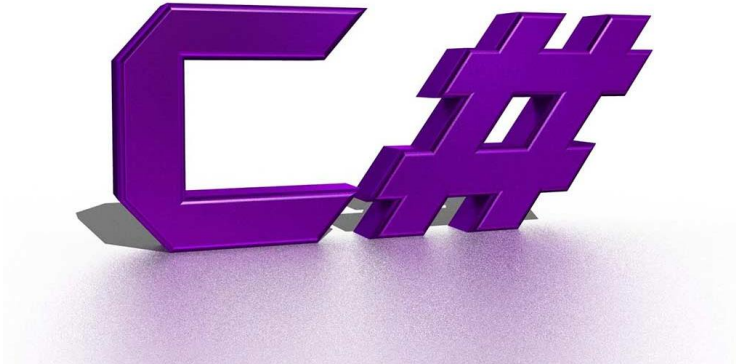




Giriş



Program Tanımı

- *Bilgisayar programı, bilgisayara belirli bir işlemi ya da işlemleri gerçekleştirebilmesi için verilen komut ya da komutlar bütünüdür.*
- *Bilgisayar programı genellikle bilgisayar programcısı tarafından bir programlama dili kullanılarak yazılır.*

Programlama Nedir ?

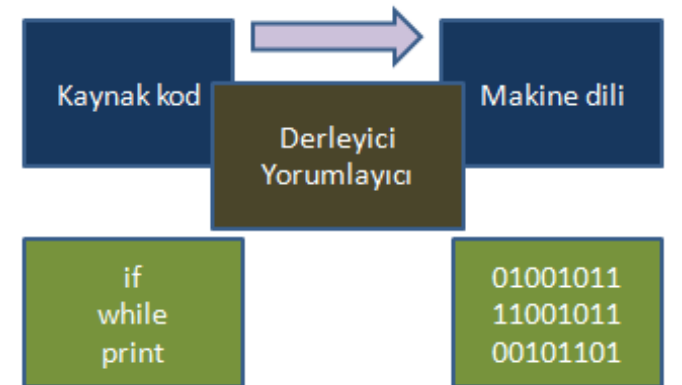


- *Programlama, (farklı tanımlar)*
 - *problemlerin makinelere tanıtılması, öğretilmesi ve öğretilen çözüm yolları ile sonuca ulaştırılmasını sağlamak amacıyla kullanılan teknik bir terimdir*
 - *Bir bilgisayara ya da cihaza nasıl davranacağını, hangi durumlarda ne tepki vereceğini ve bu tepkiler sonucunda nelerin etkileneceğini öğretme işlemidir.*
 - *Bilgisayarın bir işlevi yapması için tasarlanmış komut zinciridir.*

Programlama Dili

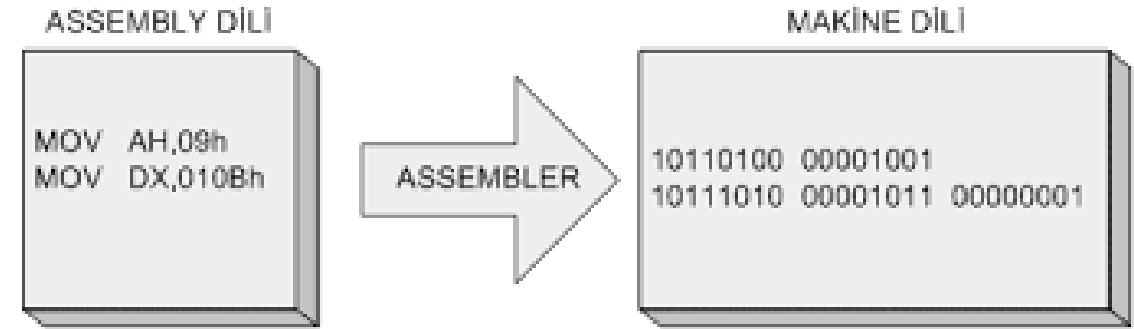


- *Programlama dili;*
 - *bilgisayar ortamında çözülecek bir sorun için, çözümün bilgisayara adım adım yazılmasını sağlayan, biçimsel olan tanımlar kümesidir.*
 - *Kısaca programların yazım kurallarına denir. Kuralların dışında bir yazım programı çalışmasına engel olur.*



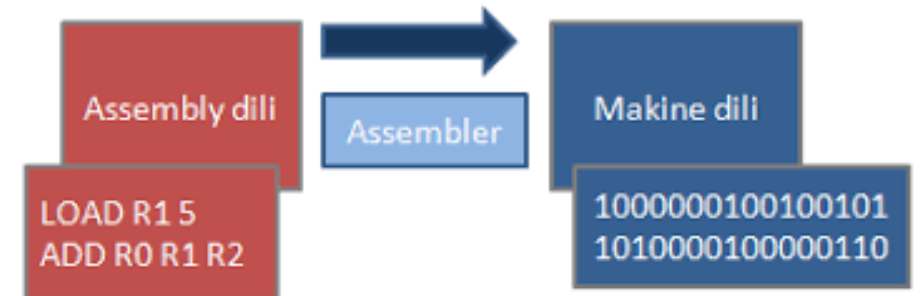
Programlama Dillerinin Temel Komutları

- *Giriş/çıkış komutları*
- *Hesaplama komutları(Aritmatik)*
- *Kontrolün yönlendirilmesi komutları(Mantıksal ve Akış Kontrol)*
- *Veriler üzerinde aktarma, saklama ve geri çağırılması için komutlar.(Bellek ve Disk işlemleri)*



Binary Code							
A	100 0001	H	100 1000	O	100 1111	V	101 0110
B	100 0010	I	100 1001	P	101 0000	W	101 0111
C	100 0011	J	100 1010	Q	101 0001	X	101 1000
D	100 0100	K	100 1011	R	101 1010	Y	101 1001
E	100 0101	L	100 1100	S	101 0011	Z	101 1010
F	100 0110	M	100 1101	T	101 0100	a	110 0001
G	100 0111	N	100 1110	U	101 0101	b	110 0010

- *Programlar dört farklı dilde yazılabilir.*
- *Bunlar sırasıyla;*
 - *makine dili*
 - *Assembly dili*
 - *orta seviyeli diller*
 - *yüksek seviyeli diller*



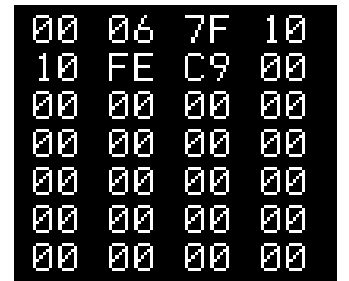
Makine Dili



- *Makine dili (machine code) farklı tanımları;*
 - *Bilgisayarların ek bir dönüşüme ihtiyaç duymaksızın doğrudan anlayabildikleri komutlar ve tanımlamalar kümesine makine dili denir.*
 - *doğrudan bilgisayarın işlemcisi (CPU) tarafından anlaşılabilen ve çalıştırılabilen komutlardan oluşan en temel program yazma aracıdır.*
 - *bilgisayarın işlemcisine ne yapacağını direk olarak söyleyen okunabilirliği düşük ve olabilecek en alt seviye programlama dilidir.*
- *Özet olarak, makine kodu ikili sayılardan oluşur.*
- *Komutlar, veriler ve adresler hep ikili sayı sistemi ile yazılır.*

- *Günümüzde programcılar makine kodu yazmak için gerekli detayları göz önünde bulundurmaksızın üst düzey diller kullanarak program yazmaktadırlar.*
- *Yani makine dili yerine programlama dili kullanılarak kodlama yapılmaktadır.*

- *Makine dili, bir bilgisayar tarafından direkt olarak anlaşılabilir yani öncesinde bir transformasyon (dil dönüşümü) gerektirmez.*
- *Kodun işlemciye direk geçmesi ve okunabilirliğinin azalması onu daha alt seviye yapar, okunabilirliği yüksekse üst seviye programlama dilidir.*
- *Makine kodunda arada hiçbir çevirme yoktur. Yani direk olarak CPU ile iletişim kurarsınız.*
- *Bir programlama dili ile yazılmış kodun bilgisayar tarafından çalıştırılabilmesi için makine diline çevrilmesi gerekir.*
- *Bu dönüşüm iki araç kullanılarak yapılabilir:*
 - *Derleyici (compiler)*
 - *Yorumlayıcı (interpreter)*

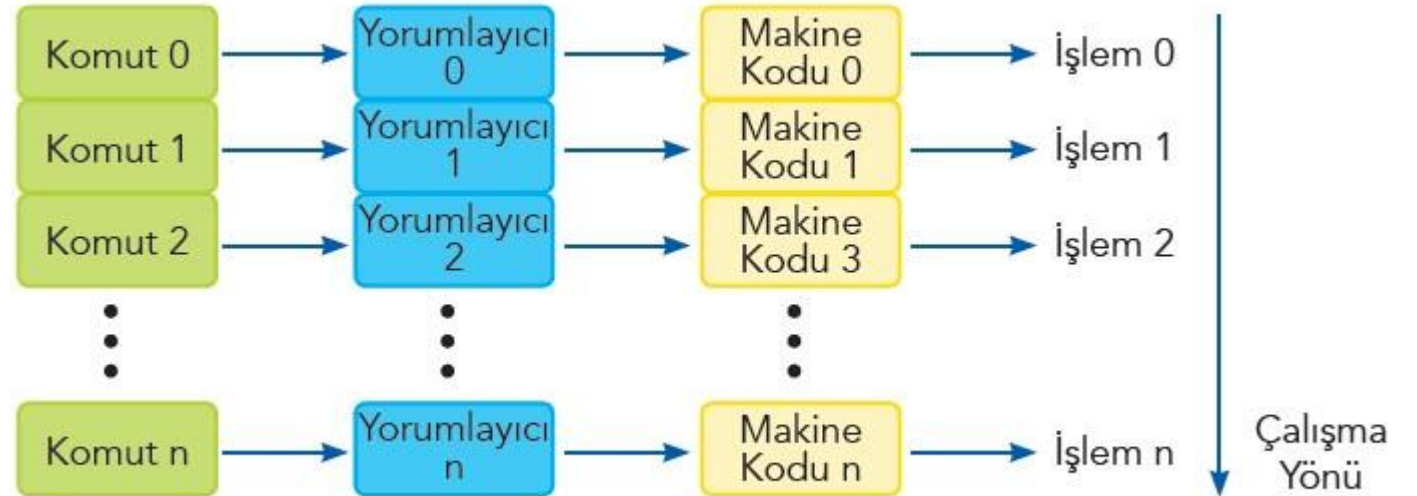


Derleyici (compiler)

- *Yüksek seviyeli programlama diliyle yazılmış kaynak kodu alan, **makinenin mimarisine** göre makine dilinde çıktı üreten bir programdır.*
- *Çıktı olarak üretilen makine kodu sonradan herhangi bir zamanda farklı girdilerle tekrar tekrar çalıştırılabilir.*
- *Derleyici, kaynak kodu derlemeden önce **kodları kontrol** eder. Herhangi bir yazım hatası veya derleyicinin sevmediği durumlar var ise bu raporlanır.*
- *Bu da tam olarak çalışan program elde etmeden önce tüm kodlama hatalarını düzeltmemizi sağlar.*
- *Ancak bir programın başarılı olarak derlenmiş olması tam olarak hatasız olduğu ve istenileni yapacağı anlamına gelmez. Programımızı çeşitli verilerle **test** edip, **mantıksal** olarak da çalıştığından emin olmamız gerekir.*

Yorumlayıcı (interpreter)

- Kodu **satır satır** veya **bloklar halinde** çalıştırıp sırası gelmeyen satırları hiç çalıştırmayan bu satırlardaki hataları hiçbir zaman göremeyen ve kodun bütününe ait iyileştirmeleri yapamayan çeviricilere denir.



Şekil 2.2: Yorumlayıcı yapısının grafiksel gösterimi

Derleyici ve yorumlayıcı arasındaki farklar

<i>Yorumlayıcı</i>	<i>Derleyici</i>
<i>Programı satır satır işler</i>	<i>Tüm programı tarar ve bir bütün olarak makine koduna çevirir</i>
<i>Kaynak kodu analiz etmekle zaman harcamaz. Ancak genel yürütme süresi daha yavaştır.</i>	<i>Kaynak kodun analizi için büyük zaman harcar. Ancak genel yürütme süresi daha hızlıdır.</i>
<i>Herhangi bir hata olana kadar programı çalıştırır. İlk hata gördüğü yerde durur. Bu nedenle hata ayıklama kolaydır.</i>	<i>Tüm kaynak kodu taradıktan sonra hata mesajı üretir. Bu nedenle hata ayıklama nispeten zordur.</i>
<i>Python, Ruby, Java HTML, XML, PHP, Script gibi diller yorumlayıcı kullanır..</i>	<i>Pascal, Visual Basic, C, C++ gibi diller derleyici kullanır.</i>

Derleyiciler (compiler) ile Yorumlayıcılar (Interpreter) özet tanım

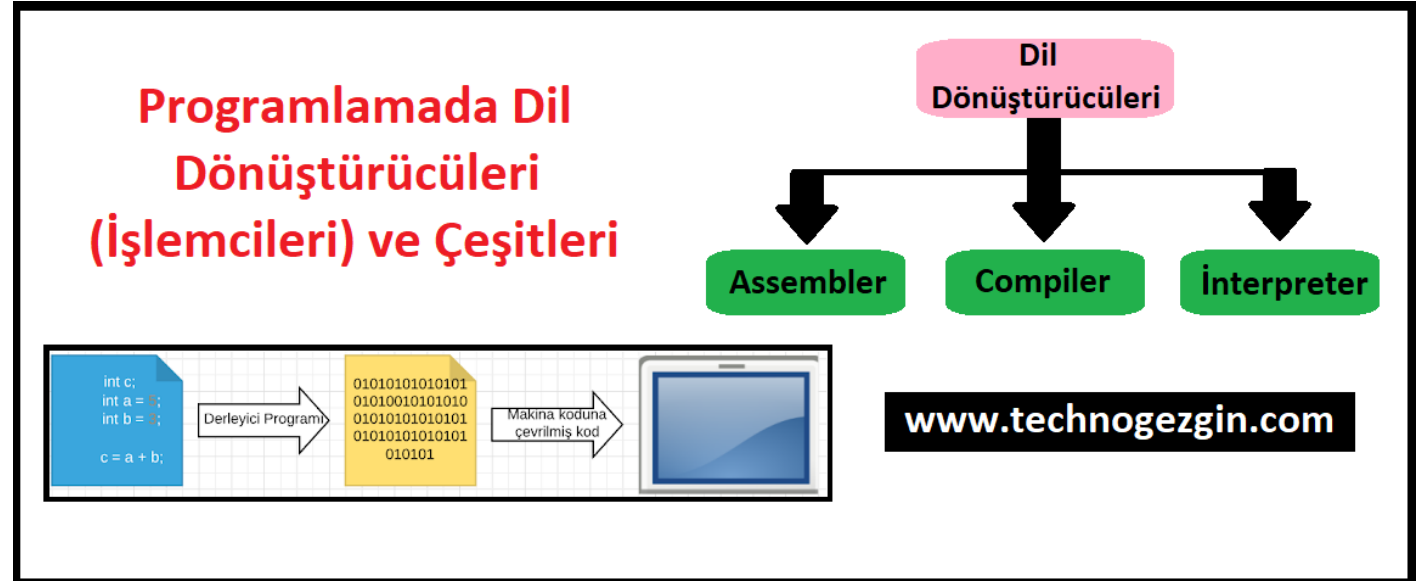
- Bir kaynak kodu hedef koda çevirdikten sonra çalıştıran ve dolayısıyla koddaki hataları yakalama işlemini ve kodun iyileştirilmesini daha kod çalıştırmadan yapan çeviricilere derleyici, kodu satır satır veya bloklar halinde çalıştırıp sırası gelmeyen satırları hiç çalıştırmayan bu satırlardaki hataları hiçbir zaman göremeyen ve kodun bütününe ait iyileştirmeleri yapamayan çeviricilere de yorumlayıcı (interpreter) adı verilmektedir.*

Assembly(Çevirici) Dili

- *Assembly “mnemonic” (hatırlatıcı ipucu)olarak adlandırılan komut yapılarından oluşan makine kodundan sonra en alt seviyelide yer alan bir programlama dilidir*
- *Bu komutlar akılda kalıcı olduğundan makine diline göre yazımı çok daha kolaydır.*
- *Assembly makine kodundan sonra en alt seviyeli programlama dilidir.*

Assembler(Çevirici-Derleyici)

- *Assembly dili ile yazılan komutların birleştirilmesi, düzenlenmesi ve makine diline çevrilmesi işlemini yapan çevrim programlarıdır.*
- *Assembler sayesinde makine dili bilmeden makine kodları kolaylıkla oluşturulabilmektedir*



C# Nedir?

- *C# isminin temeli C, C++ ve Java dilinden gelir.*
- *C dilinde bir tamsayı değişkenin değerini artırmak için “++” operatörü kullanılır.*
- *Bu yüzden C dilinden sonra geliştirilen ve Nesne Yönelimli (Object Oriented) özellikleri taşıyan yeni dil için, C'nin bir fazlası anlamında C++ adı kullanılmıştır.*
- *C++'tan sonra geliştirilen C# dili ise adını; ((C++)++) ifadesinde artı işaretlerinin bir araya gelmesinden alır.*

- *C# C /C++ ve Java dillerinde türemiş güçlü, basit, esnek, tip güvenli, modern bir programlama dilidir.*
- *1990'lı yıllarda tasarımı yapılmaya başlanmış ve 2000 li yıllarda piyasaya sürülmüştür.*
- *C#, Microsoft'un geliştirmiş olduğu yeni nesil programlama dilidir.*
- *Yine Microsoft tarafından geliştirilmiş .NET Teknolojisi için geliştirilmiş dillerden biridir.*

Neden C# ?

- *Kolay öğrenme(anahtar kelimelerin mantığını öğrenmek)*
- *%100 Nesne yönelimli Programlama Dili*
- *Yüksek verimli (hatalar yapıldığında yazıldığında (tip güvenliği)yakalanabiliyor)*
- *Güncellenen-yenilenen*
- *Büyük projeleri kodlarken dilin basitliği ve mimarisel bütünlüğü daha esnek ve hızlı kodlama imkanı sağlar.*
- *C/C++, Java dillerinden birini biliyor ya da aşınaysanız çok kısa zamanda c# dili ile kod yazmayı öğrenebilirsiniz.*

Nesne Yönelimli Programlama (NYP)

Object Oriented Programming(OOP)nedir ?

- *Nesne tabanlı programlama (OOP);*
 - *yazılım tasarımını işlevler ve mantık yerine, veri veya nesneler etrafında düzenleyen bir programlama dili modelidir.*
- *Günümüzde yazılım geliştirme teknolojileri içerisinde büyük bir önem taşır.*

Nesne Yönelimli Programlamanın Faydaları

- Nesne oluşturma bir sınıf içerisinde toplanır ve tüm projelerde kullanılabilirliğe olanak sağlar.*
- Sınıfların bir kez oluşturulması sayesinde uzun kodları tekrardan yazmak yerine kısa kodlamalar ile çalıştırılabilir.*
- Uzun kodların tekrar yazılmasının engellenmesi sayesinde geliştirme süreci kısalır.*
- Nesneler birbirinden bağımsız olduğundan bilgi gizliliği konusunda avantaj sağlar.*
- Sınıflar sayesinde tüm projelerde değişiklik yapmak yerine tek bir sınıfta değişiklik yapıp tüm projelerde çalışması sağlanır. Bu zaman kaybını büyük ölçüde azaltır.*

C# ile Neler Yapılabilir, Nerelerde Kullanılır

- *Konsol uygulaması geliştirilebilir.(siyah ekranda çalışan)*
- *Windows form, Web uygulamaları (Tüm iş süreçlerinde küçük büyük orta işletmelerin uç tan uca tüm yazılım ihtiyaçları c# ile yazılabilir)*
- *İşletmelerin başka platformlar ile beraber çalıştığı durumlarda diğer sistemlere entegre olabilen yazılımlar geliştirilebilir.*
- *Mobil uygulama geliştirilebilir.(Windows Phone)*

C# ile Neler Yapılabilir, Nereelerde Kullanılır

- *İşletmelerin kullandığı tüm platformlara ait mobil uygulamaları var ise hem bunlar ile **güvenli** veri(data) akışı sağlayan uygulamalar geliştirilebilir.*
 - *buradaki güvenli kavramı önemli veri transferi sırasında araya girip paketler üzerinde işlem yapmaya çalışanlar için c# güvenlik kısımları bulunmaktadır.*
- *ASP.NET İnternet Uygulamaları;*
 - *Microsoft tarafından geliştirilmiş bir web uygulama gelişimi teknolojisidir*

C# ile Neler Yapılabilir, Nereelerde Kullanılır

- *Web API (Application Programming Interface-Uygulama programlama ara yüzü)*
 - *bir takım verileri sunan servisler-web üzerinde o servislere veri çekme, veri gönderme işlemleri yapmamızı sağlar*
- *Tek bir veri tabanı üzerinden farklı uygulamaların çalıştırılması.*

(İşletmelerin genelde mobil tarafları giderek artmaktadır hem kendi satış temsilcileri hem de müşteri taraflı olarak bulunmakta bu farklı ortamları tek bir veri tabanı üzerinden veri kaynağı olarak kullanımını c# sağlayabilir.)

C# ile Neler Yapılabilir, Nereelerde Kullanılır

- *Belirli bir süre içerisinde çalışmasını istemiş olduğumuz iş bloklarını c# ile kodlayabilir ve Windows servisi olarak ilgili sunucuya kurabilir ve çalışmasını sağlayabiliriz.*
- *Sonuç olarak c# bir çok platformda kullanılabilir.*

C#'ın kısa bir tarihi

- *Başlangıç: C# 1.0 (2002)*
- *C# 2.0'dan (2005)*
 - *Generic*
- *C# 3.0'*
 - *LINQ (Language Integrated Query - Dil Destekli Sorgulama)(Veritabanı işlemlerini SQL sorgu kullanmadan c# sorguları üzerinden yapılması)*
- *C#'ın 4.0*
 - *Dynamic yapılar(Dinamil anahtar kelimesi)*
- *C# 5.0*
 - *Asenkron özellikler*
- *C# 6.0*
- *C# 7.0*
- *C# 8.0*

Sürüm	Yayınlama Yılı
Sürüm 1.0	2002
Sürüm 2.0	2005
Sürüm 3.0	2007
Sürüm 4.0	2010
Sürüm 5.0	2012
Sürüm 6.0	2015
Sürüm 7.0	2017
Sürüm 8.0	2019

Neden Visual Studio Kullanıyoruz

- *C# ile yazdığımız bir program text dosyasında yazılıp c# derleyicisinde derlenebilir.*
- *Visual Studio kullanmak zorunda değiliz ama visual studio bize kod hatırlatması, kelimenin tamamlanması anahtara kelimeleri yanlış yazdığımızda uyarıyor düzeltmemize yardım ettiği için editör olarak visual studio kullanıyoruz.*
- *Çalıştır dediğimizde c# derleyicisinde kodu derliyor console uygulaması ise konsol ekranı, Windows ekranı ise Windows ekranı açılıyor ve programımız çalışıyor.*

.Net Framework

- *Framework, yazılım geliştiricilerin kullandığı önceden hazırlanmış kütüphanelerin bulunduğu ve bunlara yenilerini ekleyebileceği yapıların adıdır.*
- *Gelişmiş frameworklerde; form kontrolü, veri tabanı bağlantısı, kullanıcı giriş çıkış, mail atma, tema motoru gibi kütüphaneler mevcuttur.*

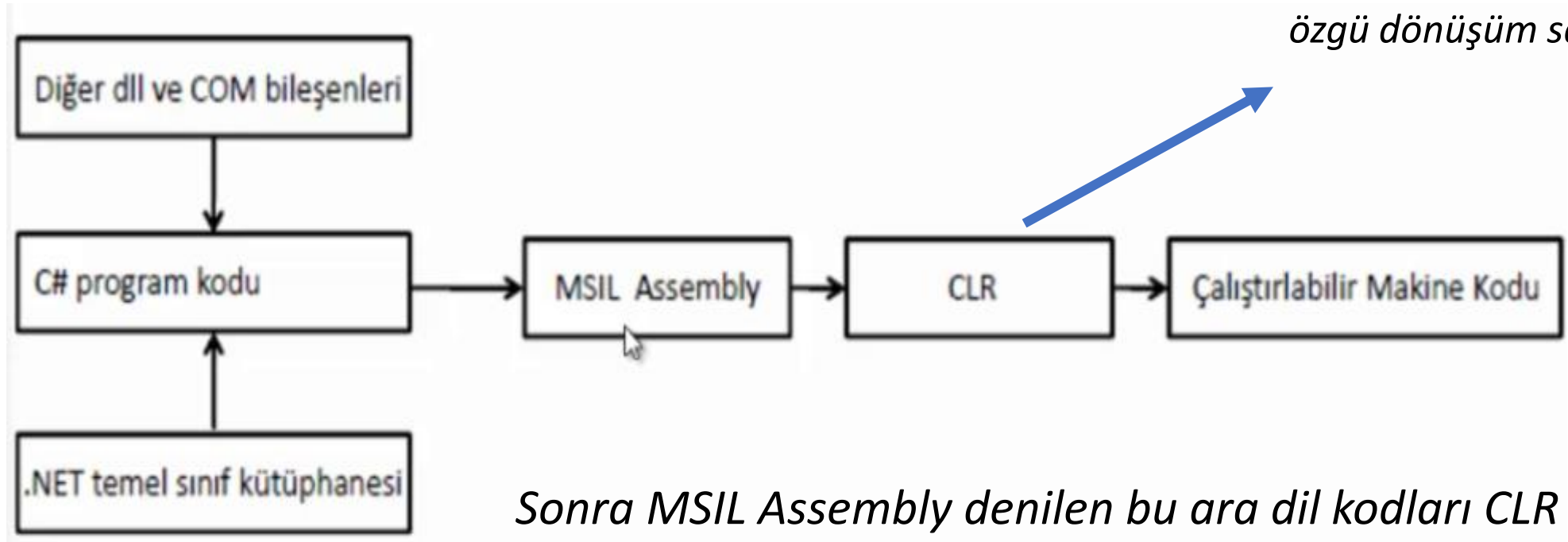
- *Microsoft .Net mantığında, işletim sisteminden bağımsız kod geliştirme özelliği ekliyor.*
- *Bu Microsoftun tüm işletim sistemleri arasında bir bağımsızlık sağlıyor demek*
- *Yazılan C# kodları makine diline dönüştürmeden önce Microsoftun **bir ara diline çevriliyor** ara dilden sonra tekrar makine diline dönüştürülüyor.*
- *Bu sayede Microsoft'un farklı işletim sistemi de olsa bu ara dil sayesinde sıkıntı oluşmadan işletim sisteminden bağımsız olarak programın çalışması sağlanmış oluyor.*
- *Bu çevirme işlemine bakalım.*

IL -Intermediate Language(Ara Dil)

(MSIL - Microsoft Intermediate Language)

- *Bütün kodların makine diline çevrilmeden önce dönüştürülen ara bir dildir.*
- *Eskiden nasıldı;*
 - *yazılan kodlar direk makine diline çevrilir ve işletim sistemi ile iletişime geçerek bu kodlar çalıştırılırdı.*
 - *Buda bize her işletim sistemine uygun derleme ve kod yazımı mecburiyeti getiriyordu.*
 - *Yani biz işletim sistemine özgü düzenleme yapıyorduk.*

Şimdi bu düzenlemeyi .NET yapıyor, işletim sistemine uygun olarak dönüştürülen ara dil ise IL oluyor.



CLR: İşletim sistemine özgü dönüşüm sağlar

Sonra MSIL Assembly denilen bu ara dil kodları CLR VE JIT derleyicileri ile okunup makine koduna dönüştürülüyor.

CLR (Common Language Runtime – Ortak Dil Platformu-Çalışma Zamanı)

- *.Net in diğer bir alt yapısıdır.*
- *Farklı kodlar ile yazılan tüm kodlar IL ile ortak bir dile dönüştürülüyor demiştik.*
- *CLR da bu ortak dili yorumlayıp makine diline dönüştürüp işletim sistemi üzerinde çalışmasını sağlıyor.*
- *Yani işletim sistemi ne olursa onun CLR ı çalışır.*
- *Diğer bir tanımlama ile CLR işletim sistemine özgüdür.*

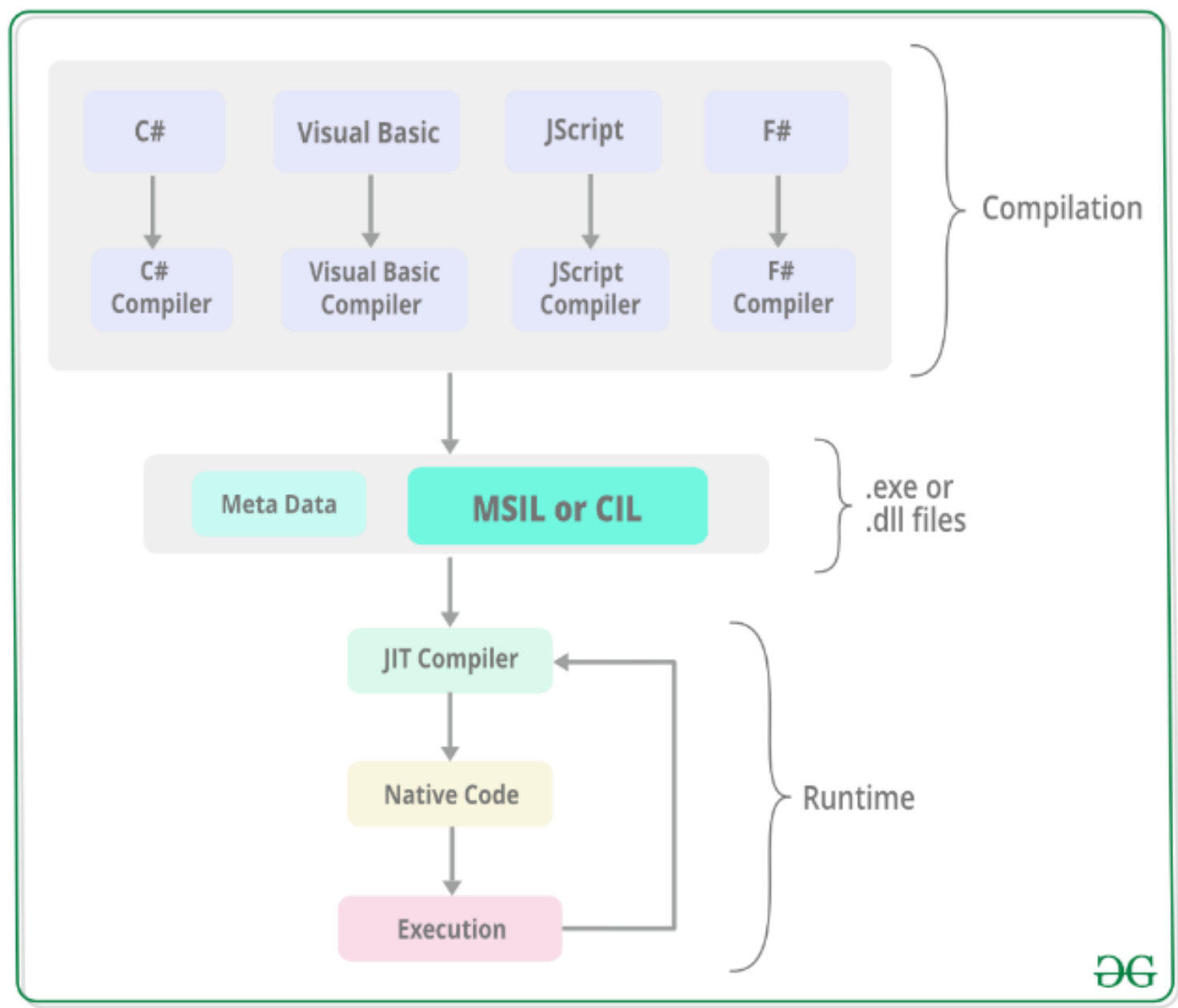
- *. Siz c# java,c ya da farklı bir dille kod yazın bu kod MSIL dönüşecek CLR bu kodu tanıdığı için bu kodu alıp makine diline dönüştürecek.*
- *IL sadece yazılan kodları birebir dönüştürülmüş hali değildir tanımlamış olduğunuz sınıflar, veri tipleri gibi bilgileri de tutar.*
- *Günlük hayattan örnek verecek olursak IL'i bir çevirmen gibi düşünebiliriz dünyada konuşulan tüm dilleri bilen ve bu dilleri ortak bir dile dönüştüren yapıdır.*
- *Yani hem tüm dilleri bilecek hem de ortak olan o dili biliyor.*

Özet

- *.NET Framework ilk olarak yazılan kodu IL/MSIL denilen ve işletim sistemine uygun (Windows sürümleri)ara bir dile dönüştürüyor, sonra CLR bu ara dili makine koduna çeviyor, JIT ise donanım özelliklerine uygun bu makine kodunun ya hepsini ya bir kısmını çalıştırılmasını gerçekleştiriyor.*

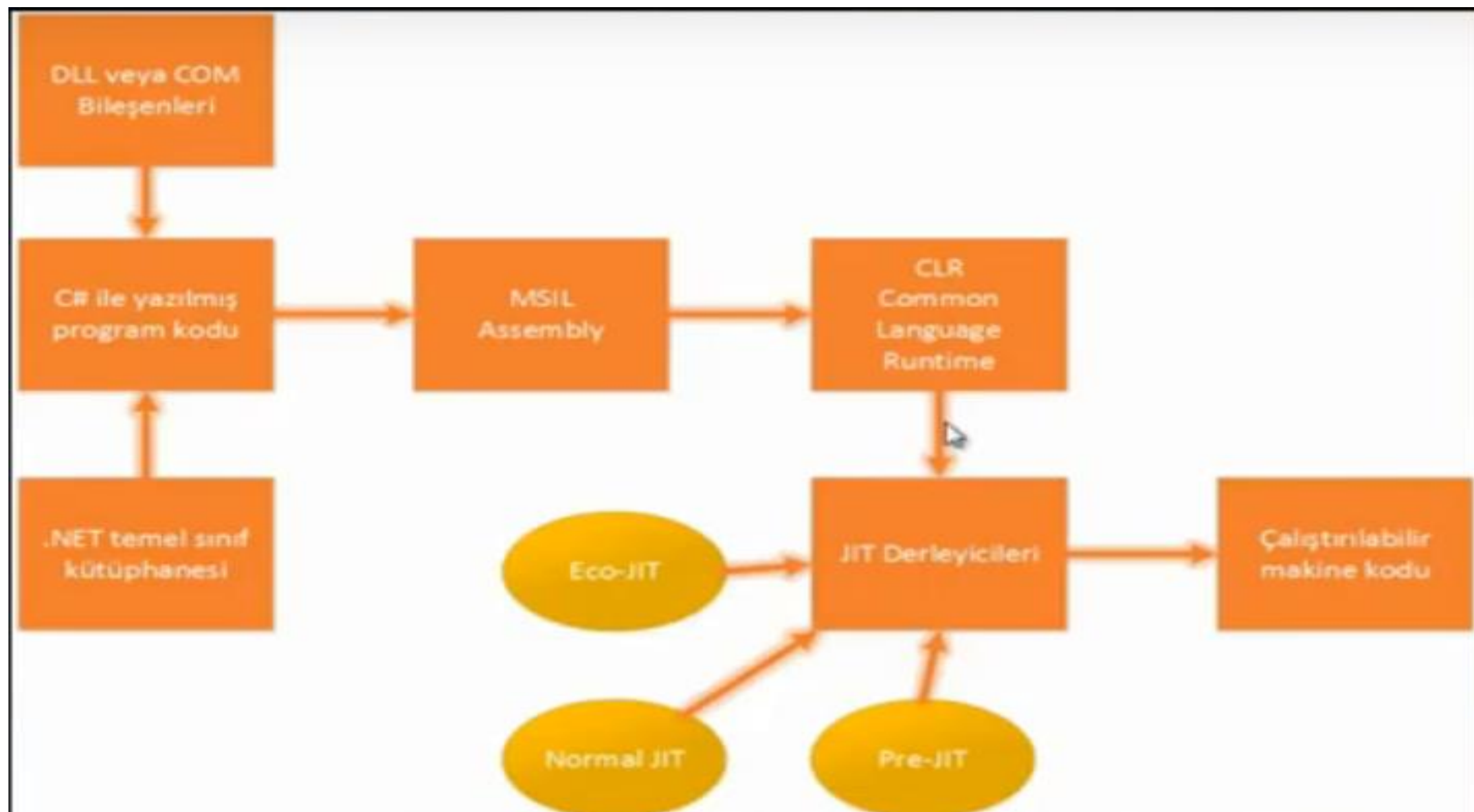
JIT(Just In Time) Tam Zamanında

- CLR üzerinde çalışan bir **derleyicidir**.
- C# üzerinde yazılan ve ara dile dönüşen kodlar hangi işletim sistemi üzerinde çalışacaksa ilgili JIT derleyici ile makine koduna dönüştürülür.
- JIT derleyicileri işletim sistemine bağımlıdır.
- Linux ün kendine özgü JIT derleyicisi olmalıdır.



Üç farklı JIT Derleyicisi vardır

- *Normal JIT*
- *Pre-JIT*
- *Eco-JIT*



Normal JIT:

- *IL kodu makine diline çevrilirken varsayılan olarak kullanılan JIT derleyicisidir.*
- *Makine diline çevrilen kodu ön bellekte tutar.*
- *İhtiyaç halinde RAM den çalıştırırlar*

Eco-JIT:

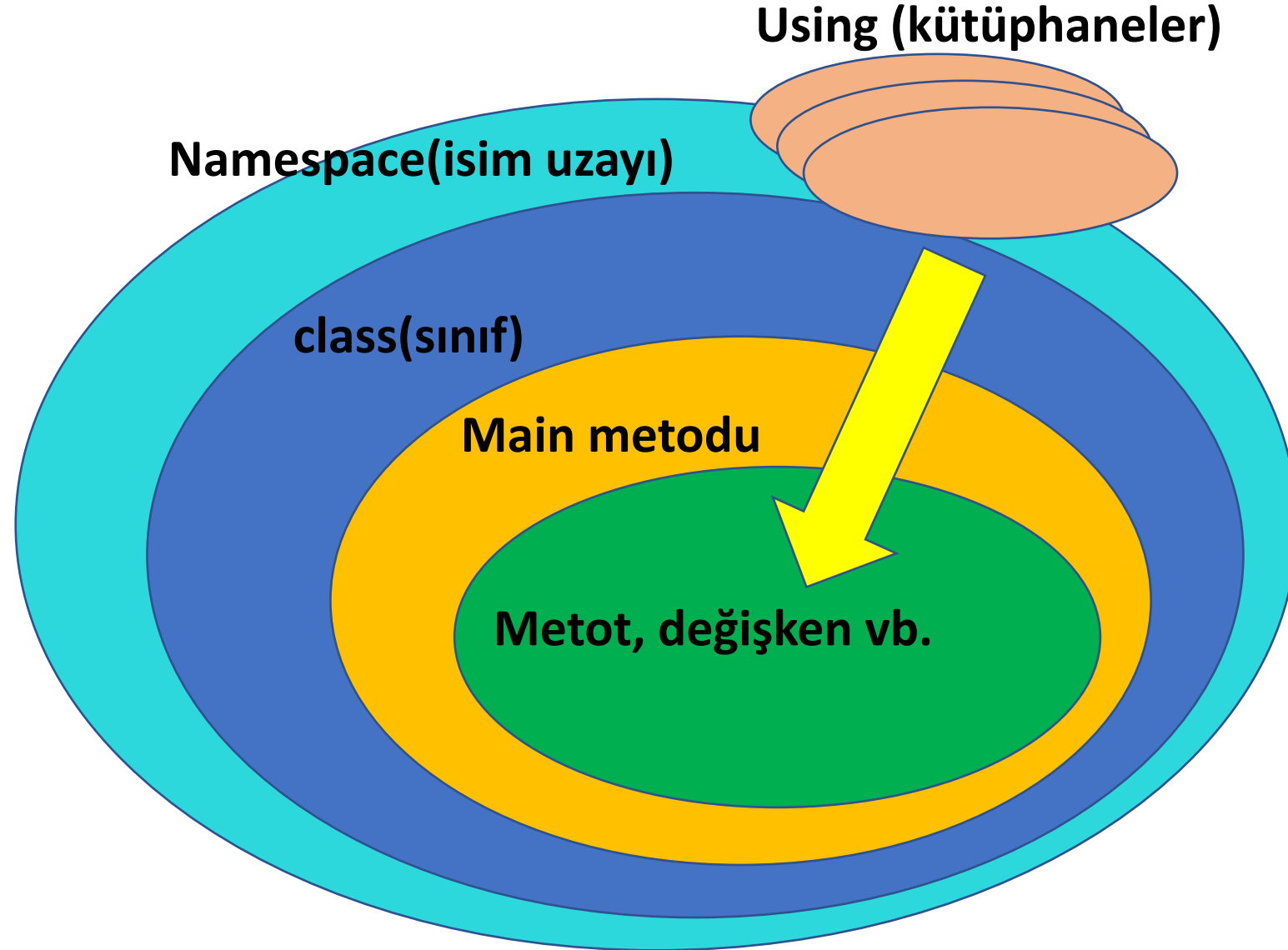
- *Ekonomik sistemle(cep telefonu, mobil tabletler vb.)r için kullanılabilir.*
- *Hafızası ve önbellekleri az olan sistemlerde .NET programlarının daha iyi çalışması için kullanılır.*
- *Hafızası az olduğundan daha önceden derlenenleri silerek işlemine devam eder, yani kodun çok az bir kısmı tutulur.*
- *Tekrardan kullanmak gerekince tekrar derlemesi gerektiğinden çalışma zamanı uzar.*

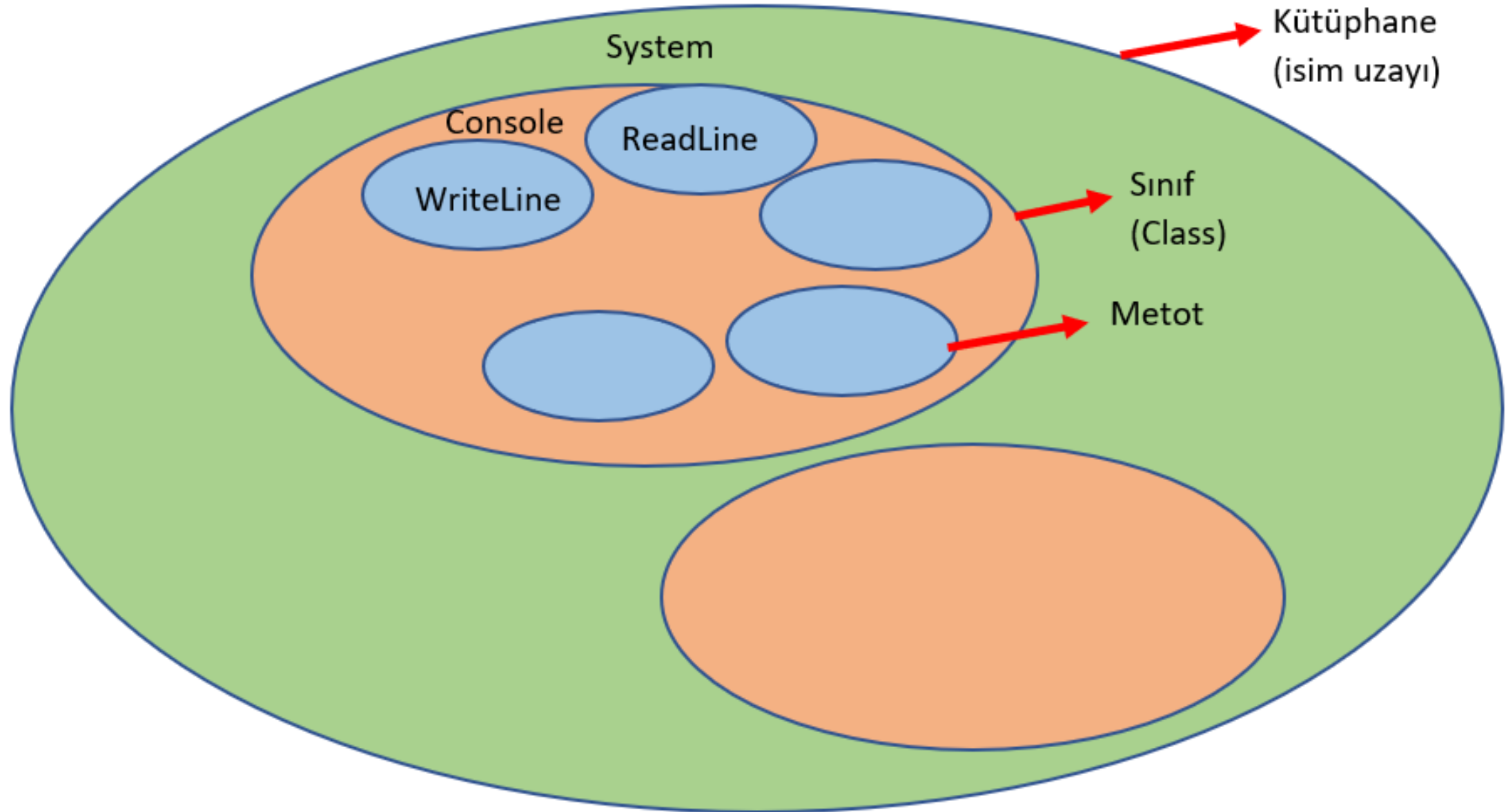
Pre-JIT:

- *Pre-Jit programları daha hızlı(Performanslarını artırmak) çalıştırması için yapılmıştır.*
- *Programın hepsi derlenir daha sonra işleme alınır bu da büyük hafıza gerektirir.*
- *Ama çalışma anında herhangi bir parçanın derlenmesini beklemediğinden Normal JIT' e göre daha hızlıdır.*

Bir hata ile karşılaştım! Bir şeyler yanlış...

- **Yazılım Dünyası == Hatalar Dünyası**
- Özet olarak **yazılımcı kime denir?** (*Sorun çözen insan...*)
- Profesyonel **yazılımcılar neden sabırlıdır?**
- Bahaneler! İyi **anlatamıyor...** Kitap da **yetersiz!** Yazar **iyi açıklayamıyor...**
- **Kendine odaklanmak ve hatalarını görebilmek...**
- **Okuma makinesi olmak!** *Kitaplar, makaleler, dökümanlar üzerine...*
- **Arama motorlarına hakim olmak!**
- Sormadan önce **araştırma kültürünü benimsemek!**
- Çözülen bir hatayı internet üzerinden **paylaşma kültürüne sahip olmak!**
- **Hatalardan öğrenmek...**





Kaynakça

- <https://furkanalaybeg.medium.com/nesne-y%C3%B6nelimli-programlama-oop-nedir-b6f805a9473f>
- *8051 & ADuC8xx EĞİTİM NOTLARI Selim Dilmaç 8051 Assembly Dilinde Programlama*
- <http://www.bilisimakale.com/2018/11/13/yorumlayici-interpreter-ve-derleyici-compiler-nedir/>
- <https://www.ahmetcevahircinar.com.tr/2016/07/24/derleyiciler-compiler-ile-yorumlayicilar-interpreter-arasindaki-farklarnelerdi>
- <http://bilgioloji.com/pages/yazilim/kod/program/giris/yorumlayici-interpreter-nedir/>
- *Hüseyin POLAT – Bilgisayar Mühendisliğine Giriş*