

## HAFTA 11

### LOOP –UNTIL Döngü Deyimi

LOOP-UNTIL döngü deyimi, yapılması gereken işlem ya da işlemlerin herhangi bir koşulun sağlanmasına gerek olmadan bir kez yapılmasını ve sonrasında, koşul sağlanmadığı müddetçe söz konusu işlem/işlemlerin tekrarlı olarak yapılmasını sağlar. Yani, önce işlem/işlemler yapılır, koşul daha sonra kontrol edilir. LOOP-UNTIL döngü deyiminin çalışması, koşulun sağlanması ile sona erer. Döngünün genel yapısı aşağıdaki gibidir:

LOOP

İşlem / İşlemler

UNTIL koşul REPEAT

**Örnek 1:** Bir çikolata firması 100 gr çikolatayı 3.5 TL'ye mal edip, 7 TL'ye satmaktadır. Firma, 500 ve 999 adet arasında sipariş veren müşterisine %1, 1000 ve 1499 adet arasında sipariş verene %2.5, 1500 adet ve üzerinde sipariş verene ise %5 indirim yapmaktadır. İşletmenin 1 ay sonundaki kârını, kâr yüzdesini ve toplam satış miktarını bulup dış ortama aktaran bir algoritma yazınız.

**NOT:** 1 gün içerisinde 1'den fazla müşteriden sipariş gelebilmektedir.

**Örnek 2:** Bir kişinin bankada 10000 TL tutarında birikimi bulunmaktadır. Bankanın aylık faiz getirisi %0.75'tir. Yatırım sahibi, parasını bankada tutmak yerine hisse senedi alıp satarak da kazancını arttırmayı düşünmektedir. Bunun için hisse senedi fiyatları % 5 ve altında düşerse, parasının tümüyle hisse senedi almayı; hisse senedi fiyatları %10 ve üzerinde artarsa da senetleri satmayı ve elde ettiği parayı tekrar bankaya yatırmayı planlamaktadır. Hisse senedi fiyatı ilk ay için 500 TL'dir. Hisse senedi satışında herhangi bir maliyet söz konusu değildir. Eğer hisse senedi alımı sonunda artan para kalırsa, kalan miktar yine bankaya yatırılacaktır. 5 yıl sonunda, yatırım sahibinin parasının ne kadar olacağını bulan ve görüntüleyen algoritmayı tasarlayınız.

**Örnek 3:** Bir sınıfta 80 öğrenci bulunmaktadır ve 3 adet zorunlu dersin not ortalaması belirlenmek istenmektedir. Öğrencilerin her ders için dönem sonu ortalamaları girilmektedir. 3 dersin not ortalamasını bulduran ve sonucu yazdıran algoritmayı tasarlayınız.

## DİZİLER

Algoritma tasarımı, şimdiye kadar değer atamaları için tek değişken kullanılmıştır. Dış ortamdan girilen ya da algoritma içerisinde türetilen tüm değerlerin korunması istendiğinde, her değer için ayrı değişken tanımlanması gerekmektedir. Korunmak istenen değer sayısının artması, değişken sayısının da çok fazla artmasına neden olur. Örneğin klavyeden girilen 10 sayıya daha sonra da ulaşmak istenirse her biri için ayrı değişken kullanılması gerekir. Eğer girilen sayılar tek bir değişkene atanırsa, değişkende kendisine atanan son değer korunacaktır; diğerleri ise kaybolacaktır. Sayı daha fazla olduğunda ise kullanılacak değişken sayısı çok daha fazla olacaktır.

Bu olumsuzluğun önüne dizi değişken kullanılarak geçilebilir. Dizi değişkende, girilen ve türetilen tüm değerler korunabilir ve bu nedenle algoritma içerisinde gerektiğinde ulaşılabilir. Dizi değişkenler tek boyutludur. Dizi tanımlaması yapılırken dizi adıyla birlikte boyutunun da verilmesi gerekmektedir.

### Örnek:

Not [30]; Ad[25]; Soyad[25] vb.

Bir şehrin bir haftalık hava sıcaklığı değerlerini depolamak için oluşturulacak dizide, SICAKLIK[7], her biri eşit uzunlukta olan 7 yer ya da saha bulunacaktır:

SICAKLIK[1]	SICAKLIK[2]	SICAKLIK[3]	SICAKLIK[4]	SICAKLIK[5]	SICAKLIK[6]	SICAKLIK[7]

Yukarıda, köşeli parantezler içindeki sayılar, dizinin indisleridir. Girilen her sıcaklık değeri, farklı indisle belirlenmiş sahalarda depolanır. Örneğin, sıcaklık değerleri aşağıdaki gibi olsun:

22 – 18 – 21 – 22 – 25 – 26 - 28

22						
SICAKLIK[1]	SICAKLIK[2]	SICAKLIK[3]	SICAKLIK[4]	SICAKLIK[5]	SICAKLIK[6]	SICAKLIK[7]

Sıcaklık dizisinin içeriği belirlendikten sonra bu değerler, üzerine başka değer atanmadığı sürece, dizide korunurlar. Değerler, indis adresi ile birlikte, başka değişkenlere aktarılabilirler:

Pazartesi ← SICAKLIK[1]

Bu şekilde Pazartesi isimli değişkene, Sıcaklık dizisinin 1 indisli sahasındaki 22 değeri atanır. Şimdi de dizi değişkenlere nasıl değer aktarılacağını inceleyelim:

```
SICAKLIK[7]
FOR I ← 1 TO 7 DO
    PRINT 'Sıcaklık değerini giriniz:'
    READ SICAKLIK[I]
REPEAT
```

Bu örnekte FOR döngüsündeki I değişkeni, dizinin hangi hücreğine atama yapılacağını belirlemektedir. FOR döngüsü, sıralı bir şekilde dizi değişkene atama yapılmasını sağlamaktadır ancak, sıralı atama zorunlu değildir. Örneğin SICAKLIK[5] ← 22 ile dizi değişkenin 5. alanına 22 değeri atanmaktadır. Bu şekilde 7 sayı girişi yapılabilir:

```
SICAKLIK[7]
I←1
WHILE I < 8 DO
    PRINT 'Değer girişi yapılacak alanın numarasını giriniz:'
    READ J
    PRINT 'Sıcaklık değerini giriniz:'
    READ SICAKLIK[J]
    I ← I+1
END
```

LOOP-UNTIL döngüsü ile de düzensiz giriş yapılabilir.

```
SICAKLIK[7]
LOOP
  PRINT 'Değer girişi yapılacak alanın numarasını giriniz:'
  READ J
  PRINT 'Sıcaklık değerini giriniz:'
  READ SICAKLIK[J]
  PRINT 'Başka değer girmek istiyor musunuz? E/H:'
  READ Cevap
UNTIL Cevap='H' REPEAT
```

Örneğin LOOP-UNTIL döngüsünde J değerleri için sırasıyla 3, 6 ve 1 girilmiş olsun. 3. Girişten sonra Cevap ← 'H' atamasıyla LOOP-UNTIL döngüsünden çıkıldığı farz edilsin. Dizi aşağıdaki gibi olabilir:

21		22			18	
SICAKLIK[1]	SICAKLIK[2]	SICAKLIK[3]	SICAKLIK[4]	SICAKLIK[5]	SICAKLIK[6]	SICAKLIK[7]

Bu durumda, öncelikle 3 indisli alana 22 değeri, 6 indisli alana 18 değeri ve 1 indisli alana da 21 değeri girilmiştir.

**ÖRNEK 1:** Fibonacci dizisi aşağıdaki gibidir:

1 – 1 – 2 – 3 – 5 – 8 – 13 – 21 – 34 ....

Dizinin ilk 100 elemanını bir dizi değişkene, sayılar arasındaki ilişkiyi kullanarak atayın.

**ÖRNEK 2:** 100 adet sayı sırasıyla klavyeden girilmektedir. Girilen sayılar için aşağıdakileri bulan ve görüntüleyen algoritmayı tasarlayınız.

- Sayıların ortalaması
- Sayıların en büyük ve en küçük değerli olanları
- Ortalamadan küçük sayı adedi

**ÖRNEK 3:** Bir torbada mavi, beyaz, kırmızı, sarı ve siyah olmak üzere 5 farklı renkte top bulunmaktadır. Bu torbadan bir top çekildikten sonra tekrar torbaya geri konulmakta, bu sayede her çekilişte torbada aynı sayıda top bulunması sağlanmaktadır. 100 çekiliş sonunda

- Her renkten kaç adet çekildiğini
- En az çekilen rengi

bulan ve sonuçları görüntüleyen algoritmayı SPARKS dilinde oluşturunuz.