

Nesneye Dayalı Programlama

Sakarya Üniversitesi
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği

Prof. Dr. Ümit KOCABIÇAK
Prof. Dr. Cemil Öz
Öğr. Gör. Nevzat TAŞBAŞI
Öğr. Gör. Sinan İLYAS

2018
1. HAFTA

Ders Hakkında

► Değerlendirme:

- % 45 – Ara Sınav
 - % 25 - Proje
 - % 30 - Ödev (3 adet)
- %100 Toplam
-
- % 55 – Yıl İçi Çalışmalar
 - % 45 – Yıl Sonu sınavı
- %100 Toplam

Ders İçeriği

- .NET Framework Mimarisi ve C#'a Genel Bakış
- Veri Tipleri, Temel Giriş-Çıkış Örnekleri
- Operatörler, Kontrol Deyimleri, Döngüler, Diziler, Düzensiz Diziler, Numaralandırmalar, Yapılar
- Grafikselle Kullanıcı Arayüzü Tasarımı
- Sınıf Kavramı, Sınıf Üyeleri, Nesne Kavramı, Sarmalama
- Metotlar, Overloading kavramı, Operatörlerin Aşırı Yüklenmesi, Kurucu ve Yıkıcılar, this anahtar kelimesi
- Kalıtım, Sealed Sınıf, Virtual metot ve özellikler, Arabirimler (Interface) Çok biçimlilik (Polimorphism)
- Koleksiyonlar, Genelleyiciler (Generics)
- UML, UML Sınıf Diyagramları, UML Kullanıcı Diyagramları
- Temsilciler (delegate) ve Olay (event) kavramı
- İstisna yönetimi ve hata ayıklama

Kaynaklar

Dersin Resmi Kitabı

- ❖ C# 4.0- The Complete Reference: Herbert Schildt (İngilizce)
- C# 4.0- Herkes İçin Herbert Schildt (Türkçe)

Yardımcı Kitaplar

- Her Yönüyle C# 6.0 : Sefer Algan (Temel Düzey)
- C# ile Tasarım Desenleri ve Mimarileri: Ali Kaya, Engin Bulut (İleri Seviye)
- Projeler İle C# 5.0 ve SQL Server 2014, Süleyman Uzunköprü (İleri Seviye + veri tabanı)

Online Siteler:

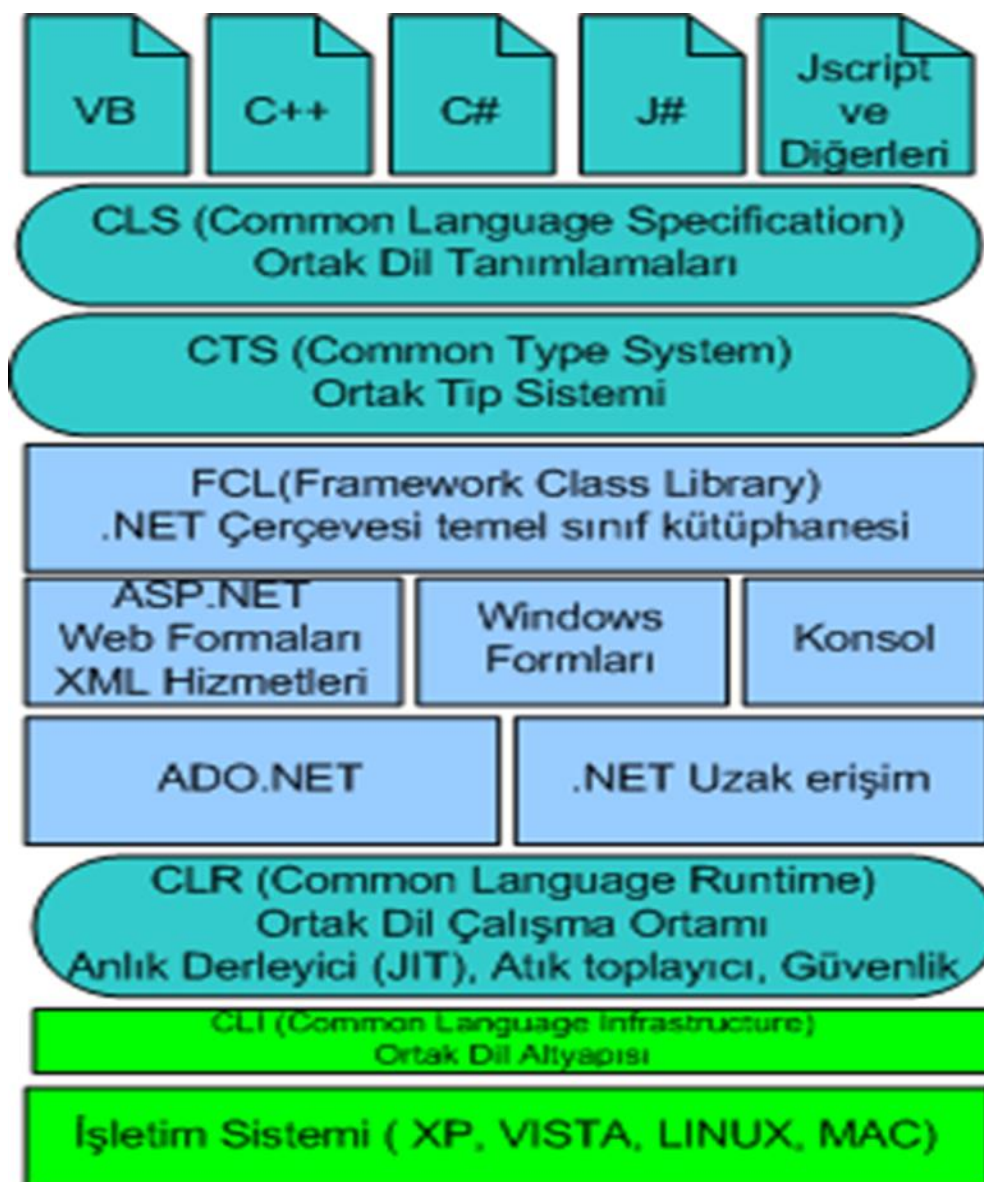
- ❖ <http://www.tutorialspoint.com/csharp/index.htm>
- ❖ <http://www.c-sharpcorner.com/>
- ❖ www.csharpnedir.com
- ❖ www.codeguru.com

.NET Mimarisı Nedir?

- ▶ .NET yeni bir yazılım geliştirme platformu
- ▶ Uygulama ve çözüm hedefleri:
Masaüstü, WEB, Mobil, Gömülü Sistemler, Endüstriyel
- ▶ Tümüleşik sistem tasarımları
- ▶ İnternet uygulamalarına ve teknolojilerine destek
- ▶ Yeni programlama dili : C#
- ▶ Yeni program geliştirme altyapısı (.NET mimari)
- ▶ Yeni programcılık mantığı (Çoklu Dil Kullanımı)
- ▶ Platformdan bağımsız uygulamalar *

*:Microsoft tabanlı sistemler : XP, Vista, Win7, Win8, Win10, WinMobile, WinCE, Linux (sınırlı)

.NET Yapısal Bileşenleri



.NET ve Alternatifler

Ölçütler	.NET Microsoft	JAVA Oracle	ColdFusion Adobe	PHP Gönüllü Grup
Derlenmiş ve Önderlenmiş Kod	Var	Var	Var	Yok
Script Diller	Var	Var	Var	Var
Nesne Yönelimli	Var	Var	Var	Var
Desteklenen Diller	C++, C#, VB, Pyton, Perl, COBOL, Delphi vb.. 25 Dil	Java	CFML ve CFScript	PHP
Internet Tarayıcı Uyumluluğu	Var	Var	Var	Var
Açık Kaynak Kodu	Kısmi	Var	Kısmi	Var

En Popüler Framework'ler (2016 için)

Top in Frameworks · Week beginning Feb 08th 2016

Name	10k	100k	Million	Entire Web
PHP	↓4,121	↓39,566	↓639,577	↓41,700,644
ASP.NET	↓2,549	↓24,356	↓297,603	↓38,210,545
Shockwave Flash Embed	↓627	↑5,442	↓150,221	↓5,378,172
Classic ASP	↓336	↓3,979	↓26,626	↓2,892,154
Adobe Dreamweaver	—403	↓4,461	↓80,850	↓2,771,343
J2EE	—1,341	↑6,955	↑72,935	↓1,982,820
DAV	↓148	↓1,603	↓20,265	↓1,031,075
ASP.NET MVC	↓696	↑3,813	↑22,603	↓905,838
ASP.NET Ajax	↓879	↓8,295	↑77,097	↓895,791
Ruby on Rails	↓641	↓3,660	↓32,892	↑860,644
Perl	↓96	↓792	↑8,734	↓835,096
Ruby on Rails Token	↑794	↓3,554	↑24,880	↓377,541
Google PageSpeed Module	↑124	↓1,060	↓8,620	—259,569
Adobe ColdFusion	↑209	↓1,830	↑14,533	↓184,074

.NET Model Gelişimi (2002-2008)

.NET Model	Program Geliştirme Ortamı	Yenilikler
1.0	Visual Studio .NET	<ul style="list-style-type: none">CLR 1.0Nesne Yönelimli Web uygulama geliştirme desteğiDLL sınıf kütüphanelerinin kullanımı
1.1	Visual Studio 2003	<ul style="list-style-type: none">ASP.NET ve ADO.NET özelliklerinin iyileştirilmesiMobil ASP.NET kontrol desteğiGüvenlik iyileştirmeleriODBC ve Veritabanları için destekInternet Protocol version 6 (IPv6) support
2.0	Visual Studio 2005	<ul style="list-style-type: none">CLR 2.0Generics ve generic collectionsPartial sınıflarNullable tiplerAnonymous metotlarASP.NET için yeni kontrol ve özellikler
3.0	Visual Studio 2005	<ul style="list-style-type: none">Windows Presentation Foundation (WPF)Windows Communications Foundation (WCF)Windows Workflow Foundation (WF) ve CardSpace
3.5	Visual Studio 2008	<ul style="list-style-type: none">AJAX desteğiLINQ desteğiDinamik Veri desteğiMulti-targeting Framework desteği

.NET Model Gelişimi (2008-2016)

.NET Model	Program Geliştirme Ortamı	Yenilikler
4.0	Visual Studio 2010	<ul style="list-style-type: none">CLR 4.0Managed Extensibility Framework (MEF)Dynamic Language Runtime (DLR)Task Parallel Library
4.5	Visual Studio 2012	<ul style="list-style-type: none">CLR 4.0 özelliklerinde iyileştirmeAsync DesteğiWindows Store uygulamalarını geliştirme desteğiWPF, WCF, WF ve ASP.NET üzerinde iyileştirmeler
4.5.1	Visual Studio 2013	<ul style="list-style-type: none">Performans ve hata ayıklamada iyileştirmeOtomatik bağlama yönlendirmeWindows Store uygulamaları için gelişmiş destek
4.6	Visual Studio 2015	<ul style="list-style-type: none">RyuJIT: 64-bit sistemler için yeni JITAçık Kaynak .Net Framework PaketleriCode sayfaları için destekOlay takibinde (event tracking) iyileştirmeler

.NET Teknolojileri

▶ Windows Formları ve Konsol

▶ WEB Teknolojileri

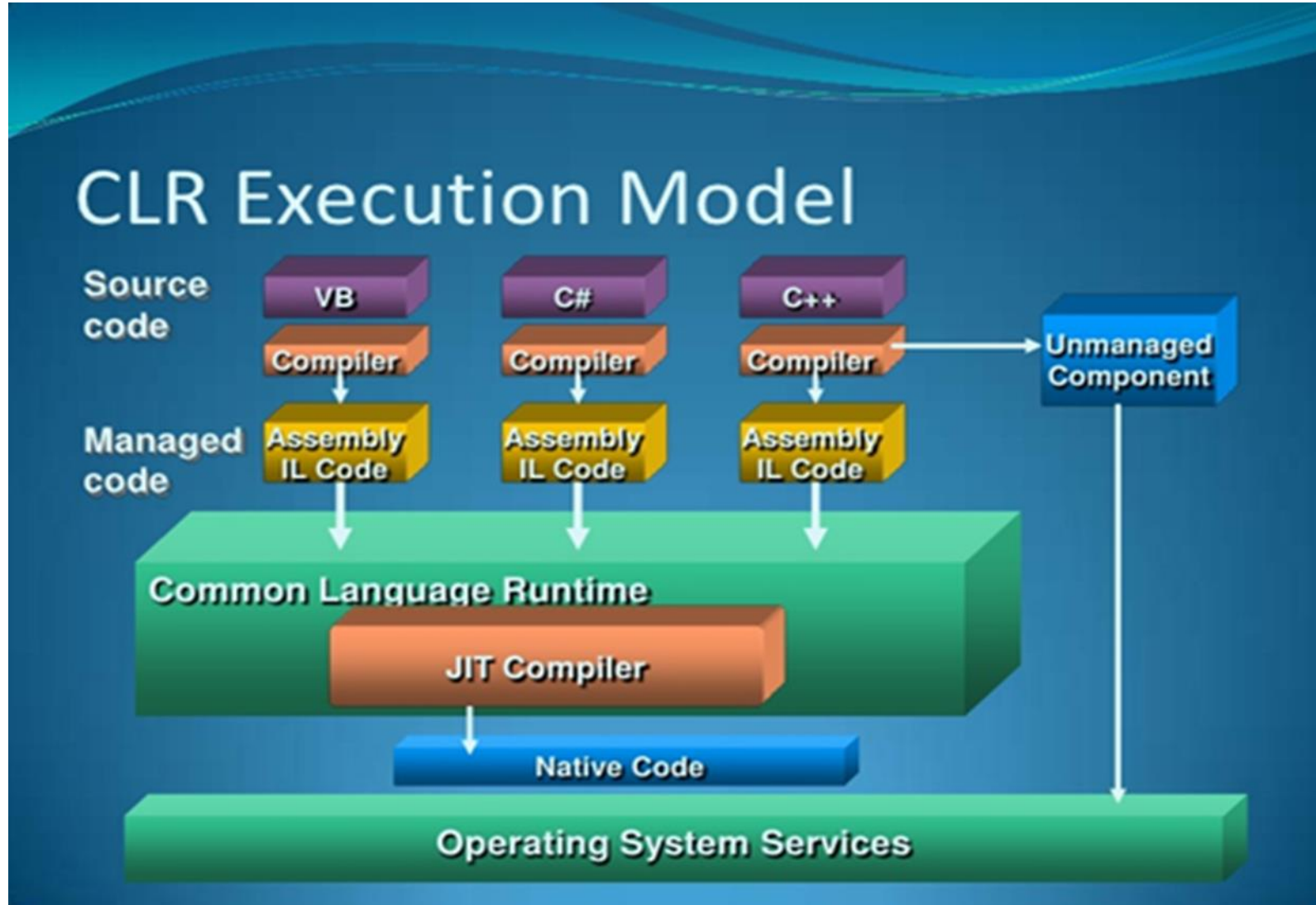
- ASP.NET
- ASP.NET MVC
- WEB Formları
- WEB Hizmetleri (XML)
 - ▶ SOAP (Simple Object Access Protocol)
 - ▶ UDDI (Universal Description, Discovery and Integration)
 - ▶ WSDL (Web Services Description Language)
 - ▶ REST (Representational State Transfer)

▶ Veritabanı Teknolojileri

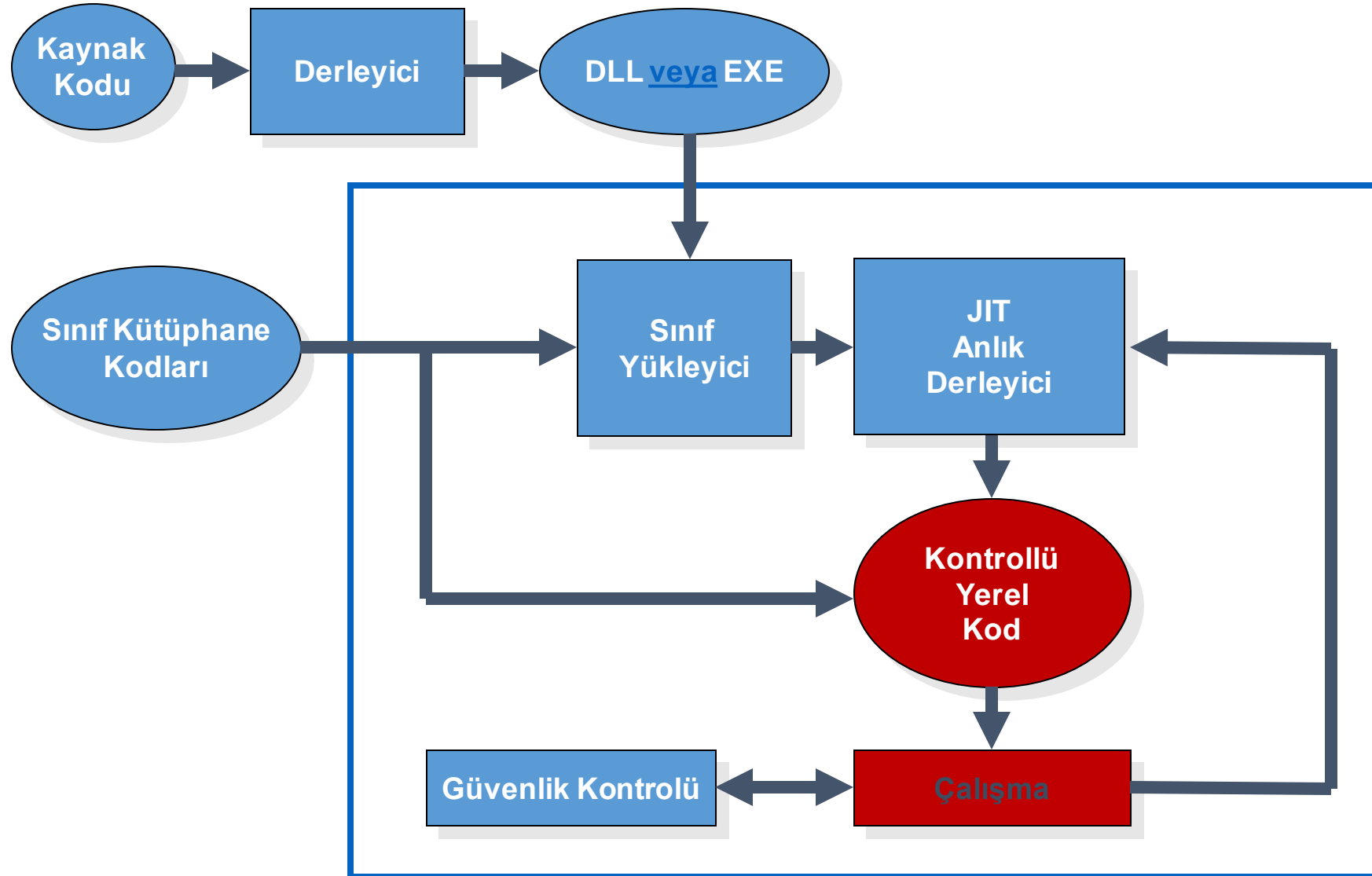
- ADO.NET

▶ WPF, Silverlight,

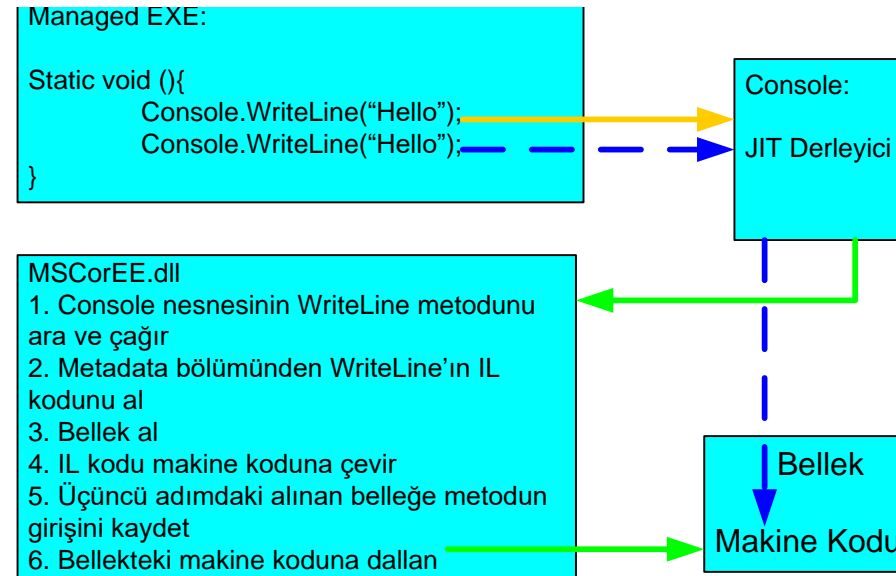
.NET Program Akışı



.NET Teknolojisi: CLR (Common Language Runtime)



Örnek C# Program Çalışma Akışı



.NET Özellikleri

- CLR hangi dil kodunu çalıştırdığını bilmez, bütün diller IL koduna çevrilir
- IL kodlar her zaman managed'tir.
- Üretilen dosya PE (Portable Executable)
- PE, CLR (.NET Framework) ile çalışır
- IL, makine dilinden daha yüksek seviyeli
- IL, nesne tabanlı makine dili olarak görülebilir
- IL kodlama yapılabilir ve ILAsm.exe tarafından derlenir
- IL kod güncel CPU'lar ile doğrudan çalıştırılmaz, gelecekte?
- NET yüklü testi: MSCorEE.dll %windir%\system32
- .NET model testi:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\policy

IL Kod Çalışması

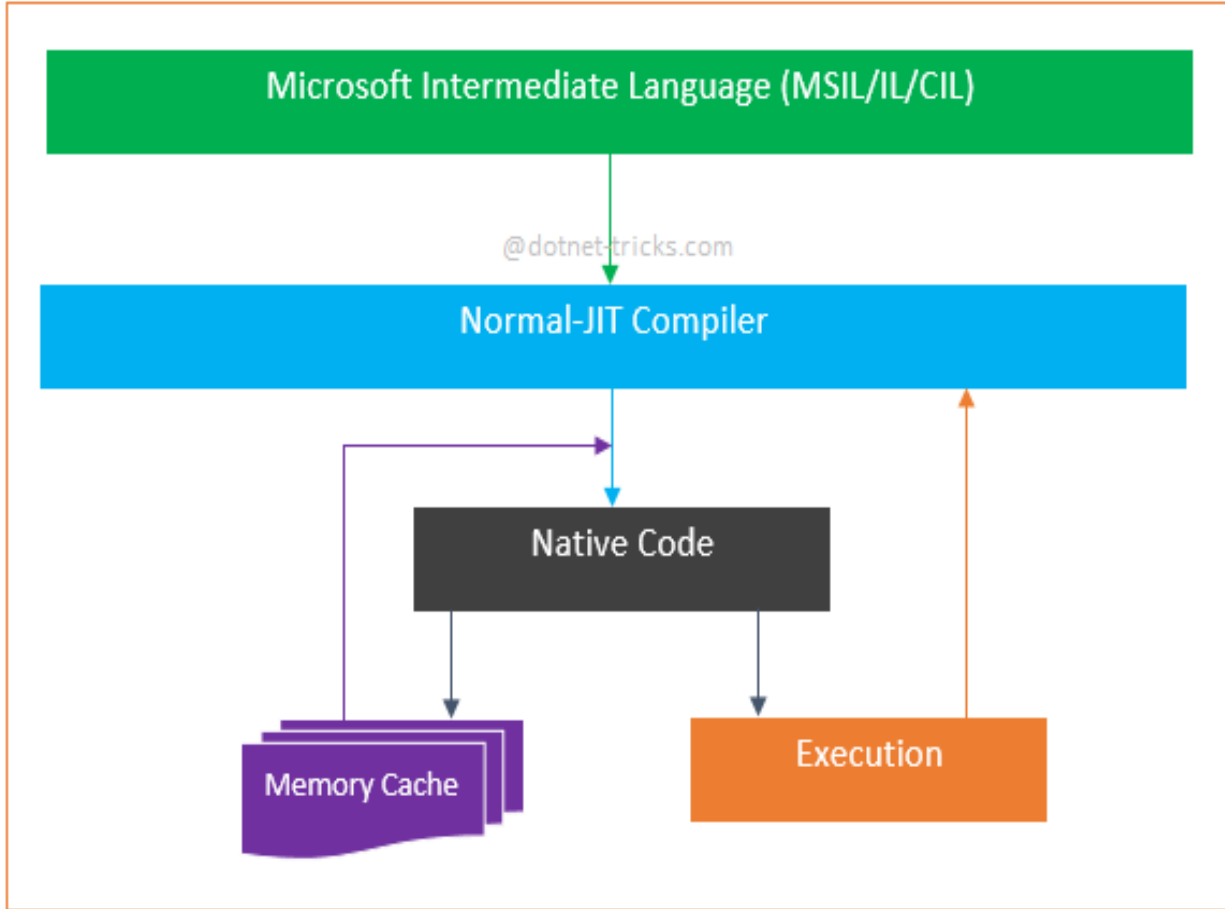
- IL kod stack (yığın) tabanlı çalışır
- İşlem verileri (operands) yığına atılır (push)
- Sonuçlar da yığından çekilir (pop)
- Derlemede kayıtçılar kullanılmaz (istisna var)
- Doğrulama (verification) IL kodun güvenli olduğunu test eder
- PEVerify.exe managed kodları doğrular
- IL kodlar veri türü ayrımı yapmaz, yığın veri boyutunu belirler

Hangisi daha Performanslı Kontrollü , Kontrolsüz?

Önyargı: Kontrolsüz kod daha hızlı çalışır. Ancak genellikle kontrollü kod hızlıdır...

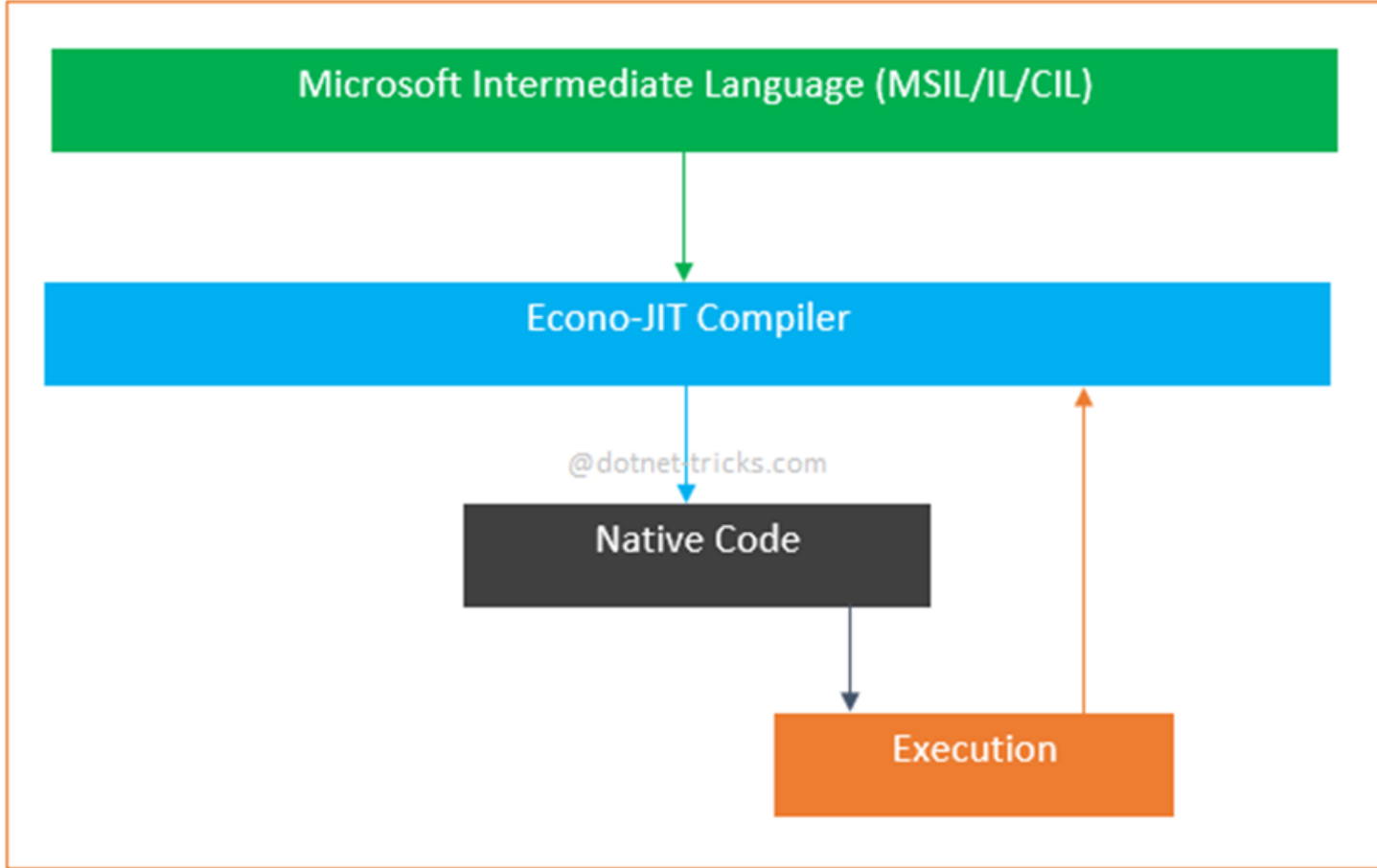
- Kontrollü kod
 - Nesne tabanlıdır
 - Tip güvenliği sağlanmıştır
 - Diller arası uyumluluğu vardır
 - Diller arası istisna yönetim uyumluluğuna sahiptir
- JIT, kontrollü kod platformunu tanır
- İşlemciye ait özel komutlar kullanılır
- Çok işlemcili sistemleri verimli kullanır
- CLR dinamik kod optimizasyonu yapar
- Windows'ta her işlem (process) için ayrı sanal bellek oluşturulur, ancak kontrollü işlemler tek bir sanal bellekte çalışır, daha az kaynak kullanır

JIT Çalışma Modları: Normal JIT (yorumlayıcı)



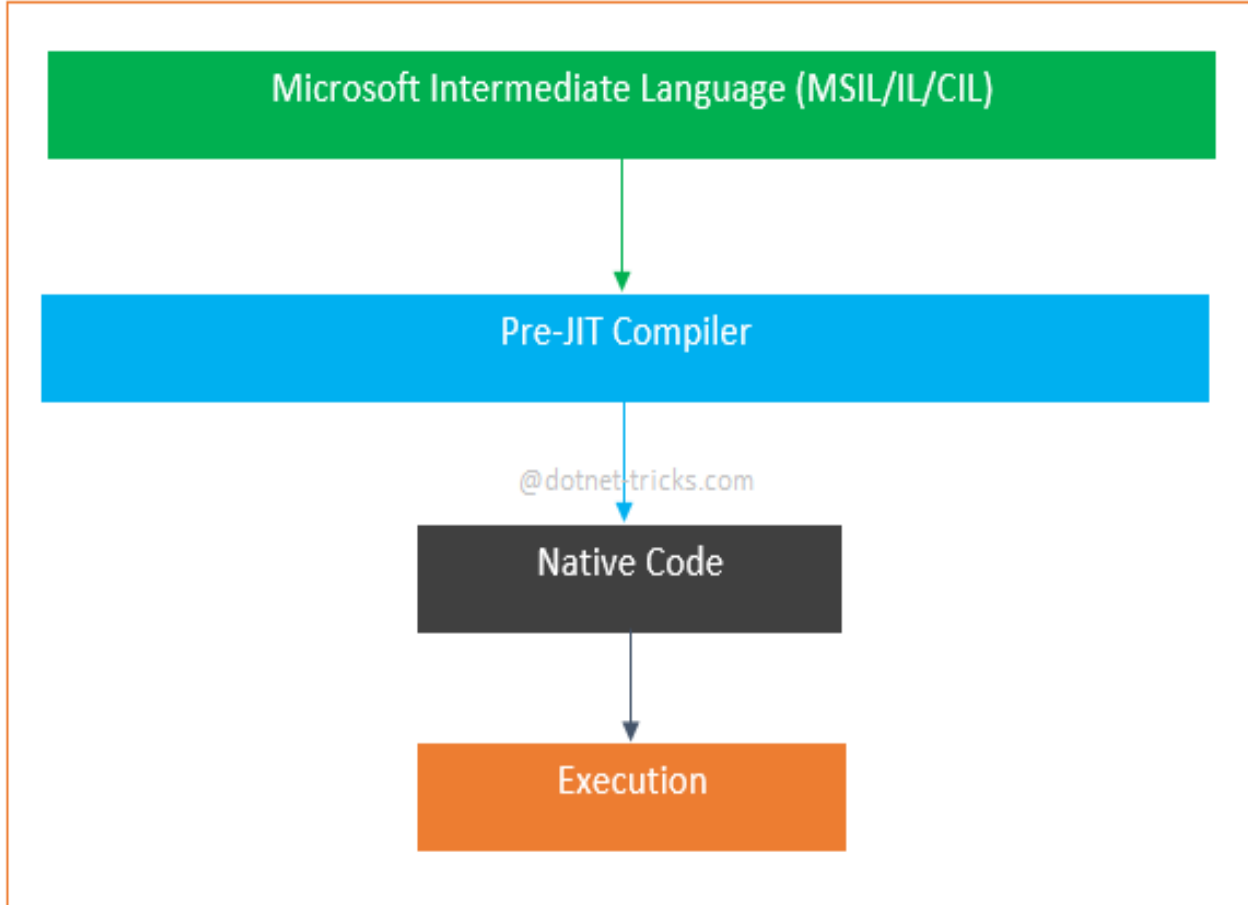
- Çalışma anında çağrılan metotlar için geçerlidir.
- Metotlar ancak ilk çağrıldıklarında derlenir ve ön belleğe yerel kod olarak kaydedilirler.
- Önbellek JITTED olarak isimlendirilir
- Aynı metot ikinci defa çağrıldığında ön bellekteki yerel kod doğrudan çalıştırılır.

JIT Çalışma Modları: Econo JIT (yorumlayıcı)



- Çalışma anında çağrılan metotlar için geçerlidir ve çağrı tamamlandıktan sonra bellekteki yerel kodlar kaldırılır.
- Econo JIT, bellek miktarı küçük olan cihazlar (cep telefonu, tablet vb.) için uygundur
- Normal JIT veya Econo JIT in hangisinin seçileceğine programcı karar VEREMEZ! Derleyici hedef platformun özelliklerine göre uygun olanı seçer!!

JIT Çalışma Modları: Pre JIT (derleyici)



- Pre JIT yüksek performans ve hızlı başlangıç gerektiren uygulamalar için kullanılır
- Tüm IL kodu yerele koda tek seferde dönüştürür, böylece yorumlayıcı tarzdan derleyici tarza geçilmiş olur
- Bu işlem projenin [ngen.exe](#) uygulaması ile komut satırından yerel koda derlenmesi ile yapılır, yerel kod diske kaydedilir
- Çalışma anında diskteki yerel kod ön belleğe alınır ve çalıştırılır
- Pre JIT kodlar genel bellekte olduğu için işlemler arasında paylaşılabılır ancak Normal ve Econ JIT modellerinde kodlar özel bellekte olduğu için paylaşamaz, bellek israfı söz konusu olur.

.NET Aradili (IL) Örneği

```
.method private hidebysig instance void button1_Click(object sender, class [mscorlib]System.EventArgs e) cil managed {  
    // Code size 60 (0x3c)  
    .maxstack 2  
    .locals init ([0] bool CS$4$0000)  
    IL_0000: nop  
    IL_0001: ldarg.0  
    IL_0002: ldfld class [System.Windows.Forms]System.Windows.Forms.TextBox SecureApplication.Form1::textBox1  
    IL_0007: callvirt instance string [System.Windows.Forms]System.Windows.Forms.Control::get_Text()  
    IL_000c: ldstr "password"  
    IL_0011: call bool [mscorlib]System.String::op_Equality(string, string)  
    IL_0016: ldc.i4.0  
    IL_0017: ceq  
    IL_0019: stloc.0  
    IL_001a: ldloc.0  
    IL_001b: brtrue.s IL_0030  
    IL_001d: ldarg.0  
    IL_001e: call instance string SecureApplication.Form1::answerToAllLife()  
    IL_0023: callvirt instance string [mscorlib]System.Object::ToString()  
    IL_0028: call valuetype [System.Windows.Forms]System.Windows.Forms.DialogResult [System.Windows.Forms]System.Windows.Forms.MessageBox::Show(string)  
    IL_002d: pop  
    IL_002e: br.s IL_003b  
    IL_0030: ldstr "Error"  
    IL_0035: call valuetype [System.Windows.Forms]System.Windows.Forms.DialogResult [System.Windows.Forms]System.Windows.Forms.MessageBox::Show(string)  
    IL_003a: pop  
    IL_003b: ret } // end of method Form1::button1_Click
```

.NET Dilleri Benzeşiyor mu?

C# ve VB.NET

//C#.NET

```
static public void Main() {  
    StreamWriter sw=new StreamWriter("date.txt ",true);  
    DateTime dt=DateTime.Now;  
    string datestring=dt.ToShortDateString()+" "+  
dt.ToShortTimeString();  
    sw.WriteLine(datestring);  
    sw.Close();  
    StreamReader sr=new StreamReader("date.txt ");  
    string filetext=sr.ReadToEnd();  
    sr.Close();  
    Console.WriteLine(filetext);  
}
```

' VB..NET

```
shared public sub Main()  
    dim sw as StreamWriter=new StreamWriter("date.txt ",true)  
    dim dt as DateTime=DateTime.Now  
    dim datestring as string=dt.ToShortDateString()+" "  
+dt.ToShortTimeString()  
    sw.WriteLine(datestring)  
    sw.Close()  
    dim sr as StreamReader=new StreamReader("date.txt ")  
    dim filetext as string=sr.ReadToEnd()  
    sr.Close()  
    Console.WriteLine(filetext)  
end sub
```

.NET'in Zayıf Yönleri

- ▶ Daha fazla sistem kaynağı kullanma (RAM, Disk, Önbellek)
- ▶ JIT Derleyici Geri-Mühendisliğe izin verir
 - Obfuscation (gizleme/örtme) araçları
- ▶ Garbage Collector'ün periyodik çalışmasının performansa etkisi
- ▶ Yer kaplama: .NET 4.5 = 4.5GB
- ▶ SSE güvenli kod desteği yok (Pentium-III)
- ▶ .NET versiyon uyumsuzlukları

.NET Sınıf Kütüphanesi

CLR ve FCL aşağıdaki uygulamaları geliştirmeyi sağlar:

- WEB hizmetleri
- WEB formları
- Windows formları
- Windows konsol
- Windows hizmetleri
- Bileşen (component) kütüphanesi

Önemli isimuzayları (namespaces)

System	Bütün uygulamaların kullandığı temel veri türleri: int, long, float, string
System.Collections	Stack, Queue, Hashtable
System.Diagnostics	Debugging
System.Drawing	Web ve Windows formlarda kullanılan 2-Boyutlu grafik nesneleri
System.IO	Klasör, dosya ve stream nesneleri
System.Management	WMI ile çoklu bilgisayar yönetimi
System.Net	Ağ haberleşmesi
System.Security	Veri ve kaynak koruması
System.Text	Kodlama karakterleri: ASCII, Unicode
System.Threading	Eşzamansız işlemciler, kaynaklara eşzamanlı erişim

CTS (Common Type System) Görevleri

- Çoklu dil bütünleşmesi, tip güvenliği ve yüksek başarımlı kod çalışması için bir çerçeve
- Çoklu diller için nesne yönelimli bir model
- Farklı dillerde yazılan nesnelerin izleyeceği kurallar Ortak Tip Sisteminin Görevleri

CTS Tipleri

- Değer : integer, float, bool, double ,
- Sınıf : class
- Temsilci: delegate
- Dizi: array
- Arayüz: interface
- İşaretçi: pointer

CLS (Ortak Dil Tanımlamaları)

• **Önlemler**

- Büyük-küçük harf ayrımı
- İşaretsiz sayılar
- Çokamaçlı işleçler
- Değişken sayıda parametreleri destekleyen metotlar

CLR/CTS/CLS İlişkisi

