BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

DERS 10

Yazılım Mühendisliği

*Ders sunumları ticari herhangi bir amaç içermez

Yazılım Mühendisliği

Büyük karmaşık yazılım sistemlerinin **geliştirilme sürecinde** karşılaşılan **problemleri incelemek**, bu sürece yön verecek **prensipler bulmak** ve sürecin sonunda **verimli ve güvenilir** yazılım ürünleri ortaya çıkarmak Yazılım mühendisliğinin amacıdır.

Yazılım Mühendisliği

- Yazılım Mühendisliği Disiplini
- Yazılım Yaşam Döngüsü
- Yazılım Mühendisliği Metodolojileri
- Modülerlik
- İş Araçları
- Test
- Belgeler
- Yazılım Sahipliği ve Sorumluluğu

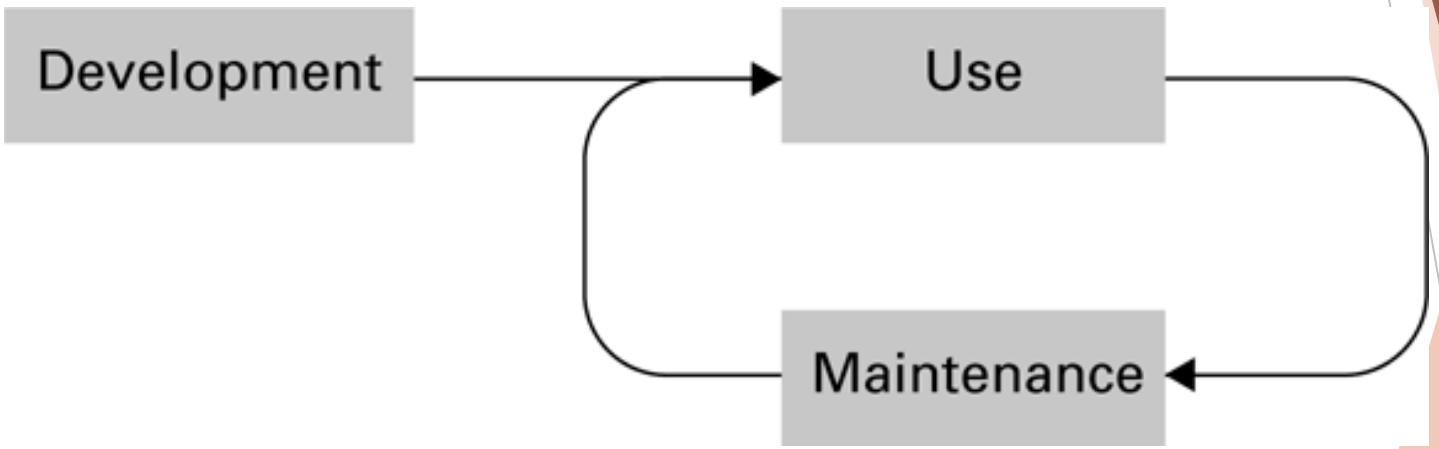
Yazılım Mühendisliği Disiplini

- Diğer mühendislik alanlarından farklı
  - Prefabrik bileşenler
  - Metrik
- Pratisyenler ve teorisyenler
- Profesyonel Organizasyonlar: ACM, IEEE, vb.
  - Mesleki etik kurallar
  - Standartlar

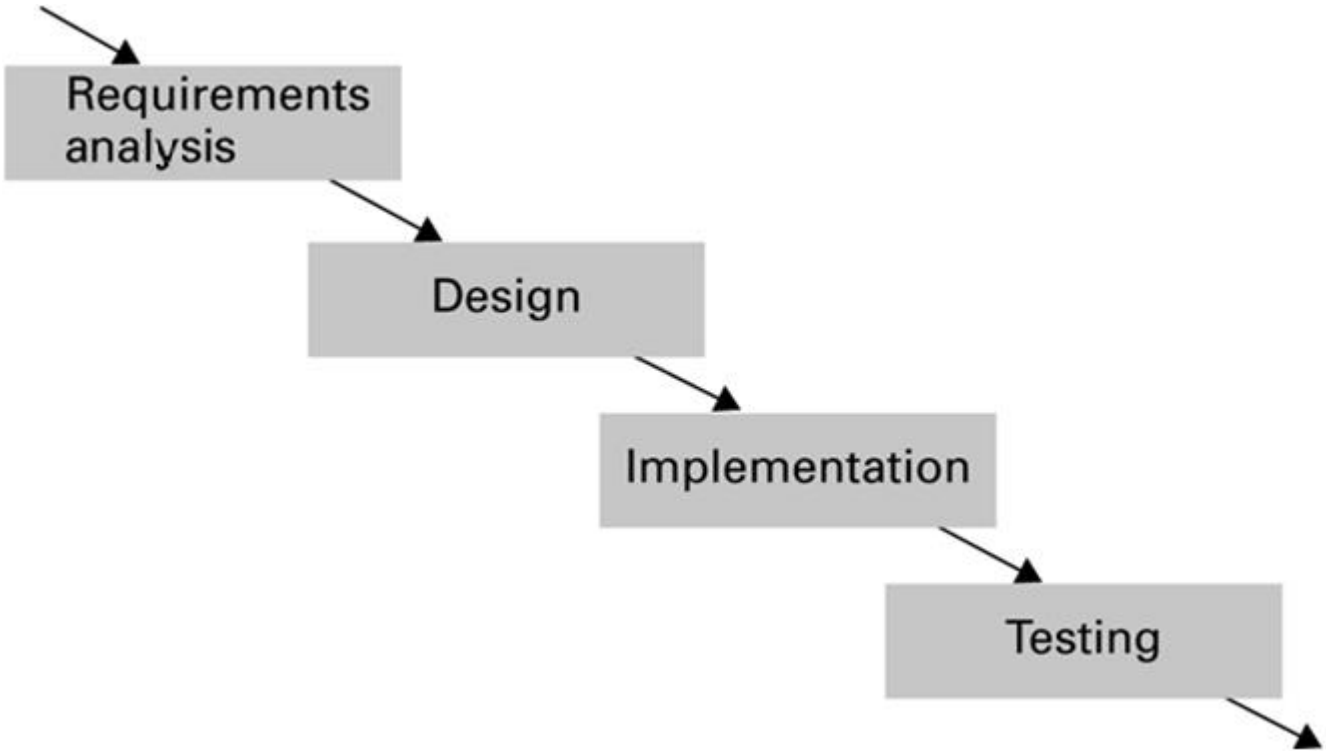
Bilgisayar Destekli Yazılım Mühendisliği (CASE) araçları

- Proje planlaması
- Proje Yönetimi
- Belgeleme
- Prototipleme ve simülasyon
- Arayüz tasarımı
- Programlama

Yazılım yaşam döngüsü



Yazılım yaşam döngüsünün geliştirme aşaması



Analiz aşaması – Gereksinim Analizi

- Gereksinimler
  - Uygulama odaklı
- Özellikler
  - Teknik odaklı
- Yazılım gereksinimleri belgesi

Tasarım Aşaması

- Metodolojiler ve
- İnsan arayüzü (psikoloji ve ergonomi)

Uygulama Aşaması

- Tasarımdan sistem oluştur
  - Yazma programları
  - Veri dosyaları oluştur
  - Veritabanları geliştirin

- “Yazılım analisti” nin “programcı” ya karşı roldür

Test Aşaması

- Doğrulama testi
  - Sistemin teknik özellikleri karşıladığını onaylayın
- Hata testi
  - Hataları bul

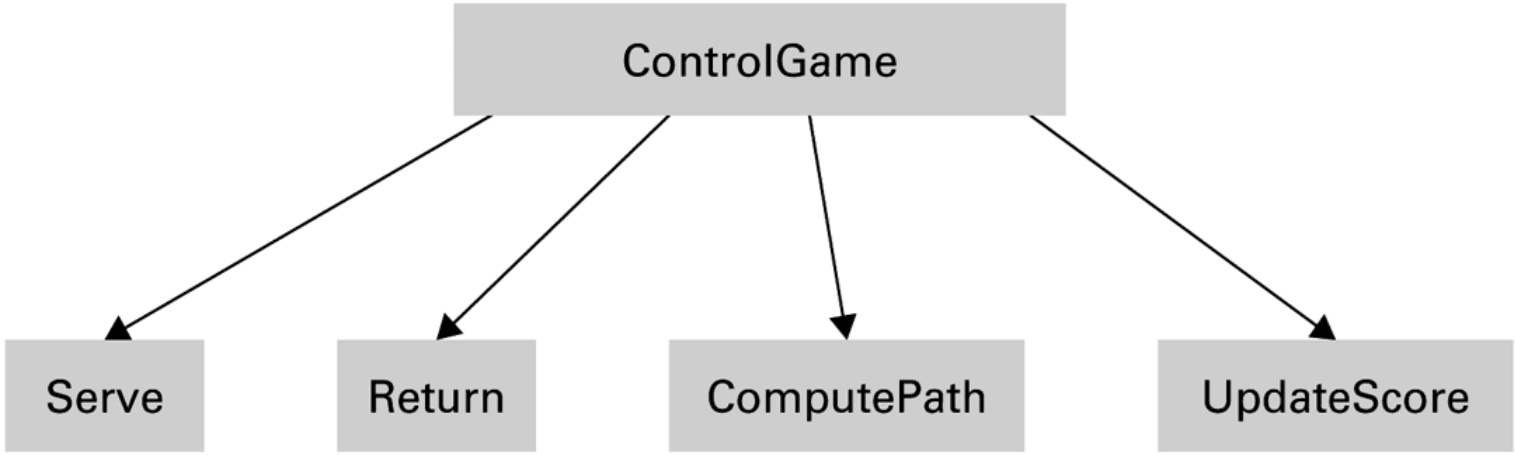
Yazılım Mühendisliği Metodolojileri

- Şelale Modeli
- Artan Model
  - Prototipleme (Evrimsel vs Throwaway)
- Açık Kaynak Geliştirme
- Aşırı Programlama

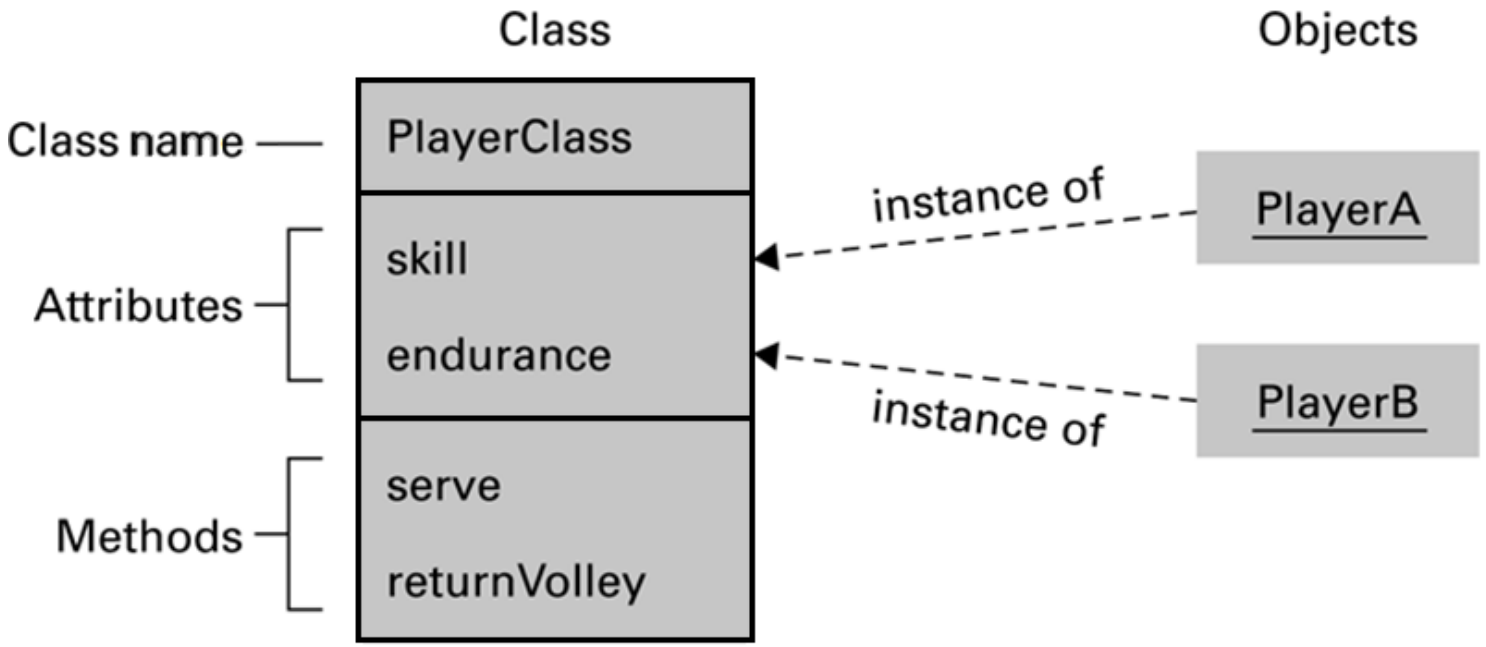
Modüllerlik

- Fonksiyonlar - Etkileyici paradigma
  - Yapı çizelgeleri
- Nesneler - Nesne yönelimli paradigma
  - İşbirliği diyagramları
- Bileşenler - Bileşen mimarisi

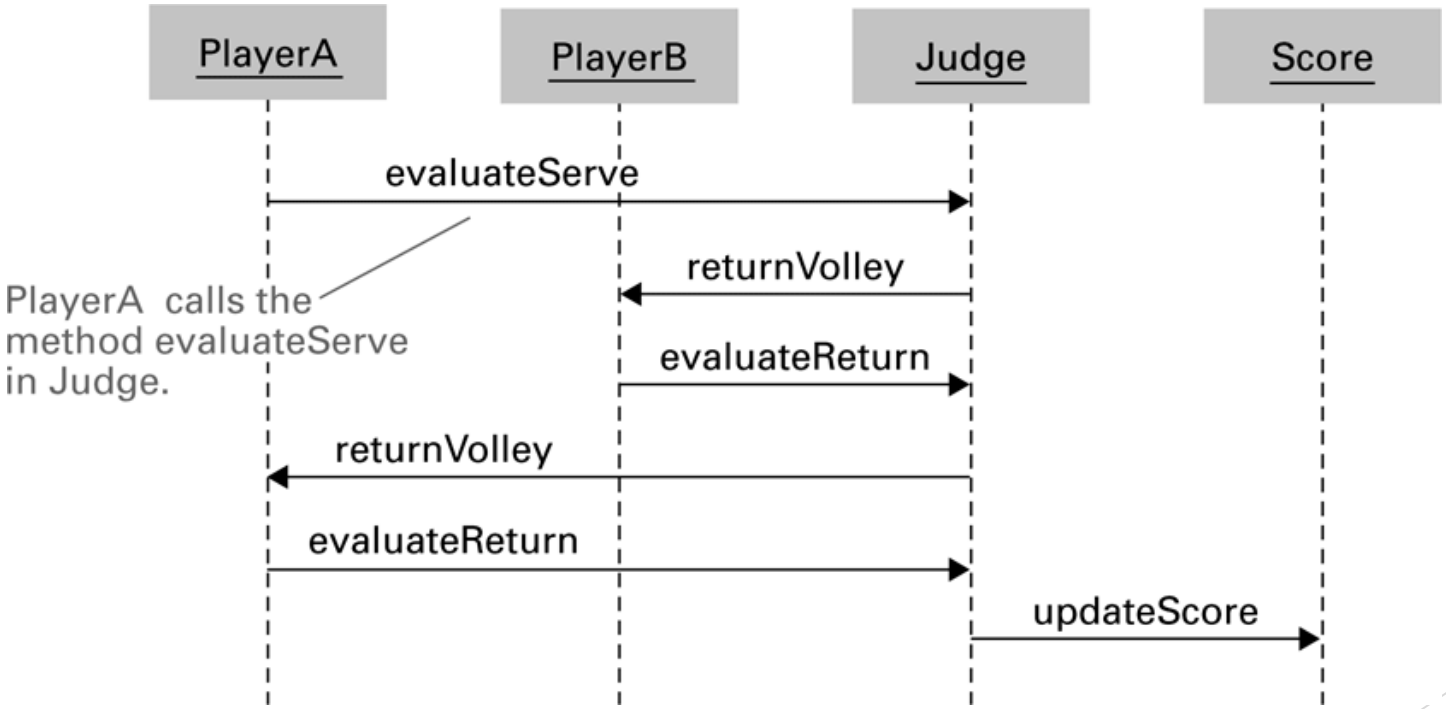
Basit bir yapı tablosu



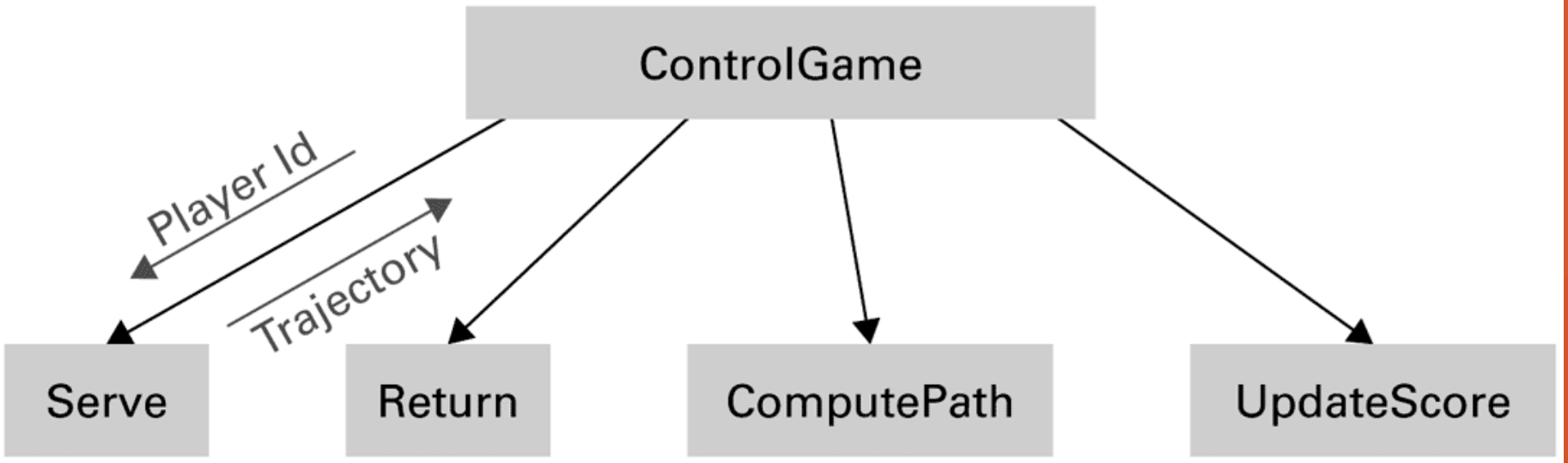
PlayerClass'ın yapısı ve örnekleri



PlayerA'nın hizmetinden kaynaklanan nesneler arasındaki etkileşimi



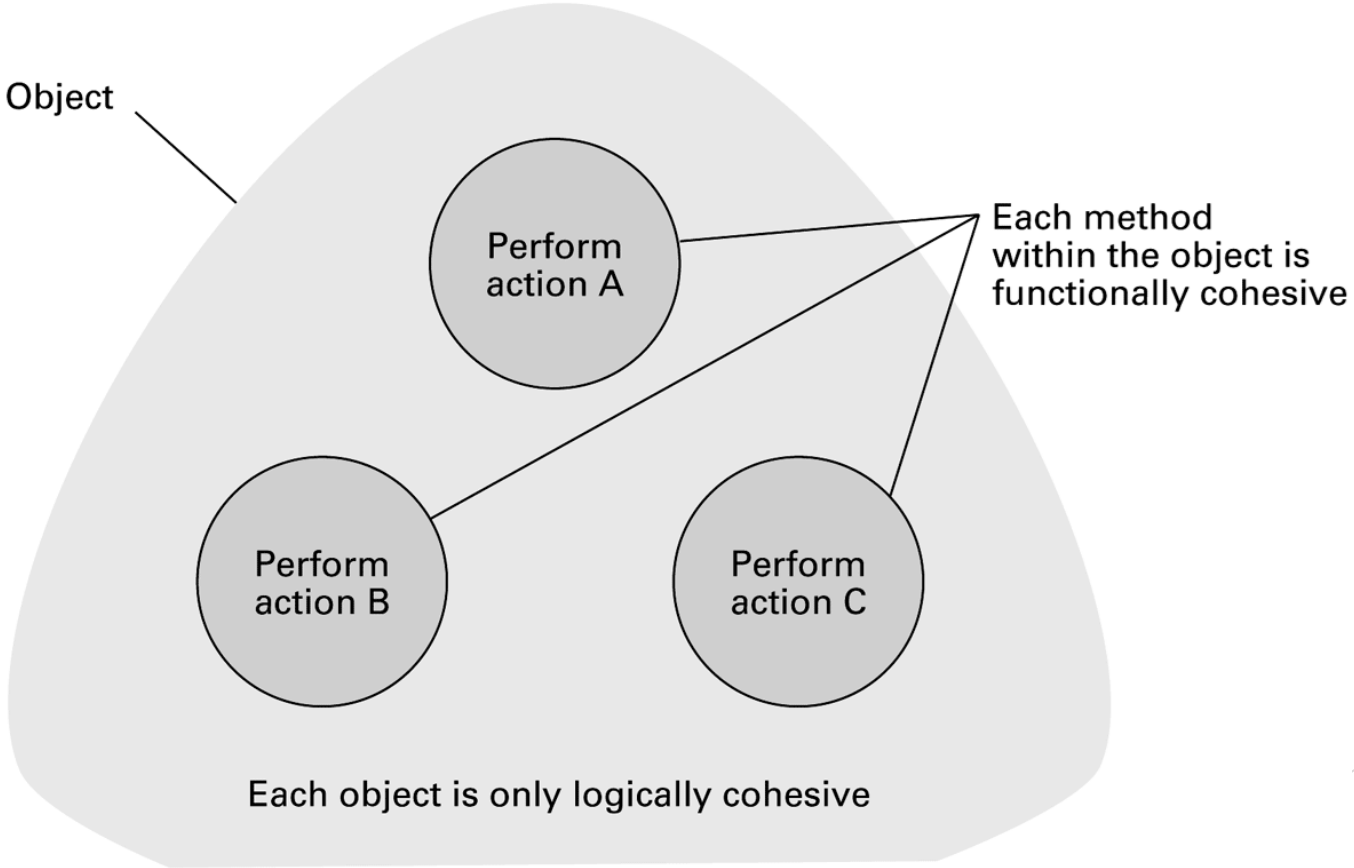
Veri birleştirmeyi içeren bir yapı şeması



Birleşme ve Bağlanma

- bağlantı
  - Kontrol Birleştirme
  - Veri birleştirme
- birleşme
  - Mantıksal uyum
  - Fonksiyonel birleşme

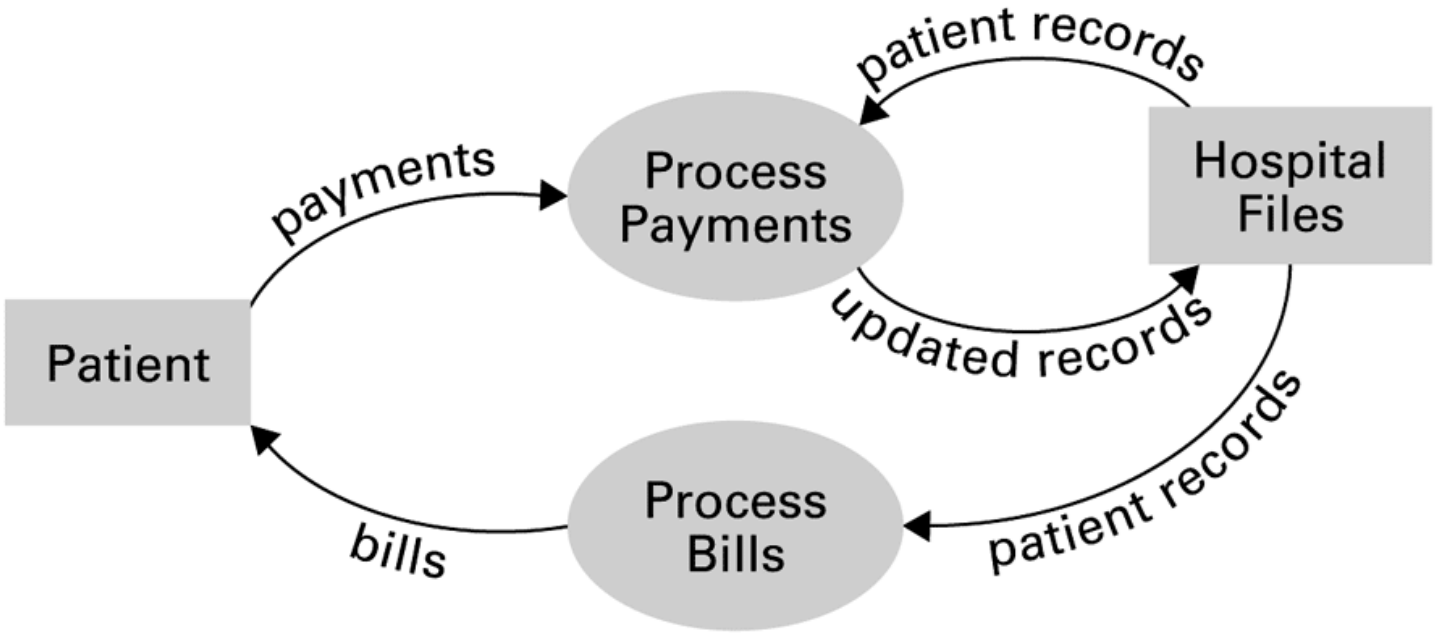
Bir nesne içinde mantıksal ve işlevsel uyum



İş Araçları

- Veri Akış Şeması
- Varlık ilişki diyagramı
  - Bire bir ilişki
  - Bire çok ilişki
  - Çoktan çoğa ilişkisi
- Bilgi sözlüğü

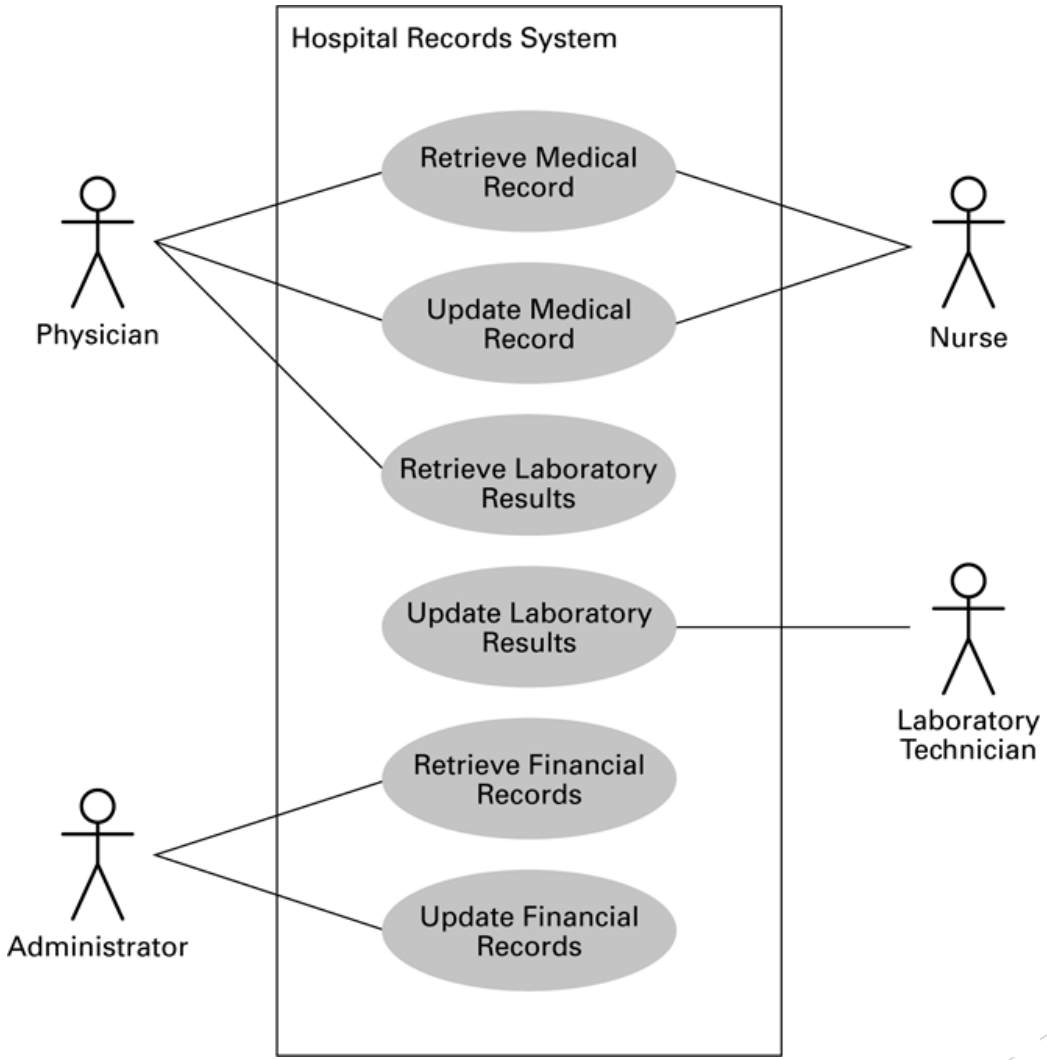
Basit bir veri akışı şeması



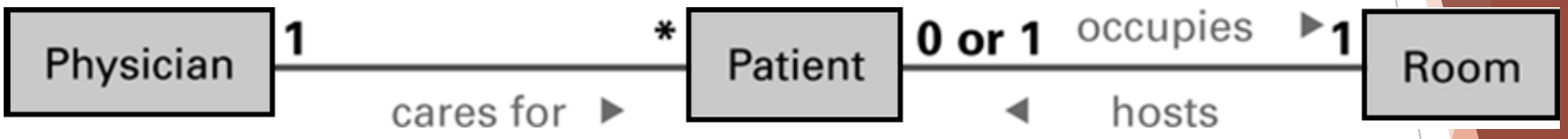
Birleştirilmiş Modelleme Dili (Unified Modelling Language - UML)

- (Use-Case) Durum-kullanım Diyagramı
  - Aktörler
- Sınıf diyagramı

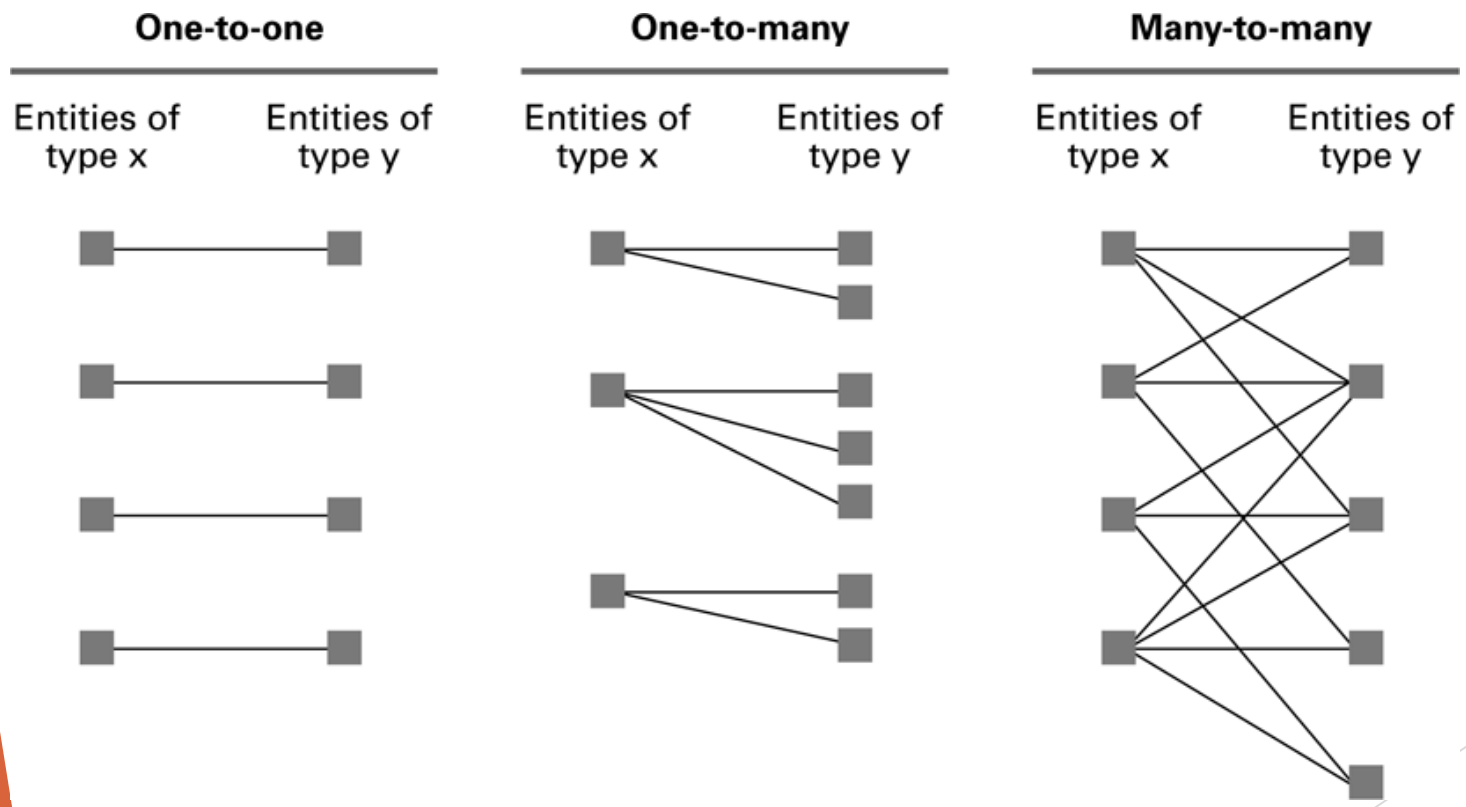
Basit kullanım durumu şeması



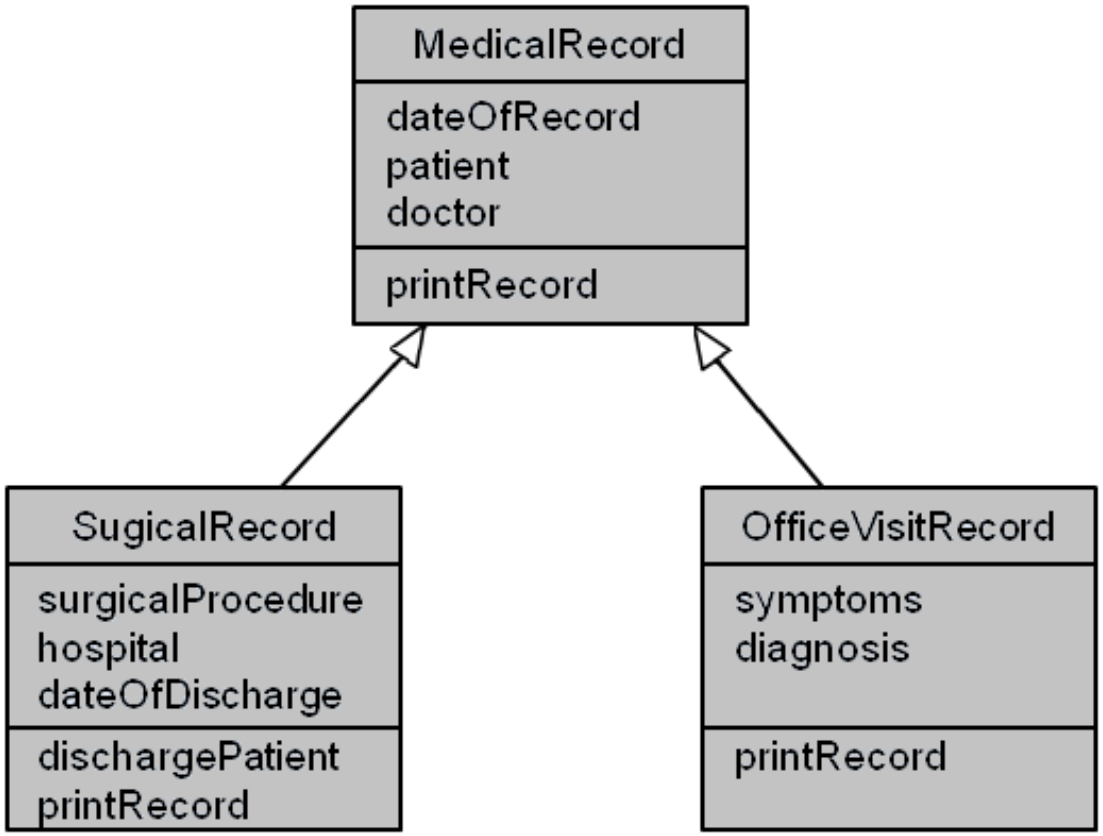
Basit bir sınıf diyagramı



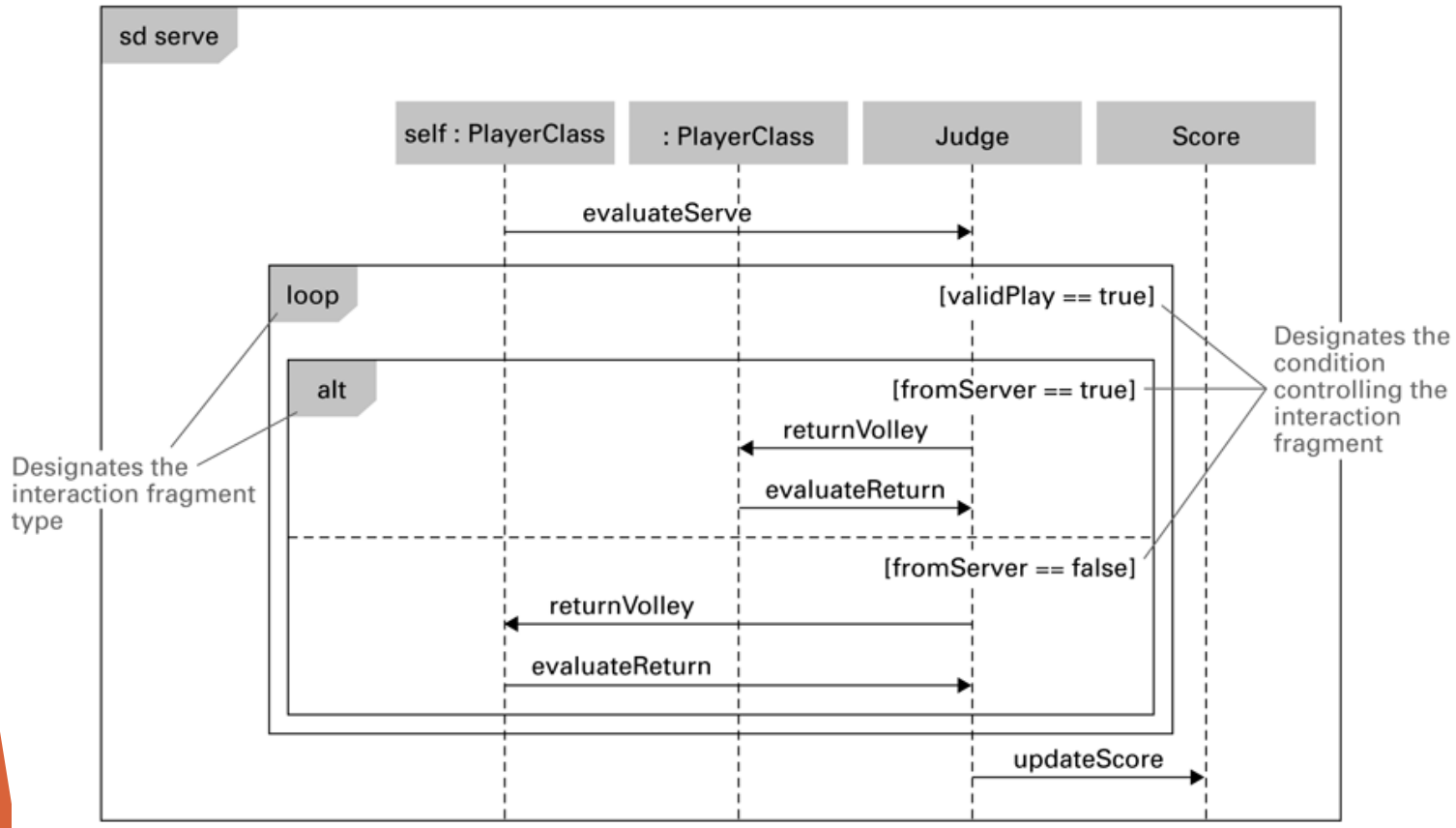
Bire bir, bire çok ve çoktan çoğa ilişkiler



Genellemeleri tasvir eden bir sınıf diyagramı



Genel atışı gösteren bir sıralama diyagramı



Tasarım örüntüleri

- Yinelenen problemleri çözmek için iyi tasarlanmış “şablonlar”
- Örnekler:
  - Bağdaştırıcı modeli: Bir modülün arayüzünü mevcut gereksinimlere uyarlamak için kullanılır
  - Dekoratör kalıbı: Aynı faaliyetlerin farklı kombinasyonları gerektiğinde karmaşıklığı kontrol etmek için kullanılır
- Mimarlıkta Christopher Alexander'ın eserinden esinlenildi

Yazılım kalite ve güvencesi

- Kaliteli Yazılım üretme
- Kalite kontrol
- Sertifikasyonlar ve standartlar ISO, IEEE, ACM

Yazılım Test Stratejileri

- Cam kutu testi
  - Pareto prensibi
  - Temel yol testi
- Kara kutu testi
  - Sınır değer analizi
  - Beta testi

Belgeleme - Dokümantasyon

- Kullanıcı Belgeleri
  - Tüm müşteriler için basılı kitap
  - Çevrimiçi yardım modülleri
- Sistem Belgeleri
  - Kaynak kodu
  - Tasarım belgeleri
- Teknik döküman
  - Kurulum, kişiselleştirme, güncelleme vb.

Yazılım Mülkiyeti

- Telif hakkı
  - Fikri mülkiyetin mülkiyetini korurken bir ürünün serbest bırakılmasına izin ver
  - Tüm eserlerde iddia:
    - Özellikler
    - Kaynak kodu
    - Son ürün

Yazılım Mülkiyeti

- Yazılım lisansı
  - Sahipliği devretmeden kullanıcıya belirli izinleri veren yasal bir anlaşma
- Patentler
  - Yeni, kullanılabilir ve benzer arka planlara sahip olmayanlar için açık olmadığını göstermelidir
  - Süreç pahalı ve zaman alıcıdır

Ders bitti

Erciyes Üniversitesi
Selçuk Üniversitesi
Sakarya Üniversitesi
Kahramanmaraş Sütçü İmam Üniversitesi
ders notları kaynak ve içerik olarak kullanılmıştır.