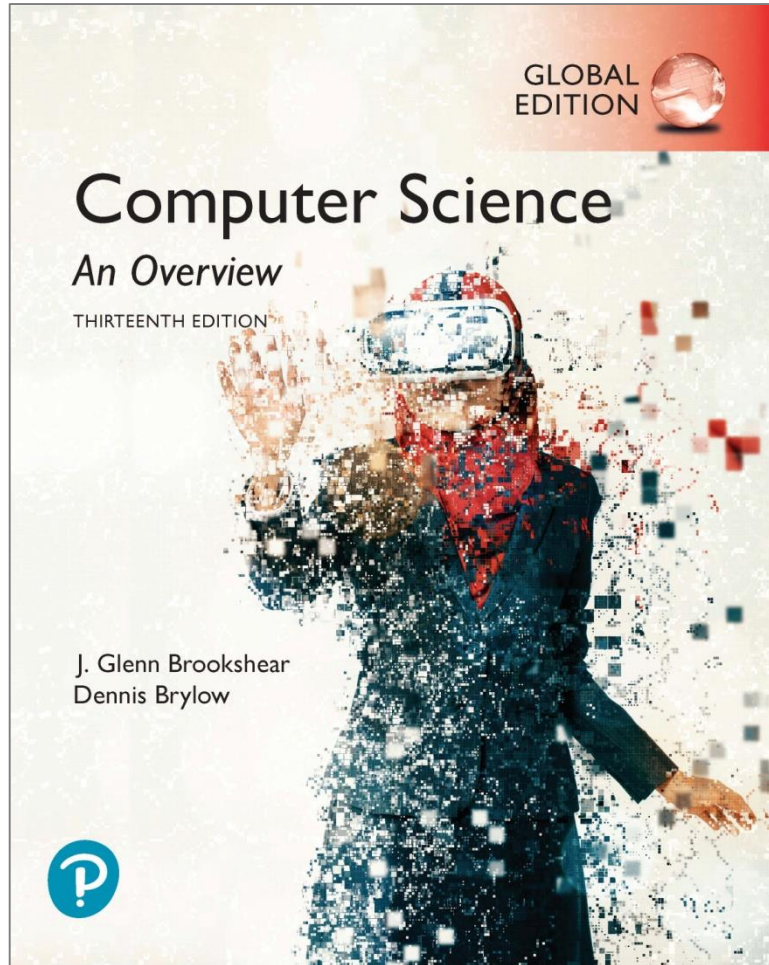


Bilgisayar Bilimine Giriş

13. Baskı, Global Edition



Bölüm 7

Yazılım Mühendisliği

7.1 Yazılım Mühendisliği Bilim Dalı

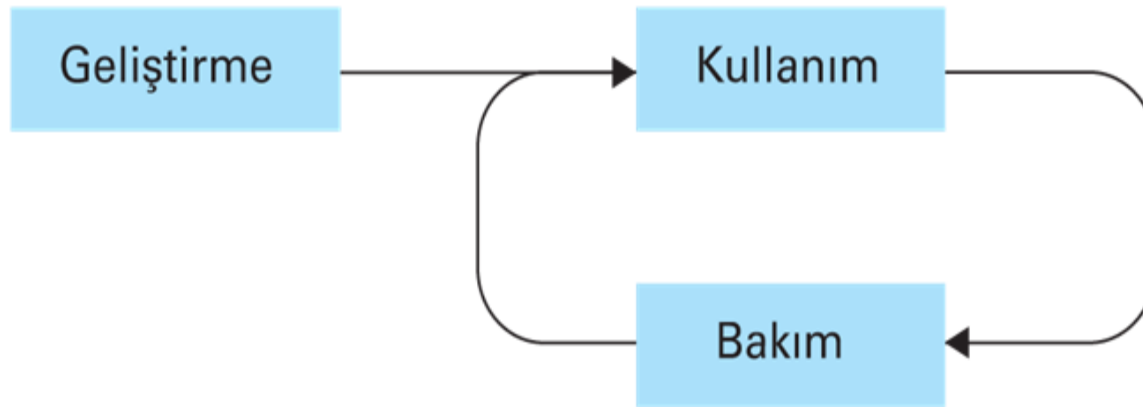
- Yazılım Mühendisliği, büyük ve karmaşık yazılım sistemleri geliştirmek için yol gösterici prensipler arayan bir bilgisayar bilimleri dalıdır.
- Yazılım mühendisliği ile diğer mühendislikler arasında fark var!
 - Ölçütler yetersiz
 - Maliyet tahmini
 - Karmaşıklık hesabı
 - Yazılım kalite ölçümü
 - Çalışan performansı/kalite ölçümü
- Profesyonel Kuruluşlar: ACM, IEEE vb.
 - Standartlar

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

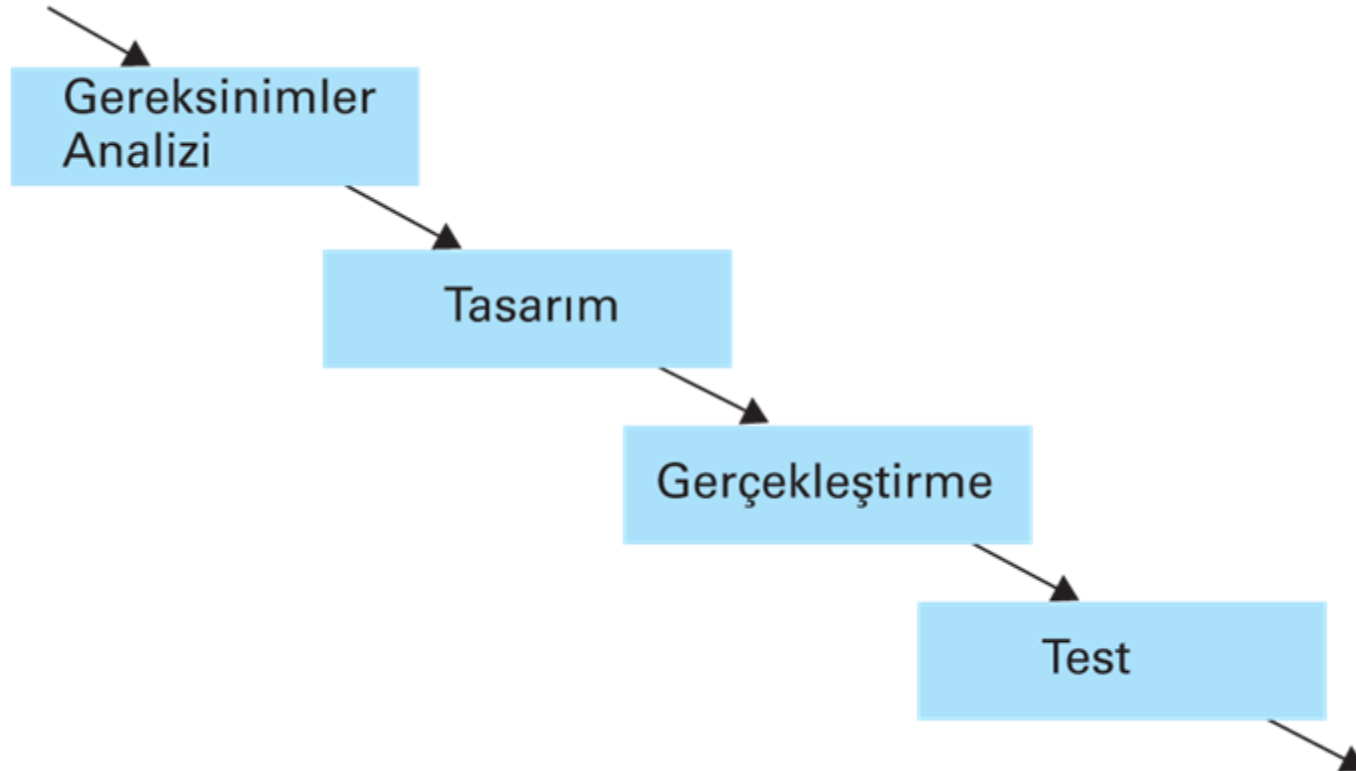
7.2 Yazılımın Yaşam Döngüsü

- Yazılım geliştirmeye çaba harcamak, değişiklikler gerektiğinde muazzam bir fark yaratabilir
- Üretilip bırakılmış ürünlerin aksine, güncelleme ve düzeltme gerektirir
- Bakım için ise programı ve içeriğini anlayan biri gerekir

Şekil 7.1 Yazılımın Hayat Döngüsü



Şekil 7.2 Yazılım hayat döngüsünün geleneksel geliştirme safhası



Gereksinim Analiz Bölümü

- Gereksinimler
 - Uygulamaya yönelik
- Şartname
 - Teknik açıdan
- Yazılım gereksinimler belgesi

Tasarım Bölümü

- Metodolojiler ve araçlar
- İnsan arayüzü (psikoloji ve ergonomi)

Gerçekleştirme Bölümü

- Tasarımdan sistem oluşturma
 - Program yazma
 - Veri dosyaları oluşturma
 - Veritabanları geliştirme
- «Programcı» veya «yazılım analisti» rolleri?
 - Yazılım analisti, gereksinim ve tasarım aşamasında yoğunlaşır
 - Programcı ise gerçekleştirme aşamasında uzmanlaşır.

Test Bölümü

- Geçerlilik testi
 - Sistemin şartnameye uyduğunu onaylama
- Zaafiyet testi
 - Yazılım açıklarının bulunması

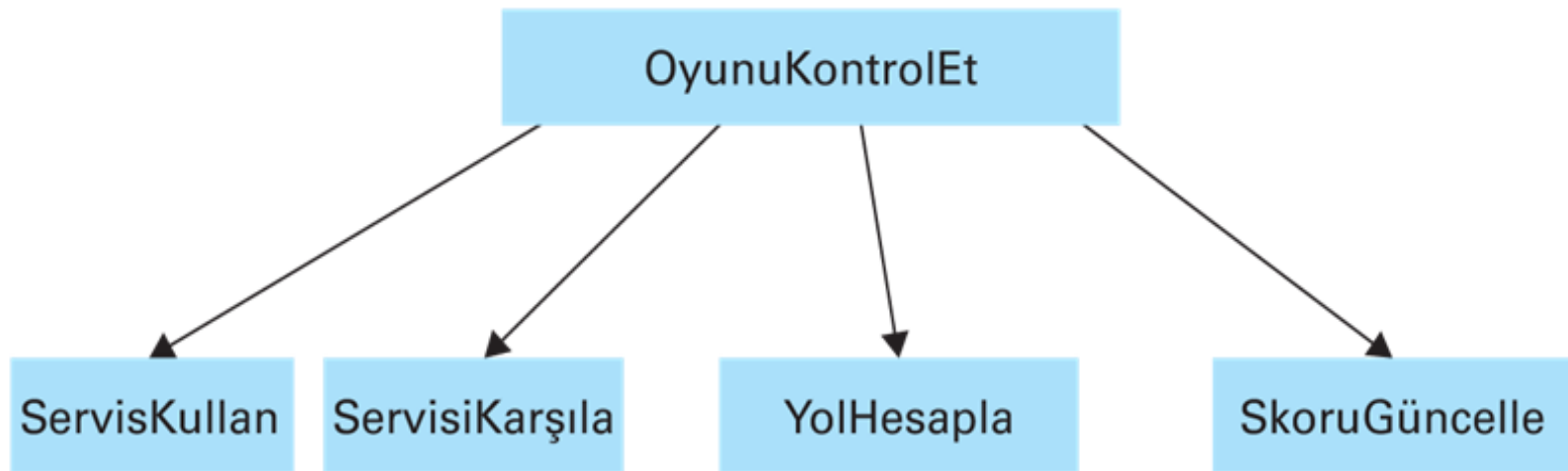
7.3 Yazılım Mühendisliği Metodolojileri

- Şelale Modeli
- Artırımlı Model
 - Prototipleme
- Tekrarlamalı model
- Açık kaynaklı geliştirme
- Çevik Metotlar

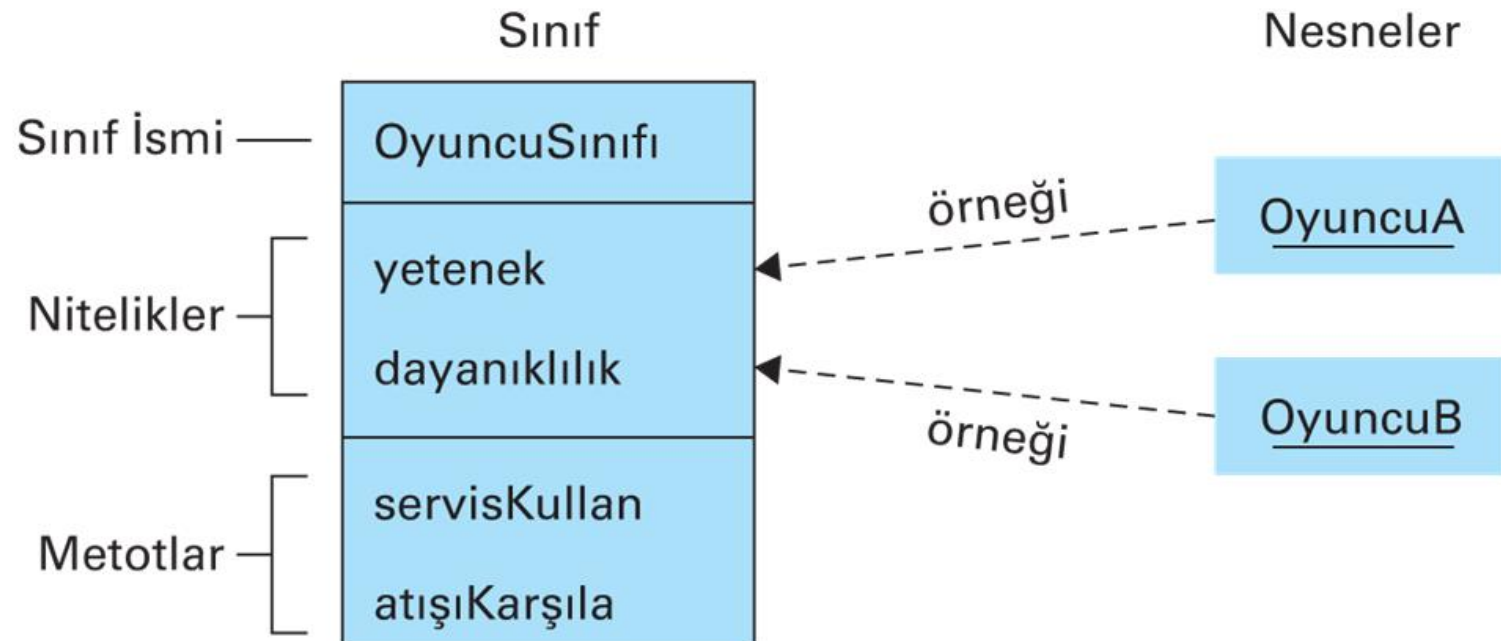
7.4 Modülerite

- Fonksiyonlar – Zorunlu paradigma
- Nesne – Nesne ilişkisine dayalı paradigma
 - Nesne İşbirliği diyagramları

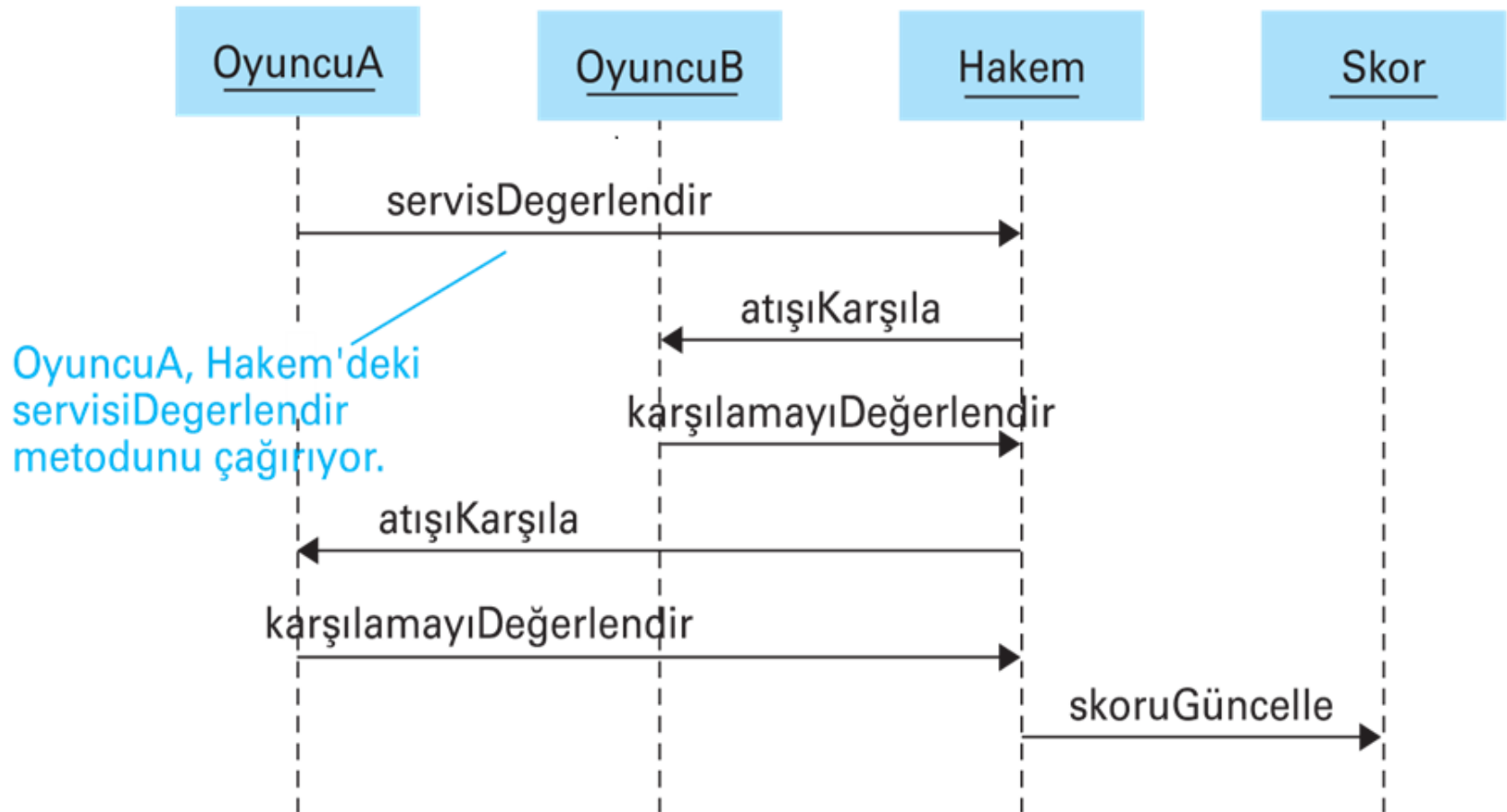
Şekil 7.3 Modülerite için Basit yapı grafiği



Şekil 7.4 OyuncuSınıfı ve örneklerin yapısı



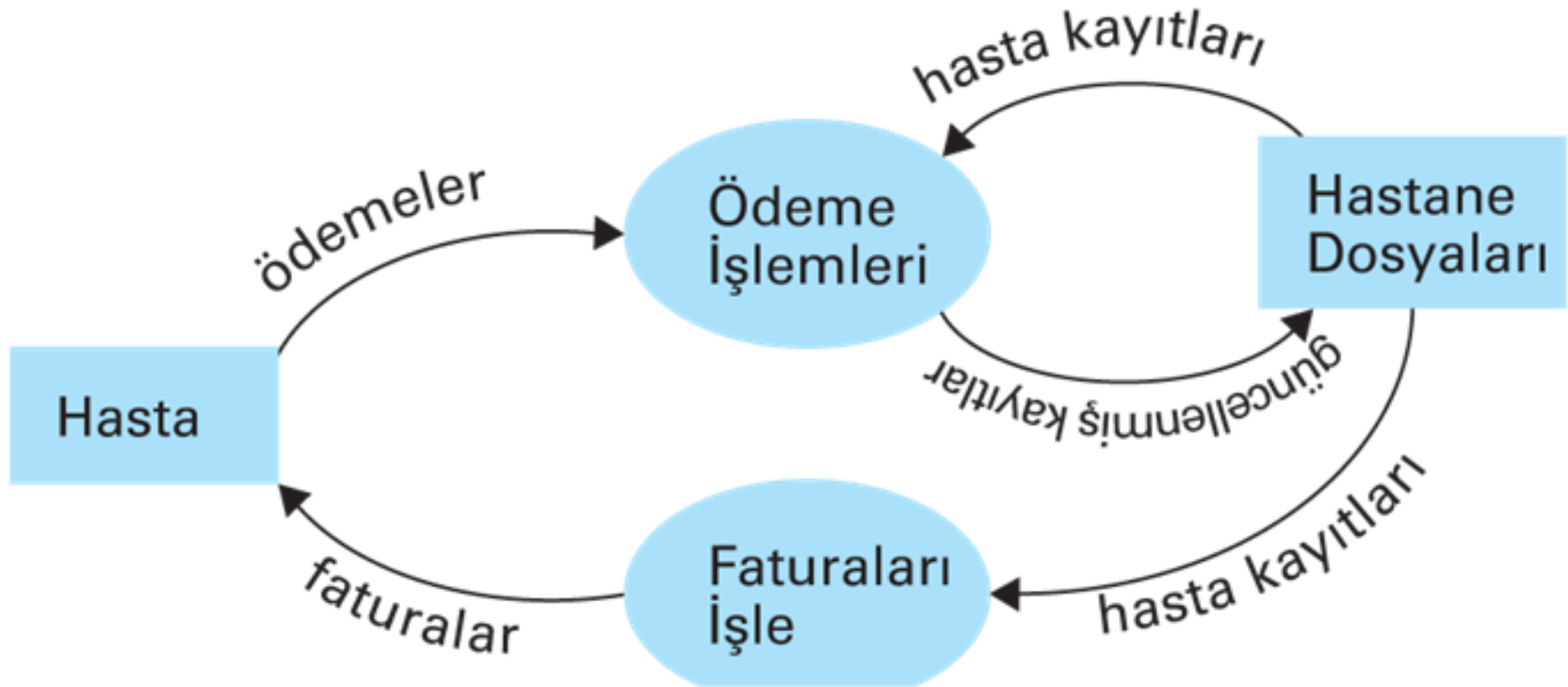
Şekil 7.5 OyuncuA'dan gelen servisin sonucunda nesneler arasındaki etkileşim



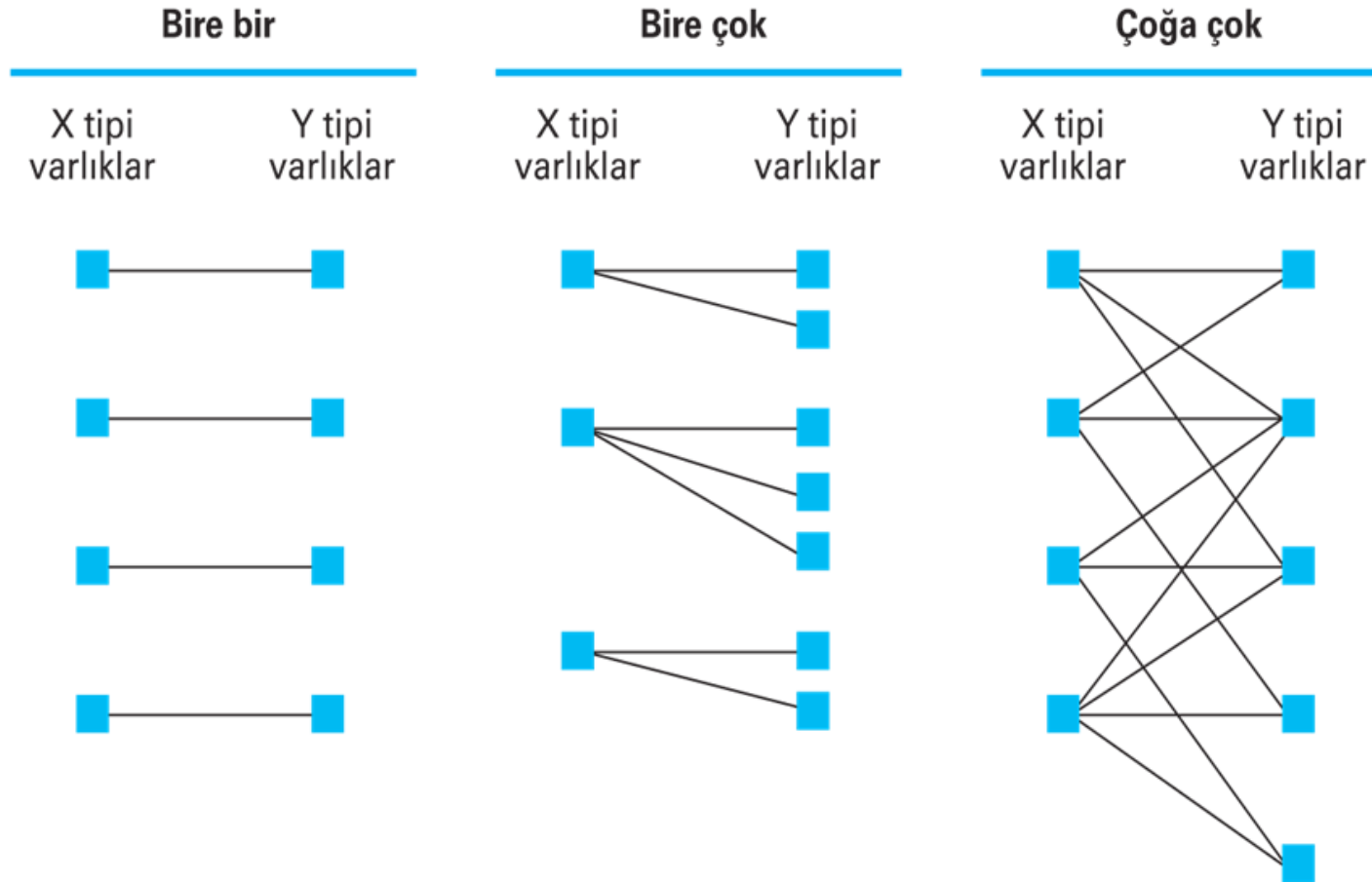
7.5 İş Araçları

- Veri akışı diyagramı
- Varlık ilişkisi diyagramı
 - Bire bir ilişki
 - Birden çoğa ilişki
 - Çoktan çoğa ilişki
- Veri sözlüğü

Şekil 7.8 Bir veri akış grafiği



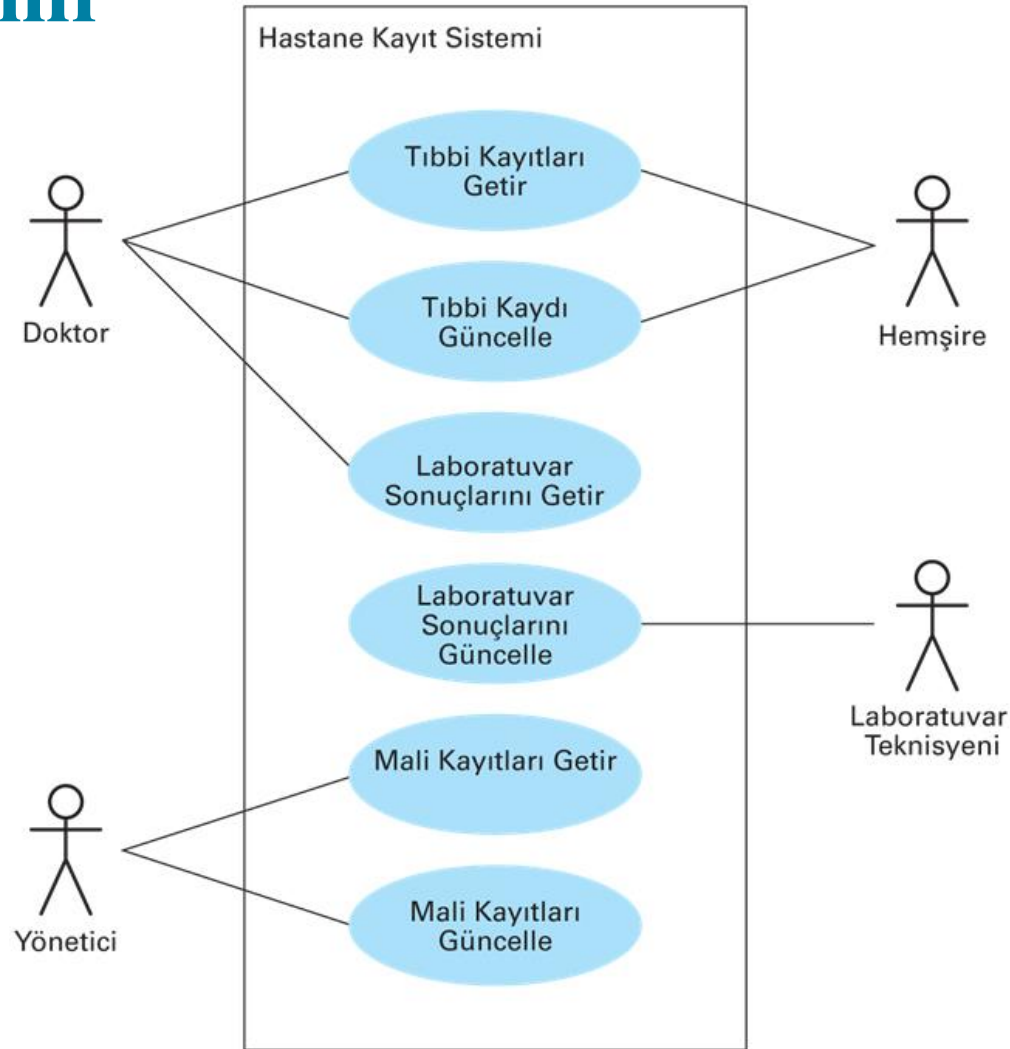
Şekil 7.11 X ve Y tipi varlıkları arasında bire bir, bire çok ve çoğa çok ilişkiler



Birleşik Modelleme Dili

- Kullanım durumu diyagramı
 - Kullanım diyagramı
 - Elemanlar
- Sınıf diyagramı

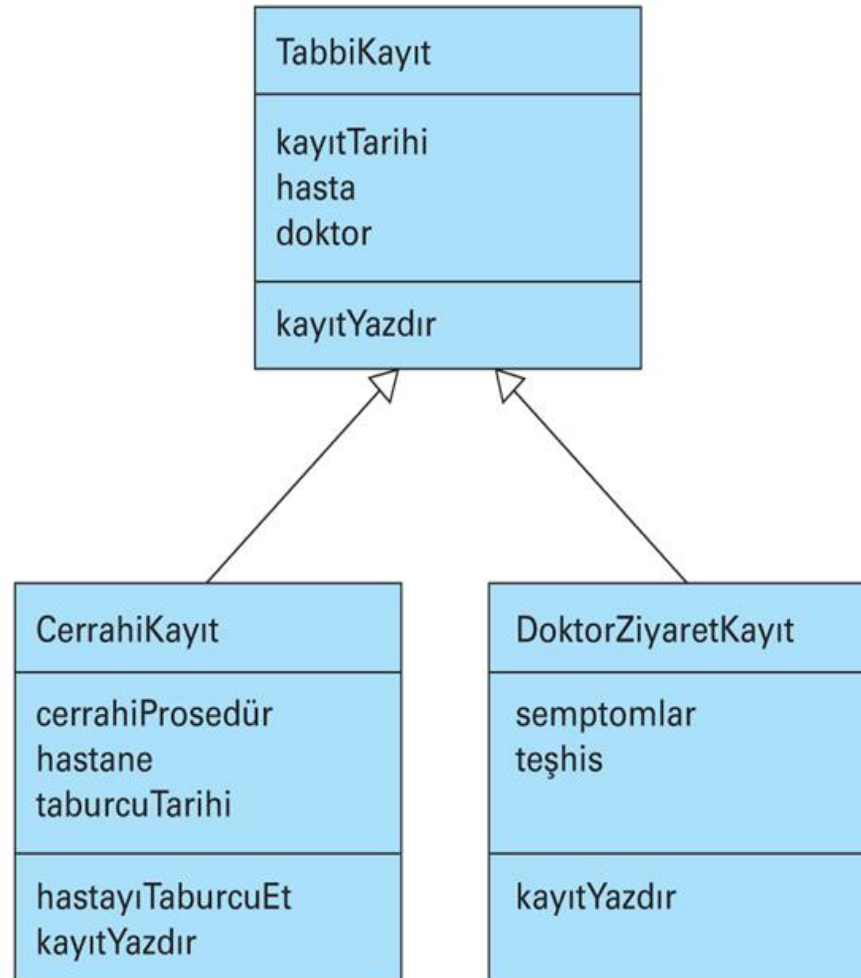
Şekil7.9 Basit bir kullanım durumu diyagramı



Şekil 7.10 Basit bir sınıf diyagramı

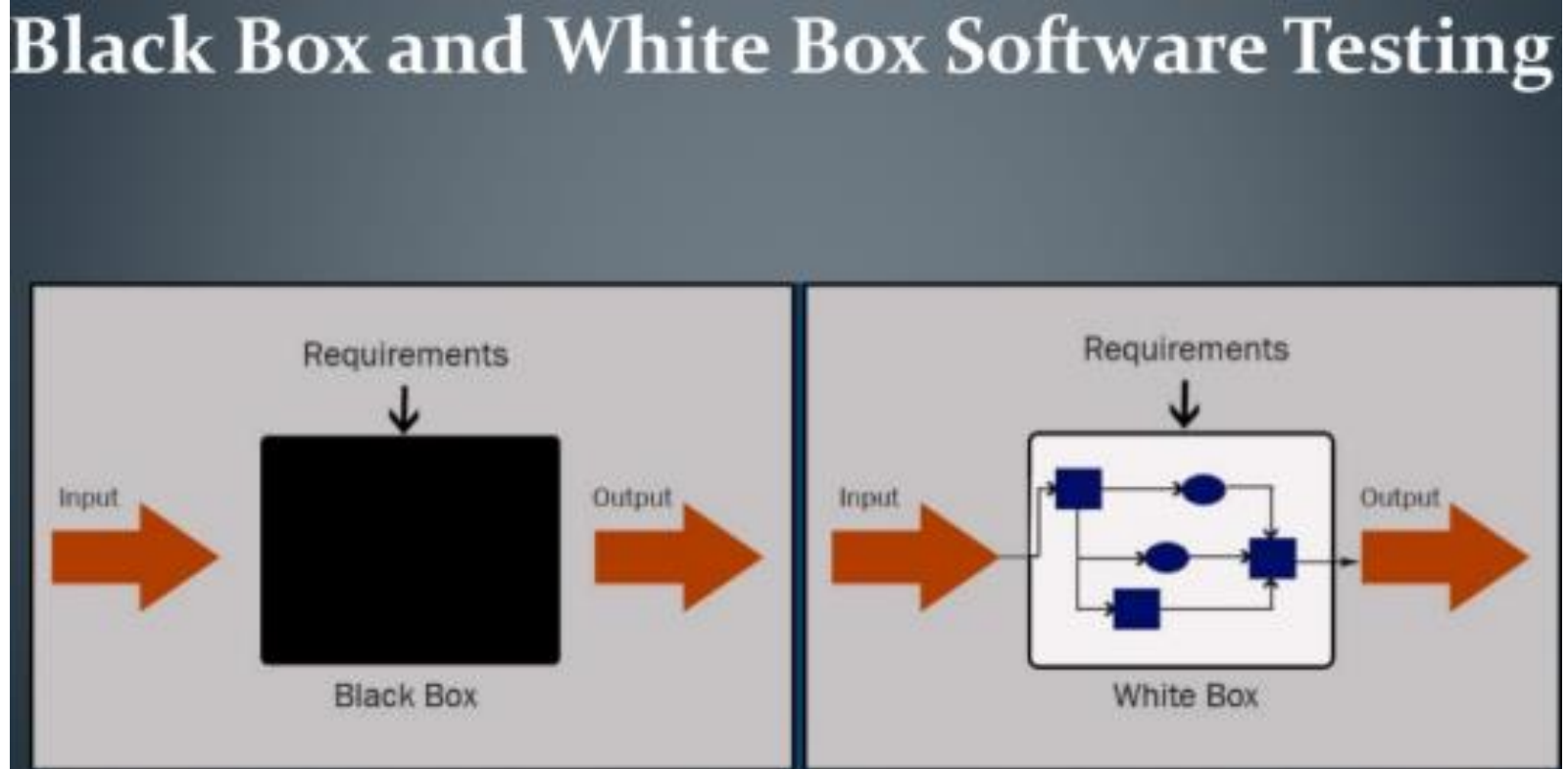


Şekil 7.12 Genellemeleri gösteren bir sınıf diyagramı



7.6 Kalite Sigortası

- Cam kutu (Beyaz kutu) testi
- Kara kutu testi



7.7 Dökümantasyon

- Kullanıcı dökümantasyonu
 - Bütün müşteriler için basılmış kitap
 - Çevrimiçi yardım modülleri
- Sistem dökümantasyonu
 - Kaynak kodu
 - Tasarım dökümanları
- Teknik dökümantasyonu
 - Kurulum, güncelleme vb. için

7.8 İnsan-Makine Arayüzü

- İnsan kullanımını kolaylaştırmak için tasarlanmış yazılım sistemi fikridir
- İnsan Ergonomisi – fiziksel yetenekleri
- İnsan kognetiği (biliş) – mental yetenekleri
 - Alışkanlıklar
 - Dikkat

7.9 Yazılım sahipliği ve yükümlülüğü

- Telif hakkı
 - Ürünün fikri mülkiyet haklarının korunması
 - Bütün işlerde vardır:
 - Şartname
 - Kaynak kodu
 - Son ürün

Fikri Mülkiyet

- Yazılım Lisansı
 - Sahipliği devretmeksizin başka bir kullanıcıya gerekli izinleri sağlayan yasal bir anlaşma
- Patent
 - Bilimsel ve teknik bir buluşun ya da böyle bir buluşu uygulama alanında kullanma hakkının kime ait olduğunu gösteren resmi belge.
 - Süreç pahalı ve zaman alıcıdır