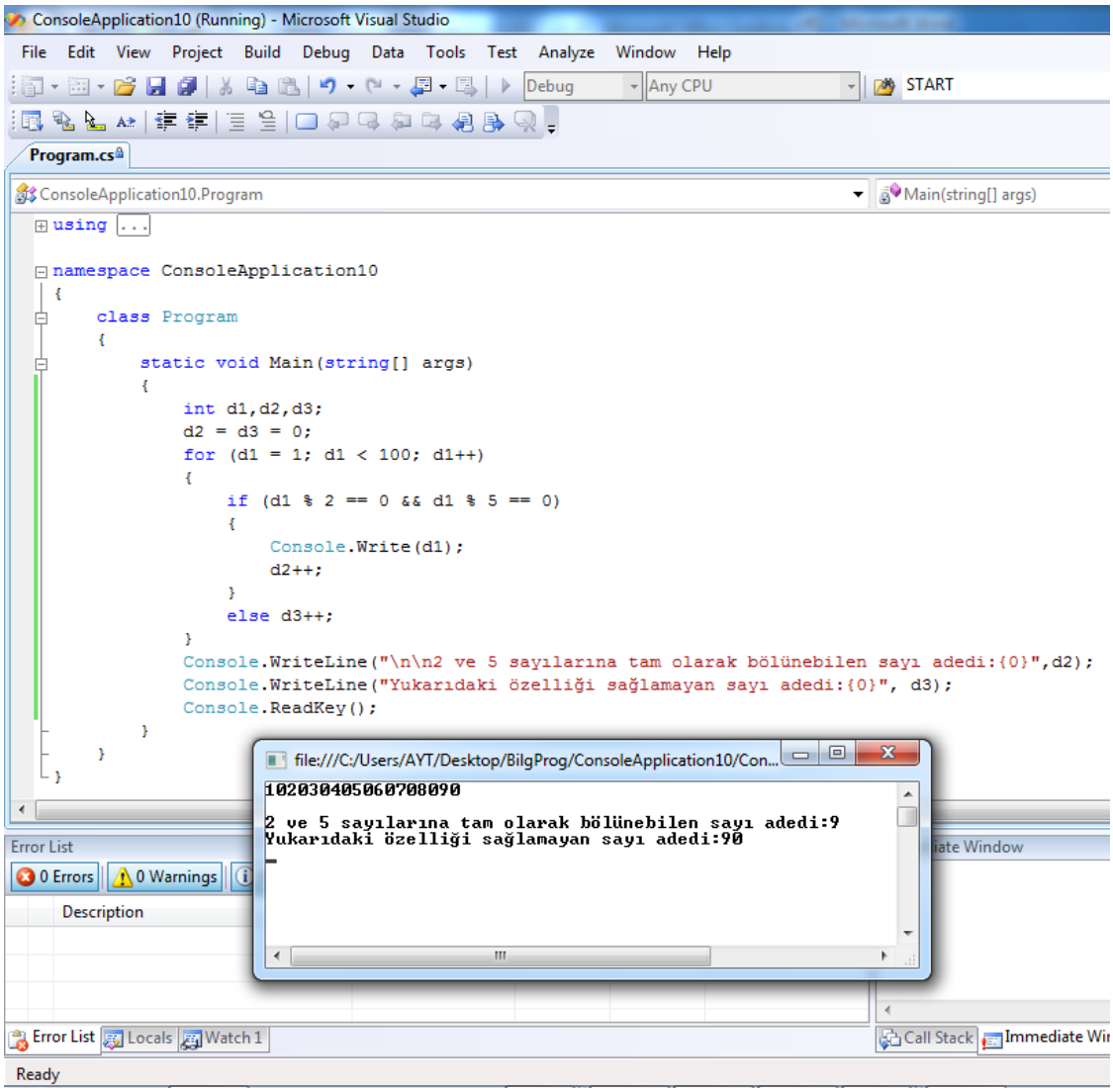
Döngü başlangıcında şart doğru sonuç vermezse döngü hiç çalıştırılmaz.

Eğer Komutlar kısmı tek komuttan oluşuyorsa **kod bloğu {...}** içinde olması zorunlu değildir.






ConsoleApplication10 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Debug
Program.cs
ConsoleApplication10.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication10
{
class Program
{
static void Main(string[] args)
{
int d1;
for (d1 = 1; d1 <= 10; d1 = d1 + 1)
Console.Write(d1);
Console.ReadKey();
}
}
}
file:///C:/Users/AYT/Desktop/BilgProg...
12345678910


ConsoleApplication10 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
START
Program.cs
ConsoleApplication10.Program
Main(string[] args)
using ...
namespace ConsoleApplication10
{
class Program
{
static void Main(string[] args)
{
int d1,d2,d3;
d2 = d3 = 0;
for (d1 = 1; d1 < 100; d1++)
{
if (d1 % 2 == 0 && d1 % 5 == 0)
{
Console.Write(d1);
d2++;
}
else d3++;
}
Console.WriteLine("\n\n2 ve 5 sayılarına tam olarak bölünebilen sayı adedi:{0}",d2);
Console.WriteLine("Yukarıdaki özelliği sağlamayan sayı adedi:{0}", d3);
Console.ReadKey();
}
}
}
file:///C:/Users/AYT/Desktop/BilgProg/ConsoleApplication10/Con...
102030405060708090
2 ve 5 sayılarına tam olarak bölünebilen sayı adedi:9
Yukarıdaki özelliği sağlamayan sayı adedi:90
Error List
0 Errors
0 Warnings
Description
Error List
Locals
Watch 1
Call Stack
Immediate Wi
Ready