



ISE 102- Nesneye Dayalı Programlama

Hafta 1 –
Ders Tanıtımı ve Giriş



Ders Hakkında

Bu dersin amacı, C# programlama dili ile programlama konusuna değinmek ve Nesneye dayalı programlama konularına C# programlama dili kullanarak giriş yapmaktır.

C#, Nesne ve Sınıf kavramları, kurucular ve yıkıcılar, kalıtım, çoklu kalıtım, UML, operatör aşırı yükleme, çok biçimlilik, fonksiyon aşırı yükleme ve fonksiyon ezmesi, Grafik Kullanıcı Arayüzü Tasarımı, Dosyalar, Kütüphane kullanımı...

Değerlendirme

- 1. Ödev %10
- 2. Ödev %10
- Proje/Tasarım %50
- Ara Sınav %30
- Yılıçi Çalışmaları %50
- Final %50



Ders Konuları

1	C#’a Genel Bakış, Kontrol deyimleri, Metotlar, Diziler	
2	Windows Formlarıyla Grafik Kullanıcı Arayüz Tasarımı	
3	WPF ile Grafik Kullanıcı Arayüz Tasarımı	
4	İstisna Yönetimi, LINQ, Jenerikler, Koleksiyonlar	
5	Nesneye Dayalı Programlama nedir?	
6	Sınıflar, Erişim Denetleyicileri, Alan ve Metotlar	
7	Kalıtım, Çoklu Kalıtım ve Arayüzler	
8	Çokbiçimlilik ve Sarmalama	
9	Özgür Sınıflar ve Arayüzler	
10	UML, UML Class Diyagramları, UML Kullanıcı Diyagramları	
11	Static metotlar, sınıflar	
12	Nesneye Dayalı Programlama Prensipleri	
13	Nesneye Dayalı Programlama Prensipleri	
14	Genel Örnekler	

Temel Kaynaklar

- Her Yönüyle C# 6.0 : Sefer Algan (Temel Düzey)
- Hands on Object-Oriented Programming with C# - Raihan Taher - 2019 - Packt
- Beginning C# Object Orineted Programming - Dan Clark-Apress
- Beginning Object-Oriented Programming with C# - Jack Purdum - Wiley
- C# 4.0- Herkes İçin Herbert Schildt (Türkçe)
- 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
- ❖ BTK Akademi Eğitim Videoları

Ders İçeriği

C# Programlama Dili

- Gelişimi
- Mimarisi
- Temelleri (Veri tipleri, koşullar, döngüler..)

Nesne Kavramına Giriş

- Sınıflar
- Nesneler
- Değişkenler
- Metotlar
- NDP karakteristikleri
 - Kalıtım
 - Encapsulation
 - Abstraction
 - Çok biçimlilik

C#'da Nesne Yönelimli Programlar Geliştirme

Görsel Programlama Giriş

Object Collaboration

İstisna Yakalama

Olaylar ve Delegeler

Generics in C#

Uygulama Geliştirme Platformu

- Derslerde MS Visual Studio 2019 kullanılacaktır.
- Aşağıdaki linkteki adımlar izlenerek kurulabilir.

<https://bidb.sakarya.edu.tr/tr/duyuru/goster/82803/microsoft-azure-kullanimi-hakkinda>

Veya aşağıdaki bağlantıdan ücretsiz sürümler indirilebilir.

<https://visualstudio.microsoft.com/tr/free-developer-offers/>

Visual Studio; Microsoft tarafından geliştirilmiş olan programlar oluşturulmasına, hata ayıklanmasına ve çalıştırılmasına yardımcı çeşitli araçlar içeren tümleşik bir geliştirme ortamıdır.

VS Türleri

Hangi sürümün hangi özellikleri sağladığını ve hangi tür iş için daha iyi olduğunu bilmek, geliştiricinin ihtiyacına uygun sürümü doğru seçmesini kolaylaştıracaktır.

<https://visualstudio.microsoft.com/tr/vs/compare/>

Visual Studio Community

VS Community sürümü ücretsiz sürümdür. Bu sürüm, diğerlerinde bulunan bazı gelişmiş özelliklere sahip değildir, ancak küçük / orta ölçekli projeler oluşturmak için kullanılabilir. Özellikle C # programlama dilini keşfetmek isteyen kişiler için kullanışlıdır.

Visual Studio Professional

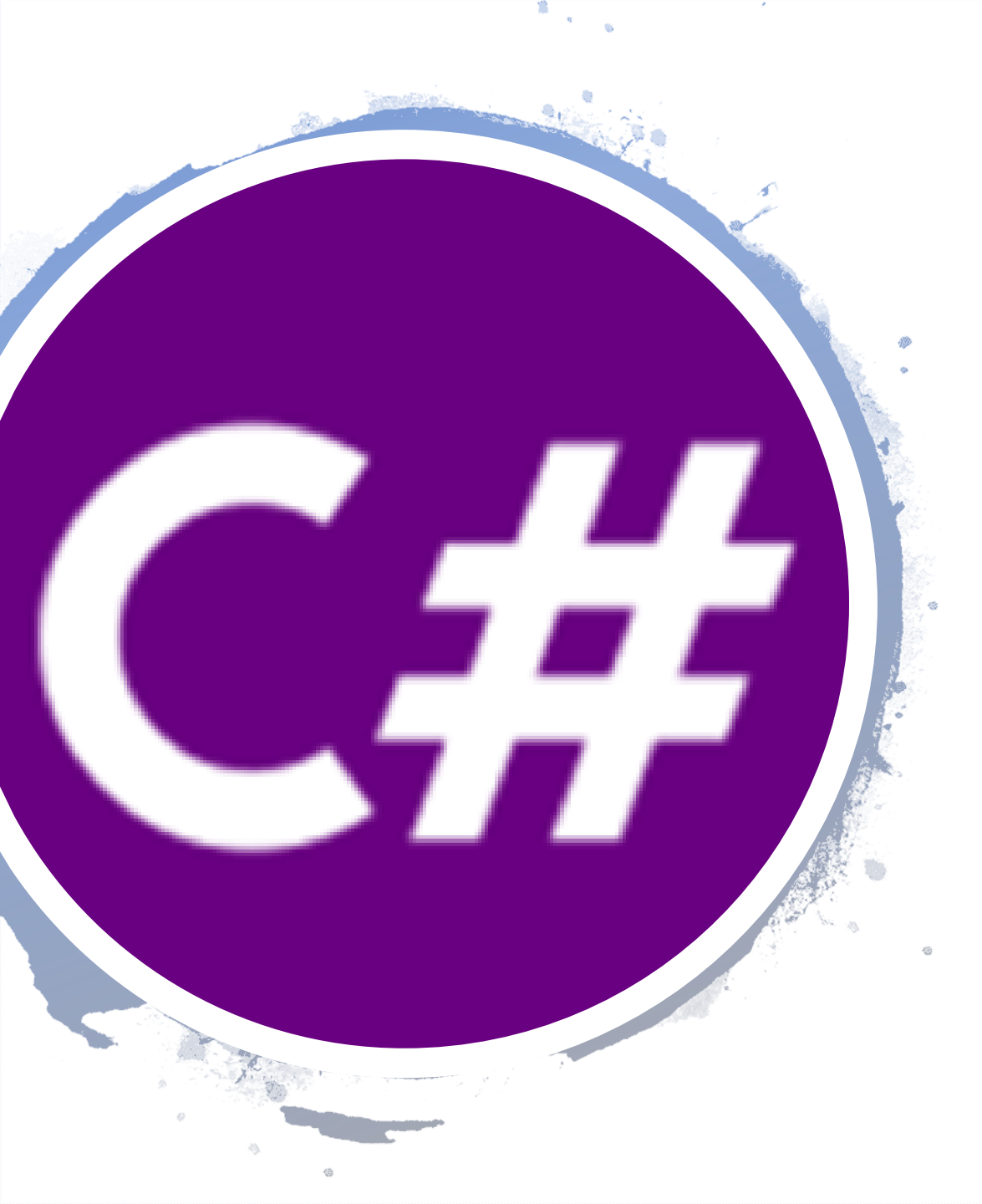
VS'nin bu sürümü, önemli hata ayıklama araçları ve yaygın olarak kullanılan tüm geliştirici araçlarıyla kendinizi geliştirmeniz içindir.

Visual Studio Enterprise

VS Enterprise sürümü, IDE'nin ticari kullanım düzeylerine ihtiyaç duyan işletmeler içindir. Test etme, hata ayıklama vb. için özel araçları destekler. Ayrıca yaygın kodlama hatalarını keşfeder, test verileri üretir ve çok daha fazlasını yapar.

Visual Studio Code

VS Code, tam bir IDE olmayan, ancak Microsoft tarafından geliştirilmiş basit bir kod düzenleyicisi olan küçük, açık kaynaklı bir araçtır. Bu düzenleyici çok hafif ve platformdan bağımsızdır. VS Code, VS IDE'nin sahip olduğu özelliklerin çoğuyla birlikte gelmez, ancak bir uygulamayı geliştirmek ve hata ayıklamak için yeterli özelliklere sahiptir.



C#

- C/C++ ve Java dillerinden türeyen, güçlü, basit, esnek, tip güvenli, modern ve .Net platformu için geliştirilmiş programlama dilidir.
- Hem alt düzey hem üst düzey programlar yazılabilir.
- C# söz dizimi, C/C++ ve Java dillerinden gelmektedir.
- C uzun yıllar yazılım geliştirme alanındaki en popüler dillerden olarak kullanıldı. Daha sonra nesne yönelimli diller karşısında zayıflayınca C++ geliştirildi. C diline nesne yönelimli özelliklerin katıldığı ve günümüzde de hala tercih edilen programlama dillerinden biridir.
- Sun Microsystems'in dağıtık uygulama modeline daha fazla destek sunan, hem web üzerinde hem de Java Runtime kurulu herhangi bir bilgisayarda çalışabilen Java'yı tanıtmayla Microsoft'da .Net ile tamamen uyumlu olan ve en gözde dillerden biri haline gelen C#'ı tanıttı.

Neden C#

- Öğrenilmesi Kolay
- Nesne Yönelimli Programlamaya Tam Destek:
Büyük ölçekli projeler hızlı bir şekilde gerçekleştirilebilir.
Hataların giderilmesi ve güncelleme kolaylığı
- Yüksek Verim:
Örneğin birçok hatanın giderilmesinde kolaylık sağlayan `if()` ifadelerinin değerin `bool` türünde olma gerekliliği
- Güç ve Kolaylık Dengesi
- XML desteği, Dağıtık uygulama geliştirme desteği
- Yönetilmeyen Kod(Unmanaged):
C# ile programlar geliştirilirken işaretçiler kullanılabilir. Geriye dönük uyumluluk
- Modern
- Güvenli:
C #, tür güvenli bir dildir. C # 'daki tüm değişkenler ve sınıflar bir türdür ve tüm türler `object` türünden türetilir ve tür güvenlidir.
- Garbage Collector yardımıyla otomatik hafıza yönetimi,
programın hafızayı sızdırmamasını sağlar ve nesnelerin yaşam döngülerini otomatik olarak yönetir ve nesneler gerekmediğinde serbest bırakır.

Neden C#

- Çok Yönlü:

Yalnızca Windows istemci uygulamaları oluşturmakla kalmaz, aynı zamanda Web ve mobil uygulamalar oluşturmak için C # kullanabilirsiniz.

- Windows Forms, WPF ve UWP kullanan Windows istemci uygulamaları.

Winforms (Windows Forms) veya Windows formları, Windows tabanlı işletim sistemlerinde kullanılan ilk GUI (Graphical User Interface) arayüzüdür.

WPF (Windows Presentation Foundation), 2006 yılında .NET Framework 3.0 ile birlikte gelen yeni GUI geliştirme arayüzüdür. Tasarım ve kodlamayı birbirinden ayırmak ve daha esnek bir görsel arayüz sunmak için XML tabanlı XAML yapısını kullanır.

UWP (Universal Windows Platform) veya Evrensel Windows Platformu, WPF sadece Windows tabanlı işletim sistemleri için kullanılan bir arayüz iken UWP işletim sistemi bağımsız olarak GUI oluşturmayı amaçlamaktadır. UWP ile geliştirilen bir arayüz daha sonra web tabanlı bir arayüze veya cep telefonu arayüzünde de kullanılması amaçlanmaktadır.

- ASP.NET ve ASP.NET Core ile web uygulamaları.
- Kitaplıklar, bileşenler ve konsol uygulamaları
- Bulut ve Azure uygulamaları
- Blockchain uygulamaları
- Birlikte çalışabilirlik, veri ve entegrasyon hizmetleri

Neden C#

- Sürekli Gelişme:

- C # dili, dünyanın en hızlı gelişen programlama dillerinden biridir. Derlemedeki mevcut C # sürümü C # 10'dur. (Kasım 2021)
- <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history>
- <https://devblogs.microsoft.com/dotnet/welcome-to-csharp-10/>

- Mobile uygulamaları destekler

- Mobil geliştiriciler, C # ve Xamarin kullanarak yerel iOS, Android ve UWP uygulamaları oluşturabilir. Xamarin ayrıca Mac ve Linux'u destekler. C# Corner, Xamarin ile ilgili 900'e yakın eğitim makalesi ve öğreticiye sahiptir.
- <https://www.c-sharpcorner.com/technologies/xamarin>

Neden C#

- Makine Öğrenmesi Uygulamaları:

ML.NET, makine öğrenimini .NET geliştiricileri için erişilebilir hale getiren, platformlararası açık kaynaklı bir makine öğrenimi çerçevesidir.

ML.NET, .NET geliştiricilerinin kendi modellerini geliştirmelerine ve makine öğrenimi modellerini geliştirme veya ayarlama konusunda .NET kullanarak uygulamalarına özel makine öğrenimini aşılamlarına olanak tanır.

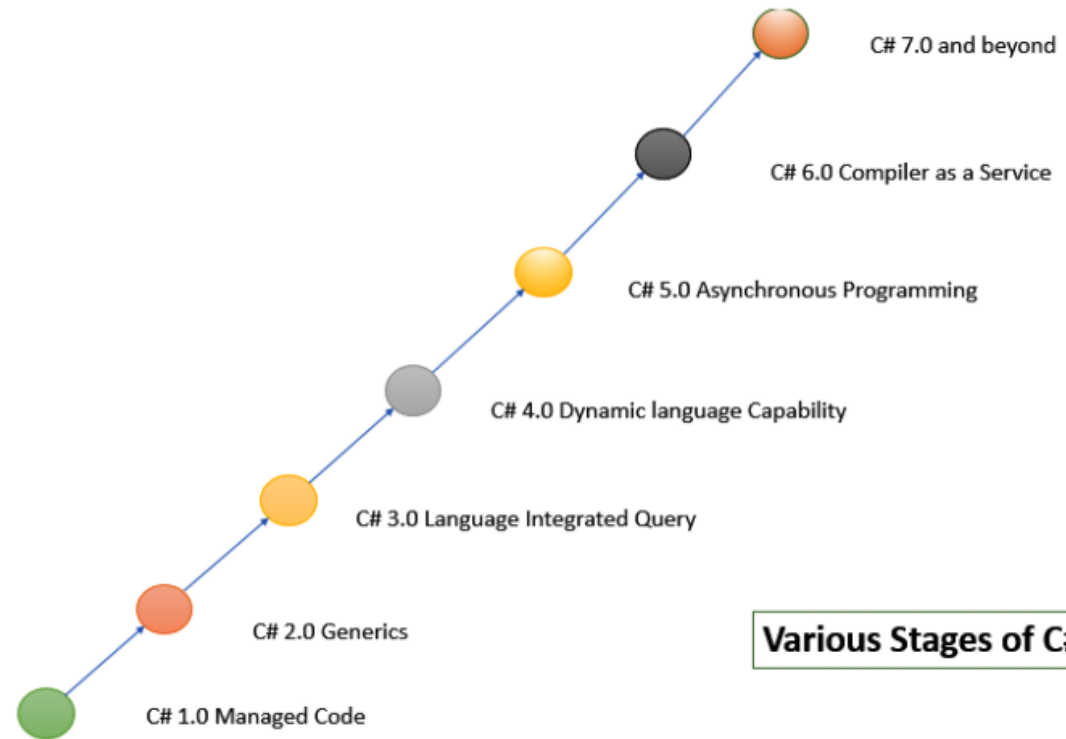
ML.NET Github'da mevcuttur:

<https://github.com/dotnet/machinelearning>

TensorFlow, makine için en gelişmiş çerçevelerden biridir TensorFlowSharp, bir API'nin C #, F# ve .NET kullanarak TensorFlow kitaplığıyla çalışmasını sağlayan açık kaynaklı bir kitaplıktır.

<https://github.com/migueldeicaza/TensorFlowSharp>

C# gelişimi



Various Stages of C# language

TIOBE INDEX
















TIOBE Index for February 2022

February Headline: TIOBE index top 3 benefits from technology changes

As of the 1st of May, the Alexa web traffic ranking engine is going to stop its services. Alexa was used to select the search engines for the TIOBE index until now. So now something has to change. Similarweb has been chosen as the alternative for Alexa. We have used Similarweb for the first time this month to select search engines and fortunately, there are no big changes in the index due to this swap. The only striking difference is that the top 3 languages, Python, C, and Java, all gained more than 1 percent in the rankings. We are still fine-tuning the integration with Similarweb, which is combined with a shift to HtmlUnit in the back-end. Some websites are not onboarded yet, but will follow soon. Now that HtmlUnit is applied for web crawling, it will become possible to add more sites to the index, such as Stackoverflow and Github. This will hopefully happen in the next few months. -- Paul Jansen CEO TIOBE Software

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Feb 2022	Feb 2021	Change	Programming Language		Ratings	Change
1	3	▲		Python	15.33%	+4.47%
2	1	▼		C	14.08%	-2.26%
3	2	▼		Java	12.13%	+0.84%
4	4			C++	8.01%	+1.13%
5	5			C#	5.37%	+0.93%
6	6			Visual Basic	5.23%	+0.90%
7	7			JavaScript	1.83%	-0.45%
8	8			PHP	1.79%	+0.04%
9	10	▲		Assembly language	1.60%	-0.06%
10	9	▼		SQL	1.55%	-0.18%
11	13	▲		Go	1.23%	-0.05%
12	15	▲		Swift	1.18%	+0.04%
13	11	▼		R	1.11%	-0.45%
14	16	▲		MATLAB	1.03%	-0.03%
15	17	▲		Delphi/Object Pascal	0.90%	-0.12%



PYPL Popularity of Programming Language Index

Worldwide, Feb 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.52 %	-1.7 %
2		Java	18.12 %	+1.2 %
3		JavaScript	8.9 %	+0.4 %
4	▲	C/C++	7.62 %	+1.1 %
5	▼	C#	7.39 %	+0.6 %
6		PHP	5.81 %	-0.3 %
7		R	4.04 %	+0.2 %
8		Objective-C	2.46 %	-1.1 %
9		Swift	2.03 %	+0.0 %
10		TypeScript	1.94 %	+0.2 %
11		Matlab	1.76 %	+0.0 %
12		Kotlin	1.58 %	-0.1 %
13		Go	1.33 %	+0.1 %
14	▲	Ruby	1.12 %	-0.1 %
15	▼	VBA	1.09 %	-0.1 %
16		Rust	0.99 %	-0.1 %
17	▲	Ada	0.75 %	+0.1 %
18	▲▲	Dart	0.71 %	+0.1 %
19	▲▲▲	Abap	0.64 %	+0.2 %
20	▼▼▼	Visual Basic	0.63 %	-0.1 %
21	▼▼	Scala	0.52 %	-0.1 %
22	▲▲	Groovy	0.44 %	+0.0 %
23	▼▼	Lua	0.39 %	-0.2 %
24	▼	Perl	0.38 %	-0.1 %

.NET Mimarisi Nedir?

- ▶ Microsoft tarafından geliştirilen, açık İnternet protokolleri ve standartları üzerine kurulmuş bir “uygulama” geliştirme platformudur. Sağladığı çoklu dil desteğiyle, tek dile bağlı kalmadan değişik tipte uygulamalar geliştirmemize olanak sağlar.
- ▶ <https://dotnet.microsoft.com/>
- ▶ .NET yeni bir yazılım geliştirme platformu
- ▶ Uygulama ve çözüm hedefleri:
Masaüstü, WEB, Mobil, Gömülü Sistemler, Endüstriyel
- ▶ Tümlleşik 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 *

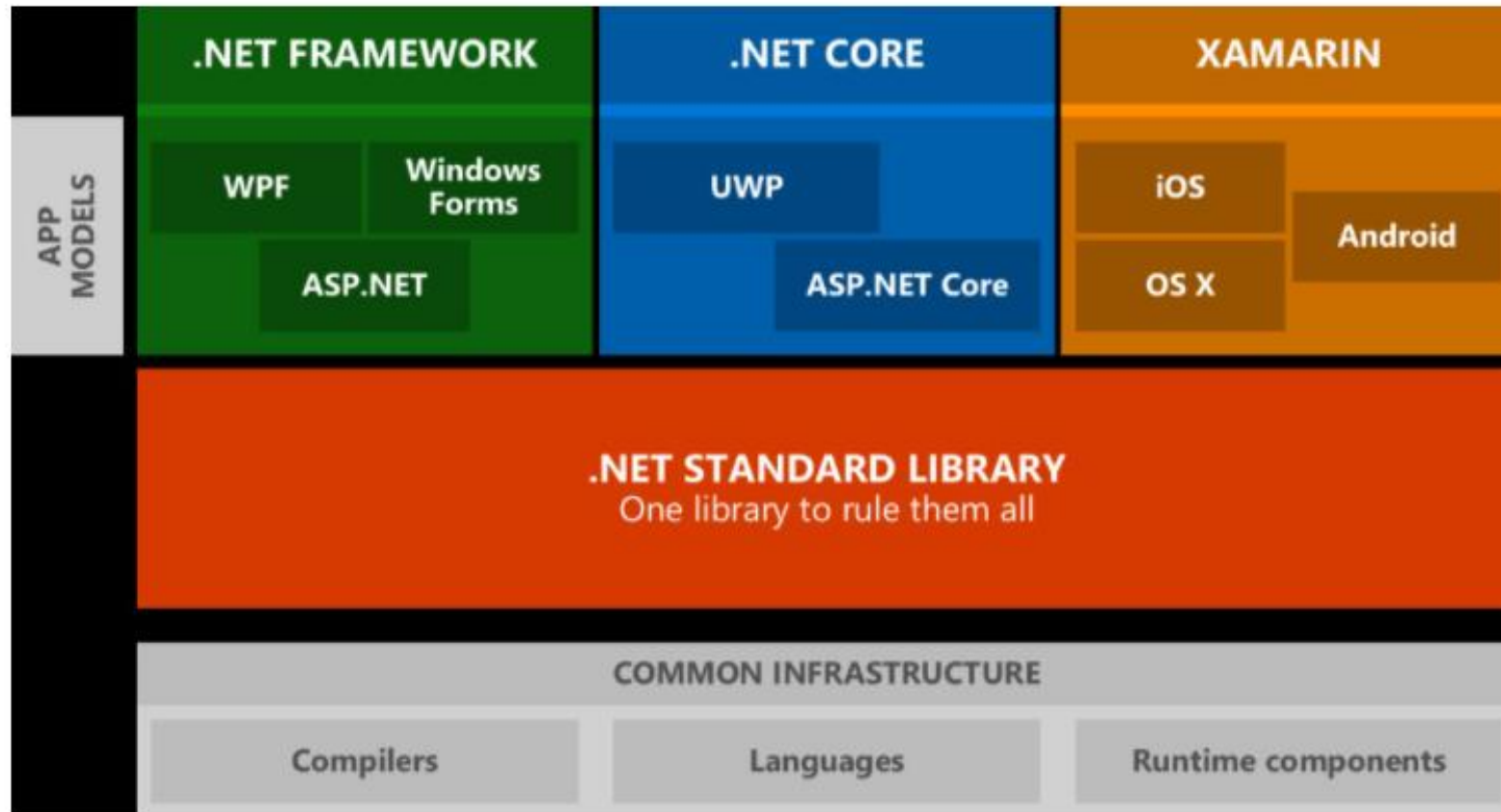
*:Microsoft tabanlı sistemler : XP, Vista, Win7, Win8, Win10, WinMobile, WinCE, Linux (sınırlı)

.Net Mimarisi

.NET – A unified platform



- <https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/intro>



Bazı Notlar

- Java'dan önce, geliştirilen yazılımlar direkt olarak makine koduna derlenirdi.
- Bu programlar her işletim sistemine özel olarak geliştirilir ve işletim sisteminin kuralları çerçevesinde derlenirdi.
- Taşınabilirlik yok derecede az.
- Java ile program kodu önce **byte code** çevrilir.
- Bu kodu Java Sanal Makinası (Java Virtual Machine) işletim sisteminin istediği koda çevirir. Yalnız Java ile yazılan programlar direkt makine koduna çevrilen yazılımlara nazaran yavaş çalışır.
- .Net çalışma mantığı da Java'ninkine benzer.
- .Net kodu önce IL'ye derler. (Ara dil) ve bu IL kodu çalıştırılmak istendiği zaman .Net CLR (Common Labguage Runtime-Ortak Dil Çalışma Platformu), JIT derleyicilerini kullanarak makine diline çevirir. Fakat CLR makine diline çevrilmiş program kodunu önbellekte tutar. Bu büyük performans artışı sağlarken sistem hafızasında büyük miktarda yer işgal eder.

Bazı Notlar

- Temel dil tanımları (CLS-Common Language Specifications) ve Ortak Tip Sistemi (CTS-Common Type System) ile .Net uyumlu dillerin hepsi aynı değişkenleri ve benzer nesne yönelimli özellikleri taşırlar.
- Örneğin C# ile yazılan DLL lerde yer alan temel veri tipleri Vb.net tekiler ile aynı özelliklere sahiptirler. Bu sayede farklı dillerde yazılan bileşenler birbirleri ile sorunsuz çalışır.
- Dil ile birlikte gelen kütüphaneler o dilin programlama aracı olarak seçilmesinde büyük rol oynar. Önceden yazılan bir kodu tekrar yazmanın bir anlamı yoktur. NYP bir amacı da kodun tekrar kullanılabilirliğini sağlamaktır. (code-reuse)
- .Net ile gelen sınıf kütüphaneleri kodların verimli şekilde çalışması için optimize edilmiştir.

2007 .NET 3.5

Visual Studio 2008
CLR 2.0
Built-In AJAX Support
LINQ
Dynamic Data
Multi-targeting Framework Support

2005 .NET 2.0

Visual Studio 2005
CLR 2.0
Generics and generic collections
Partial classes
Nullable types
Anonymous methods
Introduced many new controls and features to ASP.NET

2002 .NET 1.0

Visual Studio .NET
CLR 1.0
Support for Object-oriented Web application development
Use of DLL class libraries

.NET 4.0 2010

Visual Studio 2010
CLR 4.0
Managed Extensibility Framework (MEF)
Task Parallel Library

.NET 3.0 2006

Visual Studio 2005
CLR 2.0
Windows Presentation Foundation (WPF)
Windows Communications Foundation (WCF)
Windows Workflow Foundation (WWF), and
CardSpace

.NET 1.1 2003

Visual Studio .NET 2003
CLR 1.1
Features Enhancement to ASP.NET and
ADO.NET
Built-in support for mobile ASP.NET controls
Security Enhancement
Built-in support for ODBC and databases
Internet Protocol version 6 (IPv6) support

2017 .NET 4.7

Visual Studio 2017
CLR 4.0
Includes improvements
High DPI support for Win. app. on Windows 10
Touch support for WPF app. on Windows 10
Enhanced cryptography support
Performance and reliability improvements

2015 .NET 4.6.1

Visual Studio 2015 Update 1
CLR 4.0
Includes new features in the following
Cryptography
ADO.NET
WPF, and WWF
Profiling
NGen

2014 .NET 4.5.2

CLR 4.0
Includes performance and debugging improvements
Support for automatic binding redirection
Expanded support for Windows Store apps

2012 .NET 4.5

Visual Studio 2012
CLR 4.0
Async Support
LINQ
Support for building Windows Store apps
Features Enhancement to WPF, WCF, WF, and
ASP.NET

.NET 4.6.2 2016

CLR 4.0
Improvements in the following areas
Base Class Library
CLR, ASP.NET, SQL, WPF, and WCF
ClickOnce

.NET 4.6 2015

Visual Studio 2015
CLR 4.0
Includes new features in the following
Changes in the Base Class Library
Resizing in Windows Forms controls
Intro. Open Source .Net Framework Packages
Support for Code page encodings
Improvements to event tracing
.NET Native

.NET 4.5.1 2013

Visual Studio 2013
CLR 4.0
Includes performance and debugging improvements
Support for automatic binding redirection
Expanded support for Windows Store apps

.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

<https://insights.stackoverflow.com/survey/2020>

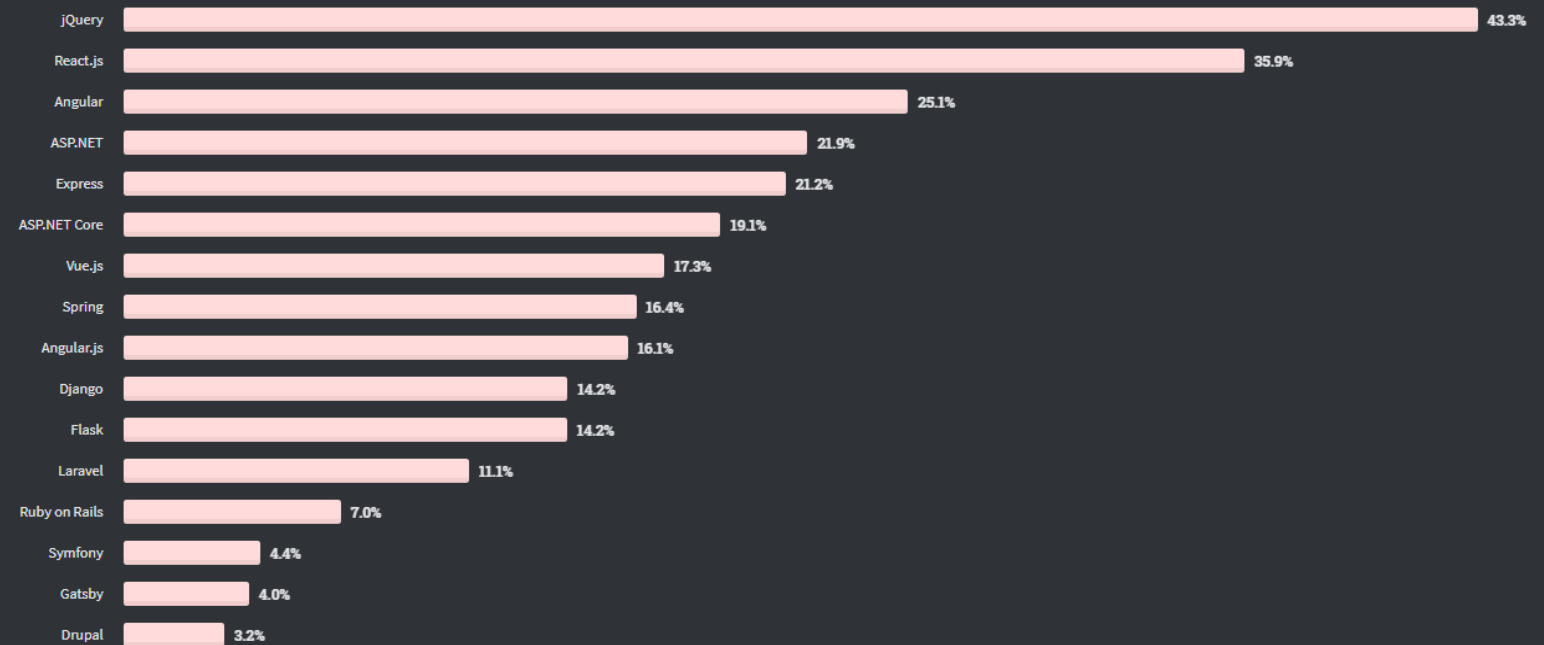
Web Frameworks

When focusing purely on web frameworks, we see that jQuery is still king, but is slowly losing ground to React.js and Angular year over year. We do see some consolidation, as more than 35% of respondents use jQuery, React, a version of Angular (combining Angular, which represents Angular 2+, and Angular.js) or a flavor of ASP.NET (ASP.NET or ASP.NET Core).

All Respondents

Professional Developers

42,279 responses; select all that apply



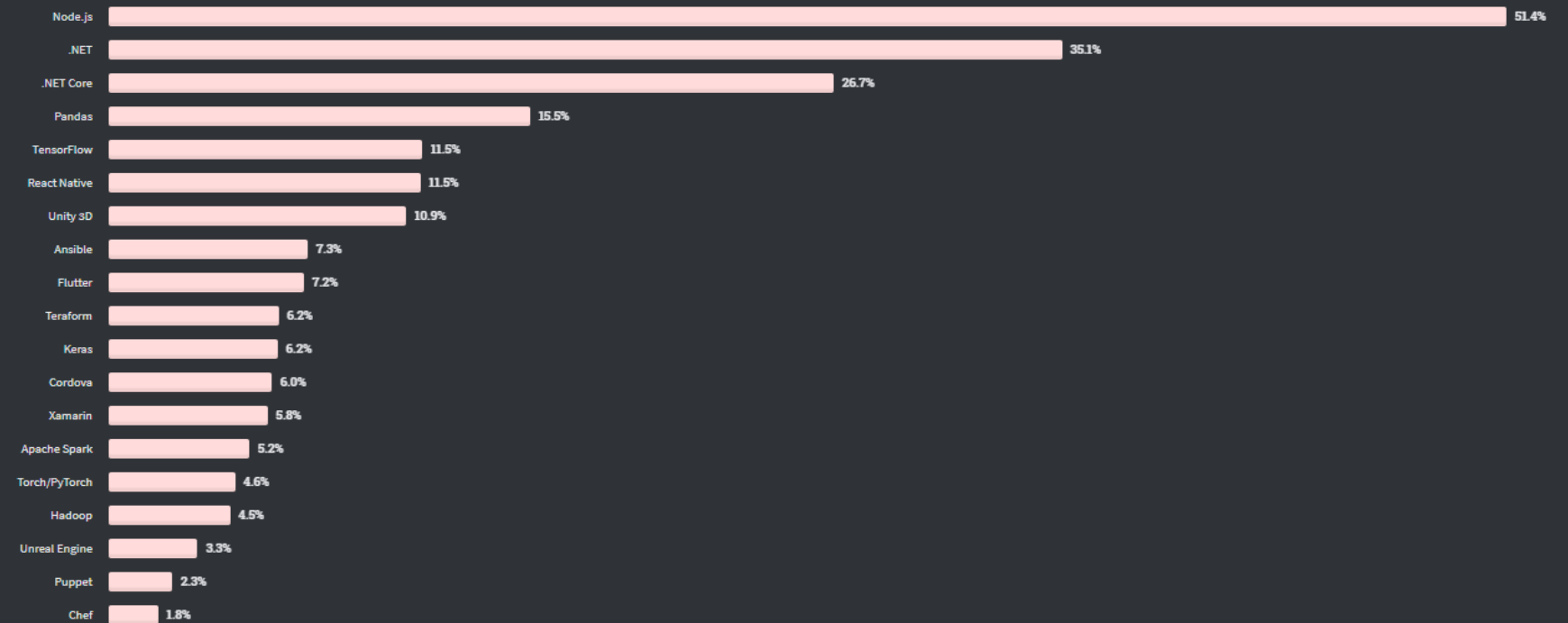
<https://insights.stackoverflow.com/survey/2020>

Other Frameworks, Libraries, and Tools

Similar to last year, we asked about many of the other miscellaneous technologies that developers are using. For the second year in a row, Node.js takes the top spot, as it is used by half of the respondents. We also see growth across the board in the popularity of data analysis and machine learning technologies such as Pandas, TensorFlow, and Torch/PyTorch.

All Respondents Professional Developers

40,314 responses; select all that apply

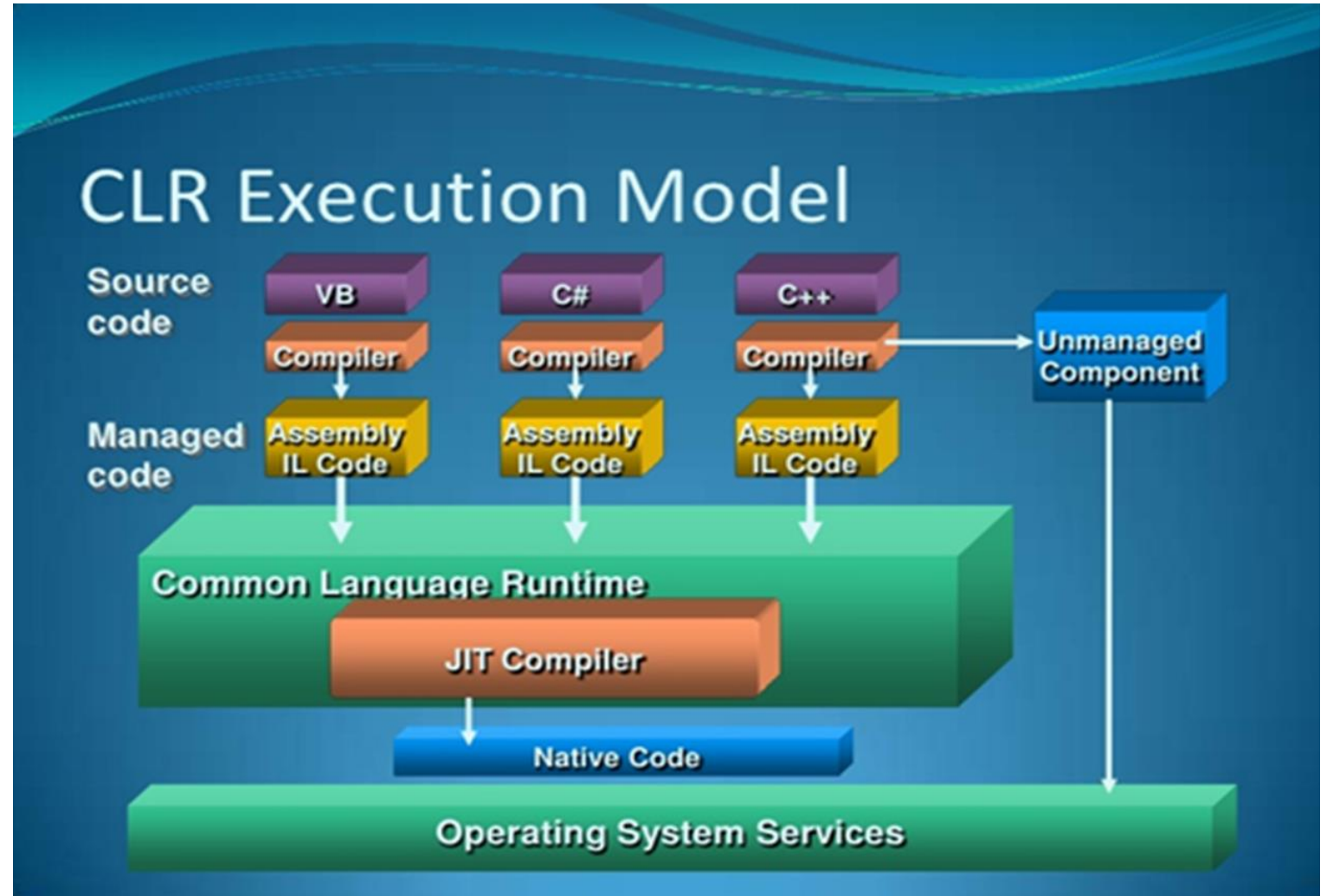


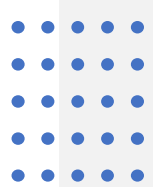
Ortak Dil Çalışma Platformu (CLR)



- .Net platformunda programların çalışmasını kontrol eden ve işletim sistemi ile program arasında yer alan birimdir.
- Normalde yazdığımız programlar derlenirken makine diline çevrilirdi ve program işletim sistemi ile direkt bağlantı kurarak çalışırdı.
- Platformdan bağımsız bir geliştirme ve yürütme ortamı isteniyorsa
 - CLR devreye girer ve .Net programlarının farklı platformlarda makineye ve işletim sistemine göre programı çalıştırır.
- C# ile yazdığımız program kodu içine ilgili diğer DLL, COM bileşenlerini ve .Net sınıf kütüphanesinde bulunan ve program kodun içinde kullandığımız kütüphanelerle birlikte derlenip , IL(Intermediate Language- Ara Dil)' e çevrilir. Burada oluşturulan assembly herhangi bir sistemde (Linux, Mac OS) CLR tarafından çalıştırılabilir.
- IL kodu CLR tarafından çağrılınca JIT derleyicileri tarafından makine diline çevrilip çalıştırılır.

.NET Program Akışı





IL (Intermediate Language)

- .Net ile kodumuzu derleyip elde ettiğimiz IL kodu işlemciye bağlı bir kod değildir. IL taşınabilir ve daha geneldir.
- IL içinde değişken tanımlamaları, değişkenlerin nasıl saklanacağı, metotların nasıl çalıştırılacağı, aritmetik işlemler, mantıksal işlemler, bellek kullanımı ve hata yakalama yöntemleri dahil birçok işin nasıl yapılacağı açıklanır.
- IL'de metadata;
 - Kullanılan verilerin yanında oluşturduğumuz sınıfların metotlarını ve bunların özellikleri ile diğer bilgileri içerir. Veri hakkında veri.
 - Metadatanın içeriği çalışma zamanında JIT derleyicileri tarafından kullanılır.

Artık IL ve Metadatasını oluşturduğumuz kodlar çalıştırılabilir bir program olmuştur.



IL Kod Çalışması

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

Örnek IL

Example program: C#

```
using System;

class Program
{
    static void Main()
    {
        int i = 0;
        while (i < 10)
        {
            Console.WriteLine(i);
            i++;
        }
    }
}
```

Intermediate language for Main method: IL

```
.method private hidebysig static void Main() cil managed
{
    .entrypoint
    .maxstack 2
    .locals init (
        [0] int32 num)
    L_0000: ldc.i4.0
    L_0001: stloc.0
    L_0002: br.s L_000e
    L_0004: ldloc.0
    L_0005: call void
[mscorlib]System.Console::WriteLine(int32)
    L_000a: ldloc.0
    L_000b: ldc.i4.1
    L_000c: add
    L_000d: stloc.0
    L_000e: ldloc.0
    L_000f: ldc.i4.s 10
    L_0011: blt.s L_0004
    L_0013: ret
}
```

```
.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    // Code size      13 (0xd)
    .maxstack 8
    IL_0000: nop
    IL_0001: ldstr      "Hello World!"
    IL_0006: call       void [System.Console]System.Console::WriteLine(string)
    IL_000b: nop
    IL_000c: ret
} // end of method Program::Main
```

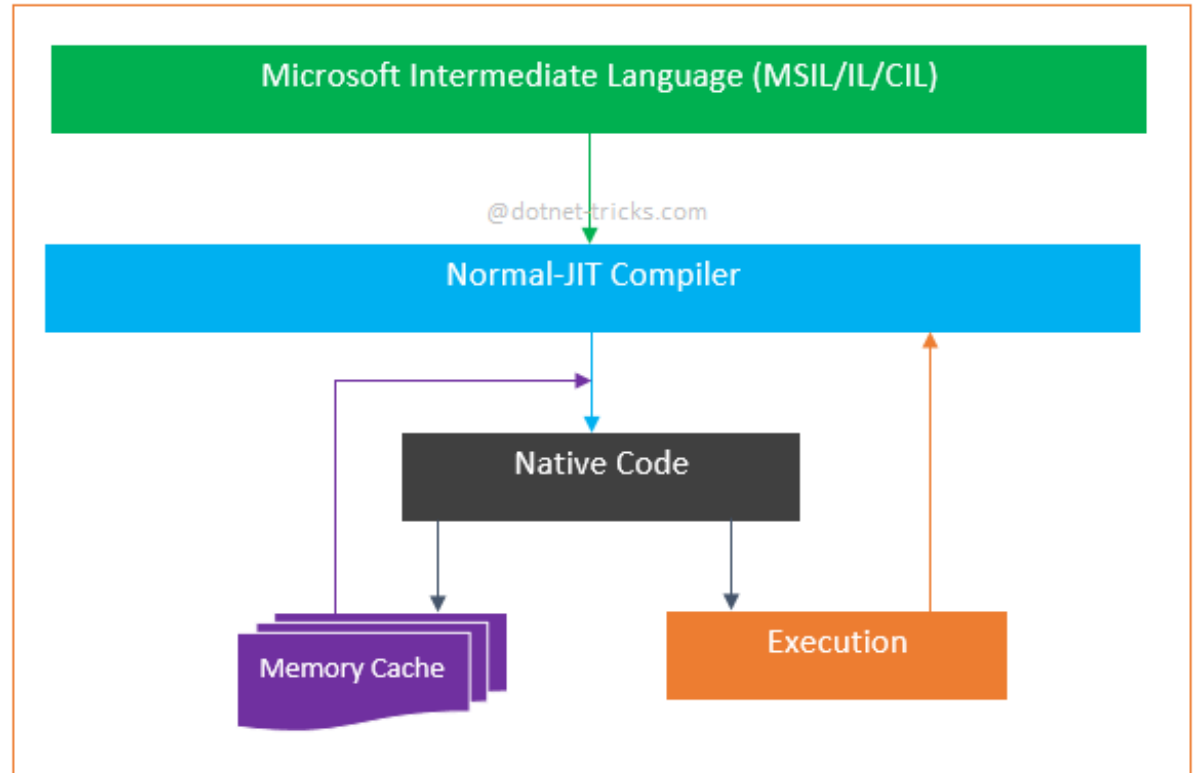


JIT derleyicileri

- C# derleyicisi ile IL'e derlediğimiz program çalışırken JIT (Just in Time) derleyicileri devreye girer. Bu derleyiciler programın çalıştırıldığı sistemin ve işlemcinin anlayabileceği makine kodunu oluştururlar.

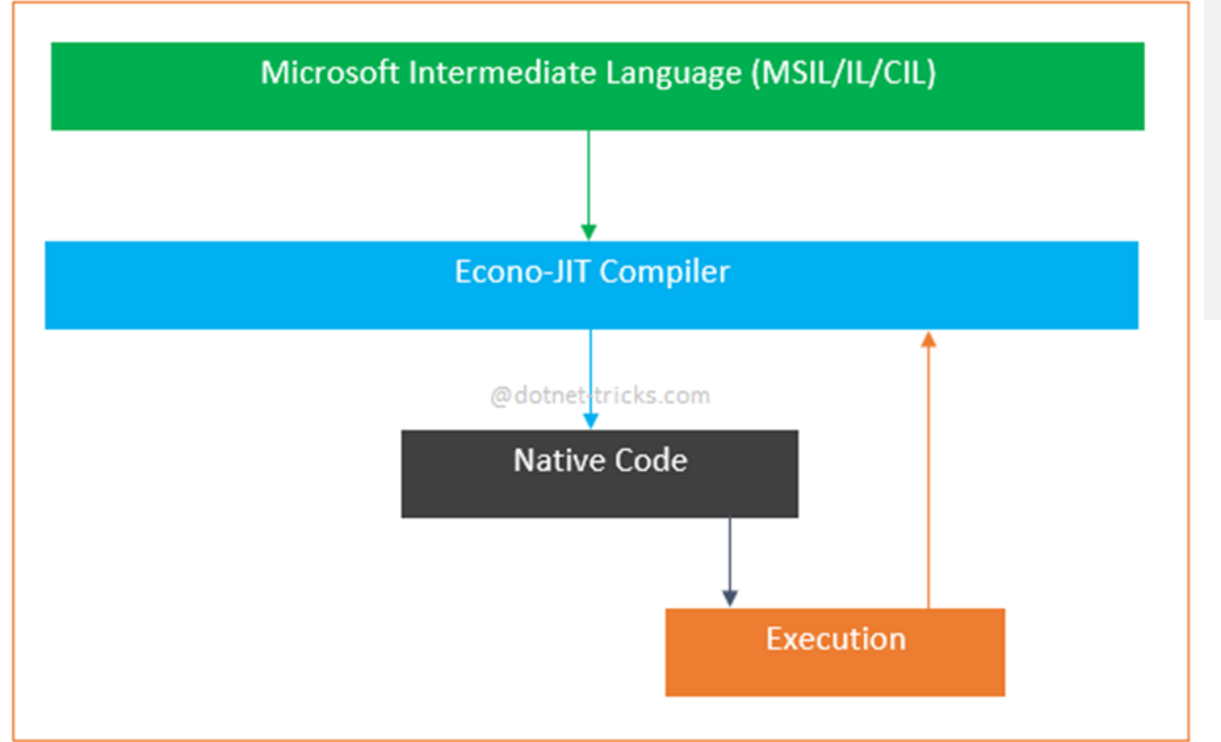
JIT Çalışma Modları: Normal JIT (yorumlayıcı)

- Varsayılan olarak kullanılan JIT derleyicisi
- IL kodunu makine koduna çevirir ve önbellekte tutar.
- 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.



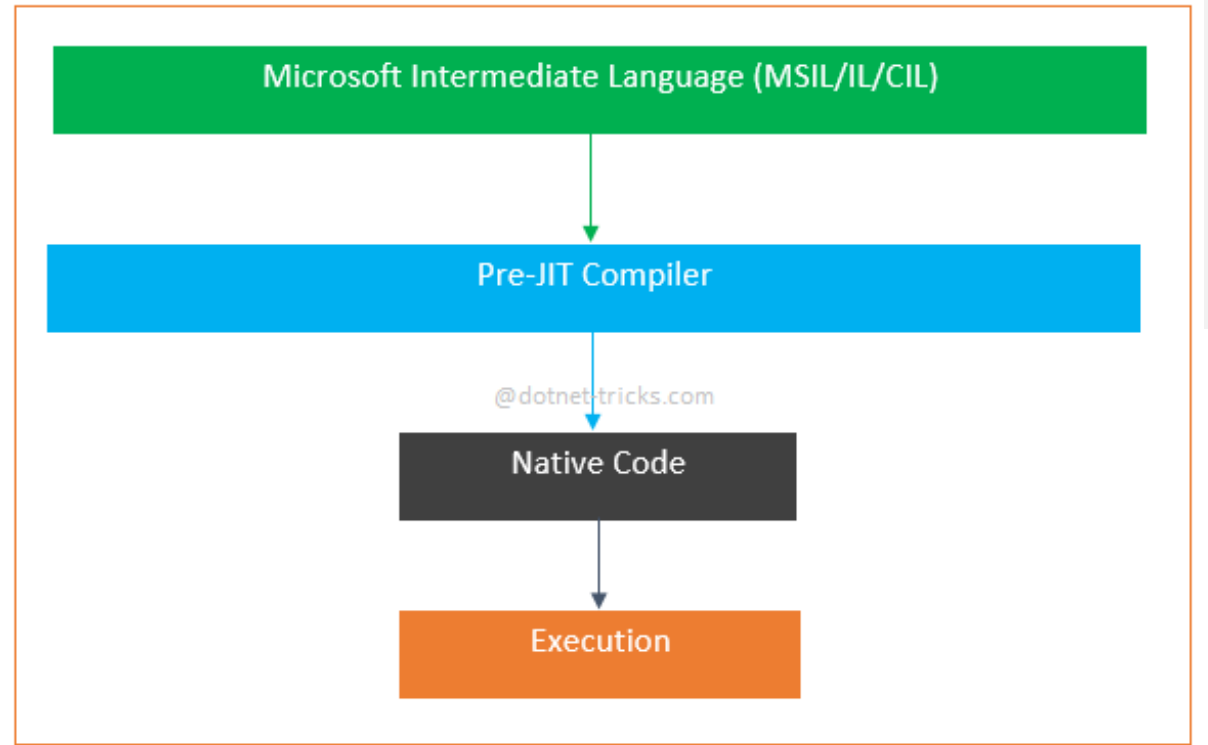
JIT Çalışma Modları: Econo JIT (yorumlayıcı)

- Bazı sistemlerin az miktarda hafıza ve önbellekleri bulunabilir.
- 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
- Ancak, hafızadan silinen kısımları (daha önce derlenmiş ve çalıştırılmış olanlar) tekrar kullanmak gerekirse yeniden makine diline derlenmesi gerekir, programın çalışma zamanı uzar.
- 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 yerel koda tek seferde dönüştürür, böylece yorumlayıcı tarzdan derleyici tarza geçilmiş olur
- 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.
- Her ne kadar hafıza gerektirse de büyük sistemlerin verimlerini arttırabilir, çalışma anında herhangi bir parçanın derlenmesini beklemeyecektir.



CTS (Common Type System)

- ▶ .Net Common Type System – Ortak Tip Sistemi : programlama dillerinin kullandığı veri tipleri arasında uyumluluk.
- ▶ .Net mimarisinde tip güvenli, yüksek performanslı ve kullanılan programlama dilinin diğer .Net uyumlu diller ile entegre bir şekilde çalışmasını sağlar
- ▶ Birçok programlama dilinin tam nesne yönelimli olmasına imkan verir.
- ▶ Herhangi bir .Net uyumlu dilde geliştirilen nesnenin diğer dillerden de sorunsuz olarak çağırılabilmesini garantiler.

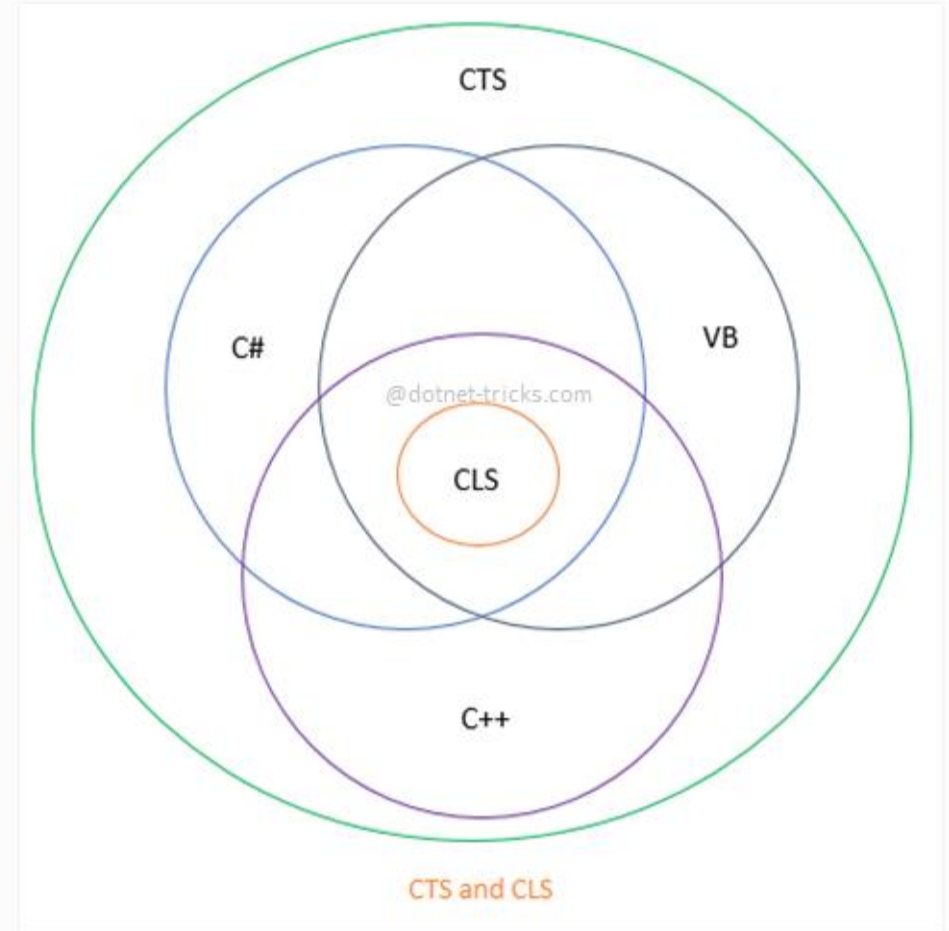
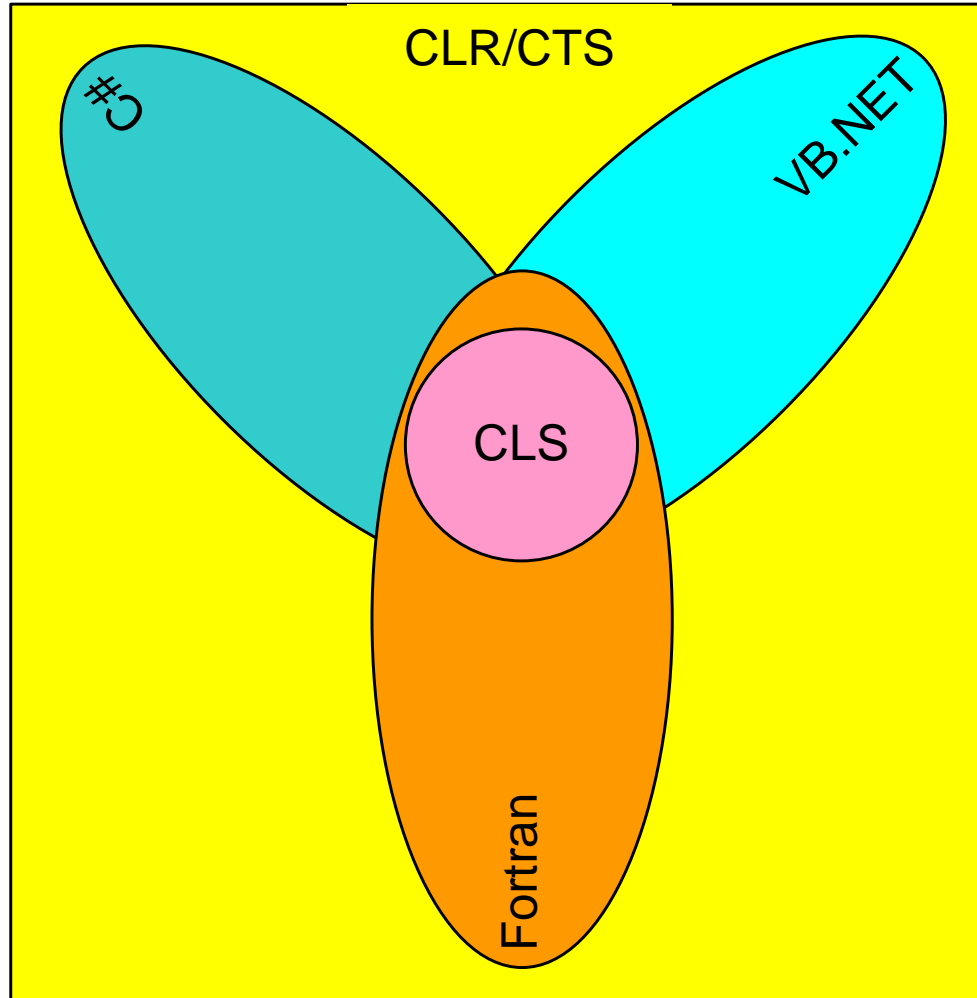
CTS Tipleri

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

CLS (Common Language Specification)

- CLS, Common Language Specification anlamına gelir ve CTS'nin bir alt kümesidir. .NET çerçevesinde çalışan her dilin uyması gereken bir dizi kural ve kısıtlamayı tanımlar. Bu kurallar dizisini izleyen dillerin CLS Uyumlu olduğu söylenir. Basit bir deyişle, CLS diller arası entegrasyonu mümkün kılar.
- .Net platformunda diller arası uyumluluğu sağlamak için veri tiplerinin uyumluluğunun yanı sıra kod yazdığımız dilin Ortak Dil spesifikasyonuna da (Common Language Specification-CLS) uyumlu olması beklenir.
- CLS uyan bir dille yazdığımız kodlar ile diller arası etkileşimi sağlarız. Bu kurallara uyan diller sayıca çok fazla olup örneğin: C#, C++.Net, VB.Net ...
- .Net mimarisinin temel sınıf kütüphanesinde yer alan kodların birçoğu CLS uyumludur
- Örneğin, bir kural, .NET Framework içinde birden çok kalıtım kullanamamanızdır. Bildiğiniz gibi C ++ çoklu kalıtımı destekler ancak; Bu C ++ kodunu C # içinde kullanmaya çalıştığınızda, bu mümkün değildir çünkü C # çoklu kalıtımı desteklemez.
- <https://www.dotnettricks.com/learn/netframework/understanding-relationship-between-cts-and-clr#:~:text=CLS%20stands%20for%20Common%20Language,CLS%20enables%20cross%20language%20integration.>

CLR/CTS/CLS İlişkisi



İsim Alanları ve .Net Sınıf Kütüphanesi

- Programlama dillerinde bir takım hazır kütüphaneler mevcuttur.
- Bazıları standart olmakla birlikte bazıları programcılar tarafından sonradan geliştirilmiştir.
- Standart bir C# kütüphane içermese de .Net Framework 'un programcılara sunduğu birtakım temel tür ve sınıflar mevcuttur. .Net kütüphaneleri bir dilden bağımsız bir yapıdadır ve bunları iyi organize edebilmek için namespace kavramı sıklıkla kullanılır.
- İsim Alanları (namespace) .Net framework sınıf kütüphanesindeki veri türleri ve sınıfları C# dilinde using anahtar sözcüğü ile birlikte kullanılır.
- Programlarımızı yazarken birbiriyle alakalı olarak geliştirdiğimiz sınıfların aynı isim alanı içerisinde olması program yazma, programdaki hataları bulma ve programa yeni özellikler ekleme aşamalarında büyük faydalar sağlar.

Önemli isimuzayları (namespaces)

System	Gerekli temel sınıflar. 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

.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 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'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
- ▶ .NET versiyon uyumsuzlukları

C# ilk Program

```
using System;
```

```
namespace HelloCSharp
```

```
{
```

0 references

```
class Program
```

```
{
```

0 references

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

```
{
```

```
    Console.WriteLine("Hello C# !!!");
```

```
}
```

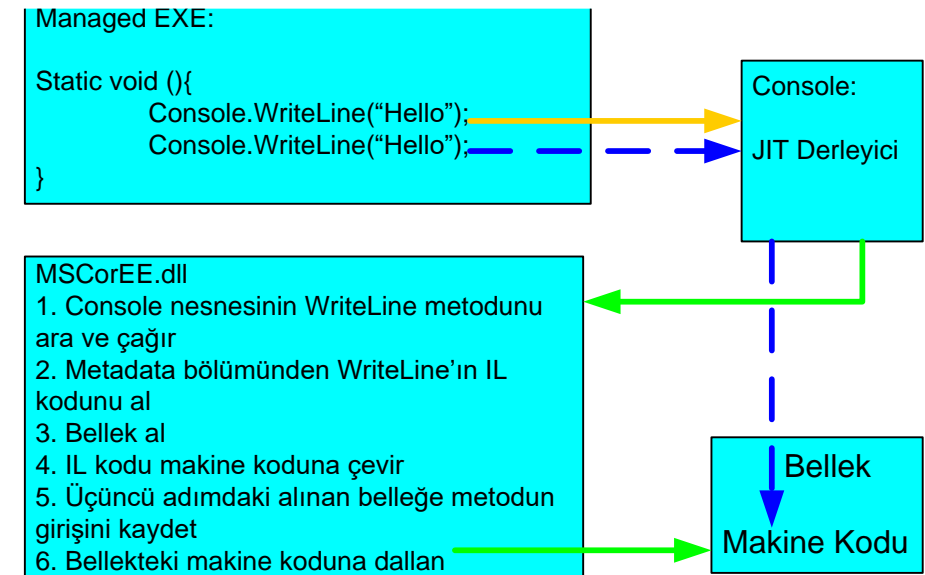
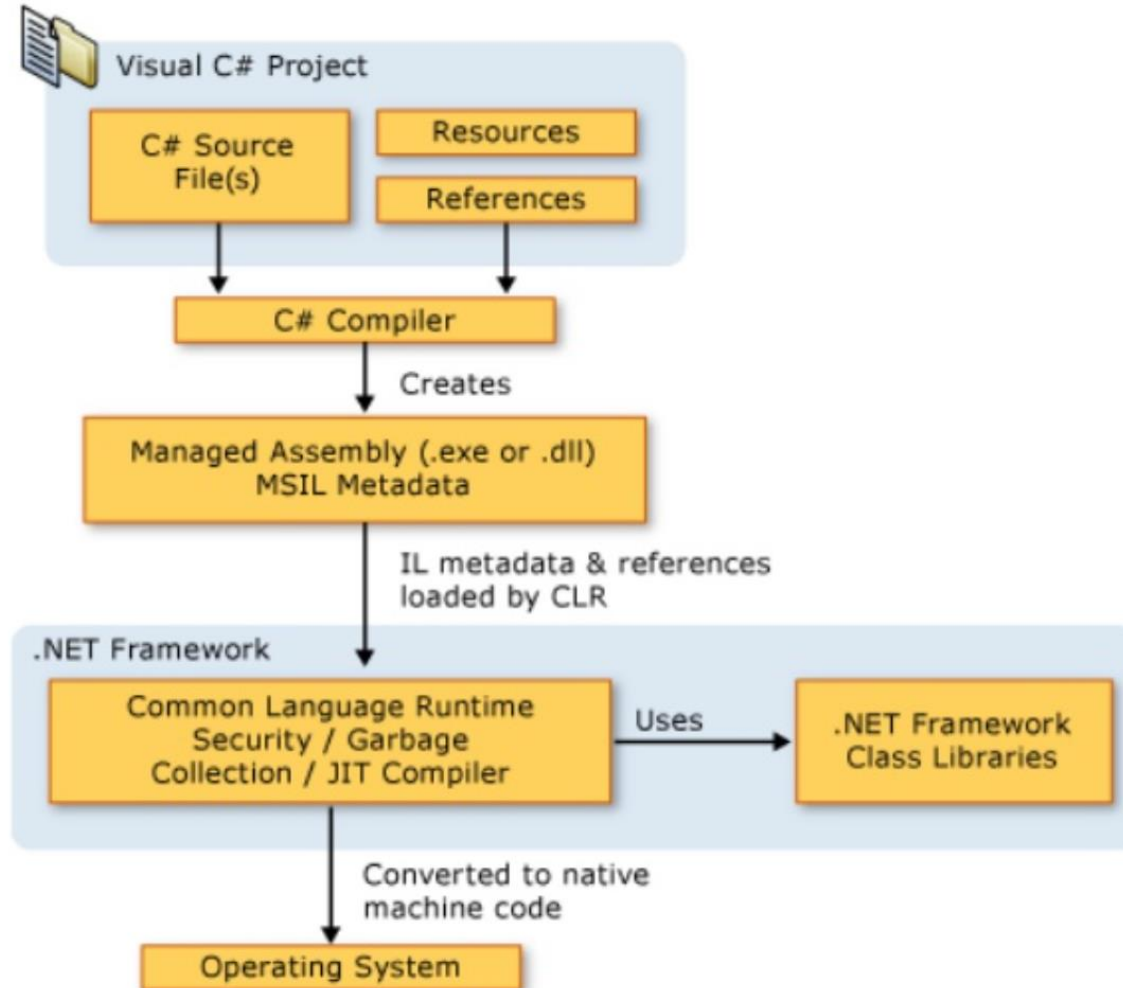
```
}
```

```
}
```

C# İlk Program

- C# %100 nesneye yönelimli bir dildir. C# dilinde yaptığımız her şey bir sınıf nesnesidir. Class anahtar sözcüğüne şimdilik dikkat etmeniz yeterlidir. Sınıflar konusuna ileride değineceğiz.
- C++ dersinden hatırlayacağınız üzere Main fonksiyonu programın çalışmaya başladığı kısımdır.
- Bütün C# programları en az bir sınıf içermelidir. Sınıf bildirimi içinde olmayan programlar derlenemez.
- Kaynak koddaki satırlar ; ile sonlandırılır. Bazı istisnai durumlar olmakla beraber derslerimizde bunları göreceksiniz.
- Bloklar {} işlevin veya sınıfın başlangıç ve bitiş noktalarını gösterir.
- İlk açılan parantez en son kapanır.
- Sınıflar isimalanı dediğimiz kavramlar ile erişilmesi kolay hale getirilmiştir.
- System isim alanı .Net sınıf kütüphanesindeki en temel alanlardandır. Altında diğer isimalanları ve sınıflar mevcuttur.
- Örnekteki WriteLine metodu .Net sınıf kütüphanesinde bulunan Console sınıfının bir elemanıdır.

Örnek C# Program Çalışma Akışı



Çalışma Soruları

Bilgisayarınızda hangi .Net versiyonları yüklü?

Bilgilerinizi ekrana yazdıran bir kartvizit programı yazınız.

Ekrana bir tuşa basınız yazdıran bastıktan sonra ise bir tuşa bastınız yazdıran C# programını yazınız.

Referanslar

- Her Yönüyle C# - Sefer Algan
- <https://www.slideshare.net/hawkmanacademy/c-101-intro-to-programming-with-c>
- <https://www.dotnetperls.com/il>
- <https://www.dotnettricks.com/learn/netframework/understanding-relationship-between-cts-and-clsharp#:~:text=CLS%20stands%20for%20Common%20Language,CLS%20enables%20cross%20language%20integration.>
- <https://www.yusufsezer.com.tr/winforms-wpf-uwp-ve-xaml/>