

COMPUTER  
*An Overview* SCIENCE *12th  
Edition*

J. Glenn Brookshear

Dennis Brylow

Çeviri Editörü

Yrd. Doç. Dr. Birim Balcı Demirci

BİLGİSAYAR  
*Giriş* BİLİMİNE *12.  
Basımdan  
Çeviri*

PEARSON

nobel

## İÇİNDEKİLER

### PROGRAMLAMA DİLLERİ

#### TARİHSEL BAKIŞ

#### GELENEKSEL PROGRAMLAMA KAVRAMLARI

#### YORDAMSAL BİRİMLER

#### DİL UYGULAMASI

#### NESNE YÖNELİMLİ PROGRAMLAMA

#### EŞ ZAMANLI EYLEMLERİ PROGRAMLAMA

#### BİLDİRİMSEL PROGRAMLAMA

# BÖLÜM 6

## PROGRAMLAMA DİLLERİ

# TARİHSEL BAKIŞ

## İlk Nesiller

Makine dilinde program yazmak, iş bitmeden önce bulunması ve düzeltilmesi (**hata ayıklama** olarak bilinir) gereken hatalara yol açtığından can sıkıcı bir iştir.

1940'lı yıllarda araştırmacılar sayısal form yerine daha akılda kalıcı olan sembolik bir formda (mnemonic) kodları temsil edip işaretleme sistemleri geliştirerek programlama işlemini basitleştirdiler.



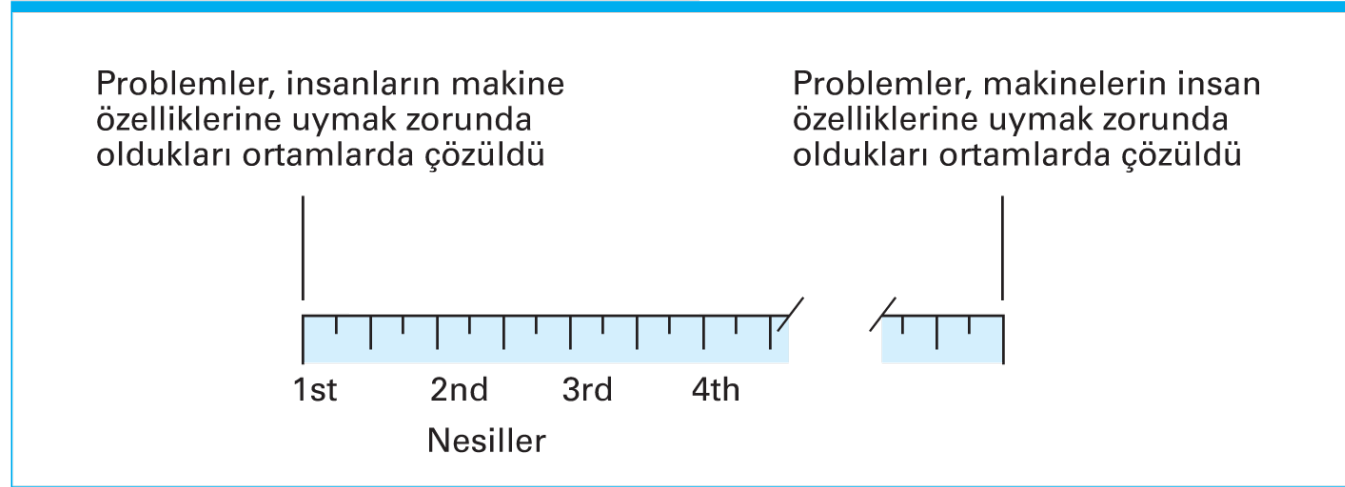
## Makine Bağımsızlığı ve Ötesi

Üçüncü nesil dillerin geliştirilmesi ile makine bağımsızlığı hedefine büyük oranda ulaşıldı. Üçüncü nesil dillerdeki deyimler özel bir makinenin özelliklerini kullanmamalarından bu yana bir makine için olan dil bir diğerine kolayca derlenebilir.

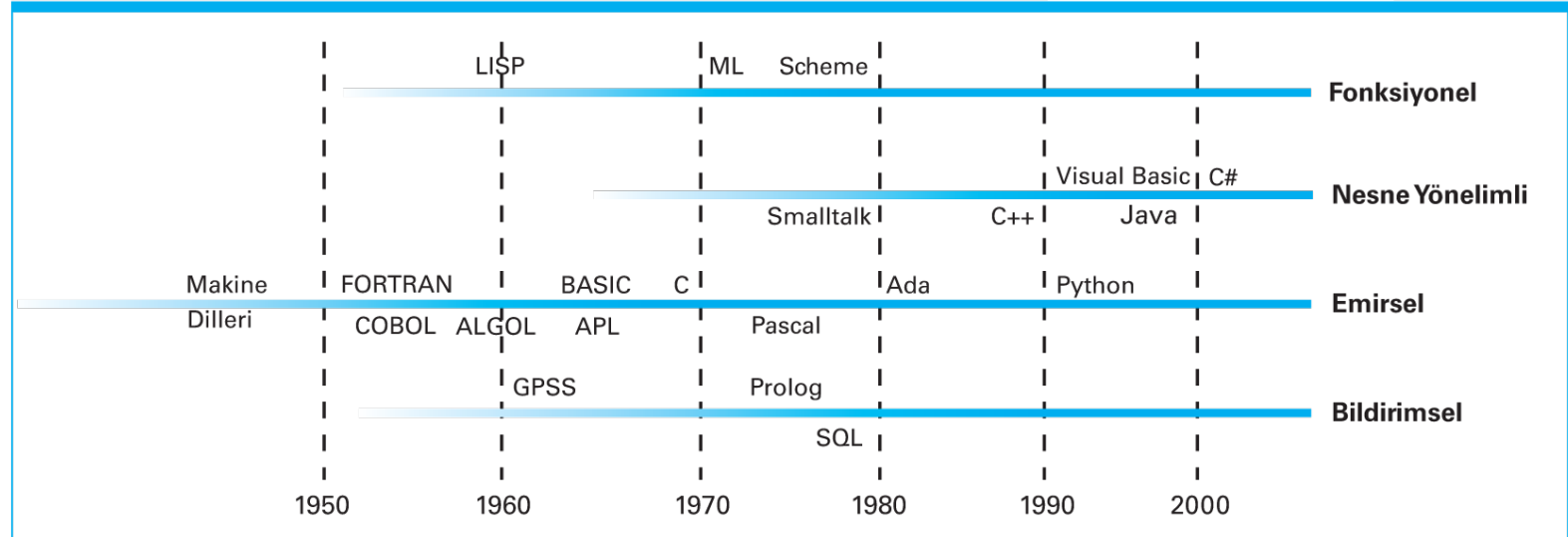
Üçüncü nesil dil ile yazılmış bir program teorik olarak uygun derleyici ile herhangi bir makinede kullanılabilir.

## Programlama Paradigmaları

**Şekil 6.1** Programlama dillerinin nesilleri



**Şekil 6.2** Programlama paradigmalarının evrimi



# GELENEKSEL PROGRAMLAMA KAVRAMLARI

Amacımız belli bir dilin detaylarında takılıp kalmak değil, sadece yaygın dil özelliklerinin çeşitli gerçek dillerde nasıl belirtildiğini göstermektir.

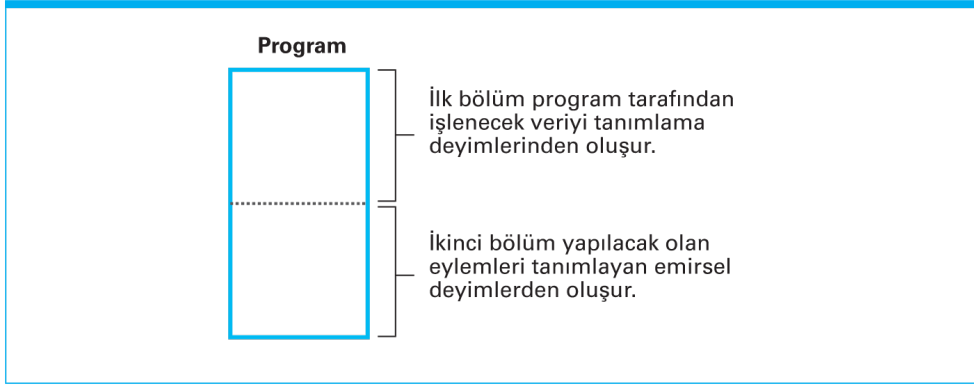
- ✓ C, üçüncü nesil emirsel bir dildir.
- ✓ C11, C dilinin genişletilmiş hali olan bir nesne-yönelimli dildir.
- ✓ Java ve C#, C11'dan türetilen nesne-yönelimli dillerdir.

FORTTRAN ve Ada yeni sürümleri çoğu nesne yönelimli paradigmayı kapsayacak şekilde geliştirilmiş olmasına rağmen orijinal olarak üçüncü nesil emirsel dil olarak tasarlandılar.

## Değişkenler ve Veri Türleri

Yüksek-seviye programlama dilleri ana bellekteki konumların sayısal adreslerden daha ziyade tanımlayıcı isimlerle adlandırılmasına imkan verir. Bunun gibi bir isim **değişken** olarak bilinir

**Şekil 6.4** Tipik bir emirsel program ya da program biriminin yapısı



Programlama dillerindeki int (tam sayı) ve char (karakter) gibi veri türlerine **ilkel veri türleri (primitive data types)** denir. Integer, float, character ve Boolean tipleri yaygın ilkel veri tipleridir.

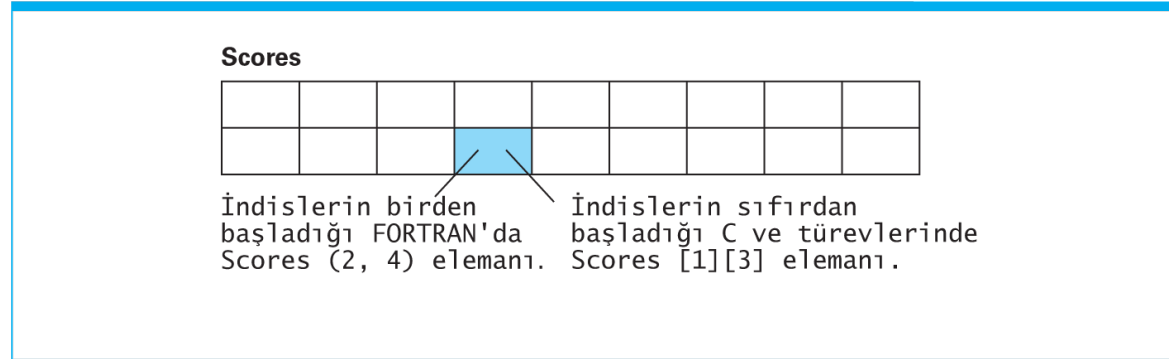


## Veri Yapısı

Veri türüne ek olarak, programdaki değişkenler çoğu zaman kavramsal bir yapı ya da veri topluluğu olan **veri yapısı (data structure)** ile ilişkilidir.

Yaygın bir veri yapısı olan **dizi (array)**; tek boyutlu bir liste, satır ve sütunlarla beraber iki boyutlu bir tablo ya da daha yüksek boyutlu tablolar gibi aynı türden bir dizi eleman topluluğudur.

**Şekil 6.5** İki satır dokuz sütundan oluşan iki boyutlu bir dizi

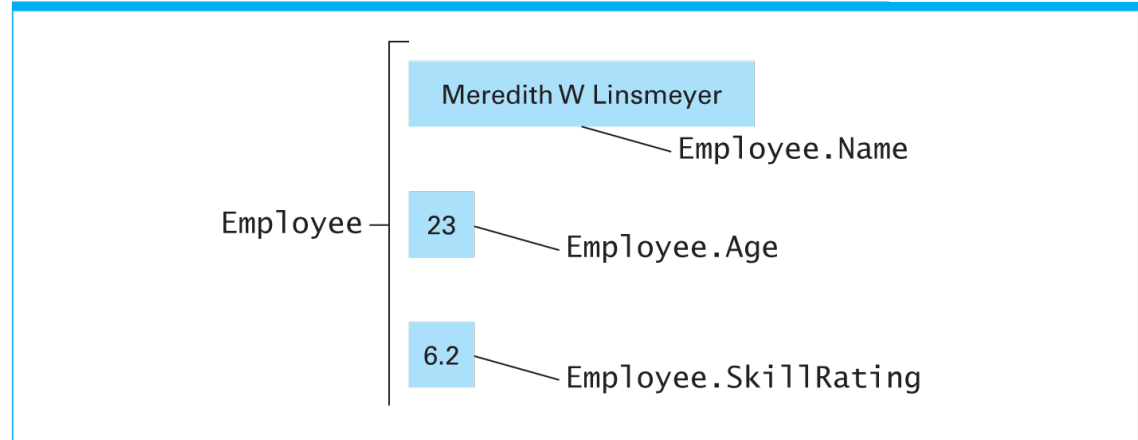


## Sabitler ve Değişmezler

Bazen bir programda sabit, önceden tanımlı değerler kullanılır. Bir değerın böyle açık bir görünümü **değişmez (literal)** olarak adlandırılır.

*EffectiveAlt = Altimeter + 645*

**Şekil 6.6** Employee yapısının kavramsal görünümü



## Atama Deyimleri

En basit emirsel deyim bir değerin bir değişkene atanmasını (ya da daha açıkça, değişken ile belirtilen hafıza alanında depolanmasını) talep eden bir **atama deyimidir**. Örneğin,

$Z = X + Y;$

deyimi, C, C11, C# ve Java'da X ve Y'nin toplamının Z değişkenine atanmasını ifade eder.

## Kontrol Deyimleri

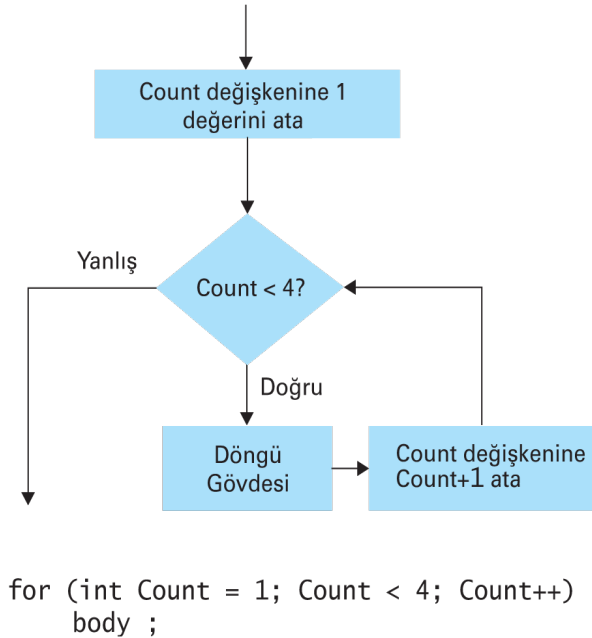
Bir **kontrol deyimi** programın çalışma gidişatını değiştirir. Bütün kontrol deyimleri içinde en dikkat çekici olanı basit *goto* deyimidir.

```
goto 40
20 Evade()
goto 70
40 if (KryptoniteLevel < LethalDose) then goto 60
goto 20
60 RescueDamsel()
70 ...
```

Tek bir deyim olarak:

```
if (KryptoniteLevel < LethalDose):
    RescueDamsel()
else:
    Evade()
```

**Şekil 6.7** Döngü yapısı ve C++, C# ve Java'daki ifadesi



## Yorumlar

Bir insan programı okumayı ya da anlamayı denediğinde ek bilgi genelde yardımcı ya da zaruri olur. Bu nedenden dolayı programlama dilleri **yorum** olarak adlandırılan açıklayıcı deyimleri eklemeyi sağlarlar.

Böylece

*/\* Bu bir yorumdur. \*/*

*ve*

*// Bu bir yorumdur.*

deyimlerinin her ikisi de geçerli yorum deyimleridir.



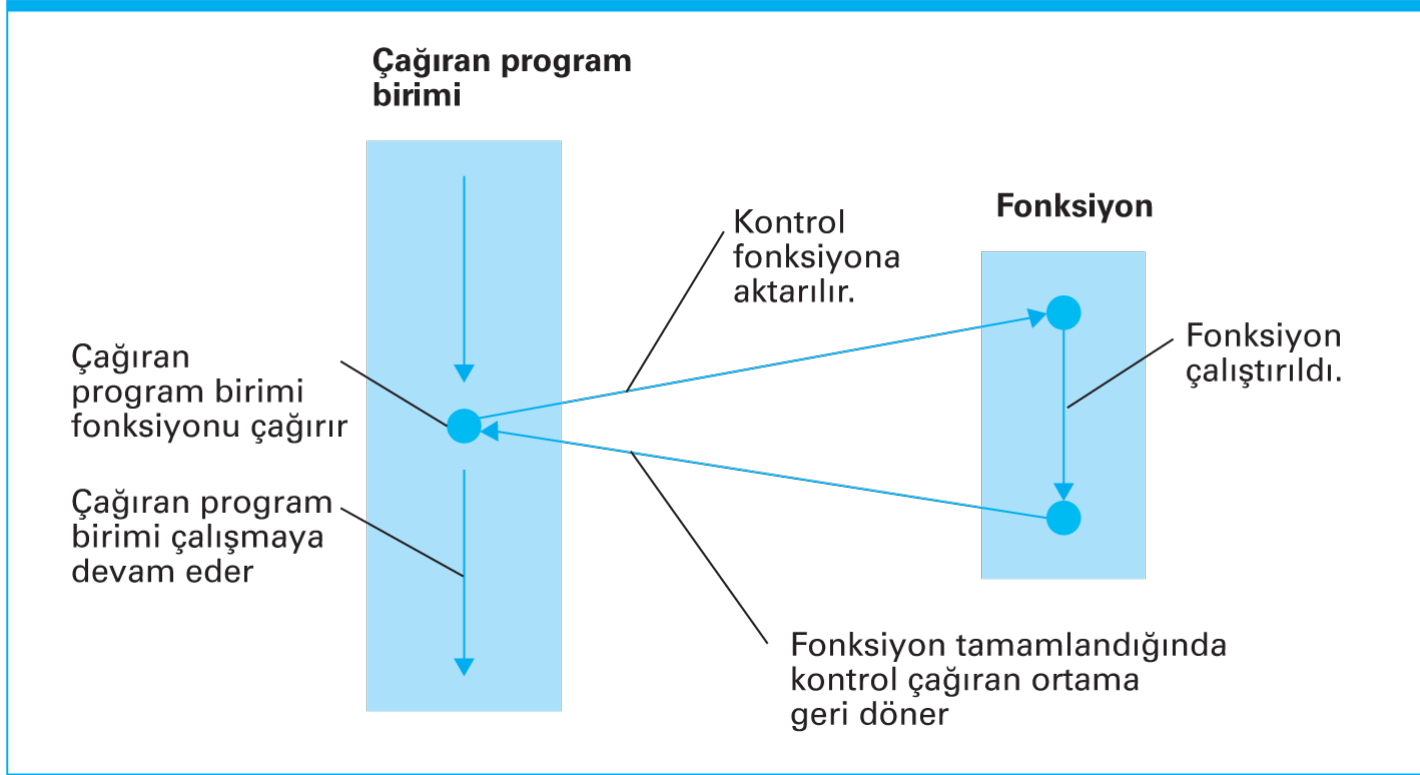
# YORDAMSAL BİRİMLER

Bu kısımda bir emirsel dilde programın modüler temsilini elde etmek için büyük bir teknik olan fonksiyon kavramına odaklanacağız.

## Fonksiyonlar

Genel anlamda bir **fonksiyon** diğer program birimleri tarafından şablon olarak kullanılabilecek bir işi yapmak için oluşturulmuş bir dizi komuttur.

**Şekil 6.8** Bir fonksiyon içeren kontrol akışı



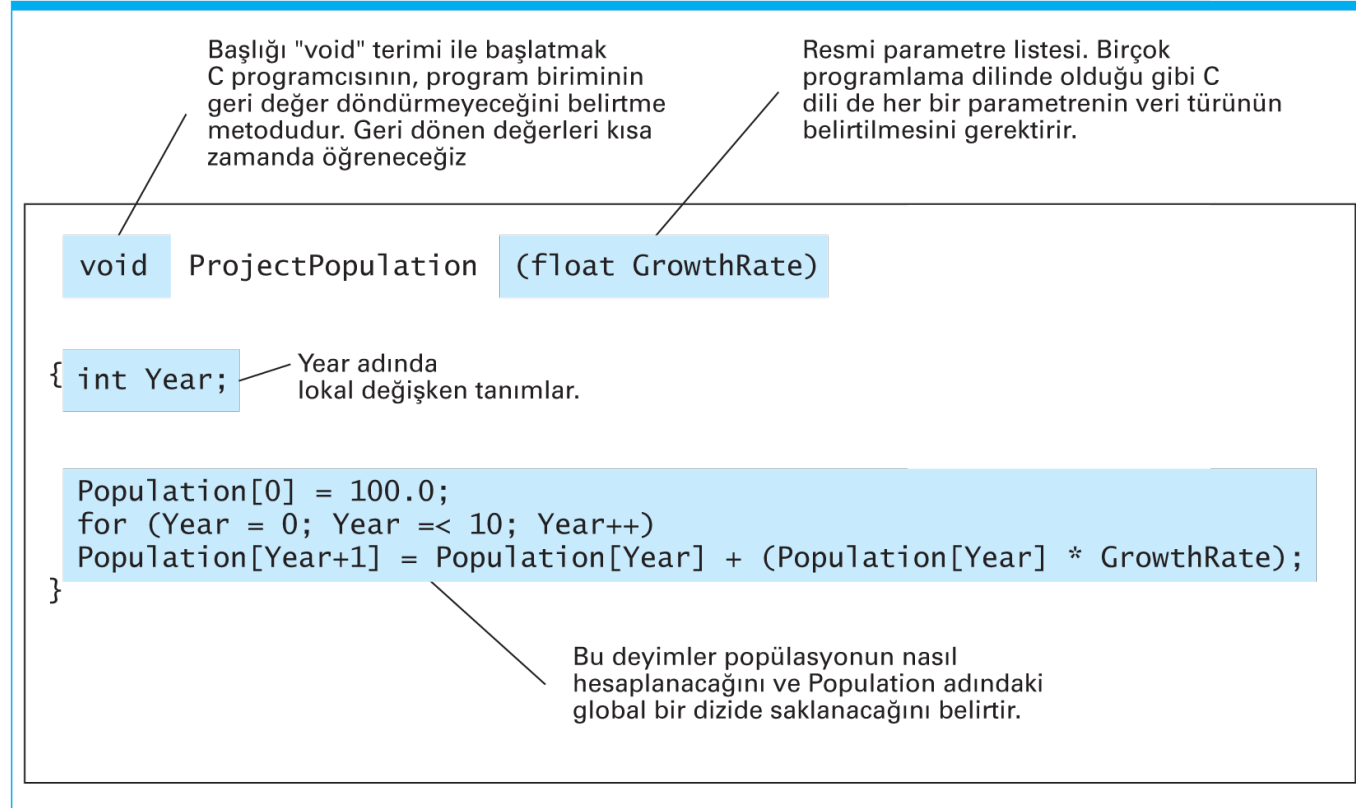
## Parametreler

Fonksiyonlar sıklıkla fonksiyon uygulandığında onları özel yapacak genel terimler kullanılarak yazılırlar.

### **def Sort(List):**

Fonksiyon içindeki bunun gibi genel terimler **parametre** olarak adlandırılır. Daha açık bir şekilde, fonksiyon içinde kullanılan terimler **resmi parametreler**, fonksiyon uygulandığında kesin anlamların bu resmi parametrelere atandığı parametreler ise **gerçek parametreler** olarak adlandırılır.

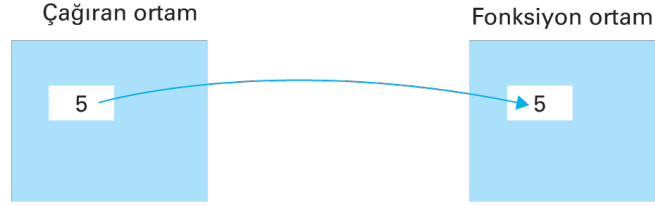
Şekil 6.9 C programlama dilinde yazılmış ProjectPopulation fonksiyonu



## Şekil 6.10

Demo fonksiyonunun çalıştırılması ve parametreleri değer ile aktarma

- a. Fonksiyon çağırıldığı zaman verinin bir kopyası fonksiyona verilir



- b. ve fonksiyon kopyayı işler.

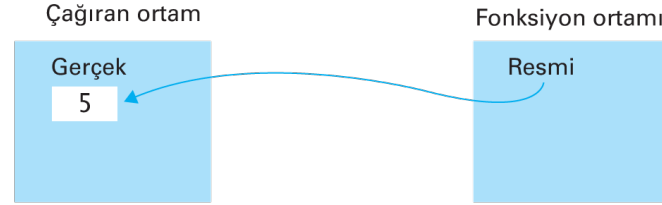


- c. Böylece fonksiyon sonlandığında çağırان ortam değişmemiş olur.

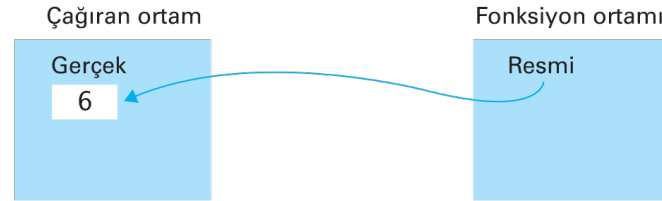




- a. Fonksiyon çağırıldığında, resmi parametre gerçek parametreye yapılan bir referans haline gelir.



- b. Böylece, fonksiyon tarafından yönetilen değişiklikler gerçek parametreye yapılır.



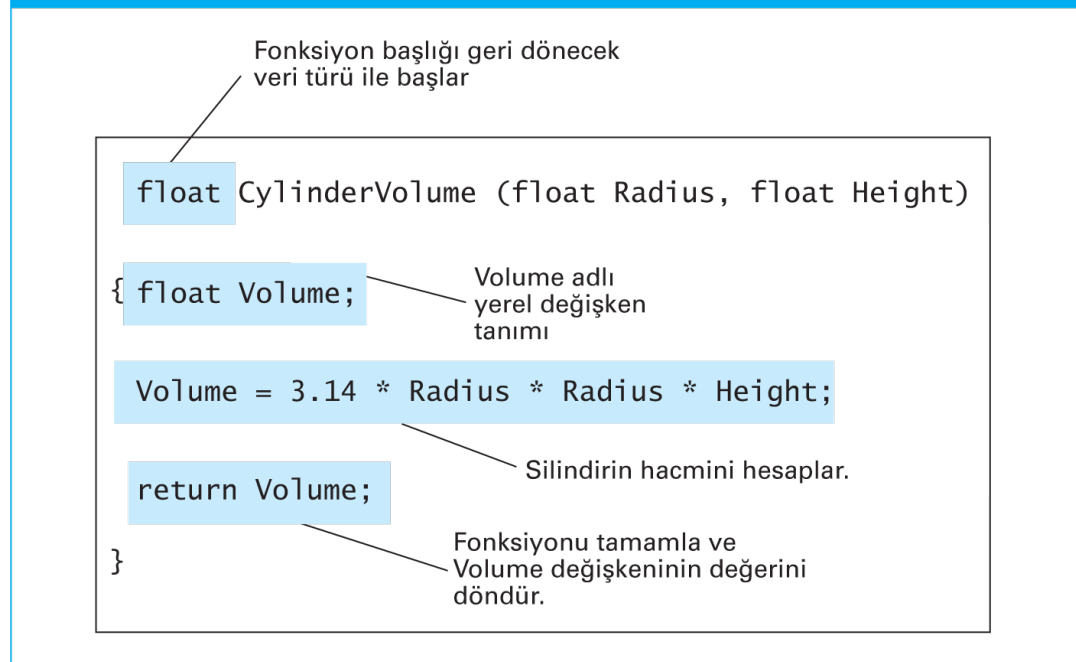
- c. ve bu nedenle fonksiyon tamamlandığında korunmuş olur.



## Verimli Fonksiyonlar

Python programcıları arasında **verimli fonksiyon** terimi geri değer döndüren fonksiyonların ayrımlarını belirtmek için kullanılır.

**Şekil 6.12** C programlama dilinde yazılmış CylinderVolume verimli fonksiyonu

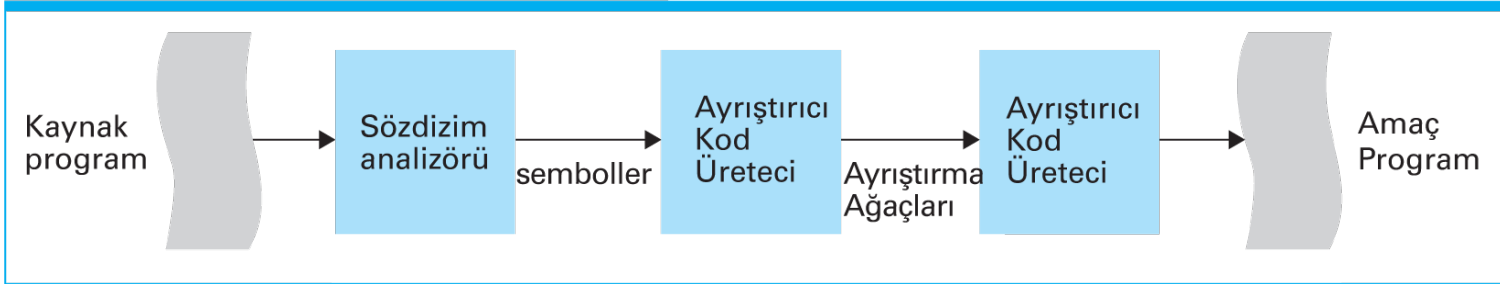


# DİL UYGULAMASI

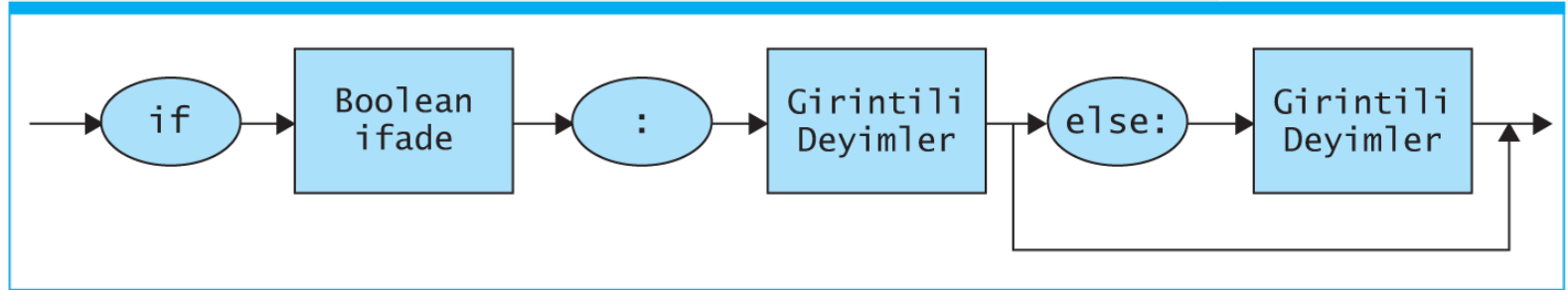
## Çeviri Süreci

Bir programı bir dilden diğerine dönüştürme sürecine **çevirme** denir. Orijinal yapıdaki programa **kaynak program (source program)**, dönüştürülen programa ise **amaç program (object program)** denir. Dönüşüm süreci üç eylemden oluşur: Sözlüksel analiz, ayrıştırma ve kod üretimi.

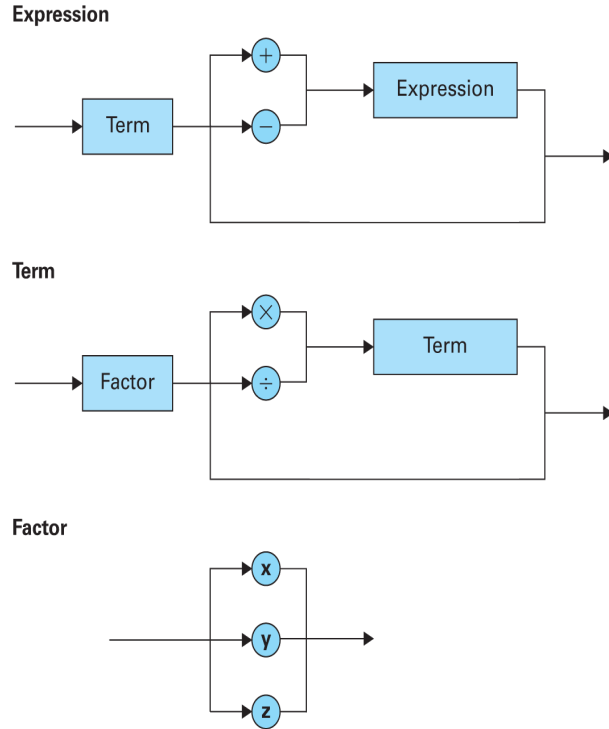
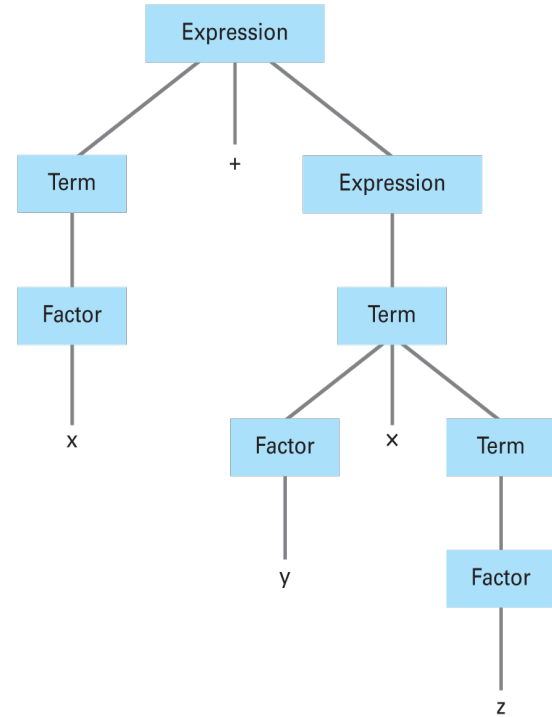
Şekil 6.13 Çevrim süreci



**Şekil 6.14** Python if-else deyiminin sözdizimi şeması

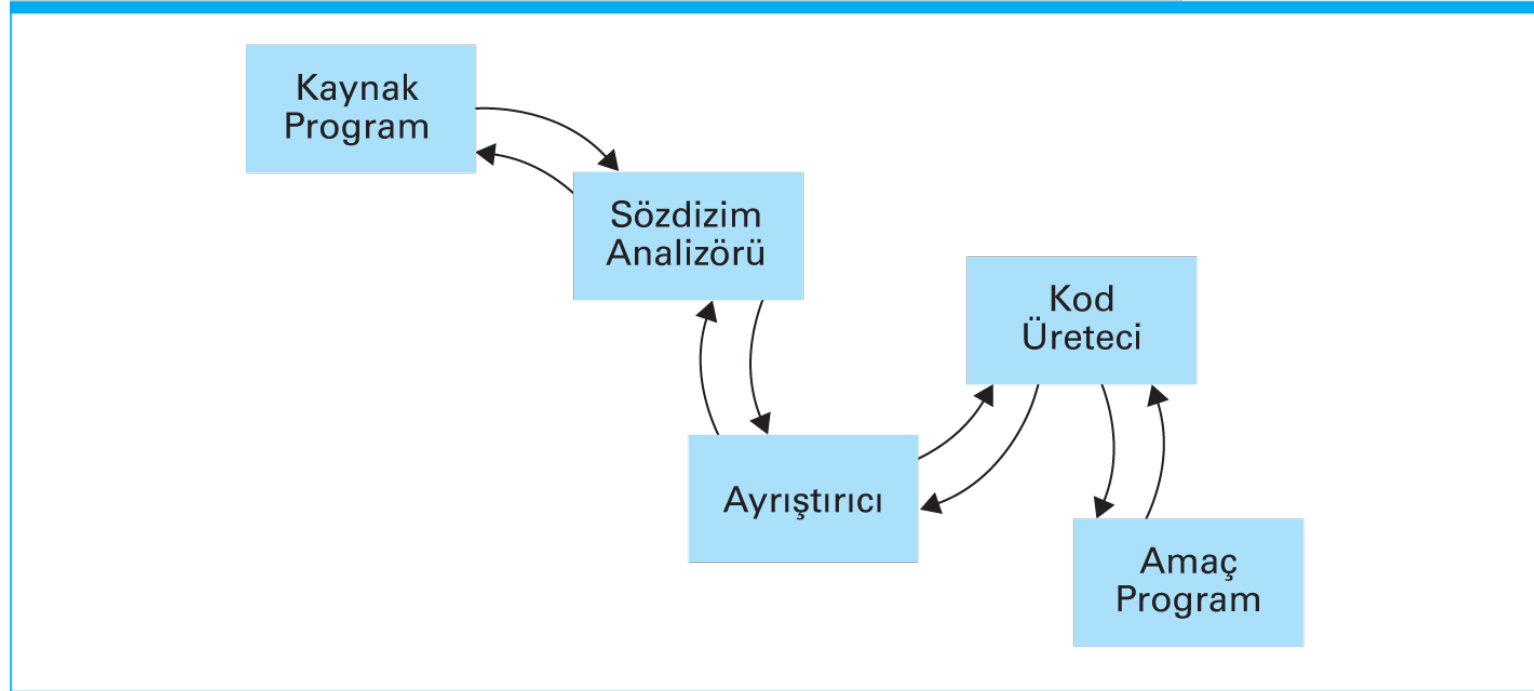


Şekil 6.15 Basit cebirsel bir ifadeyi tanımlayan sözdizimi şeması

Şekil 6.16 Şekil 6.15'teki sözdizimi şemalarını temel alan  $x + y \times z$  ifadesi için bir ayrıştırma ağacı



**Şekil 6.18** Çevrim işlemine nesne-yönelimli bir yaklaşım



## Yazılım Geliştirme Paketleri

Yazılım geliştirme sürecinde kullanılan editör ve dönüştürücü gibi yazılım araçları tek bir entegre yazılım geliştirme sistemi olarak bir paket içinde gruplanmıştır.

Programcı bu uygulama paketini kullanarak, program yazmak için bir editöre, makine diline kodları çeviren bir dönüştürücüye, arızalı programın çalışmasını takip ederek, programın nerede hataya yol açtığını bulmak için kullanılan çeşitli hata ayıklama araçlarına hazır erişim elde eder.

# NESNE YÖNELİMLİ PROGRAMLAMA

Bir probleme nesne yönelimli yaklaşmak, problemdeki nesneleri belirtmek ve bu nesneleri kendi kendine yetecek birimler olarak tanımlamaktır.

## Sınıflar ve Nesneler

Nesne yönelimli paradigmada nesne topluluğundan oluşan bir şablon **sınıf (class)** olarak adlandırılır.

**Şekil 6.19** Bir bilgisayar oyunundaki lazer silahını tanımlayan bir sınıfın yapısı

```
class LaserClass
```

```
{ int RemainingPower = 100;
```

```
void turnRight ( )  
{ ... }
```

```
void turnLeft ( )  
{ ... }
```

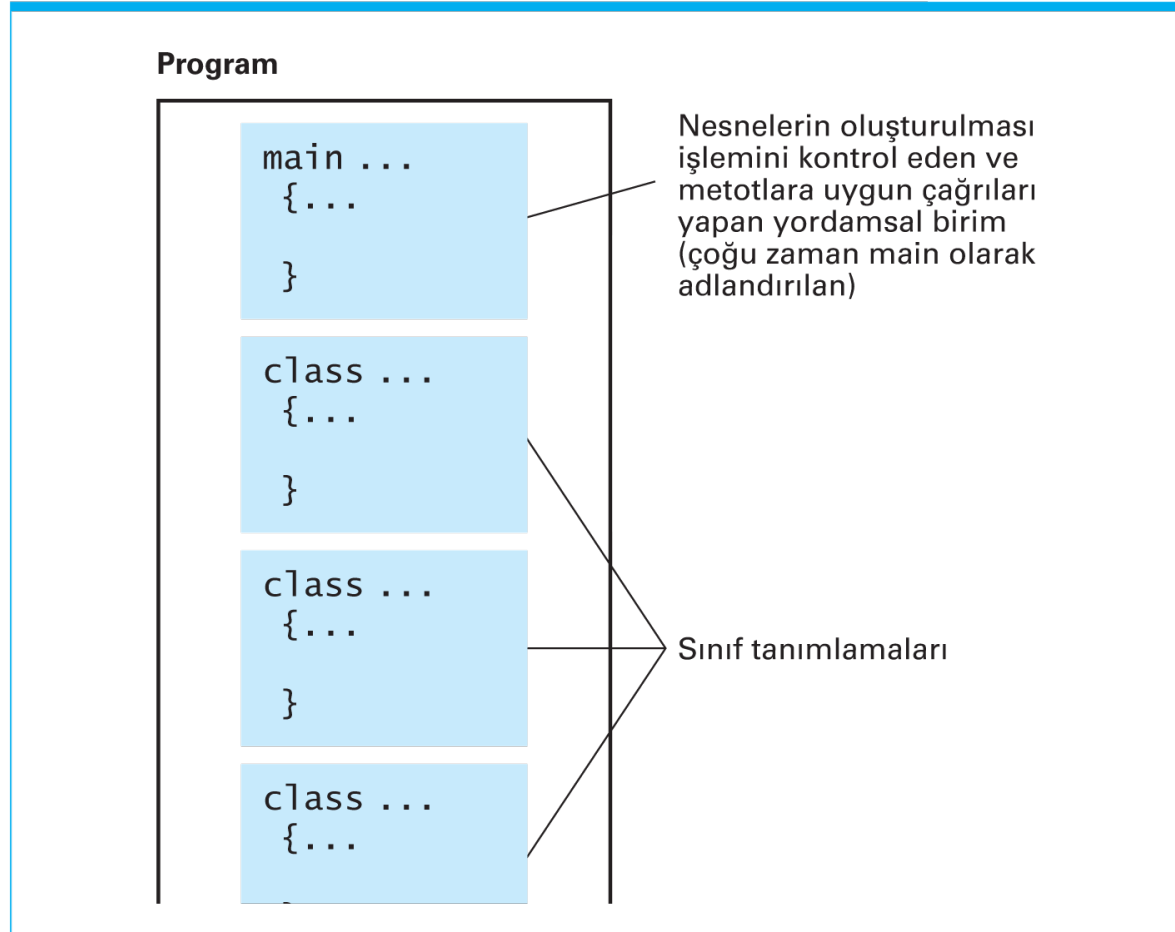
```
void fire ( )  
{ ... }
```

```
}
```

Bu türden olan her  
bir nesnenin içinde  
bulunan verinin tanımı

Bu türden bir nesnenin  
çeşitli mesajlara nasıl cevap  
vereceğini tanımlayan metotlar

**Şekil 6.20** Tipik bir nesne-yönelimli programın yapısı



## Yapıcılar

Bir nesne oluşturulduğunda, çoğu zaman özelleştirme eylemlerine ihtiyaç duyulur.

Şekil 6.21 Yapıcı fonksiyonlu bir sınıf

```
class LaserClass  
{ int RemainingPower;
```

Nesne oluşturulduğunda  
yapıcı RemaininPower'a  
bir değer atar

```
LaserClass (InitialPower)  
{ RemainingPower = InitialPower;  
}
```

```
void turnRight ( )  
{ ... }
```

```
void turnLeft ( )  
{ ... }
```

```
void fire ( )  
{ ... }
```

```
}
```

## Ek Özellikler

**Şekil 6.22** Bir Java ya da C# programında olduğu gibi kapsülleme kullanan LaserClass tanımlaması

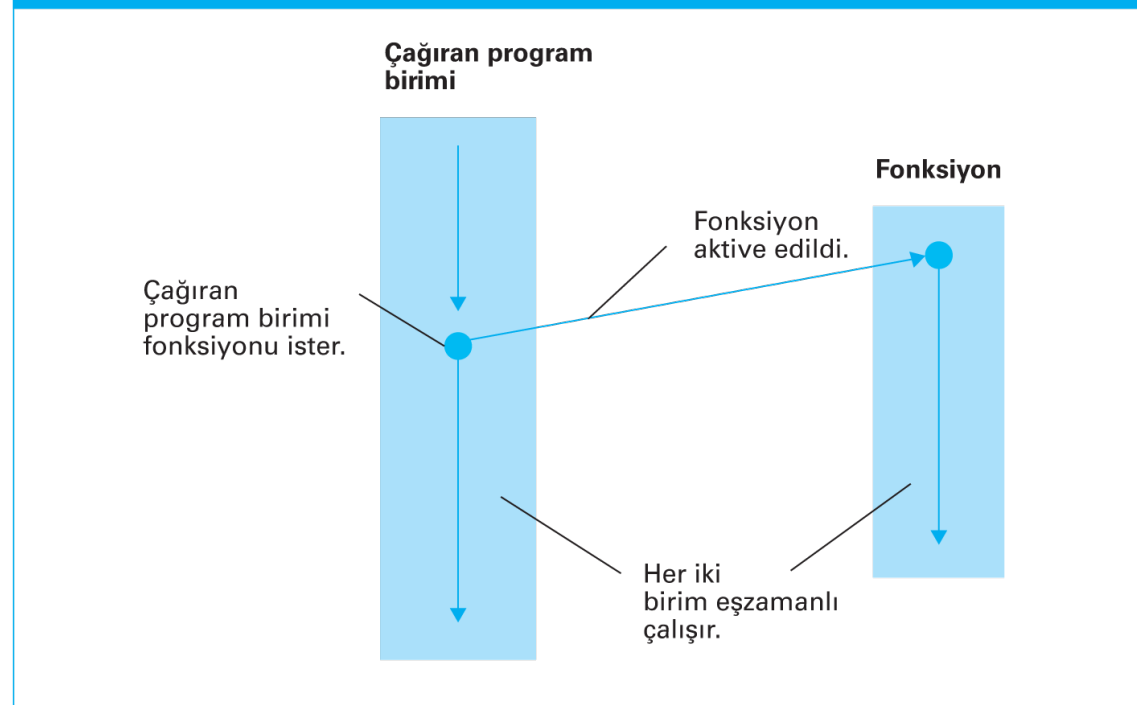
Sınıftaki bileşenler diğer program birimlerinden erişilebilir olup olmamalarına göre public ya da private olarak tasarlanmıştır.

```
class LaserClass
{private int RemainingPower;
  public LaserClass (InitialPower)
  {RemainingPower = InitialPower;
  }
  public void turnRight ( )
  { ... }
  public void turnLeft ( )
  { ... }
  public void fire ( )
  { ... }
}
```

# EŞ ZAMANLI EYLEMLERİ PROGRAMLAMA

Birçok canlandırmanın aynı anda çalıştırılması **paralel işleme** ya da **eş zamanlı işleme** olarak adlandırılır.

Şekil 6.23 İş parçacığı oluşturulması



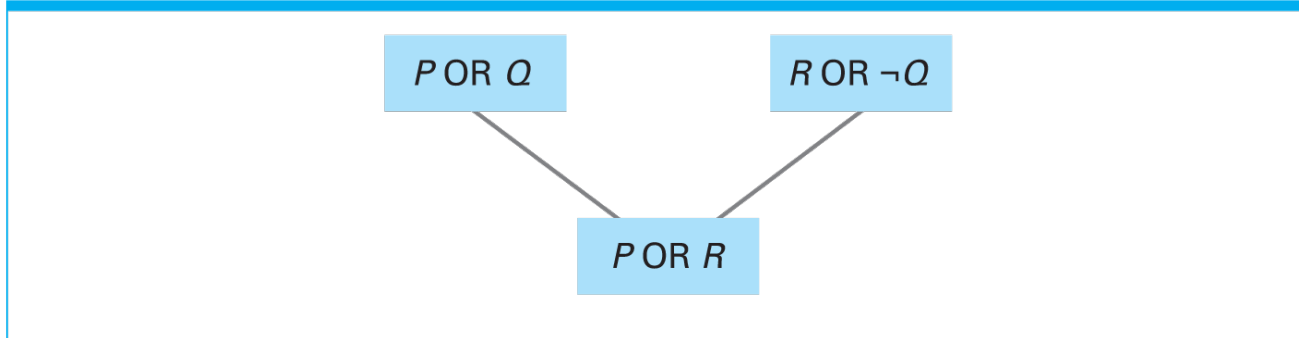


# BİLDİRİMSEL PROGRAMLAMA

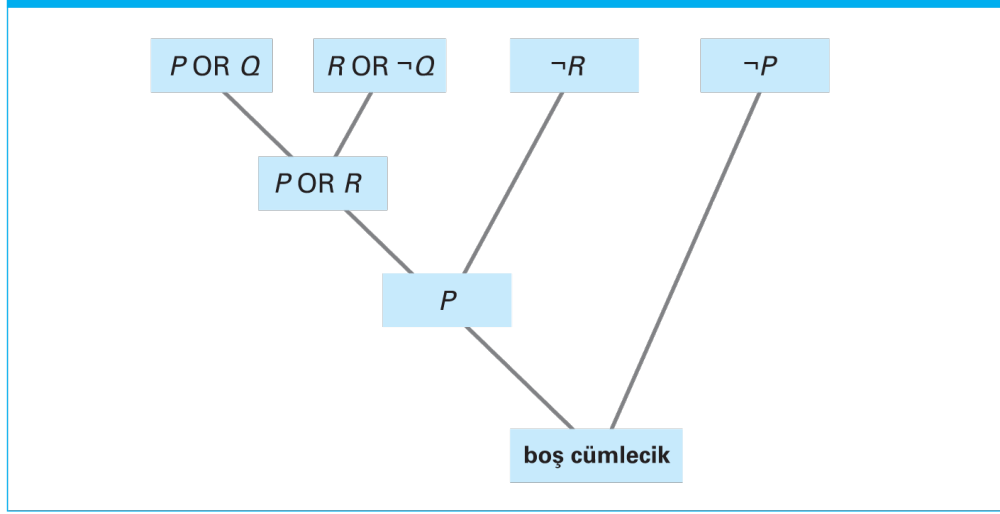
## Mantıksal Çıkarım

Kermit'in hasta ve sahnede olduğunu bildiğimizi varsayalım. Bize Kermit'in sahnede olmadığı söylendi. Kermit'in hasta olduğu sonucuna varabiliriz. Bu örnek **çözüm** olarak adlandırılan, sebep-çıkarm prensibidir. Çözüm, **çıkarm kuralları** olarak adlandırılan, deyim topluluğundan sonuç türetmek için kullanılan birçok teknikten biridir.

**Şekil 6.24** ( $P \text{ OR } R$ ) deyimini üretmek için ( $P \text{ OR } \neg Q$ ) ve ( $R \text{ OR } Q$ ) deyimlerini çözme



Şekil 6.25  $(P \text{ OR } Q)$ ,  $(R \text{ OR } \neg Q)$ ,  $\neg R$  ve  $\neg P$  deyimlerinin çözülmesi



### Prolog

Prolog (PROgramming in LOGic kısaltması) programlama dili arka planında tekrarlayan çözümleri temel alan problem-çözücü algoritma barındıran bir bildirimsel programlama dilidir. Bunun gibi diller **mantık programlama** dilleri olarak adlandırılır.