DÖNGÜLER

For Döngü yapısı

Bir işlemin belirli sayıda yapılması için kullanılır. Şartlar sağlandığı sürece komutlar işletilmektedir. Başlangıç işlemlerinde komutlar sadece bir  defa çalışmaktadır. Devam şartlarında, değer True oldukça komutlar çalıştırılır. **For** döngüsünde sayaç, belirtilen aralıkta olana kadar işlem yapılır. For döngüsünün yapısı aşağıdaki gibidir.

**Komut dizilimi**

{

For ( Döngü değişkeni=ilk değer;Koşul; Artırım değeri)

{

İşletilecek komutlar;

}

**☞ Örnek Windows application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int N = 21;

for (int i = 0; i < N; i++)

{

textBox1.Text = i.ToString(); }

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int x;

for (x = 0; x <= 8; x++)

{

if (x != 4)

{

MessageBox.Show("X = " + x.ToString());}}}

**☞ Örnek Windows Application**

Klavyeden girilen herhangi bir cümlenin ekrana 10 defa yazdırılmasını sağlayan programı yazınız.

privatevoid button1\_Click(object sender, EventArgs e)

{

label1.Text = "";

}

privatevoid button2\_Click(object sender, EventArgs e)

{

string cumle;

cumle = textBox1.Text;

label2.Text = "";

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

{

label2.Text = label2.Text + i + ". " + cumle + "\n"; } }

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int N =1;

for (int i = 0; i < N; i++)

{

MessageBox.Show("Merhaba");}

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

listBox1.Items.Add("Tekrar eden Sayılar");

for (int i = 1; i <= 44; i = i + 2)

{

listBox1.Items.Add(i);}

**☞ Örnek Windows Application**

1 den 100 kadar olan sayıların toplamını veren program kodları aşağıdaki gibi yazılabilir.

privatevoid button1\_Click(object sender, EventArgs e)

{

int toplam = 0;

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

{

toplam = toplam + i;

}

label1.Text = toplam.ToString();}

**☞ Örnek Windows Application**

1 den 100 kadar olan çift sayıların toplamını veren program kodları aşağıdaki gibi yazılabilir.

privatevoid button1\_Click(object sender, EventArgs e)

{

int toplam = 0;

for (int i = 0; i <= 100; i+=2)

{

toplam = toplam + i;

}

label1.Text = toplam.ToString();}

**☞ Örnek Windows Application**

Klavyeden inputbox ile iki değer girişi alınsın.

* 1. Birinci değer girişi Label üzerine yazılacak metin değerini
  2. İkincisi ise alınan metnin kaç kere yazılacağını belirlesin.   
     Programın C# kodlarını yazınız.

privatevoid button1\_Click(object sender, EventArgs e)

{

string metin;

label1.Text = "";

metin = Microsoft.VisualBasic.Interaction.InputBox("Metin Girişi Yapınız","","",120,120);

int A=Convert.ToInt32(Microsoft.VisualBasic.Interaction.InputBox("Kaç kere yazsın","","",120,120));

for (int i = 1; i <= A; i++)

{

label1.Text = label1.Text + i + " " + " ." + metin + "\n";}}

**☞ Örnek Windows Application**

Textbox içine girilen kelimeyi aşağıdaki gibi yazan C# program komutlarını yazınız

privatevoid button1\_Click(object sender, EventArgs e)

{

string kelime = "ismail";

for (int i = 0; i < kelime.Length; i++)

{

listBox1.Items.Add(kelime.Substring(0, i +1));}}

**☞ Örnek Windows Application**

Textbox içine girilen kelimeyi aşağıdaki gibi yazan C# program komutlarını yazınız

privatevoid button1\_Click(object sender, EventArgs e)

{

int karakter = 0;

karakter = textBox1.Text.Trim().Length;

for (int i = 0; i < textBox1.Text.Length; i++)

{

listBox1.Items.Add(textBox1.Text.Substring(0, textBox1.Text.Length - i));} }

**☞ Örnek Windows Application**

Textbox1 içine girilen yazıyı textbox2 içine tersten yazdıran program komutlarını yazınız..

privatevoid button1\_Click(object sender, EventArgs e)

{

int a, i;

string ad, tad = "";

a = textBox1.Text.Length;

ad = textBox1.Text;

a--;

for (i = a; i >= 0; i--)

{ tad = tad + ad[i].ToString(); }

textBox2.Text = tad;}}

For döngülerinin iç içe kullanımı

Şimdiye kadar gördüğümüz örneklerde for döngüsünü tek başına kullandık. Aynı koşul kontrol mekanizmalarında olduğu gibi döngüler de iç-içe kullanılabilirler.

İç-içe kullanılacak döngü sayılarında herhangi bir kısıtlama söz konusu değildir. İstediğimiz kadar sayıda döngüyü iç-içe kullanabiliriz. Çoğu zaman For döngülerindeki her etap için başka bir döngünün kurulması gerekir.

Örneğin bir müşterinin birden fazla telefon numarası bir dizi içinde tutuluyorsa, bütün müşterilerin telefonlarını listelemek için iki döngü kullanılması gerekir.

İlk döngü tek tek müşterileri almak için, alt döngü ise her müşterinin telefonlarını almak için kullanılmalıdır.

For ( Döngü değişkeni1=ilk değer;Koşul; Artırım değeri)

{

For ( Döngü değişkeni2=ilk değer;Koşul; Artırım değeri)

{

**komutlar;**

}

}

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

for (int x = 1; x <= 3; x = x += 1)

{

for (int k = 1; k <= 3; k = k += 1)

{

for (int y = 1; y <= 3; y += 1)

{

listBox1.Items.Add(" " + x + " " +" "+ k + " " +" " + y);

}

}

}

}

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

string yaz = "";

for (int i = 1; i <= 5; i++)

{

for (int k = 0; k <= 5 - i; k++)

yaz = yaz + "ismail";

listBox1.Items.Add(yaz);

yaz = "";} }

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

{

for (int x = 4; x <= 6; x = x += 1)

{

for (int k = 1; k <= 10; k = k += 1)

{

listBox1.Items.Add(x + "\*" + k + "=" + (x \* k)); }}}}

While Döngüsü

Koşul gerçekleştiği sürece çalışan döngüdür. Verilen koşul gerçekleştiği sürece işlem yapılır. Sayacı değiştirmek için kod yazılması gerekir.

**Komut dizilimi**

while(Koşul)

{

İşletilecek Komutlar

}

1’ den 10’na kadar olan sayıların toplamını hesaplamak için, 1 ve10 arasındaki sayılar tek tek yazılıp toplanabilir. İyi bir yöntem olmasa da sonucu verir. Ancak kullanıcının girdiği bir sayıya kadar toplam almak için bir döngü gerekir.

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int toplam = 0;

int sayac = int.Parse(textBox1.Text);

while (sayac >= 0)

{

toplam += sayac;

sayac -= 1;

textBox2.Text = toplam.ToString() +" "+ sayac.ToString();}}

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int s, f;

string sonuc;

f = 1;

s = Convert.ToInt32(textBox1.Text);

while (s >= 1)

{

f = f \* s;

s--;

}

sonuc = Convert.ToString(f);

MessageBox.Show(sonuc, "Faktöriyel", MessageBoxButtons.OK); }

**☞ Örnek Windows Application**

Textboxlara girilen sayıların üs hesabı yapan c# programı.

privatevoid button1\_Click(object sender, EventArgs e)

{

int s, u, sonuc = 1;

s = Convert.ToInt32(textBox1.Text);

u = Convert.ToInt32(textBox2.Text);

while (u >= 1)

{

sonuc = sonuc \* s;

u--;

}

textBox3.Text = sonuc.ToString();}

**☞ Örnek Windows Application**

Burada kullanıcının girdiği sayıdan itibaren sıfıra kadar giden bir döngü kurulur. Döngü sayacın sıfırdan büyük olduğu her durum için çalışacaktır. Sayaç sıfırlandığında ise döngüden çıkılır.

privatevoid button1\_Click(object sender, EventArgs e)

{

int toplam = 1;

while (toplam <= 555)

{

toplam++;

}

label1.Text = toplam.ToString();}

**☞ Örnek console Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int count = 1;

while (count <=4)

{

MessageBox.Show("Saiyı değer= " + count);

label1.Text = Convert.ToString("Saiyı değer= " + " " + count);

count = count + 1; }}

do …While Döngü yapısı

do while’da komutlar en az bir kere çalışır. While’da ise önce koşula bakar doğru ise çalıştırır.

Komut Dizilimi

do

{ komutlar;}

while(şart);

Burada kullanıcının girdiği sayıdan itibaren sıfıra kadar giden bir döngü kurulur. Döngü önce bir kere çalışacaktır. Sonra sayaç sıfırdan büyük olduğu her durumda çalışacaktır. Sayaç sıfırlandığında ise döngüden çıkılır.

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int toplam = 1;

do

{

toplam++;

}

while (toplam <= 555);

label1.Text = toplam.ToString(); }

**☞ Örnek Console Application**

privatevoid button1\_Click(object sender, EventArgs e)

{ int count = 1;

do

{

MessageBox.Show(" Bir kere yazar= "+ count);

count++;

}while (count <=4); }

Foreach Döngüsü

foreach, dizi (Array) ve koleksiyon ( collection ) tabanlı nesnelerin elemanları üzerinden ilerleyen bir döngüdür. Genellikle dizilerle birlikte kullanılan döngüdür. Genel kullanım şekli şöyledir;

Dizilimi

foreach (Tip Değişken in Koleksiyon)

{

İşletilecek komutlar;

}

|  |  |
| --- | --- |
| *Tip* | *Tip koleksiyonun veri tipi ile aynı veya uyumlu olmak zorundadır.* |
| *Değişken* | *foreach döngüsü içerisinde koleksiyonda bulunan sıradaki elemanı temsil eder.* |
| *Koleksiyon* | *ArrayList ya da dizi gibi aynı tip verileri barındıran koleksiyon.* |

**☞ Örnek Windows Application**

Buton tıklandığı zaman, haftanın günlerini listbox içinde görüntüleyen program kodlarını yazınız.

privatevoid button1\_Click(object sender, EventArgs e)

{

string[] dizi = { "Pazaratesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar" };

foreach (string x in dizi)

{

listBox1.Items.Add(x); }}

Sonsuz Döngüler

While döngüsü sayaç ile kullanılırken, sayacın değiştirilmesine dikkat edilmesi gerekir. Eğer sayaç değiştirilmezse, While ifadesindeki koşul hep true değeri alacağı için sonsuz döngüye girilir.Sadece sayacın kontrol edilmediği durumlarda değil, koşulların yazılmalarındaki mantık hataları da sonsuz döngüye sebebiyet verir.

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

int i = 0;

while (i < 10 | i > 5)

{

label1.Text = "Sonsuz döngüye girildi";

i += 1; } }

For döngüsünde sayaç, artırma ifadesindeki değerden fazla bir sayıda azaltılırsa yine sonsuz döngüye girilir. Bu döngünün çalışması, int veri tipinin alabileceği minimum değere ulaşınca hata ile sonlanır.

**☞ Örnek Windows Application**

privatevoid button1\_Click(object sender, EventArgs e)

{

for(int i = 0; i <= 9; i += 3)

{

MessageBox.Show("Sonsuz döngü");

i -= 4; }}

**☞ Örnek Windows Application**

1’den 49 kadar olan sayılyı rasgele bulan ve bunu mesajbox’da görüntüleyen program kodlarını yazınız.

privatevoid button1\_Click(object sender, EventArgs e)

{

int sayi;

Random rs = newRandom();

sayi = rs.Next(1, 49);

MessageBox.Show(sayi.ToString()); }

**☞ Örnek Windows Application**

*a \* x2 + b \* x + c = 0 şeklindeki 2. Dereceden bir denklemin köklerini bulan program kodunu yazınız.F*orma 5 adet T***extBox*** ve 2 adet ***Button***ekleyiniz. ***Button1***’in Text özelliğini **Kökleri bul** ve ***Button2’***nin Text özelliğini **Temizle** ismini veriniz.

**Button1\_Click** olayına aşağıdaki kodu yazınız.

public Form1()

{ InitializeComponent();}

privatevoid button1\_Click(object sender, EventArgs e)

{try

{

double b = 3;

double a,c, delta, x1, x2, z;

a = Convert.ToDouble(textBox1.Text); //textBox1'in içine yazılanı double'a çevirip a'ya atar.

b = Convert.ToDouble(textBox2.Text);

c = Convert.ToDouble(textBox3.Text);

delta = b \* b-4 \* a \* c;

if (delta < 0)

{

delta = Math.Abs(delta); //Math.Abs(delta) kodu, delta'nın mutlak değerini alır. 12

z = Math.Sqrt(delta) / (2 \* a); //Math.Sqrt(delta) kodu, delta'nın karekökünü alır.

if (b == 0) //delta < 0 ve b == 0

{

textBox4.Text = "+" + z.ToString() + "i";

textBox5.Text = "‐" + z.ToString() + "i";

}

else//delta < 0 ve b!= 0

{

textBox4.Text = Convert.ToString((-b / (2 \* a))) + " + " + z.ToString() + "i";

textBox5.Text = Convert.ToString((-b / (2 \* a))) + " ‐ " + z.ToString() + "i";

} }

else//delta >= 0

{

x1 = (-b + Math.Sqrt(delta)) / (2 \* a);

x2 = (-b - Math.Sqrt(delta)) / (2 \* a);

textBox4.Text = x1.ToString();

textBox5.Text = x2.ToString(); } }

catch

{

MessageBox.Show("Hata oluştu."); }}

privatevoid Form1\_Load(object sender, EventArgs e)

{

textBox1.Text = "1";

textBox2.Text = "0";

textBox3.Text = "‐4";

textBox4.Text = "birinci kök";

textBox5.Text = "ikinci kök";

}

privatevoid button2\_Click(object sender, EventArgs e)

{

textBox1.Text = "1";

textBox2.Text = "0";

textBox3.Text = "-4";

textBox4.Text = "birinci kök";

textBox5.Text = "ikinci kök"; }