

Döngüler(Loops)

for Dongüsü
Foreach Döngüsü

Arif GÜNEL

- TDK'ya göre Döngü, “Herhangi bir olayın birden fazla tekrarlanması olayına döngü denir.”
- Döngüler bir programlamada olmazsa olmaz yapılardır. Döngüler bir program içerisinde belirli işleri yani belli kodları **belli sayıda** ya da **istenen hedef(şart) sağlanana** kadar(şart gerçekleştiği sürece döngüyü yapan, şart sağlamadığında döngüden çıkan) yapmamızı sağlayan yapılardır.
- Sonsuz döngüler ya da belli sayıda döngüler oluşturulabilir. Bir işlemi defalarca(100-200 ..) yapılması istendiğinde ya da bir dizinin elemanları üzerinde sırayla işlemler yapmak istediğimizde, sonsuz bir döngü oluşturup parola kontrolünde(parola bilinmediği müddetçe döngünün içerisinde kalırız) döngüleri kullanırız.

Özet

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

C# Programlama dilinde 4 adet döngü vardır. Bunlar;

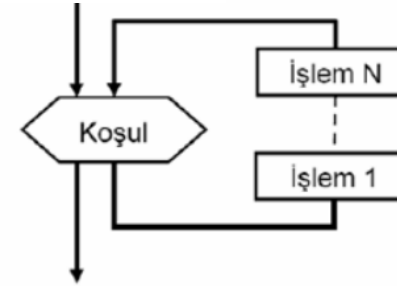
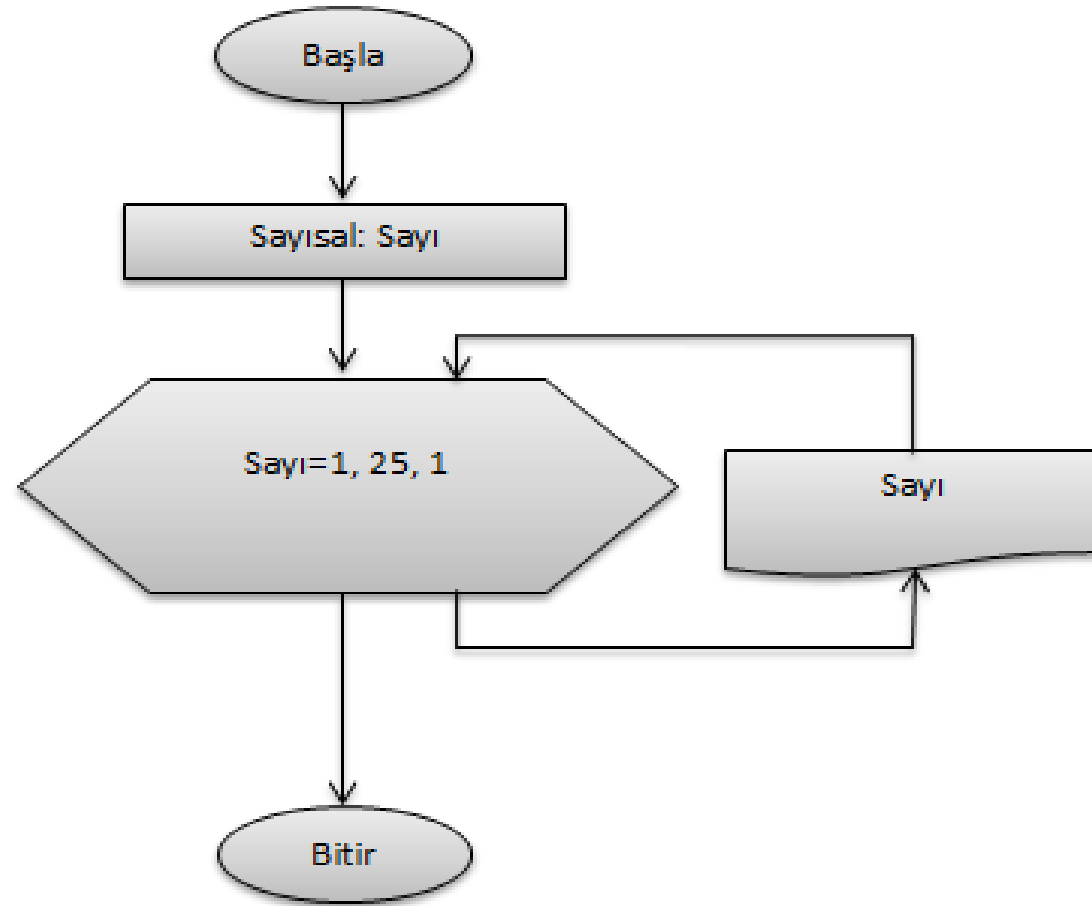
- *Belli sayıda döngü oluşturan*
 - *For*
 - *foreach*
- *Şartlı döngü oluşturan*
 - *While*
 - *Do while*

for Döngüsü

```
for(Başlangıç; Koşul; Artış)
{
    //koşul doğru olduğu sürece yapılacaklar
}
```

- *En sık kullanılan döngü diyebiliriz. Döngünün kaç defa döneceğini bildiğimiz durumlarda for döngüsü kullanılır.*
- *For döngüsünde döngü şartlarını üç ifade ile belirleriz.*
 - *ilk ifade de **başlangıç** değeri istenilen hedefin sabit olarak belirlenmesi, ilk değer atanması ve değişken tanımlanması yapılır,*
 - *ikinci döngü **koşulu**(**şart** sağlanana kadar kod bloğunu çalıştır),*
 - *üçüncü de ise **artış miktarını** (her kod bloğunun çalışmasından sonra sayacın artma miktarı) yazarak for döngüsünü oluştururuz.*

Ö12; 1 den 25'e kadar olan sayıları ekrana yazdıran programın akış şemasını çiziniz.



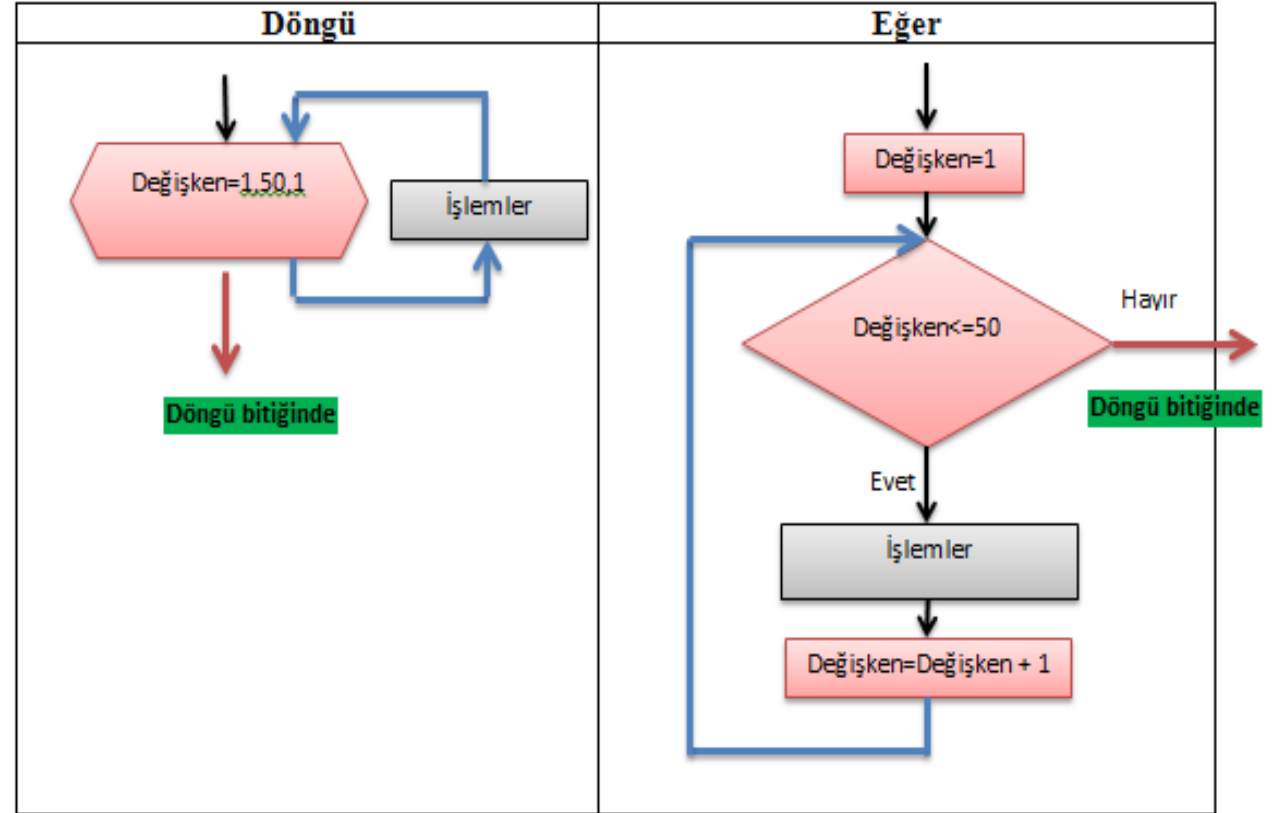
Koşul

Değişken=**Başlangıç değeri**,**Bitiş Değeri**,**Artım Miktarı**

- *Mesela 1'den 100'e kadar olan sayıların toplamını yapan bir for döngüsü yazalım*

```
int Toplam = 0;
```

```
for (int i = 1; i <= 100; i++)  
{  
    Toplam += i;  
}
```



0 ila 10 arasındaki rakamları yazdırma

- *For döngüleri tanımlanırken `int i=0` gibi bir döngü değişkeni tanımlanır; baktığımızda normal bir değişken olarak tanımlandığını görüyoruz*
- *Sonra `i<10` ile şartımızı yazıyoruz. Yani `i` değeri 10 dan küçükse kodu çalıştır diyoruz*
- *Kod çalışınca `i` değişkeninin değerini 1 artırıyoruz (`i++`)*

```
for (int i = 0; i < 10; i++)
```

Not: For yazıp “Tab” tuşuna basarsak bize kalıbı otomatik olarak getirir.

- *For satırından sonra süslü parantezler arasına (scope) döngüdeki şart sağlandığında çalışması istenilen kodlar yazılıyor. Kod tek satırdan oluşuyorsa süslü parantez yazmaya gerek yok ama programın okunması vb durumlar için yazılması daha uygundur tavsiye edilir.*

```
{
```

```
    Console.WriteLine(i); //write yazarsak yan yana yazar
```

```
}
```

```
    Console.ReadLine();
```

- *For döngü satırına break point koyup adım adım çalışmasına bakalım*
- *Burada döngü 10'a geldiğinde şart sağlamayıp döngüden çıkacaktır.*

```
for (int i = 0; i < 10; i++)  
{  
    Console.WriteLine(i); //write  
}  
Console.ReadLine();
```

C:\Users\Arif\Desktop\DESLER\MASAÜSTÜ UYGL

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```


Ornekk20. 0 ForKull Mod8

0 ile 100 arasındaki sayılardan sekize tam bölünenleri alt alta yazdıran programı tasarlayalım.

- İlk olarak 0 ile 100 arasında dönen bir döngü yazalım.

```
for (int i = 0; i < 100; i++)
```

- Sonra bu döngünün içerisine eğer 8 ile tam bölünüyorsa bu i sayısını yazdıran if şartı yazalım.

```
{
    if (i % 8 == 0)
    {
        Console.WriteLine(i);
    }
}
Console.ReadLine();
```

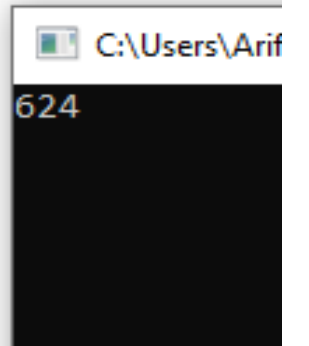
```
for (int i = 0; i < 100; i++)
{
    if (i % 8 == 0)
    {
        Console.WriteLine(i);
    }
}
Console.ReadLine();
```

0
8
16
24
32
40
48
56
64
72
80
88
96

Örneğe ekleme;

- *Biraz önceki örnekte 1 ile 100 arasındaki 8 e tam bölünen sayıları yazdırdık şimdi toplamlarını yazdıralım*

```
int toplam = 0;
for (int i = 0; i < 100; i++)
{
    if (i % 8 == 0)
    {
        toplam = toplam + i;
        //toplam += i;
    }
}
Console.WriteLine(toplam);
Console.ReadLine();
```



Ornek: ForKull_Faktoriyel

- Kullanıcının girdiği sayının faktöriyelinin hesaplayan programı yazalım.*

```
Lütfen faktöriyel hesabı yapmak istediğiniz değeri ekrana yazınız  
5  
5 değerinin faktöriyel sonucu : 120
```

```
int kullanicidanGelen = 0;

    int faktoriyel = 1;

    Console.WriteLine("Lütfen faktöriyel hesabı yapmak istediğiniz değeri ekrana yazınız");

    kullanicidanGelen = Convert.ToInt32(Console.ReadLine());

    for (int i = kullanicidanGelen; i > 1; i--)
    {
        faktoriyel = faktoriyel * i;
    }

    Console.WriteLine("{0} değerinin faktöriyel sonucu : {1}", kullanicidanGelen, faktoriyel);

    Console.ReadLine();
```

Ornek: ForKull_SonsuzDongu

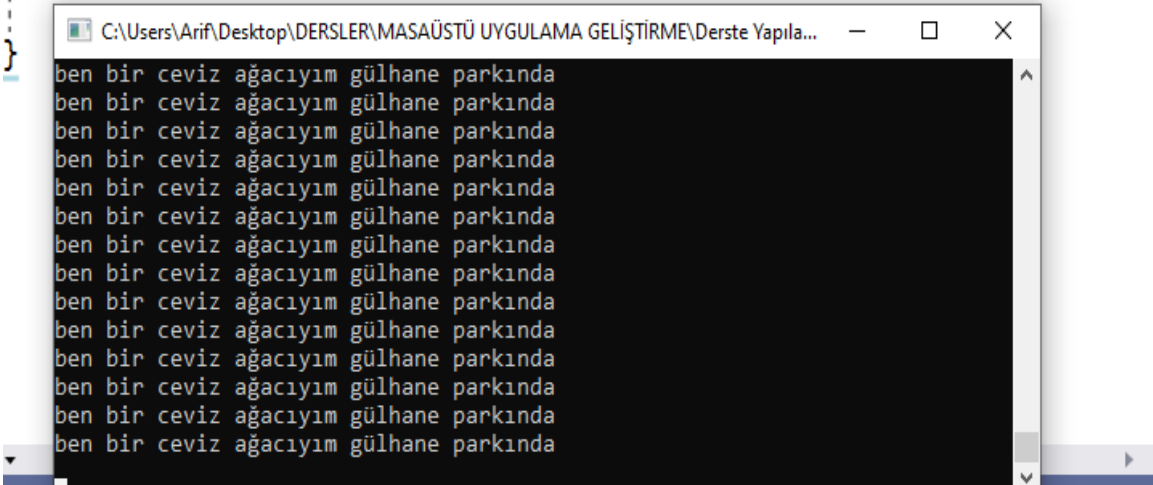
- *For ile sonsuz bir döngü oluşturmak için içi boş şartı falan olmayan bir for kalıbı yazıyoruz*

```
for( ; ; )
```

- *İçine ekrana yazı yazdıran kodu ekleyelim*

```
Console.WriteLine("ben bir ceviz ağacıym gülhane parkında");
```

```
for( ; ; )  
{  
    Console.WriteLine("ben bir ceviz ağacıym gülhane parkında");  
}
```



The screenshot shows a Windows console window titled "C:\Users\Arif\Desktop\DESLER\MASAÜSTÜ UYGULAMA GELİŞTİRME\Derste Yapıla...". The console output displays the string "ben bir ceviz ağacıym gülhane parkında" repeated approximately 15 times, demonstrating the execution of the infinite loop.

Break

- *Break* anahtar kelimesi döngüyü kırıp for döngüsünün dışına çıkmayı sağlar.
- *If* ile sayacın 10 a gelip gelmediğini kontrol edelim ve geldiyse “*break*” anahtar kelimesi döngüyü kırıp döngüden çıkalım.

```
int sayac = 0;
for ( ; ; )
{
    sayac++;
    if (sayac==10)
    {
        break;
    }
    Console.WriteLine(sayac);
    Console.WriteLine("ben bir ceviz ağacıym gülhane parkında");
}
Console.ReadLine();
```

```
1 ben bir ceviz ağacıym gülhane parkında
2 ben bir ceviz ağacıym gülhane parkında
3 ben bir ceviz ağacıym gülhane parkında
4 ben bir ceviz ağacıym gülhane parkında
5 ben bir ceviz ağacıym gülhane parkında
6 ben bir ceviz ağacıym gülhane parkında
7 ben bir ceviz ağacıym gülhane parkında
8 ben bir ceviz ağacıym gülhane parkında
9 ben bir ceviz ağacıym gülhane parkında
```

Continue

- Break anahtar kelimesi yerine `continue`; anahtar kelimesi yazarsak, döngüde şart sağlandığında `continue`' den sonra gelen kodları çalıştırmayıp yani atlayıp for döngüsüne devam eder.
- Burada sayaç 10 olduğunda yazmayıp 11 den devam ederek sonsuz döngüye girer.
- Ekrandan yazan satırların 9 dan 11 e geçtiğini görüyoruz

```
void Main(string[] args)

int sayac = 0;
for ( ; ; )
{
    sayac++;
    if (sayac==10)
    {
        continue;
        //break;
    }
    Console.WriteLine(sayac);
    Console.WriteLine("ben bir ceviz ağacıym gülhane parkında");
}
Console.ReadLine();
```

```
1 ben bir ceviz ağacıym gülhane parkında
2 ben bir ceviz ağacıym gülhane parkında
3 ben bir ceviz ağacıym gülhane parkında
4 ben bir ceviz ağacıym gülhane parkında
5 ben bir ceviz ağacıym gülhane parkında
6 ben bir ceviz ağacıym gülhane parkında
7 ben bir ceviz ağacıym gülhane parkında
8 ben bir ceviz ağacıym gülhane parkında
9 ben bir ceviz ağacıym gülhane parkında
10 ben bir ceviz ağacıym gülhane parkında
11 ben bir ceviz ağacıym gülhane parkında
12 ben bir ceviz ağacıym gülhane parkında
13 ben bir ceviz ağacıym gülhane parkında
14 ben bir ceviz ağacıym gülhane parkında
```

Ornekk23.0 ForKull 5Ve7yeBol

- Bu örnekte 0 ile 1000 arasındaki sayılardan hem 5 hem de 7 tam bölünebilen sayıların ekrana yazıldığı, bu sayılardan kaç adet bulunduğunu ve bu sayıların toplam değerini bulan programı yazalım.*

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int adet=0,toplam = 0;
        for (int i = 0; i <= 1000; i++)
        {
            if ((i%5==0)&&(i%7==0))
            {
                Console.WriteLine(i);
                toplam += i;
                adet++;
            }
        }

        Console.WriteLine("0 ile 1000 arasında hem 5'e hem de 7'ye tam bölünen {0} adet sayı bulundu.Bu sayıların toplamları= {1}",adet,toplam);
        Console.ReadLine();
    }
}
```



0
35
70
105
140
175
210
245
280
315
350
385
420
455
490
525
560
595
630
665
700
735
770
805
840
875
910
945
980
0 ile 1000 arasında hem 5'e hem de 7'ye tam bölünen 29 adet sayı bulundu.Bu sayıların toplamları= 14210

İç içe For Kullanımı

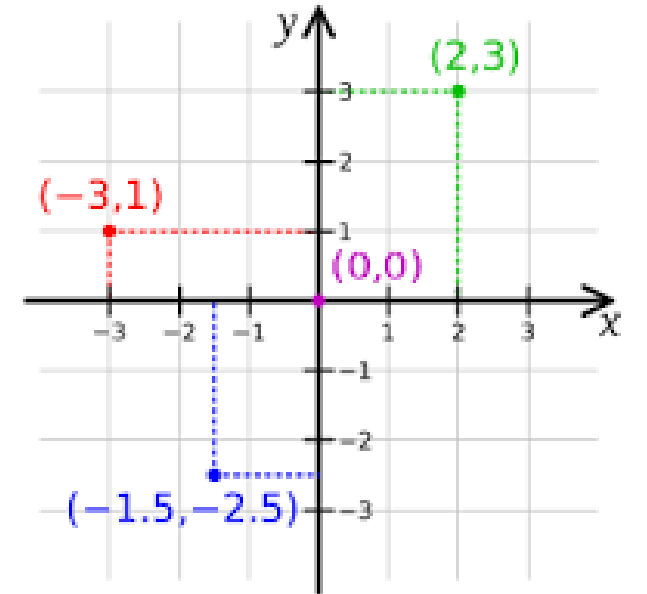
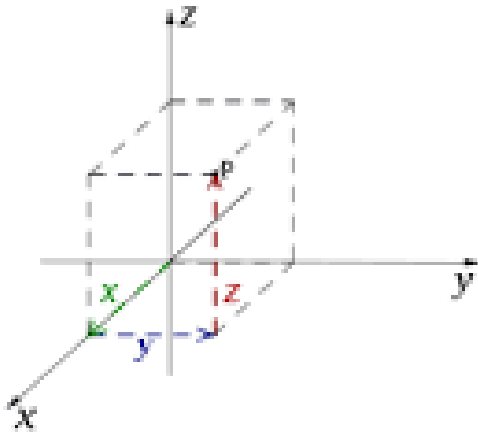
(İç içe döngüler - Nested Loops)

- *Bilgisayar bilimlerinde hemen hemen bütün programlama dillerinde kullanılan döngülerin iç içe kullanılmasına (kucak kucağa, nested) verilen isimdir.*
- *Buna göre bir döngü (loop) bir dönüşünü tamamlamadan diğer bir veya birkaç döngü dönme işlemi gerçekleştirebilir.*

$(0,0)$ (x,y)

- İki döngü kullandığımız durumda dış döngü satır değerlerinde değişiklik yaparken iç döngü sütunlarda geçiş yapar.
- Daktilo ile yazı yazmak gibi ilk satır yazıldıktan sonra ikinci satıra geçilmesi.

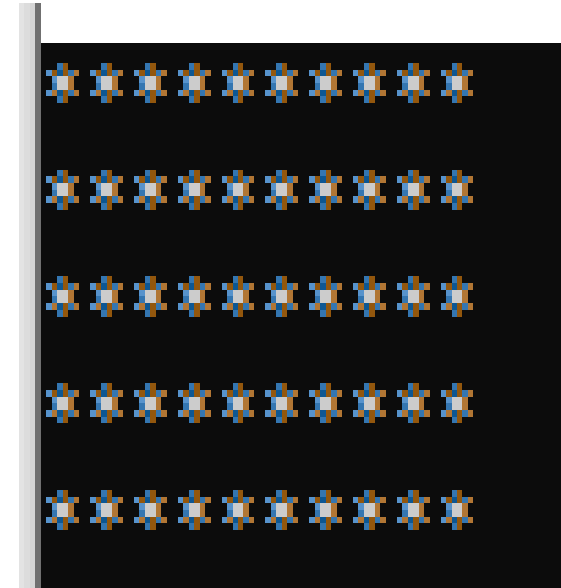
Koordinat ve Koordinat Sistemi



Ornekk24.0 CiftForKull 1

- 5 satır 10 sütundan oluşan bir yapı oluşturalım. Her bir satır değeri için sütun değerini 10' a kadar artıyoruz.

```
for (int satir = 0; satir < 5; satir++)  
{  
    for (int sutun = 0; sutun < 10; sutun++)  
    {  
        Console.Write("*");  
    }  
    Console.WriteLine();  
}  
Console.ReadLine();
```



Ornekk25. 0 CiftForKull 2

- İç içe for kullanımı anlamak için dışarıdan girilen satır ve sütun sayılarına göre ekrana yıldız(*) basan programı inceleyelim
- İlk olarak satır ve sütun isminde değişkenler oluşturalım.
`int` satir, sutun;
- Kullanıcının satır ve sütun sayılarını girmesini isteyip bu değerleri ilgili değişkenlere atayalım.

```
Console.WriteLine("Lütfen satır sayısını giriniz : ");
```

```
satir = Convert.ToInt32(Console.ReadLine());
```

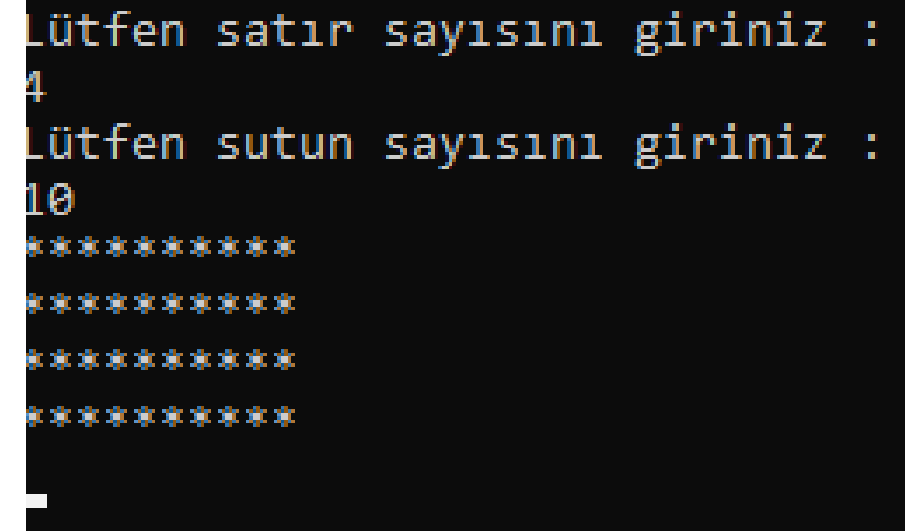
```
Console.WriteLine("Lütfen sütun sayısını giriniz : ");
```

```
sutun = Convert.ToInt32(Console.ReadLine());
```

- İki adet for oluşturacağız her satır için sütun değeri kadar dönüp sonra satır sayısını arttıracak. Yani her dış döngü değeri için iç döngüyü ilk değerden son değere kadar yapacak. İç döngüden çıkınca bir satıra geçmesi için writeLine metodu kullanılıyor.

```
for (int i = 0; i<satir; i++)//dış döngü
{
    for (int j = 0; j<sutun; ++j)//iç döngü
    {
        Console.Write("*");
    }
    Console.WriteLine();
}

Console.ReadLine();
```



```
Lütfen satır sayısını giriniz :
4
Lütfen sütun sayısını giriniz :
10
*****
*****
*****
*****
```

Ornekk26._0_CiftForKull_3_DikUcgen

- *İç içe iki adet for döngüsü oluşturup her bir satır değeri kadar sütun sayısı*

oluşturup yıldız basan program.

- ***BreakPoint koyarak adım adım inceleyelim***



```
for (int i = 1; i <= 20; i++)//dış döngü
{
    for (int j = 1; j <= i; j++)//iç döngü
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
Console.ReadLine();
```



İnceleyelim;

dış döngü olan for kapsama alanı parantezler ile belirlenen ve diğer for döngüsünde içine alan kısım. Yani dış for döngüsünün şartının sağlandığı her değer için içerideki kısım baştan sona çalışıyor.

CiftForKull_4_CarpimTablosu

- Çarpım tablosunu ekrana yazdıran programı yazalım. İki adet for ile 1 den başlayıp 9 kadar olan sayılar ile çarparak ekrana yazdıralım

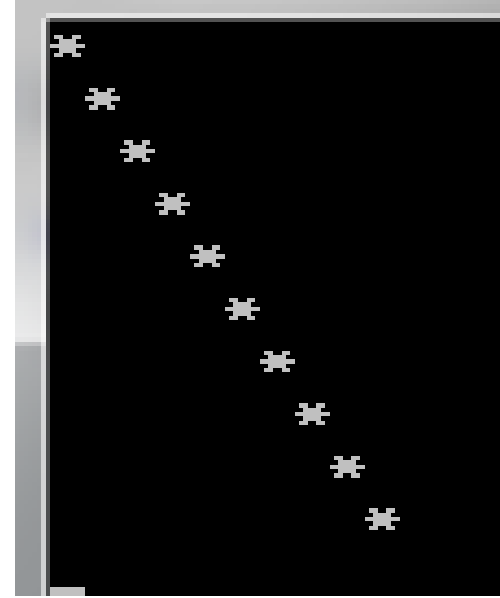
C:\Users\Arif\Desktop\DERSLER\MASAÜSTÜ UYGULAMA GELİŞTİRME\Derste Yapılan Örnekler\MasaUstuUy

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

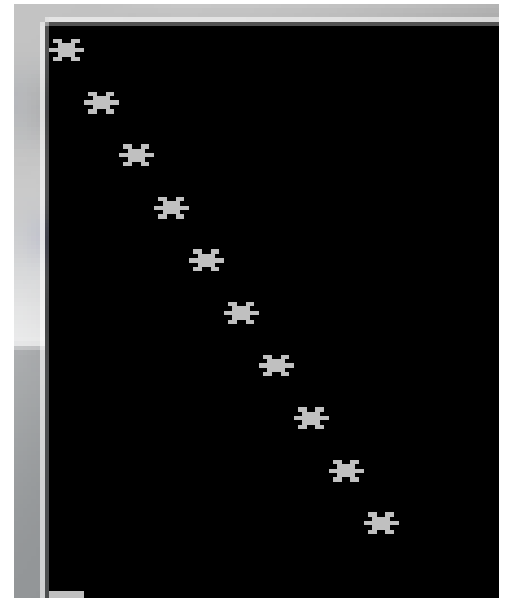

```
for (int i = 1; i <= 9; i++)  
    {  
        for (int y = 1; y <= 9; y++)  
        {  
            int carpimSonuc = i * y;  
            Console.Write("{0}*{1}={2}\t", i, y, carpimSonuc);  
        }  
        Console.WriteLine();  
    }  
    Console.ReadLine();
```

CiftForKull_5_if Kul_solaYatik -Break

- *Yandaki ekran çıktısını elde edecek programı yazalım.*
- *If else ve **break** kullanarak yapalım.*
- *Satır sayısının sütun sayısına eşit olduğu durumlarda yıldız basacak. Birinci döngü (dış)satırları ikinci döngü(iç) sütunları kontrol edecek.*



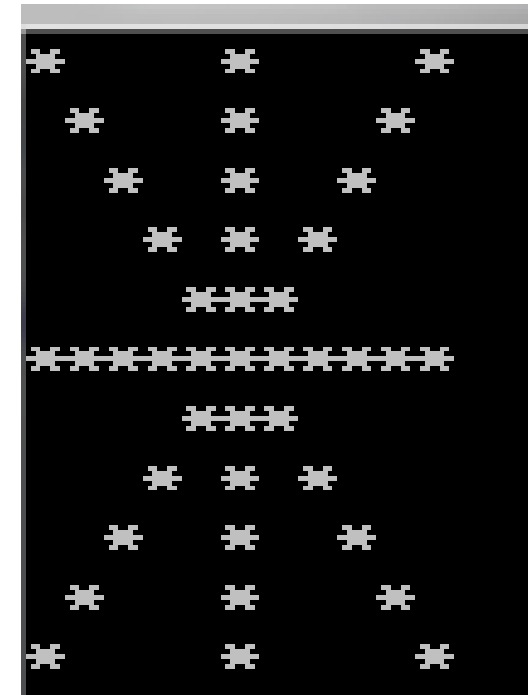
```
for (int satir = 0; satir < 10; satir++)  
{  
    for (int sutun = 0; sutun < 10; sutun++)  
    {  
  
        if (satir == sutun)  
        {  
            Console.WriteLine("*");  
            break; //İçinde bulunduğuy döngüyü sonlandırır.  
        }  
        else  
        {  
            Console.Write(" ");  
        }  
    }  
}  
  
Console.ReadLine();
```



- *Şart yazacağımız şartımız satır ve sütun sayıları eşit ise yıldız bas, eşit değilse boşluk basacak.*
- **Break** anahtar kelimesini bakmıştık daha önce döngüyü kırıp bir üste çıkmayı sağlıyordu.
- *Döngüdeki bir şart gerçekleştiğinde diğer durumlara bakmadan döngüden çıkma durumlarında kullanılır.*
Break anahtar kelimesi ile yıldızı basınca o satırı bırakması ve üst döngüye(satir) geçmesini sağlıyor.
- *Yıldız basacağı yerde WriteLine, boşluk basacağı yerde sadece Write kullanıyoruz. WriteLine ile bir alt satıra geçmeyi sağlamış oluyoruz.*

Ornekk29.0 CiftForKull 6 ifKul Yildiz

- 11 x11 bir matris oluřturacađız



```
for (int satir = 0; satir <= 10; satir++)
{
    for (int sutun = 0; sutun <= 10; sutun++)
    {
        if ( satir==sutun || sutun==5 || satir==5 || satir+sutun==10)
        {
            Console.Write("*");
        }
        else
        {
            Console.Write(" ");
        }
    }
    Console.WriteLine();
}
Console.ReadLine();
```

For Bölümünün sonu;

- *For döngüsü belli işlemleri yapmaya yarar dedik*
- *Kalıbında hiçbir şey yazmayıp sadece iki adet; kullanırsak sonsuz döngü oluştururuz.*
- *Break anahtar kelimesi ile döngüden çıkarız*
- *For içerisinde yeni bir değişken tanımlayıp kullanacağımız gibi dışarıdan bir değişken üzerinden de işlem yapılabilir ama genelde pek kullanılmaz.*

Foreach Döngüsü

- *C# özel bir döngü listelerde dönmek için kullanılır.*
- *Dizileri incelendikten sonra anlamak daha kolay. Hazır bir dizi üzerinde otomatik döngü oluşturuyoruz*
- *Bu dizinin elemanlarında dön diyoruz dizinin tüm elemanlarında sırasıyla tek tek işliyor.*

- *foreach Döngüsü for döngüsü gibi yaygın kullanılan bir diğer döngü de foreach döngüsüdür.*
- *foreach, dizi (array) ve koleksiyon (collection) tabanlı nesnelerin elemanları üzerinden ilerleyen, iterasyon(tekrarlama, yineleme) gerçekleştirerek bu elemanlara erişip iş katmanınızı oluşturabileceğiniz bir döngüdür*

Ornekk40. 0 foreach

Basit olarak bir dizi oluşturalım ve dizinin tüm elemanlarını yazdıran bir örnek yapalım.

Haftanın günlerinin olduğu dizinin tüm elemanlarını ekrana yazdıran örnek.

- *İlk olarak günler isminde dizi tanımlayıp değerlerini girelim*

```
String[] gunler =new [] { "pazartesi", "salı", "çarşamba", "perşembe", "cuma", "cumartesi", "pazar" };
```

- *Foreach döngüsü oluşturalım hedef dizi olarak “günler” dizisi gösteriyoruz*

```
foreach (string item in gunler)
{
    Console.WriteLine(item);
}
Console.ReadLine();
```


```
pazartesi
salı
çarşamba
perşembe
cuma
cumartesi
pazar
```

Örnek: dizinin eleman sayılarını toplamı

- *Dizinin tüm elemanlarını tek tek dolaşıp ayrı bir değişkende toplayan program.*

```
int[] sayilar = { 12, 33, 43, 90, 45, 65 };  
int toplam = 0;  
foreach (int i in sayilar)  
{  
    toplam += i;  
}  
Console.WriteLine("Dizinin tüm elemanlarını toplam değeri {0}",toplam);  
Console.ReadLine();
```

```
int[] sayilar = { 12, 33, 78, 44, 34, 56 };  
int toplam = 0;  
foreach (int i in sayilar)  
{  
    toplam += i;  
}  
Console.WriteLine("Dizinin tüm elemanlarının toplamı .{0}", toplam);  
Console.ReadLine();
```

 C:\Users\Arif Gunel\source\repos\HAFTA8_PROG\foreach2\bin\Debug\foreach2.exe

Dizinin tüm elemanlarının toplamı .257

Örnek : Foreach ile rasgele değer atama ve okuma

- *Random sınıfı kullanarak dizinin elemanlarına foreach döngüsü ile tek tek değer atayıp sonra tekrar foreach ile tek tek değerleri okuyalım.*
- *İlk olarak Random sınıfı ile rnd nesnesi oluşturalım*

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

- *10 elemanlı sayilar isminde bir dizi tanımlayalım*

```
int[] sayilar = new int[10];
```

- *Değişken tanımlayıp dizinin indis numaralarında gezmesini sağlayalım*

```
int sıraNo = 0;
```

- *Foreach döngüsü ile dizinin elemanlarında gezip rasgele değer üretip atayalım*

```
foreach (int i in sayilar)//Diziye değer atar
{
    sayilar[sıraNo] = rnd.Next(0, 100); // böylede olur rnd.Next(100);
    sıraNo++;
}
```

- *Foreach döngüsü ile dizinin elemanlarında gezip değerleri okuyalım*

```
foreach (int i in sayilar)//dizinin elemanlarını ezerek değerleri okur
{
    Console.WriteLine(i);

}
Console.ReadLine();
```

```
Random rnd = new Random();
int[] sayilar = new int[10];
//10 elemanlı int veri tipinde sayilar isiminde tek boyutlu dizi tanımladık
int sıraNo = 0;
foreach (int i in sayilar)//değer atadığımız foreach döngüsü
{
    sayilar[sıraNo] = rnd.Next(0, 100);
    //dizinin her bir elemanın rastgele değer ürettip, atadık
    sıraNo++;
}
foreach (int i in sayilar)//dizinin elemanlarını gezerek değer okuma
{
    Console.WriteLine(i);
}
Console.ReadLine();
```

C:\Users\Arif Gunel\source\repos\HAFTA8_PROG\foreach3_rastgeledegeratama

28
17
20
93
77
56
99
30
39
37

Kodların tamamı;

```
Random rnd = new Random();  
int[] sayilar = new int[10]; //10 elemanlı int veri tipinde sayilar  
isiminde tek boyutlu dizi tanımladık  
int sirano = 0; //dizinin elemanları(indis numarası) gezmemizi sağlar  
foreach (int i in sayilar) //değer atadığımız foreach döngüsü  
{  
    sayilar[sirano] = rnd.Next(0, 100);  
    //dizinin her bir elemanın rastgele değer ürettip, atadık  
    sirano++;  
}  
foreach (int i in sayilar) //dizinin elemanlarını gezerek değer okuma  
{  
    Console.WriteLine(i);  
}  
Console.ReadLine();
```

Ekleme yapalım; -rasgele atanan değerlerin toplamını ve ortalamasını yazdıralım

- *Toplamı tutacak olduğumuz bir değişken tanımlayalım*

```
int toplam = 0;
```

- *İkinci foreach döngüsünde değerleri toplayalım*

```
toplam += i;
```

- *ekrana toplam ve ortalama değerlerini yazdıralım*

```
Console.WriteLine("girilen sayıların toplamı : {0}", toplam);
```

```
Console.WriteLine("girilen sayıların ortalaması : {0}", toplam / 10);
```

Kodların tamamı;

```
Random rnd = new Random();
int[] sayilar = new int[10];
int sirano = 0;
int toplam = 0;
foreach (int i in sayilar)//Diziye değer atar
{
    sayilar[sirano] = rnd.Next(0, 100);
    sirano++;
}
foreach (int i in sayilar)
{
    Console.WriteLine(i);
    toplam += i;
}
Console.WriteLine("girilen sayıların toplamı : {0}", toplam);
Console.WriteLine("girilen sayıların ortalaması : {0}", toplam / 10);
Console.ReadLine();
```

Ekleme yapalım - rasgele atanan değerlerin en büyük ve en küçük olanı bulalım- yazdıralım

- *En büyük ve en küçük olan değerleri tutmak için değişken tanımlıyoruz. Başlangıç değerleri olarak en küçük ola 100 en büyük olana 0 değerini veriyoruz. Böyle yapmamızın nedeni if ile dizinin tüm elemanlarını karşılaştırdığımızı zaman karşılaştırılan değerleri en büyük ve en küçük değişkene atayabilmemiz için. Farklı bir şekilde yapmak istersek değişkenleri tanımladığımız aşamada dizinin ilk elemanının içindeki değeri de atayabiliriz*
- *(int kucuk = dizi[0]; int buyuk = dizi[0];) ama bu işlemi diziye rasgele değer atadıktan sonra yazmalıyız.*

```
int enBuyuk = 0;  
int enKucuk = 100;
```

- *if ile değişkenleri dizinin her elemanı ile karşılaştırıyoruz. İkinci foreach içine yazıyoruz..*

```
foreach (int i in sayilar)  
{  
    Console.WriteLine(i);  
    toplam += i;  
    if (i>enBuyuk)  
    {  
        enBuyuk = i;  
    }  
    if (i<enKucuk)  
    {  
        enKucuk = i;  
    }  
}
```

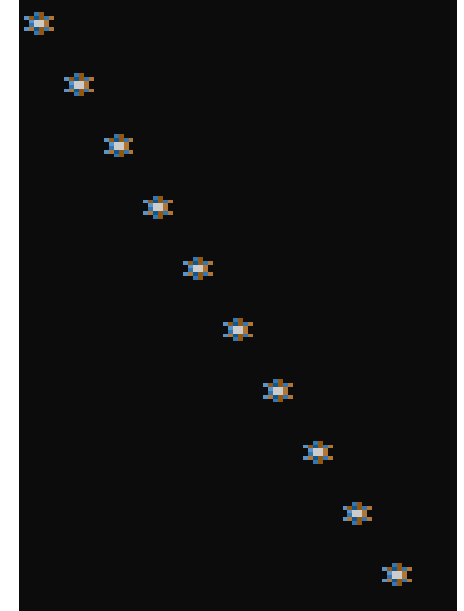
- *Son olarak en büyük ve en küçük değerleri ekrana yazdırıyoruz*

```
Console.WriteLine("Dizinin en küçük elemanı : {0}", enKucuk);
```

```
Console.WriteLine("Dizinin en büyük elemanı : {0}", enBuyuk);
```

ÖDEV 1:

- *Yandaki ekran çıktısını elde edecek programı yazalım.*
- *If else kullanarak yapalım.*
- *Satır sayısının sütun sayısına eşit olduğu durumlarda yıldız basacak. Birinci döngü (dış)satırları ikinci döngü(iç) sütunları kontrol edecek.*



ÖDEV 2: Çarpım Tablosu

- İki adet for ile 1 den başlayıp 9 kadar olan sayılar ile çarparak ekrana Çarpım tablosunu yazdıralım

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

Haftaya işlenecek konu;

- *While Döngüsü*
- *Do while Döngüsü*

Kaynakça

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