**KONTROL YAPILARI**

**KONTROL YAPILARI**

İşlem bloklarında koşullara göre program akışını kontrol etmeye yarayan ifadelerdir. Kontrol yapıları 3 gruptan oluşur.

· Sırasal

· Bölümlere ayrılan (dallanan)

· Döngü

1. Sırasal

WAIT ifadesi kullanılır. Koşulsuz olarak çalışır. Belli bir süre boyunca programın çalışması duraklatılır.

2. Bölümlere Ayrılan (Dallanan)

Koşula göre bir veya daha fazla ifade bloğu tanımlanabilir. IF ve CASE ifadeleri kullanılır.

3. Döngü

İfade blokları DO-ENDDO ve WHILE-ENDWHILE ifadeleri arasında bir veya daha fazla çalışır.

**WAIT**

Programın çalışmasını saniye cinsinden belirtilen süre kadar durdurur. Program süre bitiminde kaldığı yerden devam eder. Kullanımı aşağıdaki şekildedir.

WAIT UP TO sure SECONDS.

**IF**

Mantıksal koşullara bağlı bir veya daha fazla işlem bloğu içerir. İlk mantıksal ifadeden başlayarak aşağıya doğru koşullar kontrol edilir ve koşulu doğru olan ifade bloğu çalıştırılır. IF ile ilk koşul yazılır. ENDIF ile IF bloğu kapatılır. IF ve ENDIF arasında ELSEIF ve ELSE koşulları kullanılabilir. Kullanımı aşağıdaki şekildedir.

IF mantiksal\_ifade1.

[islem\_blogu1]

[ELSEIF mantiksal\_ifade2.

[islem\_blogu2]]

…

[ELSE.

[islem\_blogun]]

ENDIF.

**CASE-ENDCASE**

Karşılaştırma yapılacak değere göre birden fazla işlem bloğu içerir. Değere uyan ilk işlem bloğu çalışır. CASE ile karşılaştırma yapılacak değer belirtilir. WHEN ile değere karşılık çalışması istenen işlem blokları tanımlanır. Kullanımı aşağıdaki şekildedir.

CASE karsilastirilan\_deger.  
[WHEN deger1 [OR deger2 [OR deger3 […]]].  
[ilsem\_blogu1]]  
…  
[WHEN OTHERS.  
[ilsem\_blogun]]  
ENDCASE.

**DO-ENDDO**

Koşulsuz döngü için kullanılır. DO ve ENDDO arasında koşulsuz olarak çalışması istenen işlem bloğu yer alır. Döngü sayısı belirtilebilir. Belirtilmeden kullanılır ise döngüden çıkmak veya yönetmek için EXIT, CONTINUE veya CHECK ifadeleri kullanılır. Kullanımı aşağıdaki şekildedir.

DO [n TIMES].  
[ilsem\_blogu]  
ENDDO.

**WHILE-ENDWHILE**

Koşullu döngü için kullanılır. WHILE ve ENDWHILE arasında koşullu olarak çalışması istenen işlem bloğu yer alır. Mantıksal ifade doğru olduğu sürece işlem bloğu çalışır. Döngüden çıkmak için işlem bloğunda EXIT ifadesi de kullanılabilir. sy-index sistem değişkeni gerçekleşen döngü sayısını tutar. Kullanımı aşağıdaki şekildedir.

WHILE mantiksal\_ifade.

[ilsem\_blogu]

ENDWHILE.

**KONTROL YAPILARI ÖRNEKLER**

1. Önce geçerli saati yazan daha sonra 7 saniye programın çalışmasını bekleterek tekrar saati yazan örnek.

WRITE: ‘Saat’, sy-uzeit.  
WAIT UP TO 7 SECONDS.  
WRITE: / ‘Saat’, sy-uzeit.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275421eb8e59.png

2. Döngü içerisinde sy-index değişkenini önce 1 değeri, daha sonra 2 değeri ile karşılaştıran IF –ENDIF örneği.

DO 3 TIMES.  
IF sy-index = 1.  
WRITE: / ‘İlk işlem bloğu’, 30 sy-index.  
ELSEIF sy-index = 2.  
WRITE: / ‘İkinci işlem bloğu’, 30 sy-index.  
ELSE.  
WRITE: / ‘Koşullara uymayan işlem bloğu’, 30 sy-index.  
ENDIF.  
ENDDO.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275422736909.png

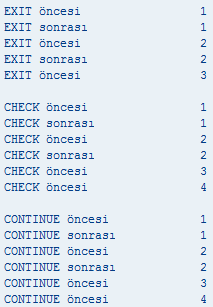
3. Döngü içerisinde farklı koşullara sahip işlem blokları kullanımını gösteren CASE-ENDCASE örneği.

DATA gv\_kosul TYPE c LENGTH 25.  
DO 3 TIMES.  
WRITE sy-index TO gv\_kosul.  
CONDENSE gv\_kosul NO-GAPS.  
CONCATENATE ‘Kosul’ gv\_kosul INTO gv\_kosul.  
CASE gv\_kosul.  
WHEN ‘Kosul1′.  
WRITE: / ‘İşlem bloğu 1.’, 25 gv\_kosul.  
WHEN ‘Kosul2′.  
WRITE: / ‘İşlem bloğu 2.’, 25 gv\_kosul.  
WHEN OTHERS.  
WRITE: / ‘İşlem bloğu OTHERS.’, 25 gv\_kosul.  
ENDCASE.  
ENDDO.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_5275422e2a9fb.png

4. DO-ENDDO ile EXIT, CHECK ve CONTINUE ifade kullanımını gösteren örnek.

DO 4 TIMES.  
WRITE: / ‘EXIT öncesi’, 20 sy-index.  
IF sy-index > 2.  
EXIT.  
ELSE.  
WRITE: / ‘EXIT sonrası’, 20 sy-index.  
ENDIF.  
ENDDO.  
SKIP.  
DO 4 TIMES.  
WRITE: / ‘CHECK öncesi’, 20 sy-index.  
CHECK sy-index < 3.  
WRITE: / ‘CHECK sonrası’, 20 sy-index.  
ENDDO.  
SKIP.  
DO 4 TIMES.  
WRITE: / ‘CONTINUE öncesi’, 20 sy-index.  
IF sy-index < 3.  
WRITE: / ‘CONTINUE sonrası’, 20 sy-index.  
ELSE.  
CONTINUE.  
ENDIF.  
ENDDO.



5. WHILE-ENDWHILE kullanımını gösteren örnek.

DATA: gv\_index TYPE i.  
WHILE gv\_index < 3.  
WRITE: / ‘WHILE ifadesi içerisinde index:’, 32 gv\_index.  
gv\_index = gv\_index + 1.  
ENDWHILE.  
WRITE: / ‘WHILE ifadesinden sonra index :’, 32 gv\_index.

