



C#

- Bir programlama dilini öğrenmek ana dil öğrenmeye benzer.
- Sizden beklenen anlatılanları anında anlamamanız değil, bol tekrar ve taklit ile anlamaya çaba göstermenizdir.
- Algoritma ve programlamanın temel kurallarını bilen çok kolay programlama öğrenebilir.

C#

- C# basit, modern, nesne yönelimli, tip korumalı, C ve C++ dillerinden türetilmiş bir dildir.
- Avantajları
 - Etkileşimli geliştirme aracıdır.
 - Görsel tasarım imkânı sağlar.
 - Derlenen bir programdır.
 - Debug yapılabilir.
- Nitelikleri
 - Basit ve tip korumalı
 - Önceki sürümlerle bağlantı
 - Nesne yönelimli ve esnek

C# Neler Yapılabilir?

- Masa üstü ve web uygulamaları yazılabilir.
- Büyük ölçekli otomasyonlar tasarlanabilir.
- Konsol çalışmaları için uygundur.
- Büyük kullanıcı kitlesi ile sorunlara kolay çözüm bulunabilir.
- Grafik ara birimini kullanan, multiprocessing işlemler yaptırılabilir.
- Bla bla bla...

C# Temelleri

```
namespace Muhendislik {  
    class Biomedical  
    {  
        static void Main()  
        {  
            System.Console.WriteLine("Merhaba Dünya");  
        }  
    }  
}
```

- Her satırdan sonra ; işareti eklenir.
- Kelime aralarında boşluklar olur.
- Tab, satırbaşları, boş satırlar, birden fazla boşluklar tek boşluk gibi algılanır.

C# Temelleri

- Namespace: Ad uzayı, container
 - Birbiriyle ilişkili olan sınıfları, yöntemleri barındıran soyut bir ambar.
 - Aynı namespace içerisinde aynı sınıftan iki tane olamaz.
 - System ad uzayı gibi
- Class: Sınıflar, objects
 - OOP doğası gereği her değişken, her fonksiyon sınıf içine tanımlanır.
 - Bu fonksiyonlar ve değişkenler o class a ait member
 - Aynı class içinde aynı fonksiyondan iki tane olamaz.
 - Console gibi
- Fonksiyonlar
 - Sınıfların içerisinde doğrudan çalışan yada sonuç döndüren yöntemlerdir.
 - Writeline gibi

C# Temelleri

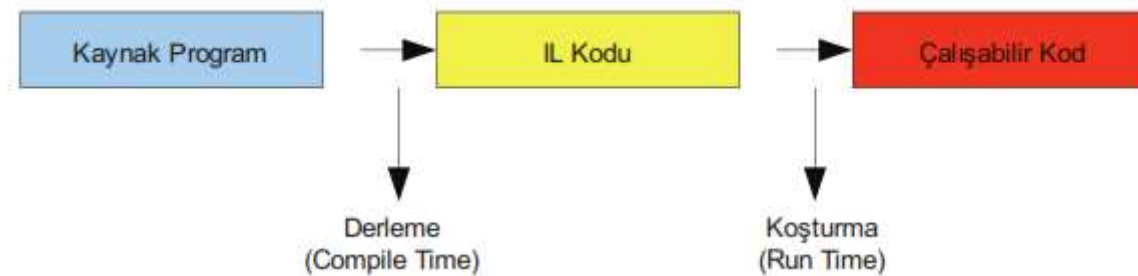
- Blocks: {}
 - C# iç içe bloklardan oluşur.
 - Class, sub-class, functions, if else, for while ...
 - Blok parantezlerinin sırası önemlidir ve hiyerarşiyi belirtir.
- Açıklama ekleme
 - Tek satır açıklama eklenecekse : //
 - Birden fazla satır veya blok açıklama olarak belirlenecekse
 - /* */

C# Bellek

- Oluşturulan tüm namespace, class, method, member hepsi bellekte tutulur. 3 tür bellek vardır.
 - CPU içindeki registerlar
 - Erişilmesi en hızlı belleklerdir. Ancak kısıtlıdır.
 - Kayıt ortamları
 - Kalıcı kayıtların yapıldığı yerdir. En yavaş birimdir.
 - RAM
 - Stack ve heap olarak ikiye ayrılır. Programa ait öğelerin tutulduğu yerdir.
 - Stack: Hızlıdır. Lifo ile çalışır. Değer tipleri burada tutulur.
 - Heap: Daha yavaş ve büyüktür. Karmaşık veriler ve referans tipli değişkenler burada tutulur.
- Garbage Collector: Nesneyle iş bitince bellekte ayrılan yer otomatik olarak boşaltılır.

Visual Studio

- 3 farklı sürümü vardır.
 - Community, Pro, Enterprise
 - Express Sürümü
- .NET Framework
 - Program geliştirme platformudur ve 2000 yılında ilk sürümü ortaya çıkmıştır.
 - Amaç işletim sistemine ve donanıma bağılılığı ortadan kaldırmaktır. Javadaki VM ye benzetilebilir.



Veri Tipleri ve Değişkenler

- Neden veri tipleri
 - Program yazarken işlem yapılan veriler ve tipleri farklıdır. Bu tipler için bellekte farklı yerler ayrılır.
- Değişkenler
 - Teknik olarak bellekte veri tipine ait değerlerin girilebileceği bir adres
 - C# dilinde global değişken yoktur.
 - Her değişken bir sınıf içinde yada sınıf içinde bir blokta tanımlanır.
 - Her değişkene bellekte tipine göre bir yer ayrılır.
 - Her değişkene bir ad verilir ve bu ad kullanılarak değişkene erişilir, değeri okunabilir, değeri değiştirilir.
 - = operatörü ile değer verilir.

Veri Tipleri ve Değişkenler

- Değişkenler

- C# tip korumalı bir dildir ve kuralları keskin hatlar ile belirlenmiştir.
- Bir değişkenin tanımlanabilmesi için veri tipinin kesin belirlenmesi gerekir.
- Dönüşüm yapılmadığı sürece o veri tipinin dışına çıkılmaz.
- Bir değerleri olmalıdır veya mutlak suretle null olarak tanımlanmalıdır.
- Tanımlama yaparken büyük-küçük harf ayrımı vardır.
- Değişken adları rakamla başlayamaz.
- Class, namespace ve kontrol isimleri gibi program tarafından kullanılan isimler verilemez.
- Aynı kod bloğu içerisinde aynı isimden birden fazla değişken tanımlanamaz.
- Özel karakter içermez örneğin /, *, -, +.
- Boşluk kullanılamaz.
- Türkçe karakter kullanılmamalıdır. Ş, ğ, ü, ö, ç, ı gibi...
- Özel sözcükler (if, else, random, vb.) kullanılmaz.
- (Benim tercihim deve notasyonu)

Tür	Adı	Boyut-Tür	Açıklama	Max ve Min aralık yada değeri
Değer Tipleri	sbyte	1 Byte	işaretli tamsayı	-128 ile 127
	short	2 Byte	işaretli tamsayı	-32.768 ile 32.767
	int	4 Byte	işaretli tamsayı	-2.147.483.648 ile 2.147.483.647
	long	8 Byte	işaretli tamsayı	-9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807
	byte	1 Byte	işaretsiz tamsayı	0 ile 255
	ushort	2 Byte	işaretsiz tamsayı	0 ile 65.535
	uint	4 Byte	işaretsiz tamsayı	0 ile 4.294.967.295
	ulong	8 Byte	işaretsiz tamsayı	0 ile 18.446.744.073.709.551.615
	float	4 Byte	tek kayan sayı	+yada $-1,5 \cdot 10^{-45}$: + ya da $-3,4 \cdot 10^{38}$
	double	8 Byte	çift kayan sayı	+yada $-5 \cdot 10^{-324}$: + ya da $-1,7 \cdot 10^{308}$
	decimal	16 Byte	ondalıklı sayı	+yada $-1,5 \cdot 10^{-28}$: + ya da $-7,9 \cdot 10^{28}$
	bool	1 Byte	— —	true ya da false
Referans Tipleri	char	2 Byte	Unicode karakterler	A b c
	string	Değişken	Karakter Katarı	— — —
	Object	Bütün veri tiplerinin türediği temel sınıf		

Const

- Const: Sabitlerdir ve tanımlanırken değer verilmesi gerekir.
 - `const int kdv=18;`

Operand

- Programlama dillerinde tek başlarına herhangi bir anlamı olmayan ancak programın işleyişine katkıda bulunan karakter ya da karakter topluluklarına operatör denir.
- Operatörlerin etki ettikleri sabit ya da değişkenlere "operand" denir.
- $a+b$, $a*b$ gibi..
- C# da bazı operatörler 1,2 ve 3 operand alabilir.

Aritmetik Operatörler

<i>Operatör</i>	<i>Açıklama</i>
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Modulo
++	1 artırma
--	1 eksiltme

Atama Operatörleri

Operatör	Açıklama
=	Sağdaki değeri soldaki değişkene atar.
+=	Soldakine sağdakini ekler, sonucu soldakine atar.
-=	Soldakinden sağdakini çıkarır, sonucu soldakine atar.
*=	Soldakini sağdaki ile çarpar, sonucu soldakine atar.
/=	Soldakini sağdakine böler, sonucu soldakine atar.
%=	Soldaki ile sağdakinin modula işleminin sonucunu soldakine atar.

İlişkisel Operatörler

Operatör	Açıklama
==	Eşit mi?
!=	Eşit-değil mi?
>	Büyük mü?
<	Küçük mü?
>=	Büyük veya eşit mi?
<=	Küçük veya eşit mi?

Mantıksal Operatörler

&&	Logical AND
	Logical OR

Diğer Operatörler

.	Nesne öğelerine erişim	(member access for objects)
[]	Array indisleme	(indexing operator used in arrays and collections)
()	Veri dönüştürme operatörü	(cast operator)
?:	Koşullu deyim operatörü	(ternary operator)
!	Değil operatörü	(not operator)
+	String birleştirme operatörü	

Operatör Öncelikleri

Birinci öncelikliler	x++, x--
Tek operand alan operatörler	+, -, !, ~, ++x, --x, (Tür)x
Çarpma ve bölme	*, /, %
Toplama ve çıkarma	+, -
Kaydırma operatörleri	<<, >>
İlişkisel ve tür testi operatörleri	<, >, <=, >=, is, as
Eşitlik operatörü	==, !=
Bitisel Ve (AND)	&
Bitisel Özel Veya (XOR)	^
Bitisel Veya (OR)	
Mantıksal Ve	&&
Mantıksal Veya	
Koşul operatörü	?:
Atama ve işlemli atama operatörleri	=, *=, /=, %=, +=, -=, <<=, >>=, &=, ^=, =



Veri giriři ve ıkıřı

- ReadLine, WriteLine komutları

Kaynak

- T. Karaçay ve A. Karaçay, *Hiç Bilmeyenler İçin C# ile Programlamaya Giriş*, 2. Baskı. Ankara: Seçkin Yayıncılık, 2016.