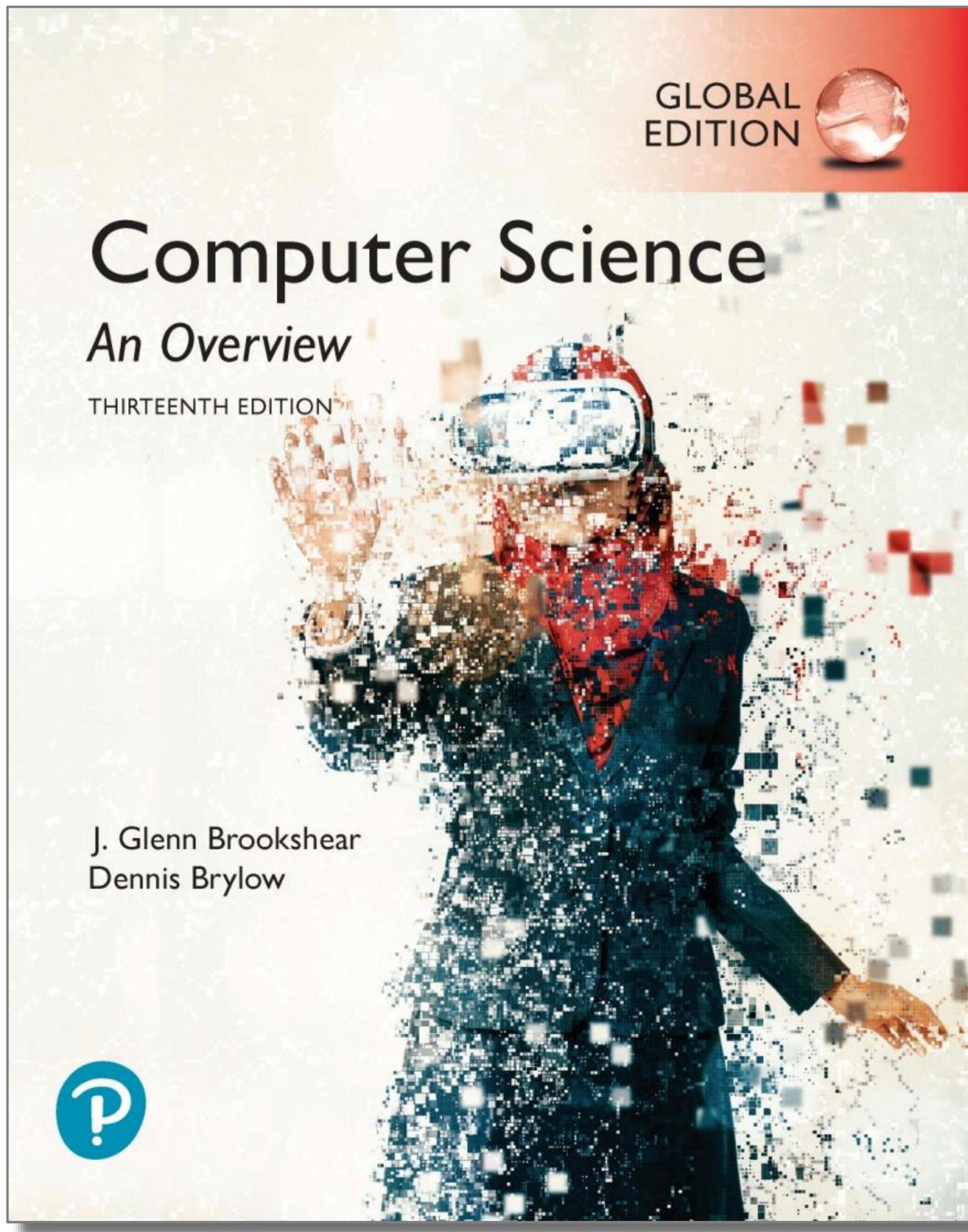


Bilgisayar Bilimine Giriş

13. Baskı, Global Edition



Bölüm 7

Yazılım Mühendisliği

Dersin dersleri howa
bilgisayar müh. dersleri -microsoft
vs

Yazılım geliştirme üretim süreçleri
süreç yönetimi



Gitar müh. dersleri

7.1 Yazılım Mühendisliği Bilim Dalı

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

SWEbook

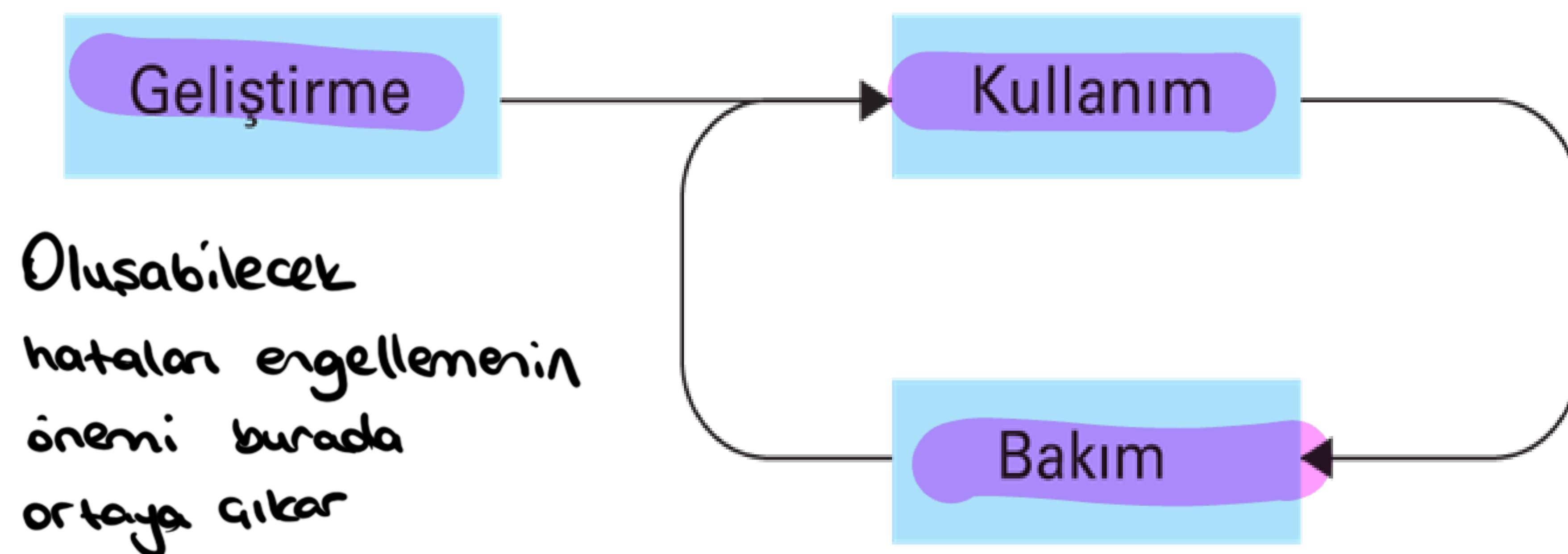
```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}
```

Piaget
e.Schüler

7.2 Yazılımın Yaşam Döngüsü

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

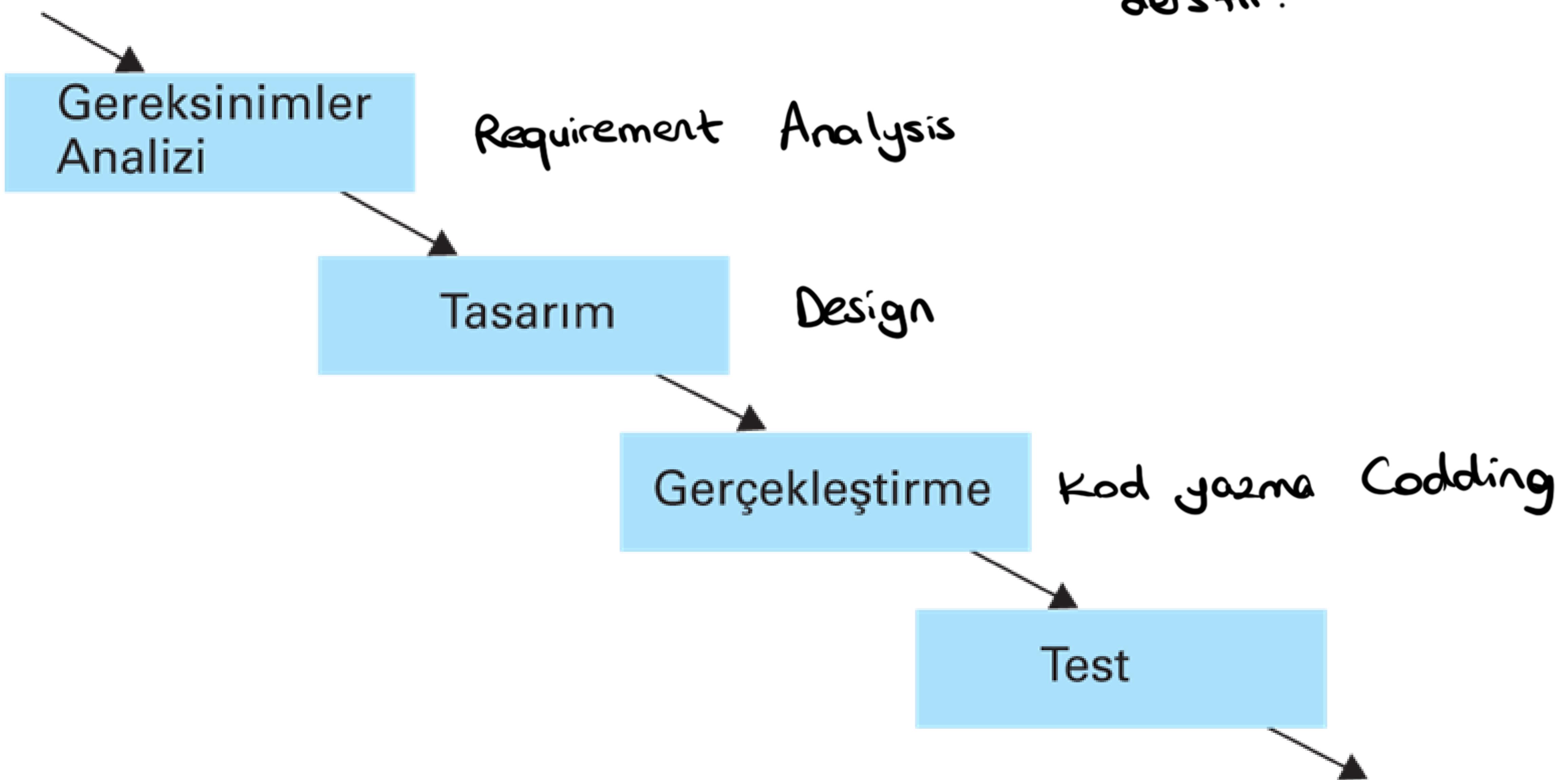
Şekil 7.1 Yazılımın Hayat Döngüsü



Firma kurulduktan bir süre ~ yaklaşık 3 yıl sonra ~ bakım garantisi verir.

Şekil 7.2 Yazılım hayat döngüsünün geleneksel geliştirme safhası

Yazılım müh.
her bir başlık bir
dersdir.



Gereksinim Analiz Bölümü



Gereksinimler

→ Her şeyi baştan oluşturma , müşteri ile iletişim
ihtiyac tesbiti

- Uygulamaya yönelik



Şartname

→ Her gerek ihtiyaç ekstra fatura

- Teknik açıdan
- Yazılım gereksinimler belgesi

③ Tasarım Bölümü

- Metodolojiler ve araçlar *Hangi programlara dili, diyagramlar nesneler, nesneler arası ilişkiler.*
- İnsan arayüzü (psikoloji ve ergonomi)



önyükt tasarım
(inşaat marketler proje)
satınması

kullancılar kitle

Gerçekleştirme Bölümü

Kod yazma bölümü

Yapay zeka AI

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

Tasarım
dökümanına
bağlı

Test Bölümü

- Geçerlilik testi (validation Test)
 - Sistemin şartnameye uyduğunu onaylama
- Zaafiyet testi
 - Yazılım açıklarının bulunması

Siber güvenlik anası

Acaba
koda doğru
mu doktuk

Jonalarla
güncellenebilir
yapılır → Jonalar bir seyi
bozuyor mu?

7.3 Yazılım Mühendisliği Metodolojileri

- **Şelale Modeli** (waterfall)

Geri dönüşsüz

- **Artırımlı Model**

- Prototipleme

Eklenteli şekilde
gelisir

- piyasaya hızlı ürün
- yarı sunum, yarı sunum



- **Tekrarlamalı model**



Kesin yolda düzülür

piyasada pincelletirme

- **Açık kaynaklı geliştirme**

- **Çevik Metotlar**

↳ Agile

Analiz ve
test ekibi
fazla yoktur.

10 kişiden az
yazılım
şirketi

Açık kaynak kodlu
linas ve fileri mülkiyet hakları ve
açık kaynak kodlu ürün → açık kayr.

~~kapatıcı kayr~~

7.4 Modülerite

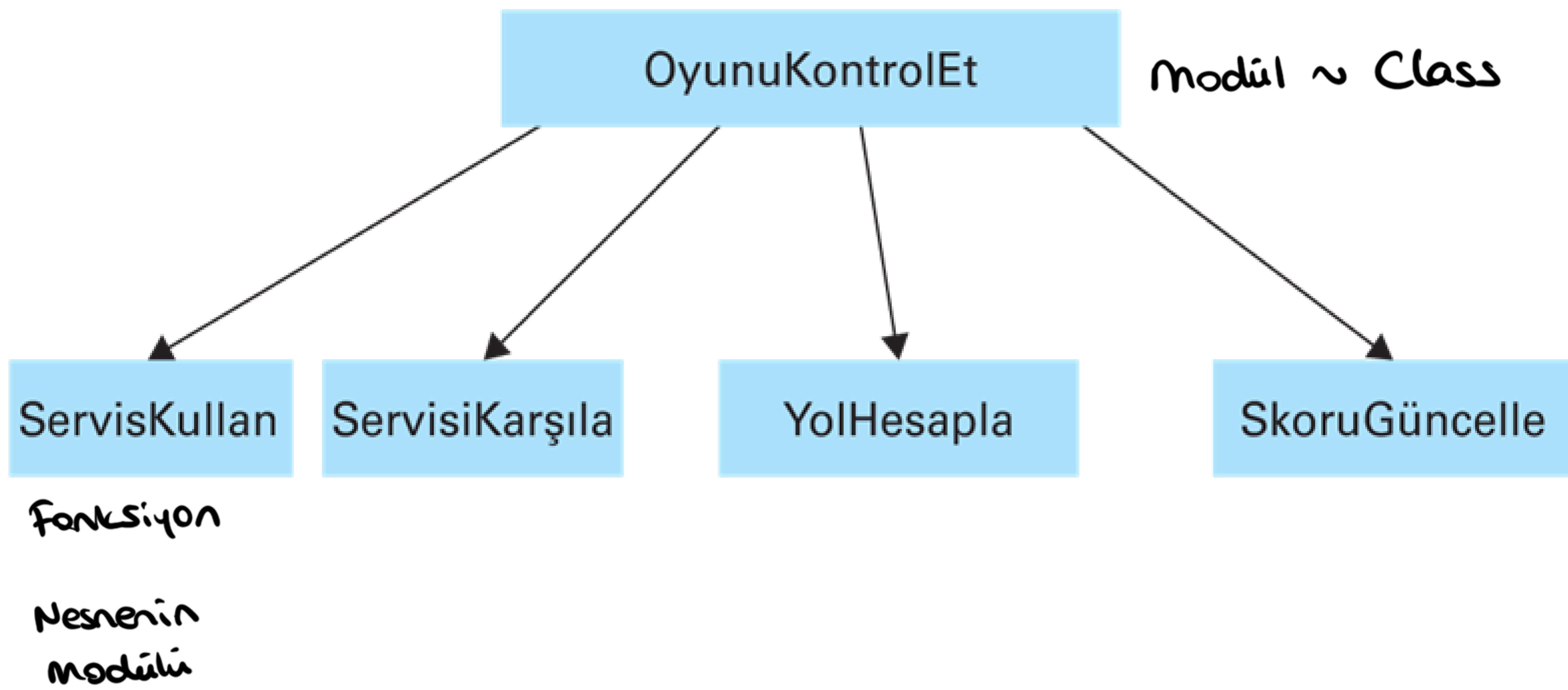
Nesneye yönelik programlama modülilik sağlar.

- 1.nov. • Fonksiyonlar – Zorunlu paradigma → C gibi
- 2.nov. Nesne – Nesne ilişkisine dayalı paradigma
– Nesne İşbirliği diyagramları → Nesneye yönelik
Prog. ise C#

"Nesne, fonksiyondan daha geniş bağımsız
bir sınımdır."

=)

Şekil 7.3 Modülarite için Basit yapı grafiği



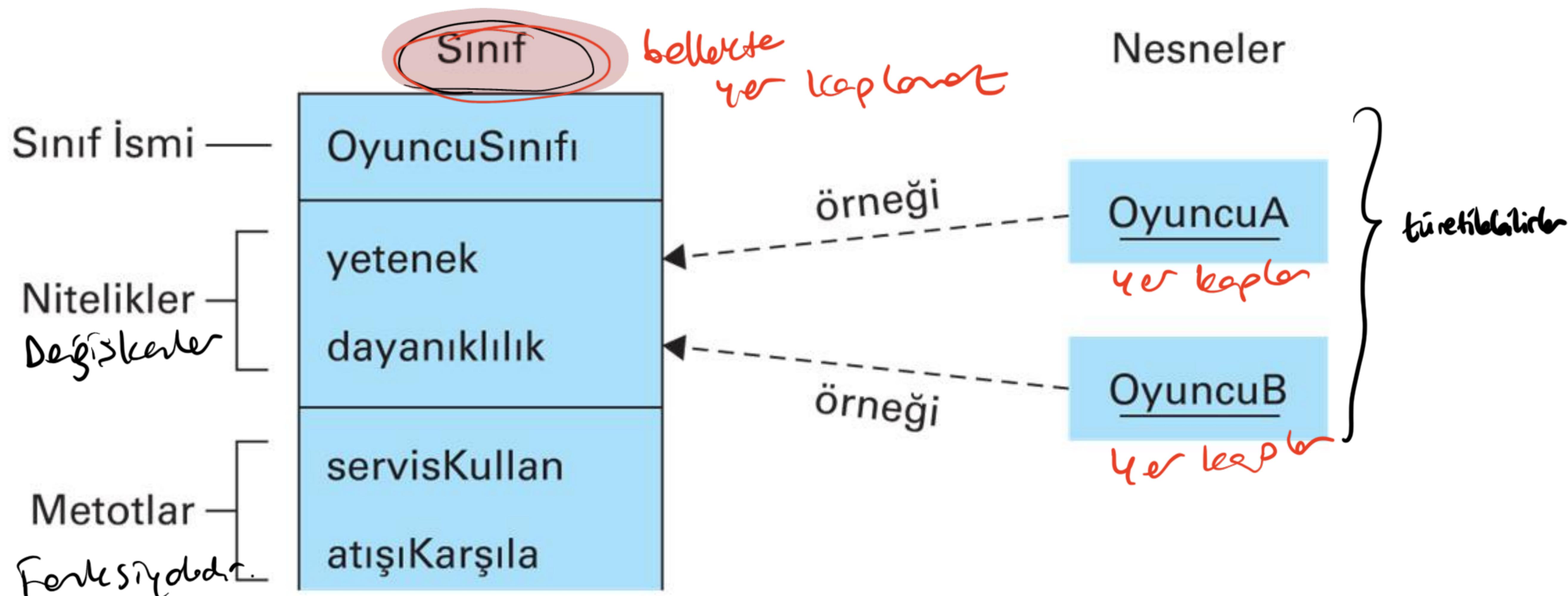
Class Diagram (Tasarruf Aşaması)

Şekil 7.4 OyuncuSınıfı ve örneklerin yapısı

Sınıflar → Sınırlı

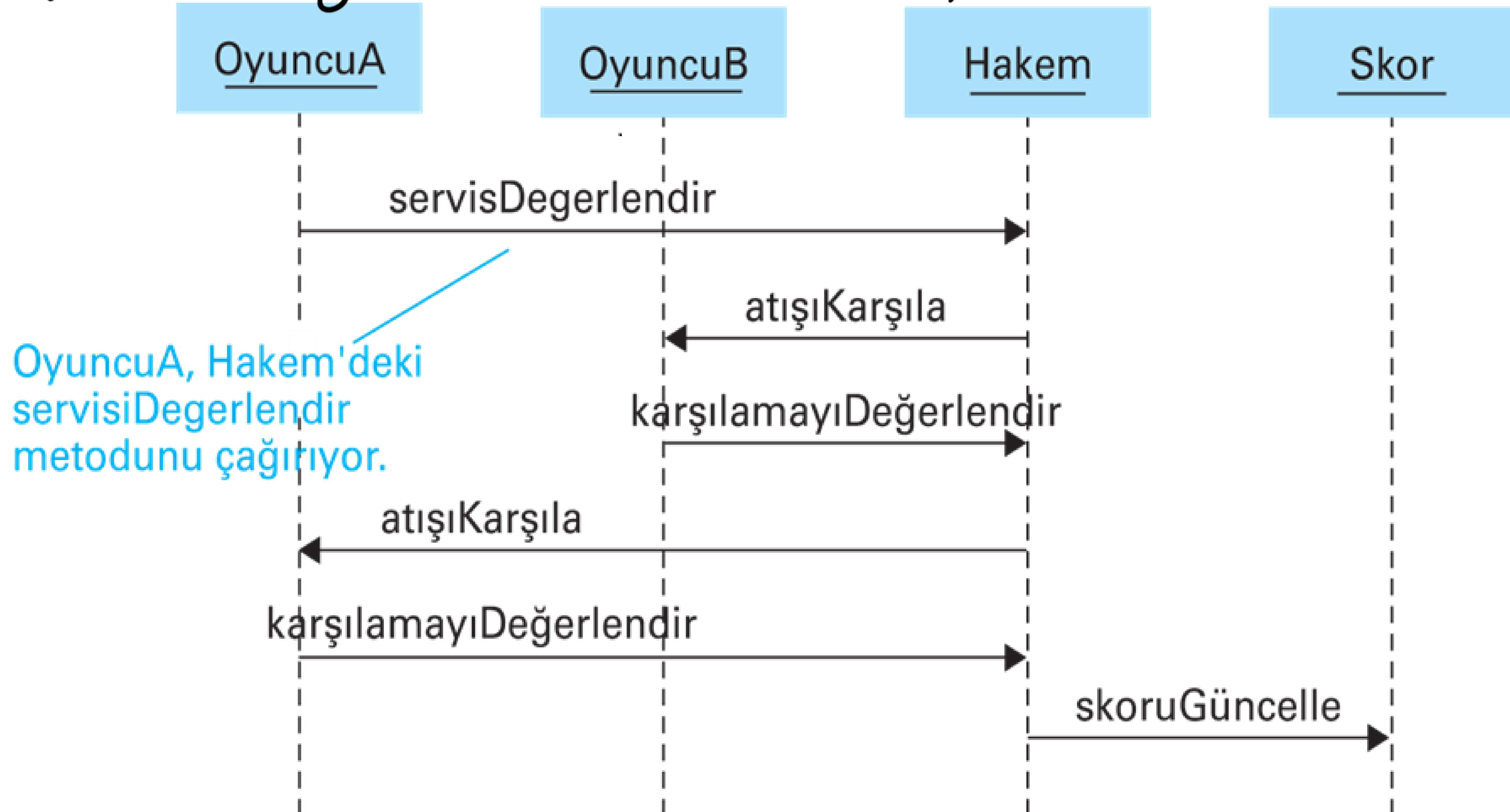
(Genel Sınıflar)

Nesneler → Tımsız, Karakteristik (Sınıflarda oluşturulan)



Şekil 7.5 OyuncuA'dan gelen servisin sonucunda nesneler arasındaki etkileşim

Sequence Diagram (Class Arası İlişki)



7.5 İş Araçları

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

→ kapay zeka daha bu diyagraları
çireret bu aralar inşa olustur

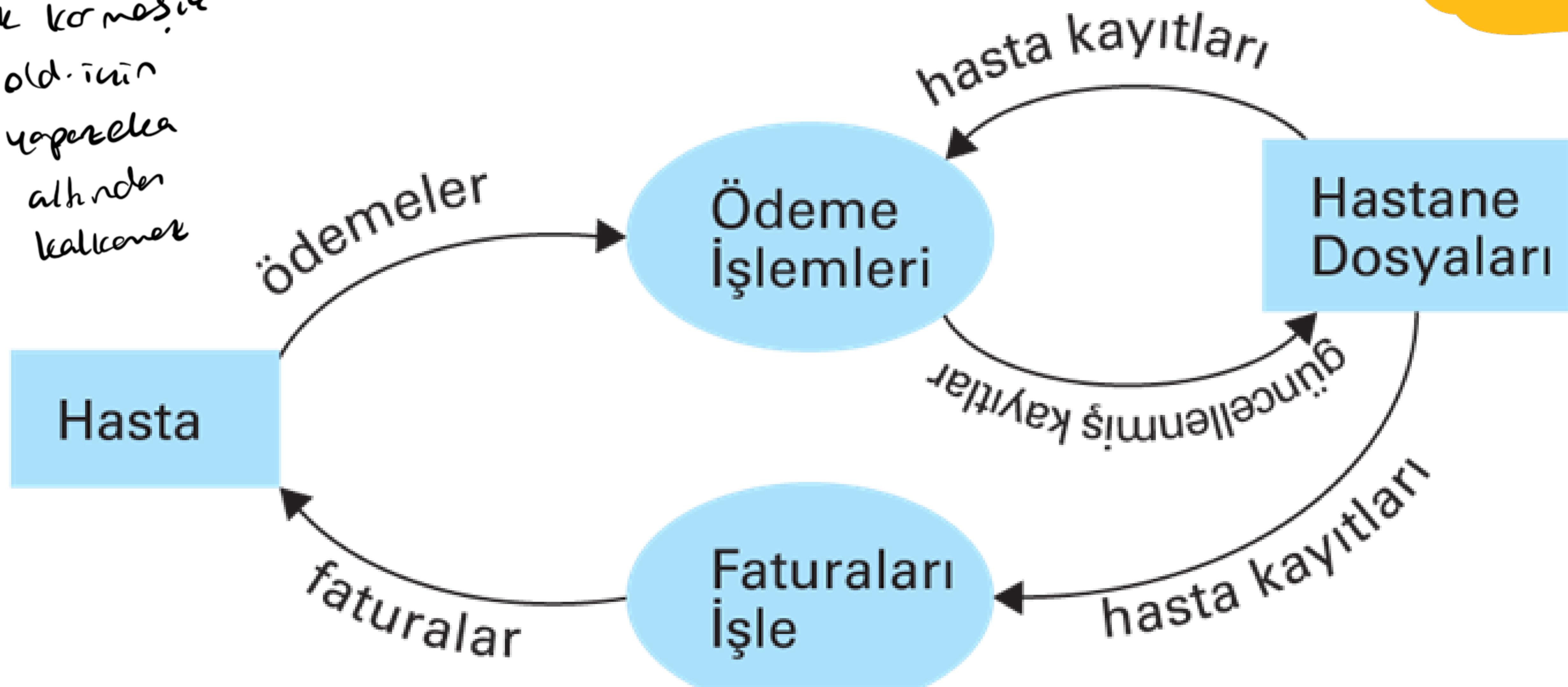
Kapay zeka
kodunu yarar.

Hastane Bilgi Sist

Şekil 7.8 Bir veri akış grafiği

hök konsos.^{ik}
old. iñin
yapozelka
alt.inden
kalkonex

veri
nereye
dönre
akacake



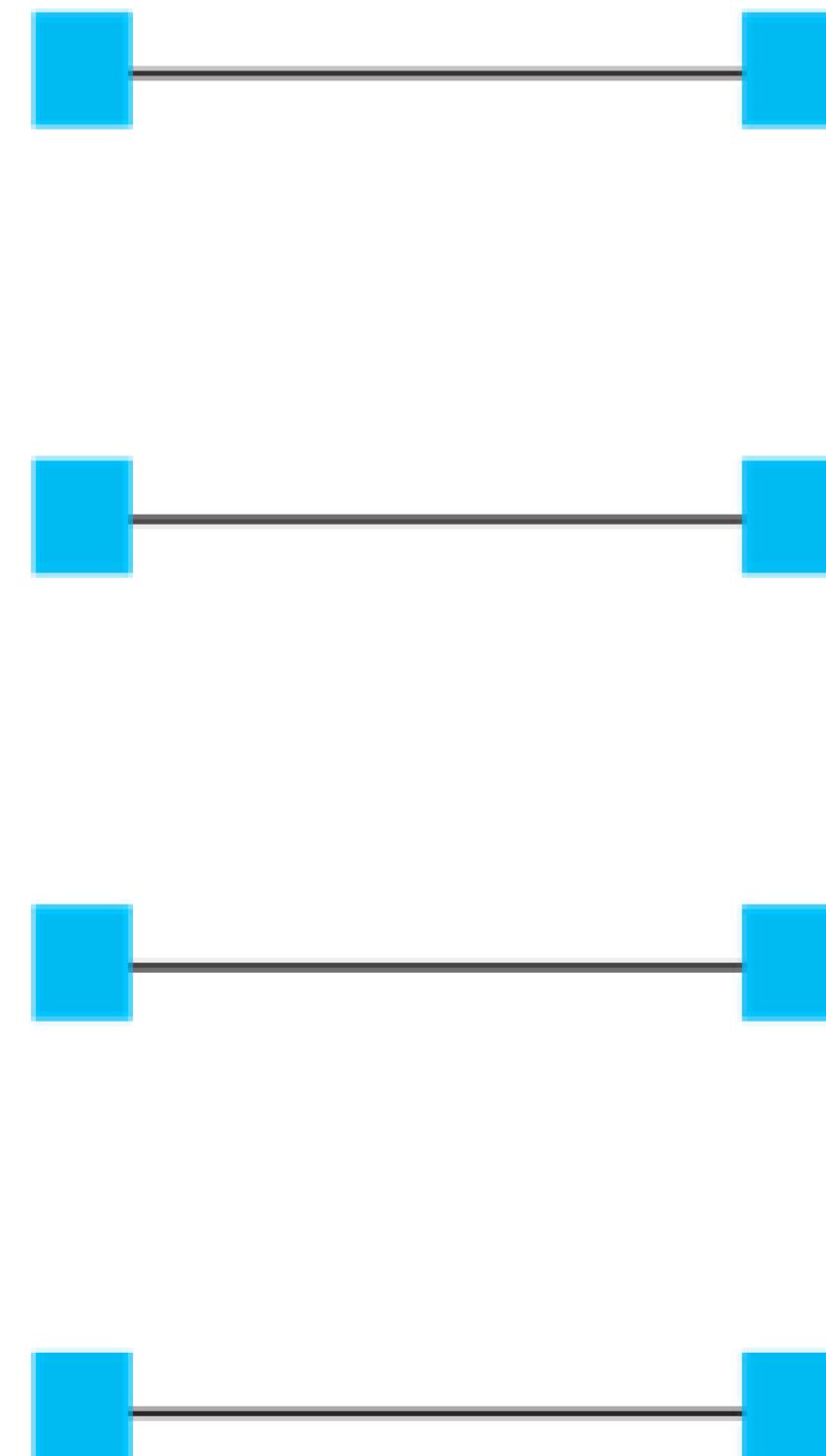
Şekil 7.11 X ve Y tipi varlıklarları arasında bire bir, bire çok ve çoğa çok ilişkiler

one-to-one

Bire bir

X tipi
varlıklar

Y tipi
varlıklar

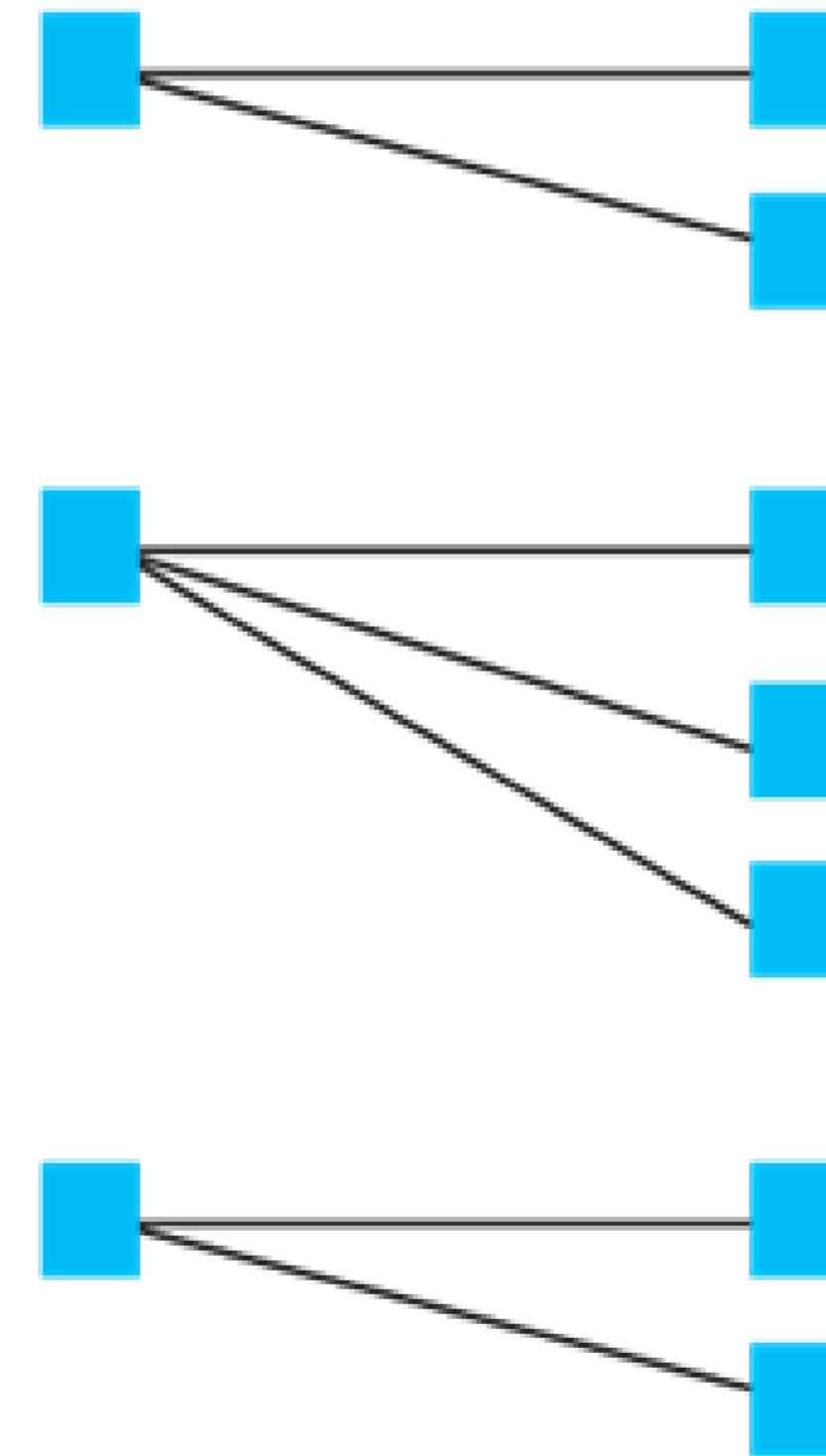


Hasta ~
~ Özel Oda

Bire çok

X tipi
varlıklar

Y tipi
varlıklar

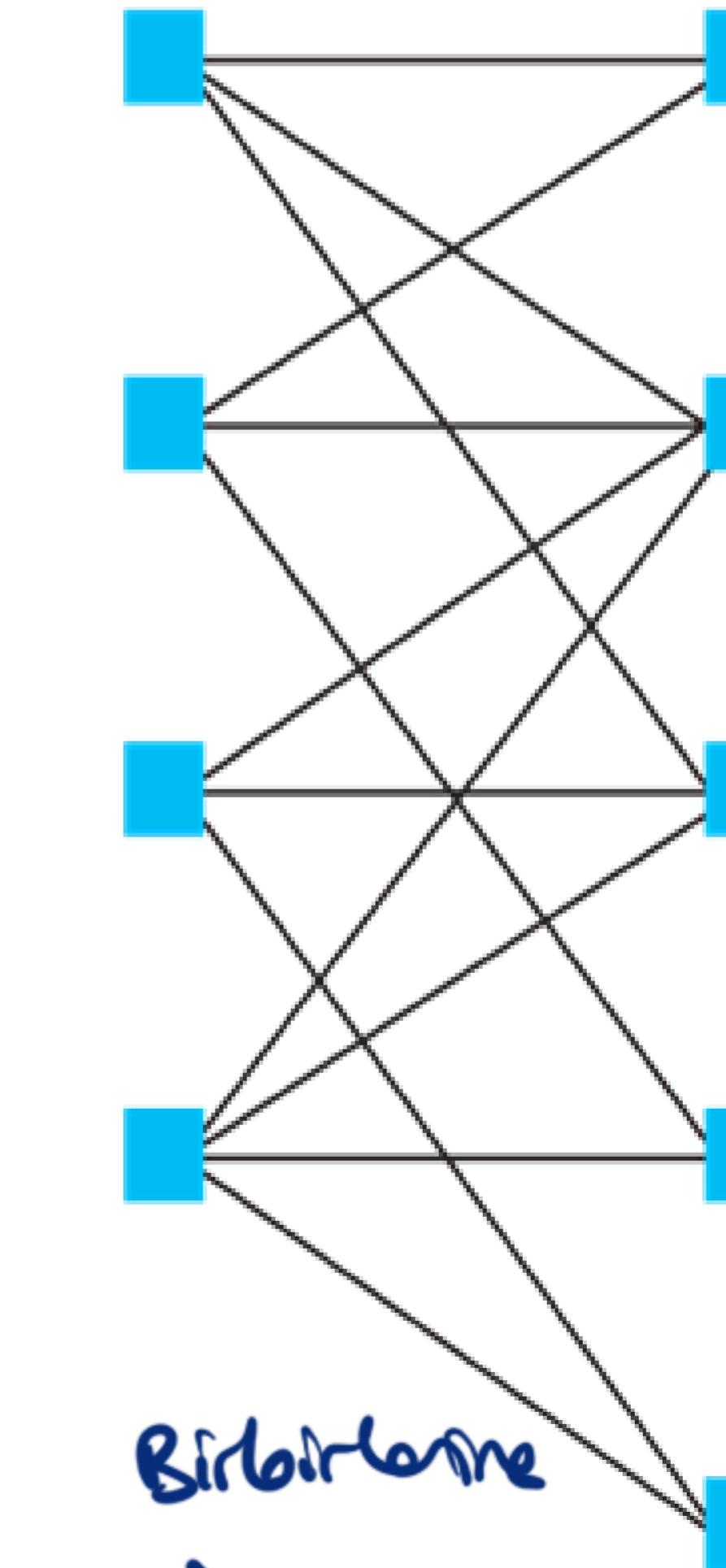


Doktorlar
→ hasta

Çoğa çok

X tipi
varlıklar

Y tipi
varlıklar



Birbirlerine
Donusan
Doktorlar ve hasta

one
sayf
21

Nesneler
arası
ilişki
hangi tip
tespit
edilmesi
gerekir.

Birleşik Modelleme Dili

(UML)
unified Modelling
Language

a) use case Diagram

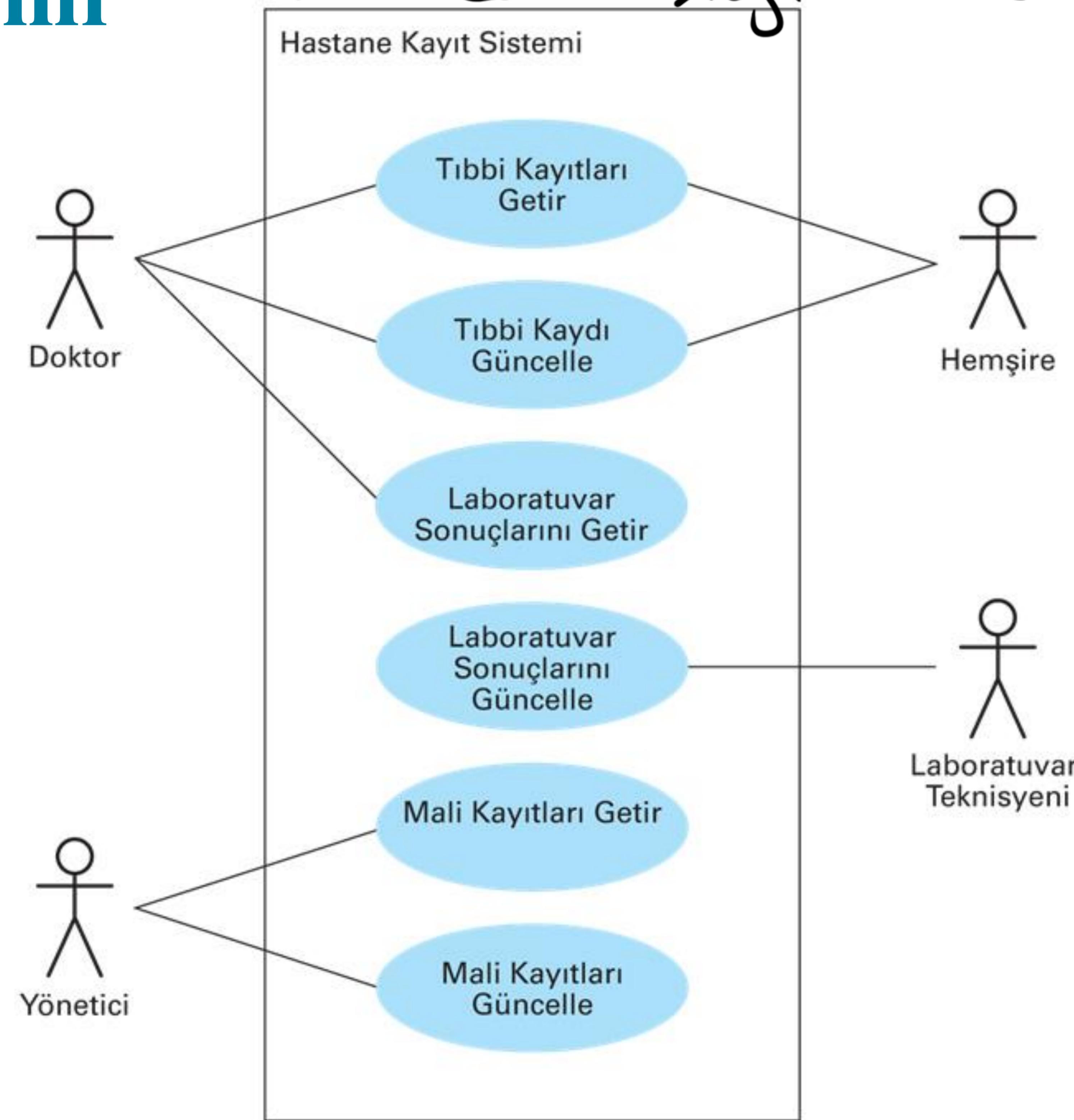
- Kullanım durumu diyagramı
 - Kullanım diyagramı
 - Elemanlar
- Sınıf diyagramı

Program tasarımının sıktı
özelliklerini gösterir

Şekil7.9 Basit bir kullanım durumu diyagramı

use case Diagram (Kullanım Durumu)
Diagram

?
Kim
nereye
erisebilecek
? 2

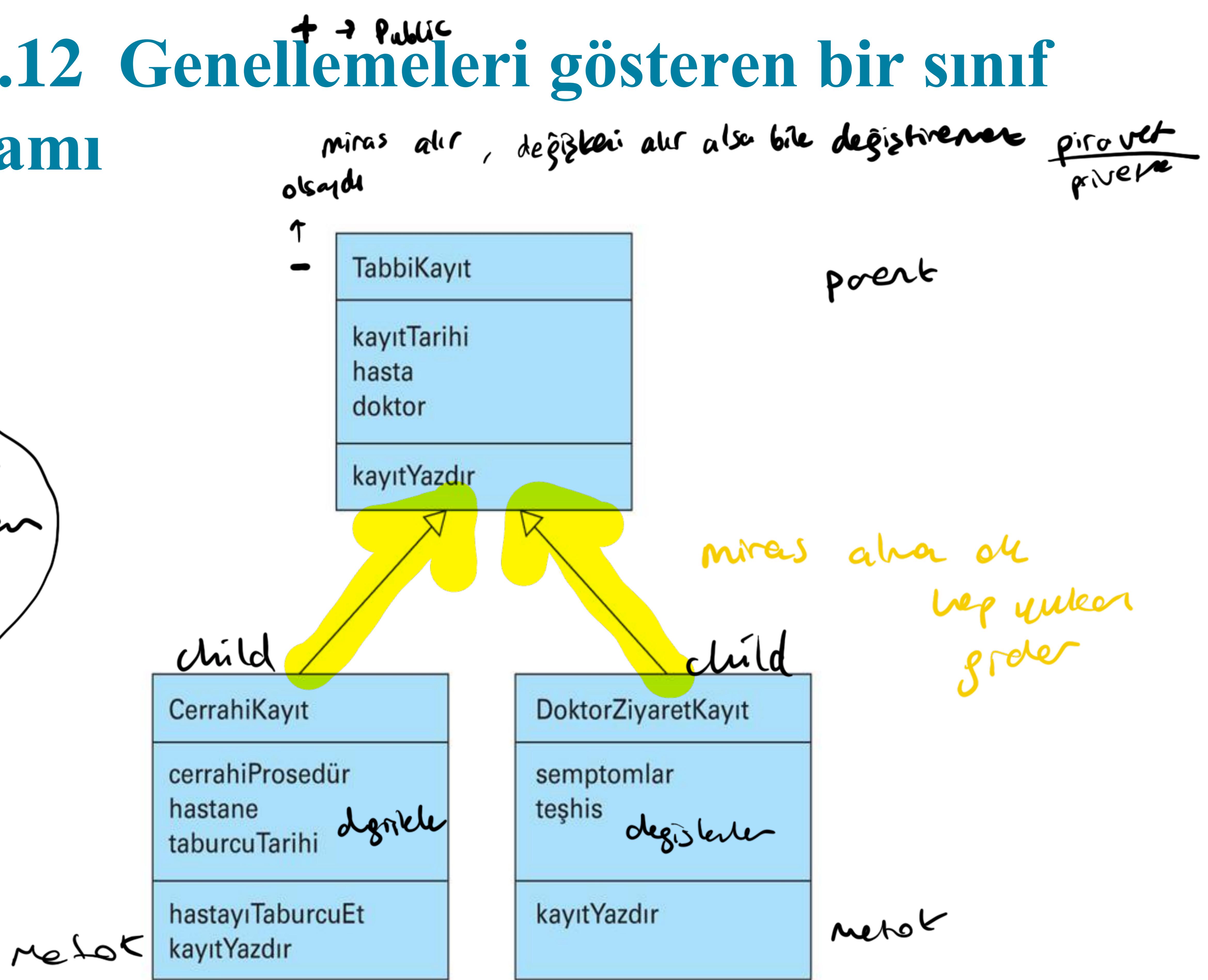


Şekil 7.10 Basit bir sınıf diyagramı



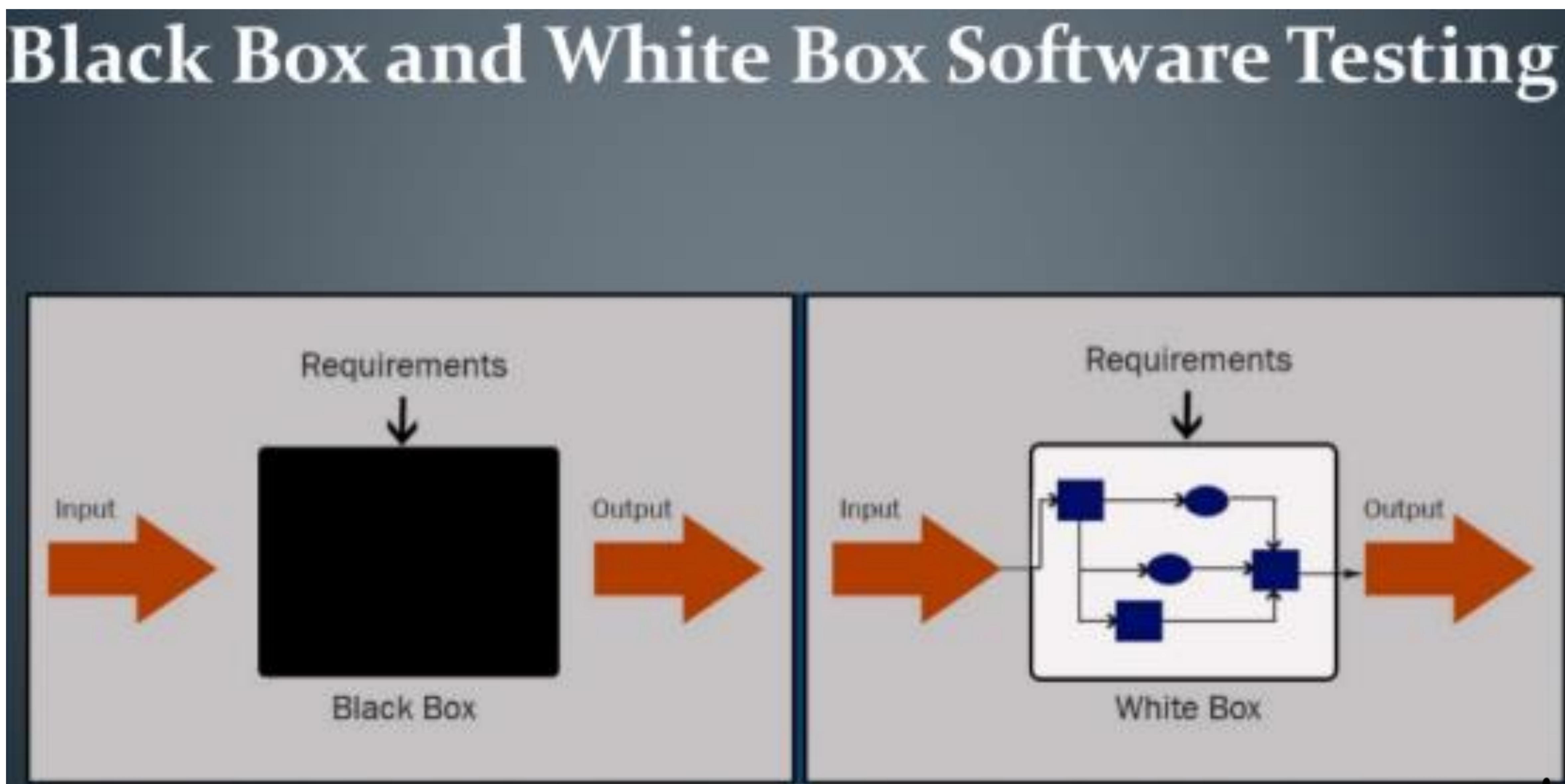
Şekil 7.12 Genellemeleri gösteren bir sınıf diyagramı

DİP NOT
En ince.
ayrıntıya kader
tartmeye bilisen
yapay zeka
yper.



7.6 Kalite Sigortası *Quality Assurance*

- 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 →
 - Class Diagram
 - Use Case Diagram
 - Akış Diagramları
 - Blocked Diagram
- Teknik dökümantasyonu
 - Kurulum, güncelleme vb. için

firma, IT ekibi için yazılar. Güncellene
yorumlar ve yeni sürümler için önemlidir.

- Comment satırları
- Açıklayıcı değişken isimleri kodun derlenmesini yavaşlatmaz.



7.8 İnsan-Makine Arayüzü

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



Ekranda,
en çələbə
bañılı şəhər

Human - Comp. Bilimi
HCE ömrək
- Google
- turku simesi
yepniş
olsaydı her
sey ör. günək
cəsədər

7.9 Yazılım sahipliği ve yükümlülüğü

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

Yerelinde suisin
PATENTİ ÖNEMLİ

① Algoritmalar patentini

ör/Zip, rar, gibi sıkıştırma
dosyaları
Sıkıştırma Patentid

② Yöntem patentii

Araştır
Çöp kütüphanesi
Zerox

Fikri Mülkiyet

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

windowsun o sürüme
kullanma sahne hakları.

• Patent alınmadığında fikri mülkiyet hukuki ,
stop up
→ ürün satılabiliyor olmazdır. → Daha sonra verilebilecek hukuk .