### FOR DÖNGÜSÜ

Eğer programda -belli koşulları sağladığı sürece- birden fazla çalıştırılmasını istediğimiz kodlar varsa döngüler kullanılır. C#'ın en çok kullanılan döngü "for"dur.

Kullanımı:

*for (döngü\_değişkeni=başlangıç\_değeri; döngü\_değişkeni ve şart; artım\_miktarı)*

for(ifade1;kosul;ifade2)

komut;

veya

for(ifade1;kosul;ifade2)

{

komut1;

komut2;

.

.

.

}

#### for döngüsünün çalışma prensibi

for(ifade1;kosul;ifade2)

{

komut1;

komut2;

.

.

.

}

**1)** ifade1 çalıştırılır.  
**2)** Koşula bakılır. Eğer koşul sağlanıyorsa;

**2.1)** küme parantezleri içindeki ya da -küme parantezleri yoksa- kendinden sonra gelen ilk satırdaki komut çalıştırılır.

**2.2)** ifade2 çalıştırılır.

**2.3)** 2. adıma dönülür.

**3)** Eğer koşul sağlanmıyorsa küme parantezleri dışına ya da -küme parantezleri yoksa- kendinden sonra gelen ilk satırdaki komuttan hemen sonraki satıra çıkılır. Dolayısıyla döngüden çıkılmış olur.

#### 

ÖRNEK: 1 den 10 a kadar olan sayıların ekrana yazdırılması.

class Program

{

static void Main(string[] args)

{

int i;

for (i = 1; i <= 10; i++)

{

Console.WriteLine(i);

}

Console.ReadLine();

}

}

ÖRNEK: 10 ile 30 arasındaki sayıların 3’er 3’er ekrana yazdıran program.

class Program

{

static void Main(string[] args)

{

int i;

for (i = 10; i <= 30; i=i+3)

{

Console.WriteLine(i);

}

Console.ReadLine();

}

}

ÖRNEK: 100’den başlayıp 0 a kadar 10’ar eksilten program.

class Program

{

static void Main(string[] args)

{

int i;

for (i = 100; i > 0; i = i - 10)

{

Console.WriteLine(i);

}

Console.ReadLine();

}

}

ÖRNEK: Başlangıç, bitiş ve artım miktarı klavyeden girilen sayıların ekrana yazdırılması.

static void Main(string[] args)

{

int i, x,y,a;

Console.WriteLine("Bşlangıç degerini giriniz");

y = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Bitiş degerini giriniz");

x = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Artım miktarı giriniz");

a = Convert.ToInt16(Console.ReadLine());

for (i = y; i <= x;i=i+a )

{

Console.WriteLine(i);

}

Console.ReadLine();

}

ÖRNEK: Klavyeden girilen ismi, yine klavyeden girilen sayı kadar ekrana yazdıran program

class Program

{

static void Main(string[] args)

{

int i,x;

string ad;

Console.WriteLine("tekrar sayısını gir");

x = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("isim gir");

ad = Console.ReadLine();

for (i = 1; i <= x; i++)

{

Console.WriteLine(ad);

}

Console.ReadLine();

}

}

ÖRNEK: 1’den 10 kadar sayıların toplamını bulan program.

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int i,tp;

tp = 0;

for (i = 1; i <= 10; i++)

tp += i;

Console.WriteLine("1 den 10 kadar sayıların toplamı:"+ tp );

Console.ReadLine();

}

}

}

ÖRNEK: 1 den 10 kadar sayıların toplamı(her defasında sayıyı ve toplam değerini ekrana yazacak)

class Program

{

static void Main(string[] args)

{

int i,tp;

tp = 0;

for (i = 1; i <= 10; i++)

{

tp += i; // tp = tp + i

Console.Write("i= "+i);

Console.WriteLine(" tp= " + tp);

}

Console.ReadLine();

}

}

SORU 1 : Klavyeden girilen sayının faktöryelini bulan programı yapın.

SORU2 : Klavyeden girilen iki sayı arasındaki sayıların toplamını bulan program.

SORU3 : Klavyeden girilen sayıya kadar olan sayıların toplamını bulan programı yapın.

### WHİLE DÖNGÜSÜ

Komut ya da komutların bir koşul sağlandığı sürece yapılmasını sağlar.

**Kullanımı**:

while(koşul)

komut;

veya

while(koşul)

{

komut1;

komut2;

.

.

.

}

#### while döngüsünün çalışma prensibi

while(koşul)

{

komut1;

komut2;

.

.

.

}

**1)** Koşula bakılır. Eğer koşul sağlanıyorsa;

**1.1)** küme parantezleri içindeki ya da -küme parantezleri yoksa- kendinden sonra gelen ilk satırdaki komut çalıştırılır.

**1.2)** 1. adıma dönülür.

**2)** Eğer koşul sağlanmıyorsa küme parantezleri dışına ya da -küme parantezleri yoksa- kendinden sonra gelen ilk satırdaki komuttan hemen sonraki satıra çıkılır. Dolayısıyla döngüden çıkılmış olur.

### DO WHİLE DÖNGÜSÜ

Şimdiye kadar gördüğümüz döngülerde önce koşula bakılıyor, eğer koşul sağlanırsa döngü içindeki komutlar çalıştırılıyordu. Ancak bazen döngüdeki komutların koşul sağlanmasa da en az bir kez çalıştırılmasını isteyebiliriz. Bu gibi durumlar için C#'ta do while döngüsü vardır.

#### Kullanımı

do

komut;

while(koşul)

veya

do

{

komut1;

komut2;

.

.

.

}while(koşul)

#### do while döngüsünün çalışma prensibi

do

{

komut1;

komut2;

.

.

.

}while(koşul)

**1)** Döngüdeki komutlar bir kez çalıştırılır.  
**2)** Koşula bakılır.

**2.1)** Eğer koşul sağlanıyorsa 1. adıma dönülür.

**2.2)** Eğer koşul sağlanmıyorsa döngüden çıkılır.

**ÖRNEK**: 1 den 10a kadar sayıların ekrana yazdırılması(While döngüsü)

class Program

{

static void Main(string[] args)

{

int i = 1;

while (i <= 10)

{

Console.WriteLine(i);

i++;

}

Console.ReadLine();

}

}

**ÖRNEK**: 1 den 10a kadar sayıların ekrana yazdırılması(While döngüsü)

class Program

{

static void Main(string[] args)

{

int i = 1;

do

{

Console.WriteLine(i);

i++;

} while (i <= 10);

Console.ReadLine();

}

}

**ÖRNEK**: 50 den 10 a kadar sayıların 5’er azalarak ekrana yazılması(While döngüsü)

class Program

{

static void Main(string[] args)

{

int i = 50;

while (i >= 10)

{

Console.WriteLine(i);

i-=5;

}

Console.ReadLine();

}

}

**ÖRNEK**: 50 den 10 a kadar sayıların 5’er azalarak ekrana yazılması(do While döngüsü)

class Program

{

static void Main(string[] args)

{

int i = 50;

do

{

Console.WriteLine(i);

i-=5;

} while (i >= 10);

Console.ReadLine();

}

}

ÖRNEK: Klavyeden girilen 5 sayıyı toplayıp sonucu ekrana yazan programı while döngüsüyle yapın.

class Program

{

static void Main(string[] args)

{

int i,s,tp;

i = 1;

tp = 0;

while (i <= 5)

{

Console.Write("sayı girin: ");

s = Convert.ToInt16(Console.ReadLine());

tp = tp + s;

i++;

}

Console.WriteLine("sayının toplamı=" + tp);

Console.ReadLine();

}

}

ÖRNEK: Klavyeden girilen 5 sayıyı tek yada çift diye ekrana yazdıran programı while döngüsüyle yapın.

class Program

{

static void Main(string[] args)

{

int i, s;

i = 1;

while (i <= 5)

{

Console.Write("sayı girin: ");

s = Convert.ToInt16(Console.ReadLine());

if (s % 2 == 0)

{

Console.Write(s);

Console.WriteLine(" girilen sayı çift");

}

else

{

Console.Write(s);

Console.WriteLine(" girilen sayı tek");

}

i++;

}

Console.ReadLine();

}

}

SORU 4 : 1 den 100 e kadar olan sayılardan 3’e bölünenleri ekrana yazdıran program.

SORU 5 : Klavyeden 0 girilene kadar olan sayıları toplayan ve ortalamasını bulup ekrana yazan programı yapın.(while yada do-while ile yapın)

### DÖNGÜLERDE KULLANILAN ANAHTAR SÖZCÜKLER

#### break

Hatırlarsanız break komutunu switchteki caselerden çıkmak için kullanmıştık. Benzer şekilde break komutu bütün döngülerden çıkmak için kullanılabilir. Örnek:

using System;

class AkisKontrolMekanizmalari

{

static void Main()

{

for(char a;;)

{

a=Convert.ToChar(Console.ReadLine());

if(a=='q')

break;

}

}

}

Bu program, kullanıcı "q" harfini girene kadar kapanmamaktadır.

#### continue

break sözcüğüne benzer. Ancak break sözcüğünden farklı olarak program continue'u gördüğünde döngüden çıkmaz, sadece döngünün o anki işlemi sonlanır. Örnek:

using System;

class AkisKontrolMekanizmalari

{

static void Main()

{

for(int a=0;a<51;a++)

{

if(a%2==1)

continue;

Console.WriteLine(a);

}

}

}

Bu program 0'dan 50'ye kadar (0 ve 50 dâhil) olan çift sayıları ekrana alt alta yazmaktadır.

#### goto

Nesneye yönelik programlamada pek hoş görülmese de kullanabileceğiniz başka bir komut "goto"dur. Aslında eskiden BASIC gibi dillerde her satırın bir numarası vardı ve bu sözcük satırlar arasında dolaşmayı sağlıyordu. Ancak böyle bir yöntem nesne yönelimli programlamaya terstir. O yüzden çok fazla kullanmamanız tavsiye edilir. Örnek kullanım:

ÖRNEK: Bu programda ikinci bölüm hiçbir zaman çalıştırılmayacaktır.

using System;

class AkisKontrolMekanizmalari

{

static void Main()

{

birinci:

Console.WriteLine("Birinci bölüm");

goto ucuncu;

ikinci:

Console.WriteLine("İkinci bölüm");

ucuncu:

Console.WriteLine("Üçüncü bölüm");

}

}

ÖRNEK: i değerine başlangıçta 2 değeri verilirse işlem SON etiketine gider ve aradaki komutlar çalışmaz.

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int i = 2;

if (i==2) goto son;

birinci:

Console.WriteLine("Birinci bölüm");

goto ucuncu;

ikinci:

Console.WriteLine("İkinci bölüm");

ucuncu:

Console.WriteLine("Üçüncü bölüm");

son:

Console.ReadLine();

}

}

}

### Döngülerle ilgili karışık örnekler

* 1'den 1000'e (sınırlar dâhil) kadar olan sayılar içerisinde 5'e tam bölünen, ancak 7'ye tam bölünemeyen sayıları alt alta listeleyen, bu sayıların kaç tane olduğunu ve toplamını yazan bir program yazınız.

using System;

class AkisKontrolMekanizmalari

{

static void Main()

{

int toplam=0, sayi=0, i=5;

for(;i<1001;i+=5)

{

if(i%35==0)

continue;

sayi++;

toplam+=i;

Console.WriteLine(i);

}

Console.WriteLine("Sayısı: "+sayi);

Console.WriteLine("Toplam: "+toplam);

}

}

* Girilen pozitif herhangi bir tam sayıyı ikilik düzene çeviren programı yazınız.

using System;

class AkisKontrolMekanizmalari

{

static void Main()

{

string mod="";

Console.Write("Lütfen ikilik sisteme dönüştürülmesini istediğiniz sayınızı girin: ");

float a=Convert.ToSingle(Console.ReadLine());

int b=(int)a;

if(a<=0||a!=b)

mod="Bir pozitif tam sayı girmediğiniz için sayınız ikilik sisteme dönüştürülmedi!";

else

for(;b>0;b/=2)

mod=b%2+mod;

Console.Write(mod);

}

}

* Konsol ekranına girilen 0 ile 100 (sınırlar dâhil) arasındaki 10 notun en büyüğünü, en küçüğünü ve ortalamasını yazan programı yazınız.

using System;

class AkisKontrolMekanizmalari

{

static void Main()

{

int bs=0, toplam=0, ks=0;

for(int a=0, b;a<10;a++)

{

Console.Write(a+1+". notu giriniz: ");

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

if(b>100||b<0)

{

Console.Write("Yanlış not girdiniz. Lütfen tekrar ");

a--;

continue;

}

if(a==0)

{

bs=b;

ks=b;

}

else

{

if(b>bs)

bs=b;

if(b<ks)

ks=b;

}

toplam+=b;

}

Console.Write("En büyük: {0}\nEn küçük: {1}\nOrtalama: "+toplam/10,bs,ks);

}

}

**Atama Operatörleri**

Operatör Eşdeğeri Anlamı

x += y x = x + y x + y’nin değerini x’e ata

x ‐= y x = x – y

x \*= y x = x \* y

x++ x = x + 1

Operatör Eşdeğeri

x‐‐ x = x – 1

x %= y x = x % y x’in y’ye bölümünden kalanı x’e ata

**Şart İfadeleri**

== Eşittir

!= Eşit Değil

> Büyük

< Küçük

>= Büyük Eşit

<= Küçük Eşit

! Değil

&& Ve

|| Veya