

NESNE TABANLI PROGRAMLAMA - I

2019- 2020

Ders Hakkında

Ders Sorumlusu

Öğr.Gör. Ekrem Eşref KILINÇ

Dersle İlgili İletişim İçin

ekremesrefkilinc@gmail.com

ekilinc@mehmetakif.edu.tr

Neler Öğreneceğiz?

- ✓ Nesneye yönelik programla nedir?
(Metot nedir?, sınıf nedir?, nesne nedir?,
nesneler nasıl türetilir?.....)
- ✓ Nesneye yönelik programlamanın
üstünlükleri nelerdir?
- ✓ .NET teknolojisi nedir? Neleri bizlere sağlar
ve avantajları nelerdir?

Dersin İçeriği

Haftalar	İçerik
1.Hafta	.NET Mimarisi ve C#'ın bu mimarideki yeri
2. Hafta	Temel Veri Türleri
3. Hafta	Tür Dönüşümleri
4. Hafta	Operatörler
5. Hafta	Akış Kontrol Mekanizmaları
6. Hafta	Diziler
7. Hafta	Ara Sınav
8. Hafta	Metotlar ve Fonksiyonlar
9. Hafta	Sınıflar, Yapılar, Numaralandırmalar
10. Hafta	İsim Alanları ve System İsim Alanı
11. Hafta	Temel I/O ve String İşlemleri
12. Hafta	Nesne Yönelimli Programlama ve Kalıtım
13. Hafta	İleri C# Konuları
14. Hafta	Final

Kaynaklar

- Her yönüyle C# 4.0
 - Sefer Algan, Pusula, 2004
- C# Temel Başlangıç Kılavuzu
 - Bradley Jones, Sistem, 2003



C# ve .NET Nedir?

Bilgisayar dillerini düzeylerine göre şu şekilde sınıflandırabiliriz :



Script Dilleri: JavaScript, Perl Script, VBScript...



Yüksek Düzeyli Diller: Vbasic, Delphi....



Orta Düzeyli Diller: C# ve Java



Düşük Düzeyli Diller: C/ C++



Assembly: Assembly



C#, C/C++ ve Java dillerinden türeyen, güçlü, basit, esnek, tip güvenli, modern ve Microsoft .NET platformu için sıfırdan geliştirilmiş bir programlama dilidir.



C#, eskiden beri programcıların yaygın bir biçimde kullandığı C/C++ ve Java dillerine benzerliği ile tanınan bir programlama dilidir. İlk aşamada çok benzer bir dil olarak görünse de bu iki dilden farklı bir çok özelliğe sahiptir.



C #, .NET için Microsoft tarafından geliştirilen ve programlama alanında C/C++ ve Java'nın güzel özelliklerini alıp, bu dillerin tehlikeli olabilecek özelliklerini dışarıda bırakan bir dildir.



C#, C/C++ dilinden farklı olarak tamamıyla nesneye yöneliktir. “int”, “double” gibi temel veri türleri dahi birer nesne olarak tanımlanmıştır.

Java dilinden farklı olarak C# dilinde işaretçiler (pointer) kullanılabilmektedir.

Peki Neden C#'ı Tercih Edelim?



Öğrenilmesi kolaydır.



Nesne yönelimli programlamaya tam destek vermektedir.



Yüksek verim.



Güç ve kolaylık arasında dengededir.



Xml desteği sunmaktadır.



Windows Forms mantığını destekler.



İnternet teknolojilerinin gelişimine uyum sağlayabilmiş modern bir dildir.

C# ile Yapılabilecekler



Konsol tabanlı uygulama geliştirme



Windows için program yazma

C# ile Windows için gelişmiş, güçlü, hızlı ve güvenli programlar yazılabilir. Fakat bunun için programın çalıştığı sistemde .NET platformunun yüklü olması gerekir.



ASP.NET için en uygun dildir.

ASP.NET, ASP gibi script yorumlamalı olmayıp tamamen nesne yönelimli haldedir. C# bu konuda büyük kolaylıklar sağlamaktadır.



C# ile Web Servisleri geliştirmek oldukça kolay ve hızlı bir hal almaktadır.

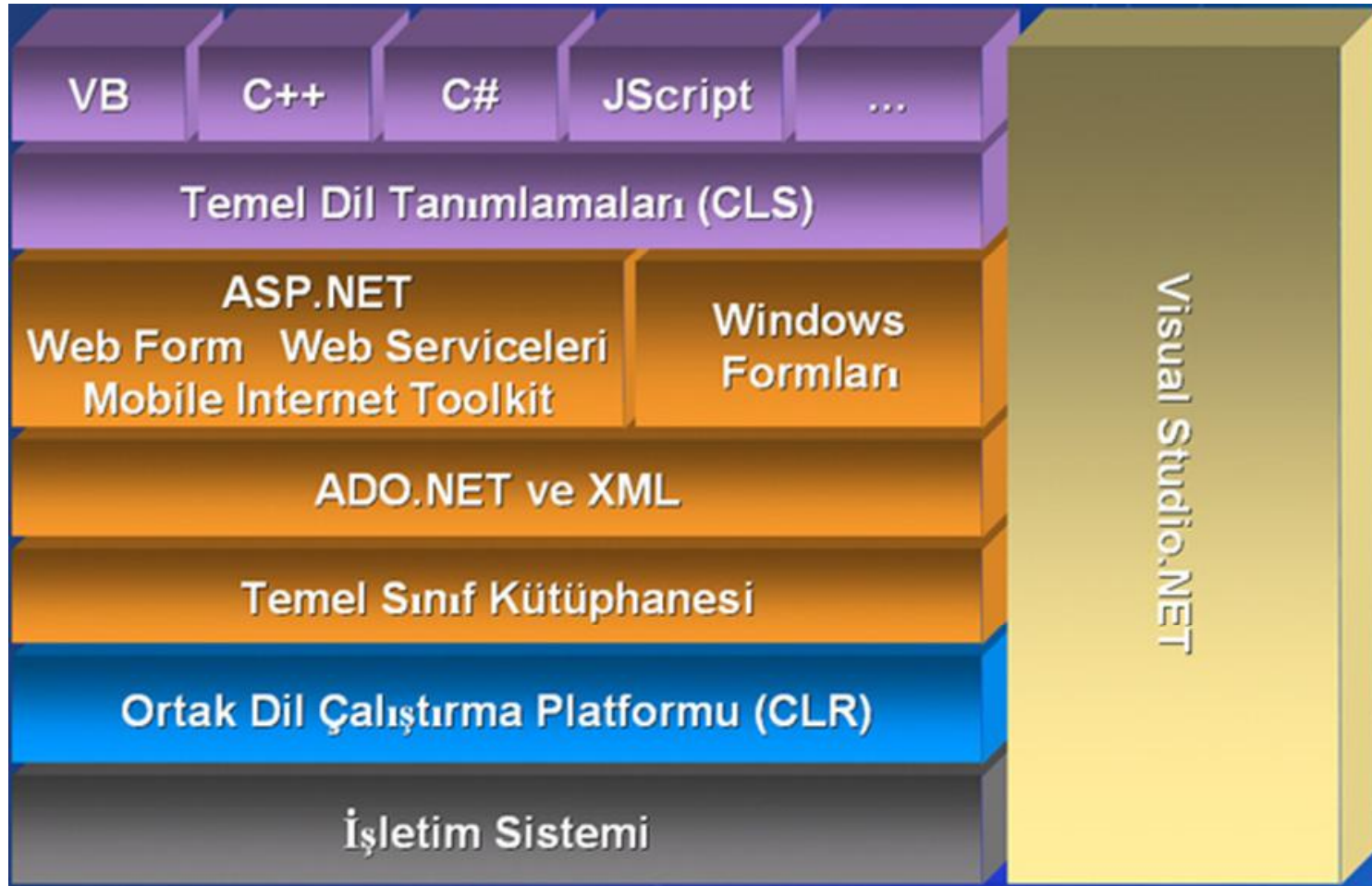


Mobil uygulama geliştirme ve DLL yazabilme.

.NET'in Bileşenleri



Microsoft .NET platformunu aşağıdaki şekilde göstermek mümkündür:





Önceden, geliştirilen yazılımlar direkt olarak makine koduna derlenir ve bu şekilde çalıştırılırlardı. Ayrıca bu programlar, her işletim sistemine özel olarak geliştirilmekte ve derleme işlemi işletim sistemine göre belirlenmekteydi.



Böyle bir yapıda taşınabilirlikten söz etmek mümkün değildir.



Java'da ise, program önce byte code'a çevrilmektedir. Bu kodu JVM(Java Virtual Machine), işletim sisteminin istediği koda çevirmektedir.



.NET platformunda da temel prensip Java ile benzerdir.



.NET kodu önce **IL** (**I**ntermediate **L**anguage)'ye derler ve bu IL kodu çalıştırılmak istendiği zaman .NET **CLR** (**C**ommon **L**anguage **R**untime-Ortak Çalışma Platformu), **JIT** (**J**ust **I**n **T**ime) derleticilerini kullanarak makine diline çevirir.



CLR makine diline çevrilmiş olan program kodunu ön bellekte tutar. Bu büyük performans artışı sağlarken, oldukça büyük miktar belleği kullanmış olur.

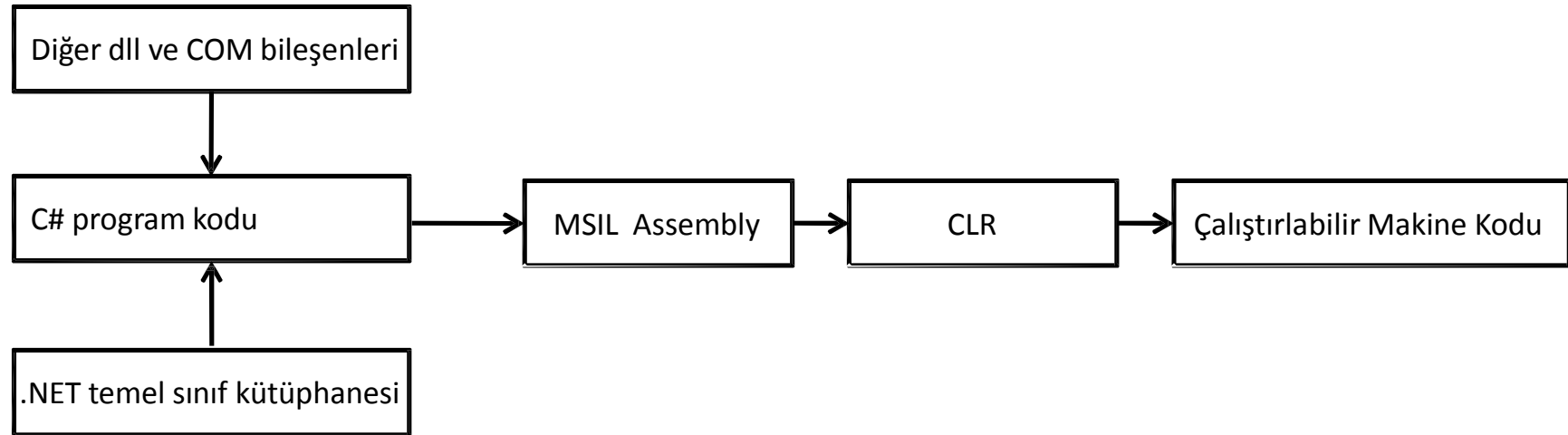


Temel Dil Tanımlamaları **CLS** (**C**ommon **L**anguage **S**pecifications) ve Ortak Tip Sistemi **CTS** (**C**ommon **T**ype **S**ystem) ile .NET uyumlu dillerin hepsi aynı **değişkenler** ve **benzer nesne yönelimli özellikleri taşır**. Örn: C# ile yazılan programdaki temel veri tipleri, VB.NET 'tekiler ile aynı özelliklere sahiptir. Böylece farklı dillerde yazılan bileşenler birbiri ile sorunsuz çalışırlar.

Ortak Dil Çalışma Platformu (CLR)



.NET altyapısında programların çalışmasını kontrol eden, işletim sistemi ile program arasında yer alan ara birimdir.



IL kodu, CLR tarafından çağrılınca JIT derleyicileri tarafından makine diline çevrilir ve çalıştırılır.

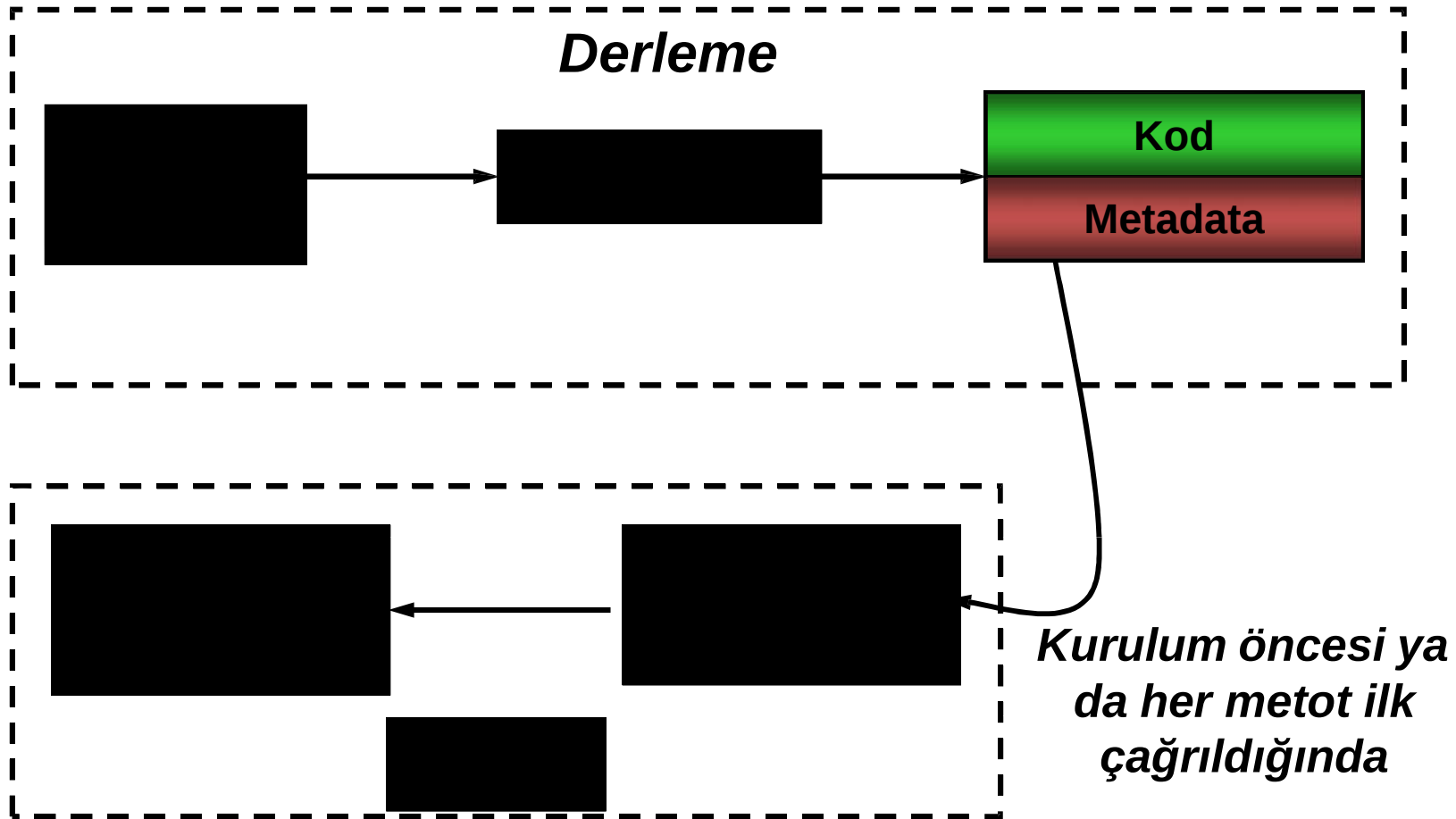
Aradil (IL)

-  İki sayıyı toplayan bir C++ programı Intel işlemcilerde farklı, Sun Sparc işlemcilerde farklı derlenir. Fakat, .NET yapısında kod parçası derlendikten sonra elde edilen IL kodu işlemciye bağımlı bir kod değildir. IL, daha genel ve taşınabilirdir.
-  IL içerisinde, değişken tanımları, değişkenlerin nasıl saklanacağı, metotların nasıl çalıştırılacağı, aritmetik işlemler, mantıksal işlemler, bellek kullanımı ve hata yakalama ve yönetimi dahil birçok işin nasıl yapılacağı açıklanır.
-  Bütün bunların yanında IL'de **Metadata** olarak adlandırılan bir birim daha vardır. Metadata, programda kullanılan verilerin tiplerinin yanında oluşturulan sınıfların metotlarını ve bunların özelliklerini ve diğer bilgileri içerir.

JIT Derleyicileri



Metadata'nın içeriği çalışma zamanında JIT derleyicileri tarafından kullanılır. IL ve Metadata'sı oluşturulan kod parçası, çalıştırılabilir bir yapıdadır.





C# derleyicisi ile IL'e derlenen programı çalıştırırken JIT derleyicileri devreye girer. Bu derleyiciler, programın çalıştırıldığı sistemin ve işlemcinin anlayabileceği makine kodunu oluşturur.

JIT derleyicilerinin Windows ortamı için 3 farklı türü vardır. Bunlar:



Normal JIT: Bu derleyici, IL kodunu makine koduna çevirir ve önbellekte tutar.



Pre-JIT: Tüm programın kodunu makine koduna çevirip sonra çalıştırır.



Eco-JIT: Derlenen ve çalıştırılan program parçaları normal JIT'de hemen hafızadan silinmiyordu. Fakat Eco-JIT'de kullanılabilir hafıza belli bir oranın altına düştüğünde, daha önceden derlenmiş ve çalıştırılmış kısımlar hafızadan silinirler.

CTS ve CLS



CTS sayesinde, programlama dillerinin kullandığı veri türleri arasında uyumluluk vardır. **Örn:**

Int tipi C++.NET ve VB.NET'teki tamsayı tiplerinin kapasiteleri aynıdır. Diğer bir deyişle VB'de geliştirilen bir dll C#'da rahatlıkla kullanılabilir.



CTS sayesinde;

- .NET'de tip güvenli, yüksek performanslı ve kullanılan programlama dilinin diğer .NET uyumlu diller ile entegre bir şekilde çalışması sağlanır.
- Birçok programlama dilinin .NET mimarisinde tam nesne yönelimli olarak .NET için program yazma aracı olmasına imkan verir.
- Herhangi bir .NET uyumlu dilde geliştirilen nesne diğer dillerde de sorunsuz olarak çalıştırılabilir.



.NET platformunu paylaşan dillerin sadece CTS'yi desteklemeleri yetmemektedir. Bunun yanında Ortak Dil Spesifikasyonu da (CLS) desteklemeleri gerekmektedir.



CLS uyumlu bir dil ile geliştirilen bir program ile farklı diller arasında etkileşim sağlanabilir. .NET'in temel sınıf kütüphanelerinin içerisinde yer alan kodların büyük bir bölümü CLS uyumludur.

Desteklenen Programala Dilleri

APL	Fortran	Pascal
C++	Haskell	Perl
C#	Java Language	Python
COBOL	Microsoft JScript®	RPG
Component Pascal	Mercury	Scheme
Curriculum	Mondrian	SmallTalk
Eiffel	Oberon	Standard ML
Forth	Oz	Microsoft Visual Basic®

Assembly



.NET platformu için yazılan bütün kodların sonucunda oluşan .exe ve .dll uzantılı dosyalara genel olarak assembly denilmektedir.



Derlenmiş kodlar ve metadata olarak adlandırılan özniteleyici kodlar Assembly içerisinde bulunurlar.

- Assembly içerisindeki metadata verileri, tür bilgileri ve başka kaynaklara olan bağlantıları saklar.
- Assembly'de ayrıca versiyon bilgisi de tutulur.
- Assembly sayesinde programlar register edilmeye gerek kalmadan direkt kopyalanarak kurulabilirler.

Application Domain

- ✓ Application domain sayesinde aynı anda çalışan birden fazla program veya process birbirinden izole edildiği halde sistemde herhangi bir aksaklığa yol açmadan aralarında veri alış-verişi yapabilirler.

İsim Alanları ve .NET Sınıf Kütüphanesi



.NET Framework'ün programcılara sunduğu bir takım temel türler ve sınıflar mevcuttur. Bütün bu sınıfları ve türleri iyi organize edebilmek için .NET, isim alanı (namespace) kavramını kullanmaktadır.



İsim Alanları, .NET Framework sınıf kütüphanesindeki veri türlerini ve sınıfları kullanabilmek için C# dilinde using anahtar sözcüğü ile birlikte kullanılır ve derleyiciye bildirilir.



.NET'teki sınıf kütüphaneleri bir dilden bağımsız bir yapıdadır.



Program geliştirilirken birbiri ile ilgili olan sınıfların aynı isim alanı içerisine konulması programdaki hataları bulma ve anlaşılabilirlik açısından oldukça önemlidir.

.NET sınıf kütüphanesinde bulunan ve en sık kullanılan sınıf kütüphaneleri şunlardır:



System: .NET ile çalışırken gerekli temel sınıfları içerir. Ayrıca diğer tüm sınıf kütüphaneleri bu isim alanı içinde kümelenmiştir. Konsol temelli uygulamalarda temel giriş çıkış işlemleri için gerekli temel sınıf “Console”, bir çok matematiksel fonksiyonu içinde barındıran “Math” sınıfı da System isim alanı içerisinde yer alır.



System.Data: Veritabanı işlemlerinin tamamı için hazır gelen sınıf kütüphanesine bu isim alanı ile erişilir.

- ✓ **System.Xml:** Veri biçimlendirme ve internetten veri paylaşımı için en çok kullanılan teknolojilerden biri olan XML ile çalışmak için gerekli sınırları içerir.
- ✓ **System.Net:** Dağıtık uygulama geliştirmek için gerekli olan ağ bileşenlerini içeren isim alanıdır.
- ✓ **System.IO:** Dosyalarla çalışmak (okuma/yazma) için gerekli işlemlerini içerir.
- ✓ **System.Windows.Forms:** Windows temelli uygulamalarda kullanılan görsel kontrolleri barındıran isim alanıdır.



Örn:

```
using System;
```

```
namespace Ornekalan1
```

```
{
```

```
    public class Program1
```

```
    {
```

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

```
        {
```

```
        }
```

```
    }
```

```
    public class Program2
```

```
    {
```

```
        public void foksiyon1()
```

```
        {
```

```
        }
```

```
    public void fonksiyon2()
```

```
    {
```

```
    }
```

```
}
```

```
}
```

```
namespace Ornekalan2
```

```
{
```

```
    public class Program3
```

```
    {
```

```
    }
```

```
}
```

C# Komut Satırı Derleyicisi



C# ile yazılan programları ya komut satırı derleyicisi ya da Visual Studio.NET yardımıyla derleyip çalıştırabiliriz.

C# derleyicisinin ismi **csc.exe**'dir.

```
ornek.cs - Notepad
File Edit Format View Help

class ornek_program
{
    static void Main()
    {
        System.Console.WriteLine("ilk uygulama...");
    }
}
```

```
C:\WINDOWS\system32\cmd.exe
C:\>path %path%;C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727

C:\>csc ornek.cs
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.42
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.

C:\>ornek
ilk uygulama...

C:\>
```

