

# GİRİŞ/ÇIKIŞ DEYİMLERİ

## 5.1. Giriş

Giriş/çıkış deyimleri bilgisayarın ana belleği ile çevre aygıtlar arasında bilgi transferi yaparlar. Herhangi bir çevre aygıtından bir verinin belleğe okutulması veya ana bellek üzerindeki bir verinin çevre aygıtlara gönderilmesi giriş/çıkış deyimleri ile gerçekleştirilir. Genellikle giriş çıkış deyimi olarak aşağıda tanımlanan deyimler kullanılmaktadır.

## 5.2 Read-Readln

Pascal 'da okuma işlemi için kullanılan komutlar *READ* ve *READLN* olup genel kullanım yapısı şu şekildedir;

```
Read(A1,A2,A3,...,An)
```

```
ReadLn(A1,A2,A3,...,An)
```

Read ile Readln arasındaki fark: Read, okuma işlemi tamamlandıktan sonra aynı satırda kalınmasını, Readln ise yeni bir satıra gelinmesini sağlar. Bir Pascal programı içerisinde bu iki komutun kullanımı aşağıdaki program parçasında verilmiştir.

```
Var  
s,a,n:real;  
.....  
begin  
    Read(s,a,n)  
end.
```

Okunacak değerlerin düzeni;

```
12.45    43.543    62.12
```

Aynı değerleri ReadLn ile okutturmak istersek;

```
Var
s,a,n:real;
.....
begin
  ReadLn(s,a,n)
end.
```

Okunacak değerlerin düzeni yukarıdaki okuma düzeni ile aynı olabileceği gibi her değer ayrı bir satırda verilebilir, her hangi satırlarda verilebilir, örnek olarak s, a ve n değerleri aşağıdaki okuma düzeninde okutturulmuştur.

```
12.45    43.543
62.12
```

### 5.3 Write-Writeln

Pascal programlama dilinde yazma işlemi için *WRITE* ve *WRITELN* komutları kullanılmaktadır. Bu komutların genel kullanım yapısı şu şekildedir;

```
Write(A1,A2,A3,...,An)
```

```
Writeln(A1,A2,A3,...,An)
```

İki kullanım şekli arasındaki fark, Read ile ReadLn arasındaki fark gibidir. İfadeler Write ile yazdırıldıktan sonra kursör aynı satırda bekleyecektir, Writeln ile yazdırılması durumunda ise kursör bir sonraki satıra geçecektir.

```
Var
s,a,n:integer;
.....
begin
  s:=3;a:=9;n:=5;
  Write(s);
  Write(a);
  Write(n);
end.
```

Programın çalıştırılmasıyla elde edilen sonuç;

```
395
```

olacaktır.

Aynı değişkenleri Writeln ile yazdırdığımızda;

```
Var
s,a,n:integer;
.....
begin
  s:=3;a:=9;n:=5;
  Writeln(s);
  Writeln(a);
  Writeln(n);
End
```

```
3
9
5
```

elde edilecektir.

## 5.4 Yazım İçin Format Belirleme

Writeln deyimi ile yapılan bilgi çıkışlarını düzenlemek için yazım formatları kullanılır. Yazdırılmak istenilen değerin tipine bağlı olarak iki farklı yazım formatı kullanılır.

**1. Yazım Formatı ( $M:n$ )** :Bu yazım formatına göre yazdırılmak istenilen  $M$  değeri  $n$  karakterlik alanda sağa dayalı olarak yazılır.  $M$  değerini oluşturan dijital veya karakterin uzunluğu  $n$  sayısından küçük ise, aradaki fark kadar sol tarafta boşluk bırakılır.  $M$  değerinin uzunluğu  $n$  sayısına eşit veya büyük ise yazdırma işlemi format belirtilmemiş gibi gerçekleştirilir.

### Örnek:

```
Const
i:integer=567;
b:boolean=True;
c:char='b';
h:string[10]='Nasılsınız';
begin
  Writeln('123456789');Writeln;
  Writeln(i:9);
  Writeln(b:9);
  Writeln(c:9);
  Writeln(h:9);
  Writeln(i:2);
  Writeln(h:5);
end.
```

**2. Yazım Formatı (M:n:k)** : Bu yazım formatı real sayılar üzerinde çalışmaktadır. Bu yazım formatıyla reel sayılar üssüz notasyonda yazdırılır. Bunu sağlamak için sadece real sayılar için geçerli olan 2. yazım formatı kullanılır. Yazdırılacak  $M$  real değeri  $n$  karakterlik alanda sağa dayalı olarak üssüz yazdırılır,  $k$  ise  $n$  karakterin sonundaki kesirli kısmı göstermektedir.

**Örneğin;**  $A$  değeri,  $A:7:2$  şeklinde yazdırılmak istenirse; program real sayı için 7 karakterlik yer ayırır. Bu alanın son iki hanesi ondalıklı kısım için sondan 3. hanesi ise desimal nokta için kullanılır.

Sayının tamsayı kısmı ayrılan alanın uzunluğundan küçük ise, aradaki fark kadar sol tarafta boşluk bırakılır. Yazdırılacak sayının tam sayı kısmı bu alana sığmıyor ise, tamsayı için ayrılan alanın uzunluğu yazdırılan sayının tamsayı alan uzunluğuna eşit büyüklükte kabul edilir. Kesirli kısmı ayrılan alandan daha küçük ise aradaki fark kadar sağ tarafa 0 dijiti ilave edilir. Kesirli kısım alana sığmıyorsa yuvarlatılarak yazılır.

Yazım sırasında kesirli alanın yazılması istenmiyorsa  $k$  sayısının "0" yazılması gerekir.  $k$  sayısının sıfır olması desimal noktanın yok olmasını sağlar.

**Örnek:**

```
const a:real=123.127927;
begin
  Writeln('123456789');
  Writeln('*****');;
  Writeln(a:9:4);
  Writeln(a:9:3);
  Writeln(a:9:2);
  Writeln(a:9:0);
  Writeln(a:7:2);
  Writeln(a:6:2);
  Writeln(a:5:2);
  Writeln(a:0:0);
end.
```

Programın çalışmasıyla elde edilen sonuç aşağıda verilmiştir. Program sonucuna göre sondan 2. ve 3. satırların aynı olduğuna dikkat ediniz.

```
123456789
*****
123.1279
123.128
123.13
123
123.13
123.13
123.13
123
```