

# C# Development with OOP

**Salih DEMİROĞ**

Founder - SLD YAZILIM

Full Stack Developer, Trainer & Consultant

[salih@sldyazilim.com](mailto:salih@sldyazilim.com)

[salihdemirog@gmail.com](mailto:salihdemirog@gmail.com)

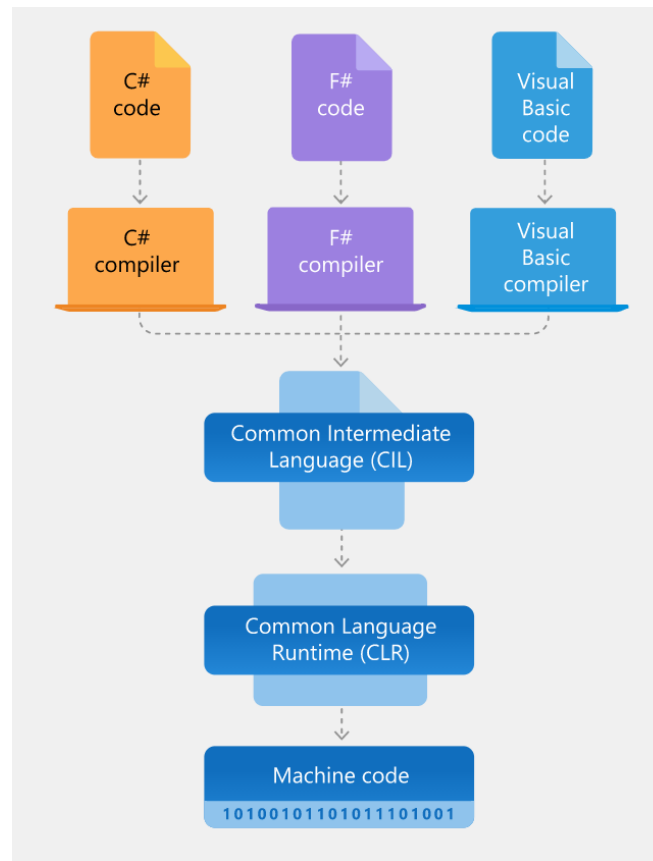
@salihdemirog



# .NET Nedir?

- ❑ .NET, Microsoft tarafından desteklenen ücretsiz, platform bağımsız ve açık kaynak bir yazılım geliştirme platformudur.
- ❑ .NET güvenli, güvenilir ve yüksek performanslı bir uygulama platformudur.
- ❑ .NET, koleksiyonlardan, ağ iletişimine ve makine öğrenimine kadar uzanan standart bir kitaplık ve API kümesi içerir.

# .Net Derleme Mimarisi



# CLR (Common Language Runtime)

- ❑ CLR, .NET ortamında uygulamaları çalıştıran, güvenli kılan ve yöneten motor gibidir.
- ❑ CLR, işletim sistemi ve .NET uygulamaları arasındaki katman olarak çalışır.
- ❑ Uygulama içerisindeki nesnelerin yönetimi, bellek yönetimi, hata yakalama ve bir çok işlem frameworkün bu motoru tarafından yürütülür.

# .NET & C# Versiyon Karşılaştırması

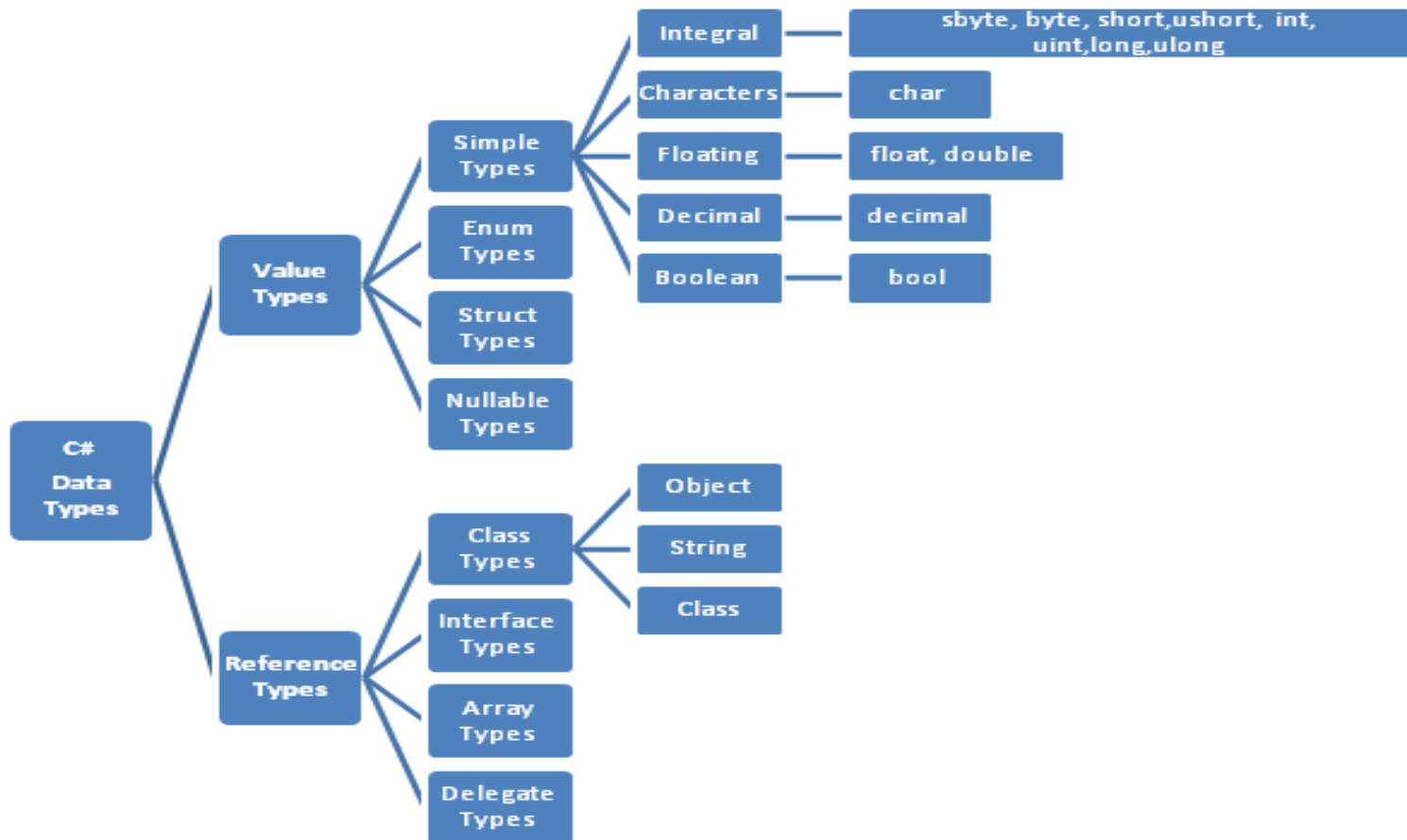
Target	Version	C# language version default
.NET	8.x	C# 12
.NET	7.x	C# 11
.NET	6.x	C# 10
.NET	5.x	C# 9.0
.NET Core	3.x	C# 8.0
.NET Core	2.x	C# 7.3
.NET Standard	2.1	C# 8.0
.NET Standard	2.0	C# 7.3
.NET Standard	1.x	C# 7.3
.NET Framework	all	C# 7.3

# C# Veri Tipleri ve Değişkenler

# Veri Tipleri

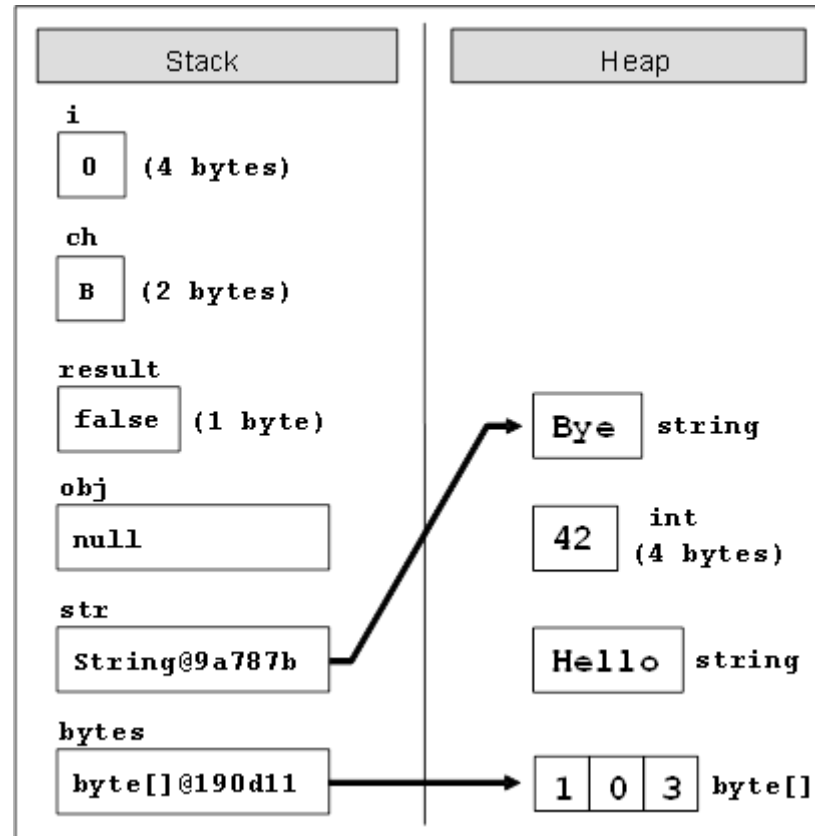
Değişken	Boyut	Açıklama	Class/Struct
byte	8 bit	0 ila 255	System.Byte
short	16 bit	-32,768 ila 32,767	System.Sbyte
int	32 bit	-+2 147 483 647	System.Int32
long	64 bit	-+9 223 372 036 854 775 807	System.Int64
float	32 bit	1.5E-45 ila 3.4E+38	System.Single
double	64 bit	5E-324 ila 1.7E+308	System.Double
decimal	128 bit	1E-28 ila 7.9E+28	System.Decimal
char	16 bit	Tek haneli unicode karakter alır	System.Char
string	Sbt dğl	Yaklaşık 2 milyar unicode karakter	System.String
bool	16 bit	true yada false değerini alır	System.Boolean
DateTime	32 bit	9999 yılına kadar olan tarih ve saat bilgisi	System.DateTime

# Değer ve Referans Tipler





# Bellek Yönetimi



# Veri Tipleri ile Çalışmak

- Tip Dönüşümleri
  - Implicit Tip Dönüşümü (bilinçsiz, kapalı, örtük)
  - Explicit Tip Dönüşümü (bilinçli, açık, belirtik)
- Aritmetik Taşma Deyimleri (checked,unchecked)
- Nullable Tipler
- Object Veri Tipi
- var ile Değişken Tanımlama

# Kontrol ve Döngü Deyimleri

- ❑ Koşullu (Conditional) İfadeler (if,else,if-else)
- ❑ Seçim (Selection) İfadeleri (switch-case)
- ❑ Yineleme (Iteration) İfadeleri (while, do while, for, foreach)
- ❑ Atlama (Jump) İfadeleri (return, break, continue)

# Diziler (Arrays)

# Collection Objects

- ArrayList
- Hashtable
- SortedList
- Stack
- Queue

# Generic Collections

- List
- Dictionary
- SortedList
- Stack
- Queue

# Object Oriented Programing

# OOP Nedir?

- ❑ Nesne yönelimli programlama, veri ve kod içerebilen nesnelere kavramına dayanan bir programlama paradigmasıdır. OOP’de bilgisayar programları, birbirleriyle etkileşime giren nesnelere oluşturularak tasarlanır.
- ❑ Nesne yönelimli programlama, uygulamaları esnek (değiştirilmesi veya yeni özellikler eklenmesi kolay), tekrar kullanılabilir, hata ayıklaması ve test edilmesi kolay hale getirir.



# Sınıf & Sınıf Üyeleri

# OOP'de Sınıflar

- ❑ Sınıflar, diğer tiplerin, yöntemlerin ve etkinliklerin değişkenlerini bir arada gruplayarak kendi özel veri tiplerinizi oluşturmanızı sağlayan bir yapıdır. Bir tipin verilerini ve davranışını tanımlar. Sınıflar, statik olarak bildirilmezse, değişkene atanmış nesnelere veya örnekler oluşturularak kullanılır.
- ❑ Birbirinin yerine kullanılmalarına rağmen, sınıf ve nesne farklı şeylerdir. Sınıf, bir nesnenin türünü tanımlar, ancak nesnenin kendisi değildir. Nesne, bir sınıfı temel alan somut bir varlıktır ve bazen bir sınıfın örneği olarak adlandırılır.

# Sınıf Üyeleri

- ❑ Methods
  - optional
  - params
  - ref & out
  
- ❑ Recursive Functions
  
- ❑ Property & Indexer
  
- ❑ Constructor & Destructor

# Statik Sınıf & Üyeler

- Static Class
- Static Constructor
- Static Methods
- Static Properties
- Static Fields

# Anonim Tipler

# OOP'nin Temel Prensipleri

# Inheritance (Kalıtım)

- ❑ Kalıtım, belirli bir işlevsellik (veri ve davranış) sağlayan bir temel sınıf tanımlamanıza ve bu işlevselliği devralan veya geçersiz kılan türetilmiş sınıflar tanımlamanıza olanak tanıyan nesne yönelimli programlama dillerinin bir özelliğidir.
- ❑ Üyeleri miras alınan sınıfa temel (base) sınıf, bu üyeleri miras alan sınıfa ise türetilmiş (derived) sınıf denir. Türetilmiş bir sınıfın doğrudan bir temel sınıfı olabilir

# Abstraction (Soyutlama)

- ❑ Soyutlama, gereksiz detayları göz ardı ederek ve bir amaç için neyin önemli olduğuna odaklanarak sınıfları tanımlama sürecidir. Belirli bir uygulama için neyin uygun olduğuna odaklanmak için gereksiz ayrıntıların soyutlanması ile ilgilidir.
- ❑ Soyutlama, bir sözleşmeyi tanımlayan ancak sözleşmenin uygulanmasını sağlamayan bir türdür. C#'ta soyutlama, abstract class (soyut sınıflar) ve interface (arayüzler) ile yapılır.



# Encapsulation (Sarmalama)

- ❑ Yönelimli programlama dilinde kapsülleme, kod parçasını daha iyi yönetmek ve korunan koddaki değişiklik nedeniyle programın diğer bölümleri üzerinde en az etkiye sahip olmak veya hiç etkiye sahip olmamak için değişkenleri, işlevleri sınıf dışından korumayı zorunlu kılan bir kavramdır

# Polymorphism (Çok Biçimlilik)

- ❑ Polimorfizm, miras alınan özellikleri veya yöntemleri birden fazla farklı şekillerde uygulama becerisidir.

# Advanced OOP

- Inheritance
- this, base keyword
- sealed keyword
- const and readonly keyword
- Interface & Abstract
- abstract, virtual, override keyword
- Extension Methods
- Enums

# Advanced OOP



# Class vs Struct

# Attributes (Nitelikler)

- Attribute, programınızdaki sınıflar, metodlar, yapılar, numaralandırıcılar vb. gibi çeşitli öğelerin davranışları hakkında çalışma zamanına bilgi aktarmak için kullanılan bir etikettir. Nitelikler, kullanıldığı öğenin üzerine/önüne yerleştirilen köşeli parantezlerle [xxxxxx] gösterilir.

# Reflection (Yansıma)

- ❑ Reflection, hakkında bilgi sahibi olmadığınız programatik nesnelere ilgili, çalışma zamanında (run-time) bilgi alabilmemize olanak tanıyan bir tekniktir.

# Delegate

- ❑ Delege, belirli parametrelere ve dönüş türüne sahip metodlara yapılan başvuruları temsil eden bir veri tipidir. Bir temsilcinin örneğini oluşturduğunuzda, onun örneğini uyumlu bir imza ve dönüş türüne sahip herhangi bir metodla ilişkilendirebilirsiniz.
- ❑ Delegeler, metodları diğer metodlara parametre olarak iletmek için kullanılır.





# Lambda Expressions



# Events

# Generics

- Generic Classes
- Generic Methods
- Generic Structs
- Generic Interfaces
- Generic Delegates
- Generic Constraints (where clauses)

# Summary

Salih DEMİROĞ

[salihdemirog@gmail.com](mailto:salihdemirog@gmail.com)

<http://www.yazilimdevi.com>