

## CONSOLE EKRANI

### CONSOLE EKRANI

- ✓ *Programlama*
- ✓ *Algoritma*
- ✓ *Hello World*
- ✓ *Değişkenler*

### KAZANIMLAR

- ✓ *Cmd Ekranı kullanımı öğrenir.*
- ✓ *Algoritma yapısını kavrar ve öğrenir.*



## ALGORİTMA

Bir sorunu / problemi çözmek veya belirli bir amaca ulaşmak için gerekli olan sıralı mantıksal adımların tümüne **algoritma** denir. Bir sorunun çözümü için izlenecek yolun tanımıdır. Kısaca algoritma mevcut bilgilerden istenilenlere erişme yöntemidir. Diğer bir ifadeyle algoritma, bir problemin mantıksal çözümünün adım adım nasıl gerçekleştirileceğinin sözlü ifadesidir.

## PROGRAMLAMA

Program, problemin çözümünde yapılması gereken işlemler bütünüdür. Kod karşılığıdır. Algoritmaların program haline getirilmesi için programlama dilleri kullanılır. Programlama dilleri kullanılarak yazılımlar geliştirilir. Programlamanın en önemli kısmı problemin çözümü için algoritma hazırlayabilmek / geliştirmektir

### Algoritmanın temel özellikleri şunlardır

#### Kesinlik

Algoritma içindeki adımlar herkes tarafından aynı şekilde anlaşılabilir olmalı, farklı anlamlara gelebilecek bulanık ifadeler içermemelidir.

#### Sıralı Olma

Her algoritma için bir başlangıç durumu söz konusudur. Çözüm, bu başlangıç durumu göz önünde bulundurularak gerçekleştirilir. Adımların hangi sırada gerçekleştirileceği çok önemlidir ve net bir şekilde belirtilmelidir.

#### Sonluluk

Algoritma sonlu sayıda adımdan oluşmalı, sınırlı bir zaman diliminde tamamlanmalıdır. Her algoritmanın bir son noktası, bitişi olmalıdır.

Şimdi algoritmanın tanımını ve özelliklerini günlük yaşamdan basit bir örnekle pekiştirelim. Diyelim ki araç trafiği olan bir yolda karşıya geçmek istiyoruz. Bu durumda çözmemiz gereken problem (buna yapılması gereken iş de diyebiliriz) karşıya geçmektir. O zaman bu problemin çözümü için bir yol bulmamız gerekiyor.

Şöyle bir ifadeye ne dersiniz?

- ✓ Önce araba var mı kontrol et, ardından yürü!

Bu ifade özünde doğrudur. Ancak yeterince açık değildir. Bunu hayatında ilk defa karşıdan karşıya geçecek birine söylersek, kim bilir nasıl anlar?

- ✓ Yolun kenarına park etmiş araba var mı?

Evet var. O zaman kaldırımdan yürüyeyim.



✓ *Yolun kenarına park etmiş araba var mı?*

*Hayır yok. O zaman yolun ortasından yola paralel yürüyeyim.*

✓ *Sol taraftan gelen araba var mı? Hayır yok. O zaman sola bakarak karşıya yürüyeyim*

*Bu böylece sürüp gider. Evet, biz yetişkin ve eğitimli insanlar "Önce araba var mı kontrol et, ardından yürü!" ifadesinden verilmek istenen mesajı açıkça alırız. Ancak bilgisayarlar öyle değildir. Hiçbir şey bilmezler. Ama onlara detaylı olarak verdiğimiz bütün emirleri yerine getirebilirler. İyi tarif edersek, herşeyi hızlıca anlayıp kolayca uygulayabilirler.*

*Şimdi problemi daha net ve kesin ifadelerle çözmeye çalışalım. Ama nereden başlayacağız? Evde miyiz? Trafik ışıklarının yanında mıyız? Yaya geçidinin önünde miyiz? Bu gibi durumlar önemlidir.*

*Bu örnekte yolun kenarındayız ve trafik ışığı yok. O zaman şöyle yapalım:*

*Önce yürüyelim, sonra sola ve sağa bakalım (tabi eğer ezilmeden karşıya geçebildiysek). Olur mu? İşimiz şansa kalır. Eğer araba yoksa olur. Araba varsa ezilme ihtimalimiz var. O zaman adımları gerçekleştirme sırası da önemli. Hadi tekrar deneyelim...*

*A1 Yolun kenarından başla*

*A2 Önce sola bakmalı, gelen araba yoksa ya da karşıya geçebileceğimiz kadar uzak bir mesafedeyse sağa bakılmalı.*

*A3 Sağ tarafta da araç yoksa ya da gelen araç yeterince uzak mesafedeyse karşıya yürünmeli.*

*A4 Dur*

*Farklı bir örnek deneyelim mi? Girilen bir sayının pozitif mi negatif mi olduğunu ekrana yazdıran algoritma..*

**A1 Başla**

**A2 Sayıyı gir(sayı)**

**A3 Eğer sayı=0 ise Adım 3 e geri dön**

**A4 Eğer sayı<0 ise ekrana , "girilen sayı negatiftir" yaz ve Adım 6 ya git**

**A5 Eğer sayı > 0 ise ekrana, "girilen sayı pozitifdir" yaz**

**A6 Dur**



## C# Nedir?

C# dilinin yaratıcısı olan Microsoft onu söyle tanıtıyor; C# basit,modern,nesne yönelimli, C ile C++ dillerinden türetilmiş bir dildir.

### C# dilinin başlıca avantajları şunlardır;

- ✓ Etkileşimli geliştirme aracıdır
- ✓ Windows ve Web uygulamaları için görsel tasarım sağlar.
- ✓ Derlenen bir programdır.(Scripting dili değildir)
- ✓ Debug yapar.

### C# dilinin başlıca uygulamaları:

**Class Library:** Başka uygulamalarda kullanılacak kütüphane sınıfları yaratır.

**Console Application:** Satır komutu arayüzü için görsel c# uygulamaları yaratır

**Asp.Net Web Application:** Web kullanıcı arayüzü için görsel C#uygulamaları yaratır.

**Asp.Net Web Service:** XML Web servisleri yaratır.

**Asp.Net Mobile Application:** PDA,cep telefonları gibi taşınabilir cihazlar için uygulama programları yaratır.

## PROGRAMLAMANIN EVRELERİ

Kullanılan dil ne olursa olsun,programlama eylemi şu evrelerden oluşur:

### 1.Kaynak programı yazma

Herhangi bir programlama dilinde text olarak yazılır.Kaynak program,kullandığı dilin sözdizimine(syntax) uymalıdır.

### 2.Derleme

Kaynak programı bilgisayarın anlayacağı bir ara dile dönüştürür.Ara dil kullanılan dile bağlıdır.Örneğin, C dilinde obj kodları,java dilinde bytecode,C# dilinde IL,vb adlar alır.

### 3.Aradil kodları yürütülebilir makine diline dönüştürülür

Programın yürütülebilir(çalışabilir,koşabilir-executable) olması için,kullanılan İşletim Sisteminin anladığı dile (makine dili) dönüşmesi gerekir.Windows İşletim Sisteminde yürütülebilir programların dosya adları .exe uzantısını alır.Örneğin,



*Deneme.cs*

*Adlı bir c# kaynak programı bütün aşamalardan geçip makine diline dönüşünce*

*Deneme.exe*

*Adını alır.Bu programı koşturmak için*

*Deneme*

*Yazıp Enter tuşuna basmak yetecektir.Aynı işi,Windows'un görsel arayüzünde yapmak için,Windows Explorer ile Deneme.exe dosyasını bulup üstüne tıklamak yetecektir.*

## **PROGRAM YAZMAYA GİRİŞ**

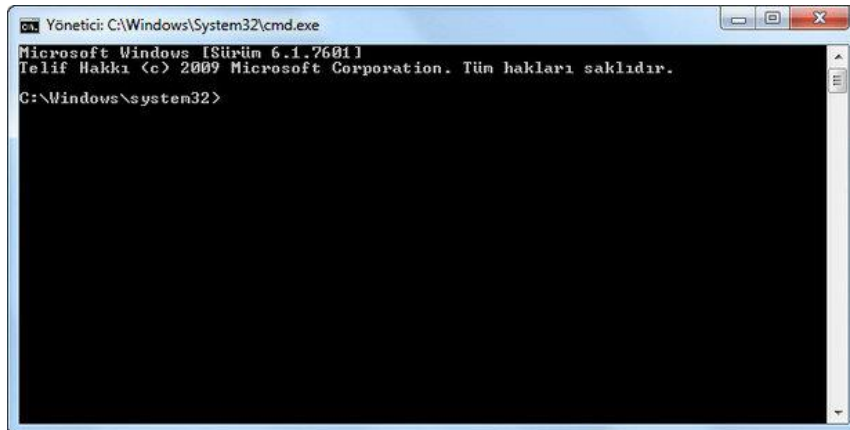
*İlk derslerimizde hiçbir görsel arayüz kullanmadan,işlerimizi doğrudan doğruya işletim sisteminin komutlarıyla göreceğiz.Böyle yapabilmek için,Windows'un Komut İstemi programını açacak ve Dos(Disk Operating System-Windows İşletim Sisteminin Anahtarları-) komutları yazacağız.Komut istemi ni aşağıdaki iki yoldan birisiyle açabilirsiniz:*

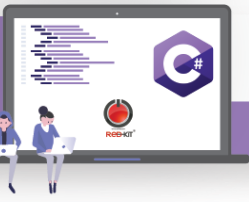
*Başlat->Çalıştır sekmelerine basınca açılan pencerede aç----- kutucuğuna cmd yazıp entere basınız.*

*Başlat->Tüm Programlar->Donatılar sekmelerinden sonra açılan alt pencereden Komut işlemi'ne tıklayınız.*

*Yada;Mouse ile ekranın el alt sol köşesine gelip sağ tıklayın veya Windows tuşu + x tuşuna basın.Çıkan menüden Komut İstemi veya Komut İstemi (Yönetici) seçin.Dos ekranı açılacaktır.*

*Bunlardan birisini yaptığımız zaman,ekranda siyah zemin üzerine beyaz yazılar yazılan Komut İstemi yani söyle bir ekran çıkacaktır..*





### 💡 **İpucu**

C# kaynak programlarının dosya adlarının uzantısı hiç olmayabileceği gibi, istenen bir uzanti da yazılabilir. Ama, bir dosyanın C# kaynak dosyası olduğunu ilk bakışta anlayabilmemiz için .cs uzantısını yazmayı alışkanlık edinelim.

**Örnek1:** Program yazmaya başlamanın ilk örneği olan ekrana Hello World yazalım.

```
Console.WriteLine("Hello World");
```

### 💡 **İpucu**

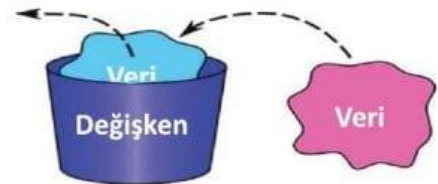
Ekrana bir şey yazdırmak için Write() yada WriteLine() fonksiyonları kullanılır. İkisi de aynı işi yapar, ancak birincisi isteneni yazdıktan sonra, yazdığı şeyin sonunda bekler, ikincisi istenineni yazdıktan sonra alttaki satırın başına geçer. Birlikte ayırmını görelim yazalım

```
Console.WriteLine("merhaba benim adım Beyza");  
Console.Write("c# ");  
Console.Write("öğreniyorum");
```

## DEĞİŞKENLER

Program yazarken çeşitli değişkenler kullanılır. Programda hangi değişkenlerin kullanılacağı değişken atama işlemiyle yapılır. Değişken atandıktan sonra ona bir ilk değer verilir, program akışı esnasında değişkenin değeri değiştirilebilir, var olan değeri silinebilir, bir değişken içindeki bir değer başka bir değişkene atanabilir. Bir değişkenin yaşam döngüsünü aşağıdaki şekilde ifade edebiliriz.

- ✓ Değişken atanır.
- ✓ Değişkene ilk değeri verilir.
- ✓ Değişken çağrılır, gerektiğinde içeriği değiştirilir.





*Pekii değişken tanımlarken nelere dikkat etmeliyiz? Hadi gelin birlikte inceleyelim...*

- ✓ Bir değeri 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.

### **Değişken Tanımlama**

Değişken tanımlamaları şu şekilde yapılmaktadır.

(Değişken Türü) ( Değişken Adı) = (Değeri)

**Örnek2:** Şimdi de ikinci örneğimizi birlikte yapalım

```
using System;
namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            string ad;
            Console.WriteLine("Adınız nedir?");
            ad = Console.ReadLine();
            Console.WriteLine("Merhaba" + ad);
        }
    }
}
```





