

# *Döngüler(Loops)*

*While Döngüsü*  
*do While Döngüsü*

*Arif GÜNEL*

# Özet

- *Şart sağlanana kadar çalışan*
- *Belli kodları defalarca çalıştıran*

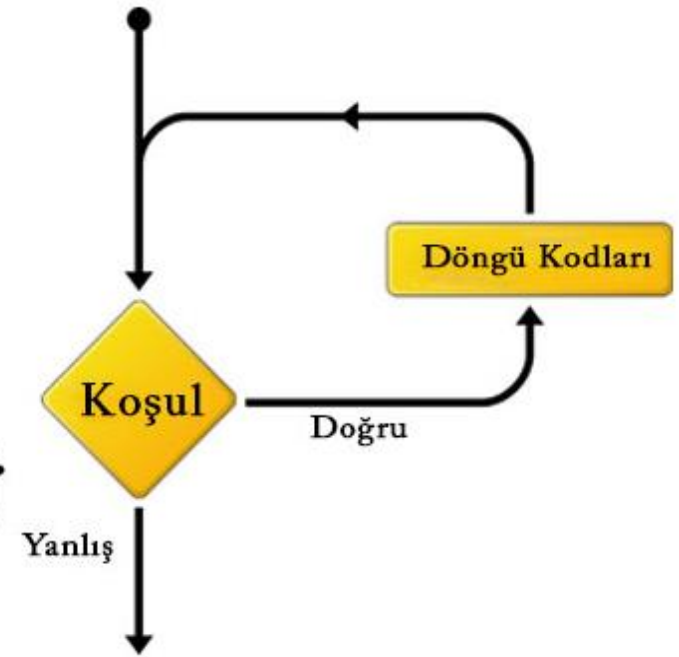
*C# Programlama dilinde 4 adet döngü vardır. Bunlar;*

- *Belli sayıda döngü oluşturan*
  - *For*
  - *foreach*
- *Şartlı döngü oluşturan*
  - *While*
  - *Do while*

# While Döngüsü

- *While döngüsü For döngüsü gibi belli sayıda döngü oluşmasını sağlar. While döngüsünün yazımı for kadar karmaşık değildir.*

```
While(şart-koşul)  
{  
  deyim  
}
```



## *For döngüsünden farkları;*

- *Değişkeni önceden tanımlamak gerekiyor. (int a gibi)*
  - *Bu sayede okuma rahat bir kod bloğu oluşuyor bir de önemli olarak değişkeni önce tanımladığımız için döngüye sokmadan o değişken üzerinde sınırsız işlem yapmamızı sağlıyor.*
- *Artım yada azaltma işlemini döngünün süslü parantezleri arası da yapıyoruz.*
  - *Bu artırma ya da azaltmayı yapmaz ise sonsuz döngü oluşur.*

- *While döngüsünde şart kısmı true yani şartı sağladığı zaman alt kısımdaki kodlar çalışır. While içindeki şart False olana kadar devam eder. Döngüyü sonlandırmak istersek break anahtar kelimesi kullanıyoruz. While döngüsü içerisinde elde edilmek istenen sonuç alındıysa döngünün diğer kısımlarına bakmadan döngüden çıkmak için Continue anahtar kelimesini kullanıyoruz.*

# Ornekk30.\_0\_While\_Oile9RakamYazma

*0 dan 10'a kadar olan rakamları ekrana yazdıran programı yazalım.*

- *İlk olarak değişkeni while döngüsünden önce tanımlıyoruz*

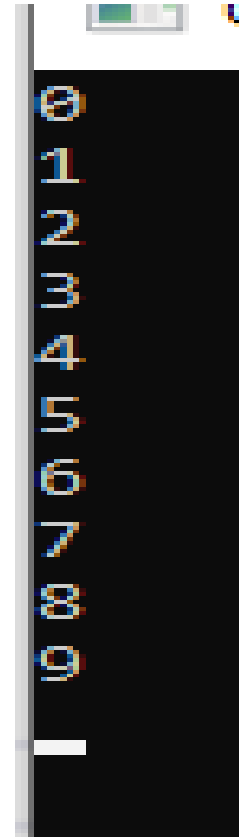
```
int sayac = 0; //while de değişken önce tanımlanıyor
```

- *Sonra while şartını tanımlıyoruz*

```
while (sayac < 10)
```

- *Sonra süslü parantezler arasına yapılması gereken işleri ve artım ya da azaltma işlerini yapıyoruz*

```
{  
    Console.WriteLine(sayac);  
    sayac++;  
}  
    Console.ReadLine();
```



# Ornekk31.0 While GeriSayma

- *Kullanıcının girdiği sayıdan geriye doğru 0 kadar yazan bir programı while şartı ile yazalım.*
- *İlk olarak kullanıcıdan değeri alalım ve değişkene atayalım*

```
Console.WriteLine("Lütfen bir sayı giriniz:");
```

```
int sayac = Convert.ToInt32(Console.ReadLine());
```

- *While döngüsü 0 kadar devam edeceği için -1 eşit olmadığı sürece döngü devam edecek.*

```
while (sayac != -1)  
{
```

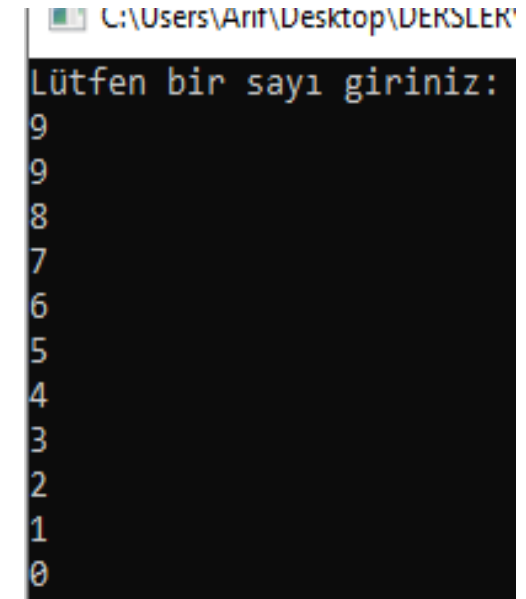


- *Ekrana değeri yazacak ve sayaç değerini 1 azaltacak*

```
{  
    Console.WriteLine(sayac);  
    sayac--;  
}  
    Console.ReadLine();
```

*Not: while döngünün şart kısmına mantıksal operatörler ile ve veya şeklinde kontroller eklenebilir.*

```
int sayi1=0;  
int sayi2=0;  
while (sayi1>12 || sayi2==25)  
{  
    if (true)  
    {  
    }  
}
```

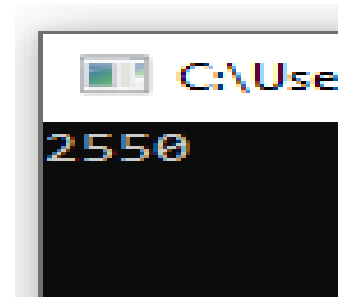


C:\Users\Arif\Desktop\DEKSLER'  
Lütfen bir sayı giriniz:  
9  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0

# Ornekk33.\_0\_While\_1ile100çiftSayiToplama

- 1'den 100'e kadar olan çift tamsayıların toplamını veren programı while döngüsü ile yapalım.

```
int sayac = 0;
int toplam = 0;
while (sayac<=100)
{
    toplam += sayac;
    sayac += 2;
}
Console.WriteLine(toplam);
Console.ReadLine();
```



## Ornekk34.0 While 1ile10TahminBulma

- *Bu örnekte sistemin 1 ile 10 arasında Random nesnesi sayesinde sistem rastgele sayı üretecek kullanıcı bu sayıyı tahmin etmeye çalışacak. Tahmini bulduğunda tahmini kaç denemede bulunduğunu ekrana yazacak.*
- *İlk olarak sistemin rastgele üreteceği sayıyı tutacak değişkeni ve tahmin etme sayısını tutacak değişkeni ve random nesnesinde bir değişken tanımlayalım.*

```
int sayiUret = 0;
```

```
int tahmminAdet = 1;
```

```
Random rnd = new Random();
```

```
sayiUret = rnd.Next(1, 10);
```

- *While ile sonsuz bir döngü oluşturup denemeler yaptıralım sürekli ve ekrana yeni tahmininizi girin yazsını yazdıralım ve girilen değeri “**tahmin**” değişkenine atayalım.*

```
while (true)

{

    Console.Write("{0} . deneme - Lütfen tahmin ettiğiniz değeri
giriniz : ", tahmminAdet);

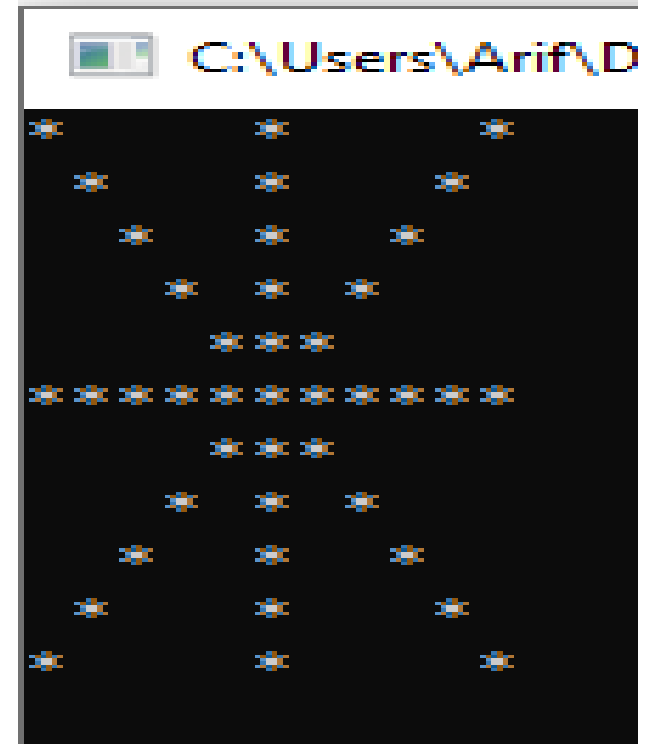
    int tahmin = Convert.ToInt32(Console.ReadLine());
```

- *If ile tahmin oluşturup üretilen sayı ile tahmini kontrol ettirelim. Tahmin doğru ise “break;” ile döngü kırılıp çıkılsın*

```
        if (tahmin == sayiUret)
        {
            Console.WriteLine("{0}. denemenizde değeri buldunuz Tebrikler !", tahmminAdet);
            break;
        }
        else
        {
            Console.WriteLine("{0} .denemeniz ,Tahmin edemediniz lütfen yeniden deneyin", tahmminAdet);
            tahmminAdet++;
        }
    }
    Console.ReadLine();
```

## Ornekk33. 0 While YildizBasma

*İç içe while döngü kullanımı for daki göre daha basit ve kullanımı kolaydır.*



- *İlk olarak iki adet değişken tanımlayalım while de kullanmak için*

```
int satir = 0;
```

```
int sutun = 0;
```

- *İç içe iki adet while döngüsü oluşturalım.İlk satır ikinci sutun değerini kontrol edecek.İlk satırın tüm sütunlarını kontrol edip bir alt satıra geçecek.*

```
while (satir <= 10)
```

```
{
```

```
    while (sutun <= 10)
```

```
    {
```

- *İf şartı ile yıldız yazması gereken yerleri belirleyeceğiz. Şart doğru ise yıldız yanlışı ise boşluk yazacak.*

```
if (satir == sutun || satir == 5 || sutun == 5 || satir + sutun == 10)
{
    Console.Write("*");
}
else
{
    Console.Write(" ");
}
```



- *While döngüsünün şart değişkenlerini artıracacağız.*

```
        sutun++;  
    }
```

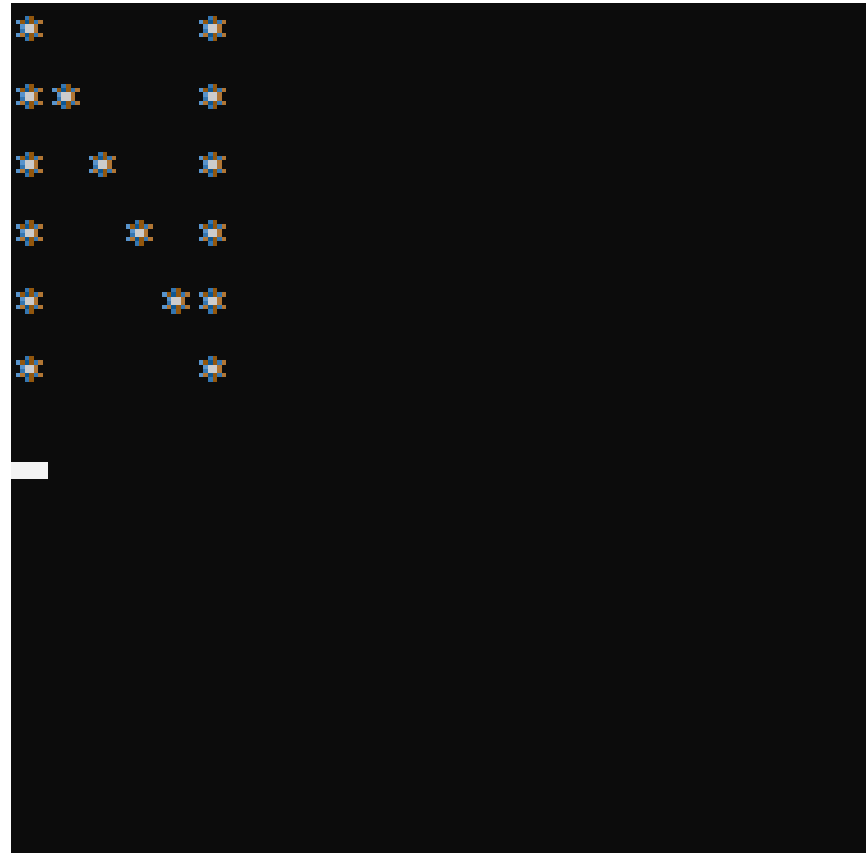
- *Bir de ilk satır bitince sütun değerini 0 a eşitliyoruz (sutun = 0;).Bir alt satıra geçmek için writeLine metodunu kullanacağız.*

```
        sutun = 0;  
        Console.WriteLine();  
        satir++;  
    }  
    Console.ReadLine();
```

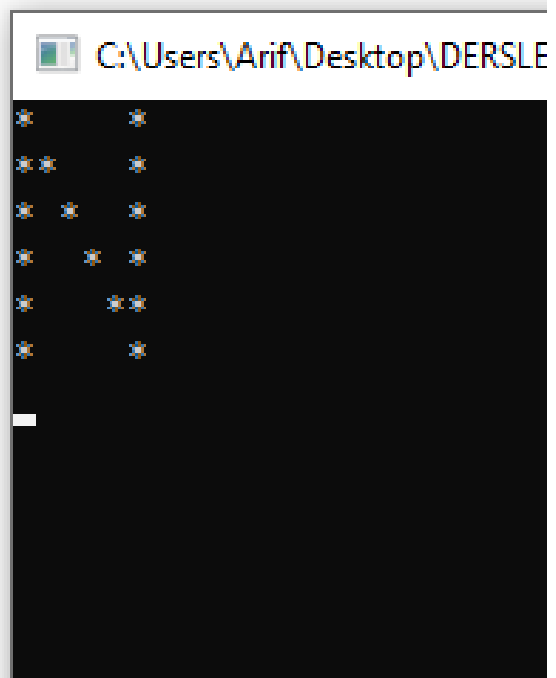


## Ornekk36. 0 While N Yazma

 C:\Users\Arif\Desktop\DESRLE



```
int satir = 0;
int sutun = 0;
while (satir <= 5)
{
    while (sutun <= 5)
    {
        if (satir == sutun || sutun == 5 || sutun == 0)
        {
            Console.Write("*");
        }
        else
        {
            Console.Write(" ");
        }
        sutun++;
    }
    sutun = 0;
    Console.WriteLine();
    satir++;
}
Console.ReadLine();
```



*Ödev: P harfini siz yapınız*

## *do (While); Döngüsü(;)*

```
do
{
    İşlemler;
} while(koşul);
```

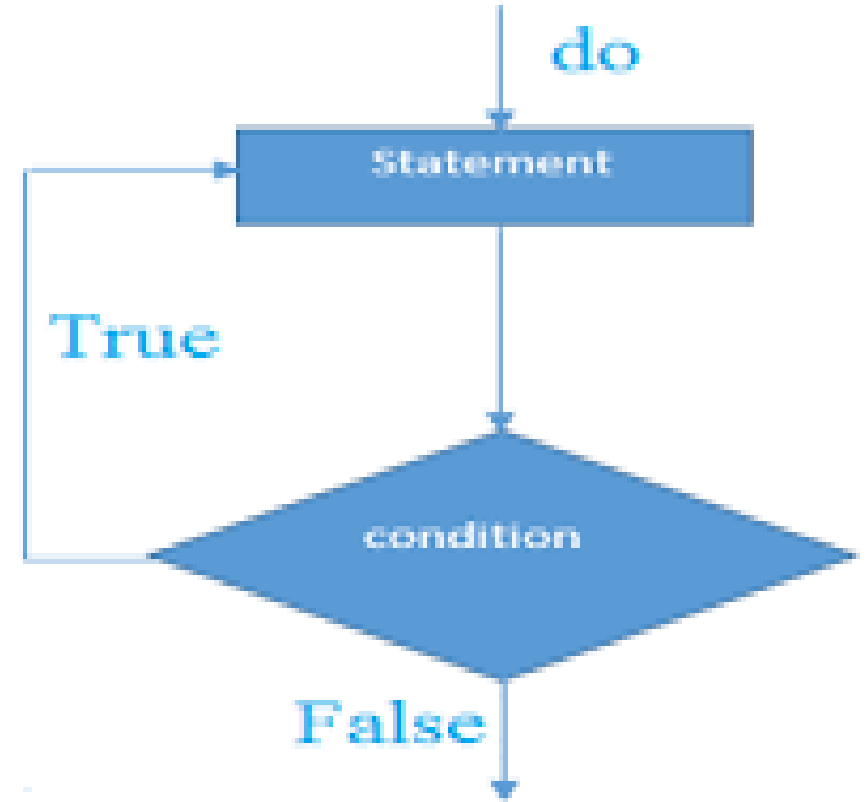
- *for ve while döngülerinde döngü bloklarının koşul sağlanmadığı takdirde hiç çalışmama durumu söz konusudur.*
- *Yani while(şart) yazdığımızda şart sağlamıyorsa hiç döngüye girmeden devam ediyordu. Bazı durumlarda döngü bloğunun en az bir kere çalışması istenir. Bu durumlarda do while döngüsü kullanılır.*

### **Do while kuralı:**

- **Şart ne olursa olsun döngü en az bir kere çalışacaktır. Şart doğru ise tekrar, yanlış ise döngüden çık.**

- *Çalışma mantığı ve yapısı while döngüsününkiyle aynıdır. Tek farklı while döngüsünde döngünün başında belirttiğimiz koşul ifadesi do-while döngüsünde sonda yer almaktadır. While döngüsünde koşul ifadesinin sağlanmadığı durumda program akışı döngüye girmeden devam eder, fakat do-while döngüsünde bu koşul ifade sonda yer aldığından programımız en az 1 kere do-while içerisindeki işlemleri uygular.*

- *While döngüsünden en önemli farkı şartın döngünün sonunda kontrol edilmesidir. Bu nedenle de do while döngüsü kullandığımızda döngü içerisindeki işlemler en az bir kere yapılacak demektir.*
- *Mesela parola kontrolü yapan bir programda en az bir kere parola girildiği için do while kullanılır. Eğer koşul doğru(true) ise tekrar başa geçip döngüye devam edilir, yanlış(false) ise döngüden çıkılır.*



## Ornekk37. 0 doWhile 1 Giris

- *Şart sağlanmadığı zaman döngü biter, doğru ise döner*


do

{

Console.WriteLine("selamlar doğru döner yanlış çıkar");

} while (1==2);

Console.ReadLine();



selamlar doğru döner yanlış çıkar



## Ornekk38. 0 doWhile 2 Parola Kontrolü

*Bu örnekte kullanıcının parolayı bulması isteniyor doğru bilene kadar yani yanlış değer girdikçe devam eden bir program.*

- İlk olarak kullanıcının gireceği parola değerini tutacak string türünden değişken tanımlayalım.

```
string parola = "" ;
```

- Do while kısmında parola girmesini isteyip girilen değeri değişkene atıyoruz

```
do
{
    Console.WriteLine("Parola giriniz:");
    parola = Console.ReadLine();
}
```

- While ile parolanın eşit olup olmadığını kontrol ediyoruz. Doğru ise programdan çıkıyoruz değilse tekrar do kısmı çalışıyor.

```
}while (parola != "duman");  
Console.WriteLine("Parolayı bildiniz.");  
Console.ReadLine();
```

```
Parola giriniz:  
arif  
Parola giriniz:  
duman  
Parolayı bildiniz.  
_
```

## Ornekk39. 0 doWhile 3 SayiTahmini

*Bu örnekte sistem 0 ile 100 arasında rastgele bir değer üretecek kullanıcı bu değeri bulmaya çalışacak.*

*Tahminleri sayıdan büyükse daha küçük değer giriniz, tahmin sayıdan küçükse daha büyük değer giriniz diye yönlendirecek.*

- İlk olarak kullanıcının tahminini ve sistemin otomatik olarak rastgele ürettiği sayıyı tutacak değişkenleri tanımlayalım.*

```
int sayi = 0;
```

```
int tahmin = 0;
```

- Random sınıfından bir rnd nesnesi üretiyoruz.Bu nesne 0 ile 100 arasında rastgele değer üretecek. New anahtar kelimesi random sınıfından türetilen nesne(rnd) için bellekte yer ayırır.*

```
Random rnd = new Random(); // rnd simli Random tipinde nesne tanımlıyoruz.
```

```
sayi = rnd.Next(100); // rnd sınıfının next metodu rasgele sayı üretir. Üretilen sayı maksimi 100 olacak şekilde ayarlandı.
```

- *Do kısmında sayıyı tahmin etmeye çalışacağız burada kullanıcının tahmini alıp if şartı ile kontrol edeceğiz.*

```
do
{
    Console.WriteLine("Tahmininizi giriniz:");
    tahmin = Convert.ToInt32(Console.ReadLine());
    if (tahmin < sayi)
    {
        Console.WriteLine("Daha büyük sayı gir");
    }
    else if (tahmin > sayi)
    {
        Console.WriteLine("Daha küçük bir sayı giriniz.");
    }
    else
    {
        Console.WriteLine("Sayıyı bildiniz. Üretilen sayı {0} idi.", sayi);
    }
} while (tahmin != sayi);
Console.ReadLine();
```

```
Tahmininizi giriniz:
35
Daha büyük sayı gir
Tahmininizi giriniz:
65
Daha büyük sayı gir
Tahmininizi giriniz:
80
Daha büyük sayı gir
Tahmininizi giriniz:
90
Daha büyük sayı gir
Tahmininizi giriniz:
95
Daha küçük bir sayı giriniz.
Tahmininizi giriniz:
93
Sayıyı bildiniz. Üretilen sayı 93 idi.
```

# Kaynakça

- <https://zenbilisim.tr.gg/Switch-Case.htm>
- <https://slideplayer.biz.tr/slide/3085379/> Emin BORANDAĞ-Celal Bayar Üni