

DELPHI'DE ŞARTLI ÇALIŞMA VE BLOK KONTROL İŞLEMLERİ

IF THEN KONTOL DEYİMİ– İstenilen şartın gerçekleşmesiyle, şarta bağlı olan kodlar işletilir. Ne dediğimi şimdi anlarsınız :) If-then deyiminin kalıbı şöyledir.

```
If Şart Then  
begin  
.  
.  
end  
else //-> Şuanlık kullanmayacağız.  
begin  
.  
.  
end;
```

Örnek: Yazdığımız programda bir butona tıklanıldığında daha önce dediğimiz şifrenin doğru olup olmadığını söylesin. Şifre **abc123** olsun. Buton'umuz için yukarıda bulunan **Standart** sekmesinde **OK** yazan dikdörtgen şeklindeki butona tıklayın ve daha sonra başka hiç bir yere tıklamadan direk formun üzerine tıklayın. Formun üzerinde bir buton oluştuğunu göreceksiniz. Bu butonun üzerine çift tıklayın, ve önünüze **kod penceresi** gelecektir.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin //-> Buton1 tıklanıldığında buna ait olan kodlar çalışacak.  
begin //-> Kodların yazılması için bunun yazılması şarttır. Kodların artık başladığını belirtir.
```

```
end; //-> Buna en yakın begin hangisiyse, o begin ile buradaki end arasındaki kodlar son bulmuş anlamına gelir.  
Begin gibi bunun da yazılması gerekir.
```

```
end. //-> Buradaki end'le bundan önceki end'i karıştırmayın. Buradaki end ise programın tamamıyla son bulunduğunu anlatır. Ve dolaylı olarak projede sadece ve sadece 1 kez ve en sonda olarak kullanılır. Birisinin sonu ";" ile birisinin sonuda "." ile biter.
```

```
procedure TForm1.Button1Click(Sender: TObject);  
begin //-> Başlıyoruz.  
If Edit1.Text='abc123' then //-> Eğer ki Edit1'in içinde abc123 kelimesi yazıyorsa (veya içindeki abc123 yazısına eşitse) bunları işle...  
ShowMessage('Şifre Doğru'); //-> Şifre doğruysa ekranımıza Şifre Doğru diye bir mesaj çıkartacak.  
end; //-> Button1 tıklanıldığında yapacağı görevler böylece bitiyor.  
end. //-> Program sonu.
```

Birde if-then kontrol'üne ek olarak bir deyim daha kullanılır bu da **else**'dir anlamı: **değilse?** Şart gerçekleşmez ise?

```
procedure TForm1.Button1Click(Sender: TObject);  
begin //-> Başlıyoruz.  
If Edit1.Text='abc123' then //-> Eğer ki Edit1'in içinde abc123 kelimesi yazıyorsa bunları işle...  
ShowMessage('Şifre Doğru') //-> Şifre doğruysa ekranımıza Şifre Doğru diye bir mesaj çıkartacak.  
else -> //-> Ya edit'in içinde abc123 yazmıyorsa?  
ShowMessage('Şifre Yanlış'); //-> Bu sefer Şifre Yanlış diye bir mesaj karşımıza gelecek.  
end; //-> Button1 tıklanıldığında yapacağı görevler böylece bitiyor.  
end. //-> Program sonu.
```

Burada dikkat etmemiz gereken birkaç nokta var onlara deyinayım. Her kod satırının sonuna ; gelmelidir. Bunun anlamı o satırdaki

kodun artık sona erdiğini ve dolaylı olarak bundan sonraki kodun işleyeceğini belirtir. Siz şimdi diyeceksiniz ki programa else eklediğiniz zaman ShowMessage('Şifre Doğru'); buradaki ; kalktı niye? Bunun anlamı demiştik ; o satırdaki kodları bitirir. Ama aslında burada kodlar bitmiyor halen devam ediyor. yani else If Edit1.Text='abc123' then komutunun parçası ve ondan önce bir ; gelirse bu onun işleyişini engelleyecektir. Bu da onun anlamsız yere yazıldığı anlama gelecektir, ve program hata verektir. Deyinmek istediğim başka bir konu ise Edit1.Text='abc123' burada abc123'ün başına ve sonuna niye '(tırnak)' işareti geldiğidir. Bunun sebebi abc123 bir string'dir yani bir karakter katarıdır. Mesela oraya '123' yazsaydınız bu bir rakam değil sadece yazı olacaktı eğer ki sadece 123 yazsaydınız rakam olacaktı. Niye böyle olmak zorunda olduğunu soracak olursanız belli bir zaman geçtikten sonra, bunu sizde olmazsa olmaz olarak kabul edeceksiniz. Şimdi gelelim bir sonraki blok-kontrol deyimine...

CASE OF DEYİMİ= Herşeyi ile **if-then** deyimine benzetmektedir. Bazı zamanlarda **if-then**'den daha pratiktir. İç içe geçmiş bir **if-then** deyimleri gibidir. Fakat pratik olmasının sebebi defalarca **if-then** deyimini yazmak yerine bir tek komutla hepsini birden kontrol etmeği sağlamaktadır. Kalıbını yazmadım aşağıdaki örnekte açıkça belli.

Örnek: Bu programda ise bir butona tıkladığımızda edit1'de yazdığımız notumuzu, bizim zayıf orta gibi durumumuzu söyleyen bir mesaj çıkarsın...

```
procedure TForm1.Button1Click(Sender: TObject);
var //-> Değişken yazmak için kullanır. Bu koddan sonra değişkenler atanmaya başlar.
notum : integer; //-> notum isminde bir tamsayı(integer) değişkeni atıyoruz. Sadece içerisinde sayı tutabilir. Bu
değişkenleri Değişik branşlardaki öğretmenler gibi düşünün bilirsiniz. Hepsinde farklı farklı işler yapar. Ve biz ona göre
onun dersinden faydalanırız. Mesela yukarıda bir string'den bahsetmiştim, bu da bir değişkendir ve içerisinde yazı
katarları tutabilir. Ne istiyorsak ona göre değişkenimizi
doğru belirlemeliyiz. Aksi takdirde program abuk sabuk işlemler yapar veya hata verip çalışmaz bile. Neyse konuya
dönelim.
begin //-> Başlıyoruz.
notum:=strtoint(Edit1.Text); //-> İşte burası çok önemli buradaki notum'un sadece rakamları tutabileceğini
öğrenmiştik. Fakat edit'in içinde hiç bir zaman rakam değil yazı vardır. Edit içine yazdığınız rakamlar rakam gibi
gözüksede aslında onlar bir rakam değildir daha
önce dediğim gibi onlar bir string(karakter katarı)dir. İşte bundan dolayı notum bir rakam edit1.text(edit'in içindeki
yazı) bir yazıdır. Bundan dolayı Edit1.Text'in içinde yeralan rakamları gerçekten birer rakam yapmamız gerekiyor.
Yukarıda olduğu gibi Edit1.Text'i iki parantez içine almak zorundayız (). Ve daha sonra bunu strtoint ile string'den
integere çevirmeliyiz. Yani yazıdan rakama. Zaten adı üzerinde String to Integer, StrToInt olmuş. Bunu tam tersi ise
IntToStr'dir. Peki bu ne demektir düşünün bakalım? Sonuç olarak birini
birine mutlaka benzetmeliyiz aksi takdirde elma ile armut toplanmaz. Program gereyi yazıdan karaktere döştürmek
zorundayız.
Case notum of //-> notum değişkeni attakilerden hangisine eşitse onu yap.
0..44: Showmessage('Zayıf Not Aldınız'); //-> Eğer Edit'de 0 ile 44 arası bir rakam girilmişse Zayıf not aldınız diye bir
mesaj kutusu çıkar. Diğerleride aynı.
45..59: Showmessage('Başarısız Not Aldınız');
60..69: Showmessage('Orta Not Aldınız');
70..84: Showmessage('İyi Not Aldınız');
85..100: Showmessage('Pekiyi Aldınız tebrikler!');
else Showmessage('Lütfen 0 ile 100 arası bir not girin'); //-> Eğerki girdiğiniz sayı 0 ile 100 arası değilse bu mesajı
çıkartıyor. Zaten 0 ile 100 arası bir rakam olsaydı çoktan yukarıdakilerden biridi uygulamıştı.
end; //-> Case-of deyimini sonlandırıyor.
end; //-> Buton Basıldığında yapacak olduğu işler artık bitti. İlk baştaki begin ile en sondaki end; i yazmanıza gerek
yok zaten Delphi onları kendisi koyar.
```

Yukarıda anlattığım gibi, eğer **Case-of** kullanmasaydık her durum için yani 0 ile 44, 45 ile 59 vs.. farklı farklı diyecektik ki şunla şunun arasındaysa şunları yap, şunla şunun arasındaysa şunları yap gibi bir çok **if-then** deyimini kullanacaktık. Başlangıçta pratik dememin sebebi buydu. Örnekte maksimum ve minimum sayılar yanlış olabilir çünkü %85'inde yazılardan 5 alıyorum da :) söylemesi ayıp. :) Neyse bir sonda ki...

FOR TO DO DÖNGÜSÜ= İçerisindeki komutları istenilene göre tekrar tekrar işler. For-to-Do döngüsü kalıp olarak şöyle kullanılır.

```
For başlangıç değeri to bitiş değeri Do
Begin
.
.
end;
```

Örnek: Programı yazamaya başlamadan önce yukarıdaki **Standart** sekmesinden **ListBox**(Liste kutusu) isimli bileşeni formun üzerine yerleştirin. Bu arada bütün örneklerimizde bir tane buton kullanıyoruz. Her zamanki gibi onuda koymayı unutmayın. Bu programla verilen sayılar arasındaki sayıları **ListBox**'a ekleyeceğiz.

```
procedure TForm1.Button1Click(Sender: TObject);
var
```

ilkdeger : integer; //-> Tam sayı değişkenimiz.

begin //-> Başlıyoruz.

For ilkdeger:=1 to 20 do //-> 1'de başlayarak bu komutları 20 kere tekrarla eğer i'ye 5 verseydik 5'den başlayacaktı yani bu olayı 15

kere tekrarlayacaktı. Burada bilmeniz gerek önemli bir husus daha var fakat bunu biraz sonra açıklayayım.

begin //-> Döngünün başlangıcı.

ListBox1.Items.Add(inttostr(ilkdeger)); -> Burayla uzun bir süre ilgileneyim. Bu komut "ilkdeger" integer'ını her seferinde her seferinde listbox(liste)a ekliyor. peki bu nasıl oluyor. Şimdi programın en başına dönelim for-to-do döngüsünün başlagıç değerini 1 olarak

atamıştık. ve progarmımız geldi geldi geldi. İlk değerimiz o anlık 1'di ve Listbox'a 1'i yazdı. Daha sonra döngü tekrar başa döndü bu sefer başlangıcı (bir anlamıda sayaç deniliyor, counter) 1 arttı ve sayacımız 2 oldu ve geldi geldi geldi Bu sefer sayac yani ilkdeger 1 artırılıp

2 olduğu için Listbox'a bu sefer 2 eklendi. Bu böyle devam edecek. Fakat nereye kadar? Tabiki 20'ye kadar çünkü döngünün bitiş

noktasını 20 olarak daha önce belirlemiştik. Sonuç olarak program Listbox'a 1'den 20'ye kadar sıralayacak.

end; //-> For-to-Do döngüsü sona eriyor.

end; //-> Button1 tıklanıldığında yapacağı görevler böylece bitiyor.

end. //-> Program sonu.

Birde if-then döngüsündeki else gibi for-to-do 'nun bir bir de break diye bir deyimi vardır. Bunu da Döngüyü önceden neler yapacağını ne kadar devam edeceğini belirtiyorduk, bu komut ile istenilen zamanda daha dönünün işi bitmeden döngüden çıkılabilir. Mesela

döngünün içine bir if-then şartı koyarsınız eğer döngü işlenirken bu şarta uyulduğu görüldüyse ve if-then şartının içinde de break

komutu yer alıyorsa o şarta bağlı olarak döngü içerisinden çıkmış olur. Başka bir konu ise For ilkdeger:=1 to 20 do 'da ortada bulunan

to eğer downto olarak değiştirilirse bu sefer geriye doğru sayım yapılır. Mesela bu sefer 100'de 20'ye kadar sayar. Yani döngü 80 kere dönmüş olur. Gelelim bir sonraki döngüye...

WHILE DO DÖNGÜSÜ= Bu döngü aslında For-to-Do döngüsünün tıpatıp aynısıdır. Tek fark while-do döngüsünde önceden bir başlangıç

veya bitiş bokaası gibi kısıtlamalar yoktur. Yani bu döngünün içerisindeki şart gerçekleşmediyi sürece döngü sonsuza kadar sürecektir. Kalıbı şöyledir.

While koşul1 do

begin

.

.

if koşul2 then

Break; //-> Yukarıda anlaşıldığı gibi, koşul2 gerçekleşmedikçe döngü devamlı dönecektir.

end;

SON NOTLAR: Eğer bu bilgilerle size yardımcı olabilmişsem ne mutlu bana. Eğer Delphi konusunda çok yeniyseniz, Ve bazı sorunlarınız varsa bana aytekin@altavista.com adresinden ulaşabilirsiniz. Kendinize iyi bakın.

Bu text herhangi bir kaynağa bakılmadan Aytekin tarafından yazılmıştır.