

[Anasayfa](#) / [Basit Dili Kuralları](#)

İşaret

Her verinin normalde kesin bir adresi vardır. Bu durum statiktir. Yani söz konusu verilerin adresleri bellidir, çünkü kodda kesin olarak belirtilmiştir.

Fakat bazen öyle senaryolarla karşı karşıya kalırsın ki, bir tek adres üzerinden farklı adreslere erişmek zorunda olursun. Söz konusu adresler bellidir, ama o an ki kod bölümünde her biriyle tek tek iletişim kurmak optimal değildir.

Bazende başka bir geçerlilik alanında bulunan bir adrese erişim gerekir, mesela yazmak için. Bundan öte kodda belirlenmemiş program gidişatı esnasında sonradan ihtiyaç duyulduğu için rezerve edilen adres alanları olabilir. Onlara erişimde tabiki gereklidir.

Bu durumların her birinde işaretler yardımcı olur. İşaret adı üstünde bir şeye işaret eder. İşaret ettiği ise bir adrestir. Tüm değişkenler ve dizilimlerde sonuçta adrestir.

Basit işaretleri üç ayrı modelde organize eder.

Model 1: Türe göre işaret

Bu işaretler gerçekten tanımlanmış olan bir değişken veya dizilim için sunulmaktadır. Türe göre olması sayesinde hata yapma olasılığını ortadan kaldırır.

Mesela Rakam türünden bir değişkene işaret eden bir İşaret tanımı asla KüçükRakam, EsnekRakam, Virgöl, Esnek, Metin veya Karakter türünden bir değişkeni kabul etmez.

Böylece kodu yazarken doğru işaretlemek zorunda kalır, hataları hemen tespit edip, düzeltebilirsin.

Bu tür bir işareti oluşturmak çok kolaydır. Normal bir değişken veya dizilim tanımlama modeliyle aynıdır, sadece "İşaretli" anahtar kelimesi önüne gelir:

```
Değişken İşaretli Rakam a
Dizilim İşaretli Metin
```

Gerçek bir değişken veya dizilim adresini bir işarete vermek şu biçimde gerçekleşiyor:

```
Değişken Rakam ben = 5
Değişken İşaretli Rakam sen

sen = ben
```

İşaret okunurken hep işaret ettiği adresin verisini verir:

```
Denetim sen ? sonuç olarak 5 verecektir
```

İşareti yazarken, ancak kendi türünden bir adres, yani değişken veya dizilim kendisine sunulabilir.

Bunlar dışında sadece 0 ve Boş komutu kabul edilir. 0 veya Boş, işareti sıfırlar. Yani işaret herhangi bir adrese işaret etmez. 0 kalır. Alternatif yöntem:

```
Yoket sen
```

aynı sonucu doğurur.

İşaretin yazım için kabul ettiği tek operatör = operatörüdür. Diğer durumlarda derleyici hata verecektir.

Model 2: Türleri temsilen işaret

Bu işaretler belli bir türe göre değildir. Tüm türlerin adresini kabul eder. Bu ancak adresin matematiksel değerine ulaşmak için bir imkandır. Bundan ziyade elde edilen adres ancak doğru türden bir işarete verilmelidir. Aksi taktirde hata oluşabilir.

Bu modelden işaret "İz" türündendir.

Örnek:

```
Değişken İz benimİşaretim
```

Kullanım kuralları Model 1 ile aynıdır.

Model 3: Bellek alanına işaret

Bu işaret "İşaret" komutu ile sunulur. "İşaret" kendine has bir kuraldır ve arabellekte gerçekten bir alan rezervasyonu yapar. Değişkenin kendisi bu alanın giriş adresini sunar. [Bellek Modülü](#) ile bu alan içinde her türlü işlem yapılabilir.

Tanımı:

```
? Boş bir işarettir
İşaret işaretimBenimBoş

? 100 Bayt kapasiteye sahip bir bellek alanına işaret eder
İşaret işaretimBenimDolu Boyut 100
```

veya

```
? Boş bir işarettir
İşaret işaretimBenimBoş

? 100 Bayt kapasiteye sahip bir bellek alanına işaret eder
İşaret işaretimBenimDolu Boyut 100 B
```

İşaret rezerve edeceği alan için bir sabit değer kabul eder, ardından gelen bir tanı ne olduğunu ifade eder:

Tanı	Manası
B, Bayt, Byte	Bayt
kb, kib, kilobayt, kilobyte	Kilobyte
mb, mib, megabayt, megabyte	Megabyte
gb, gib, gigabayt, gigabyte	Gigabyte
tb, tib, terabayt, terabyte	Terabyte

Eğer bu tanı girilmez ve boş bırakılırsa, otomatik olarak bayt kullanılır.

İşaret eğer boş olarak belirlencek ise, o halde normal bir değişken gibide belirlenebilir:

```
Değişken İşaret a; b; c
? veya
Belirle İşaret a; b ; c
```

Tabi boş bir işareten dizilimde belirlenebilir:

```
Dizilim İşaret a; b; c
? veya
Belirle Dizilimsel a; b; c
```

Bellek alanına işaret eden bir işaret ile bellek işlemleri yapılabilmesi için ayrıntılı bilgi [Bellek Modülü](#) bölümündedir.

From:

<https://basit.web.tr/dokuwiki/> - Basit Yazılım Dili Dokümantasyonu

Permanent link:

<https://basit.web.tr/dokuwiki/doku.php?id=isaret>

Last update: 12.11.2020 01:09

