

# Sistem Analizi ve Tasarımı

6.Ders

Göksel Biricik

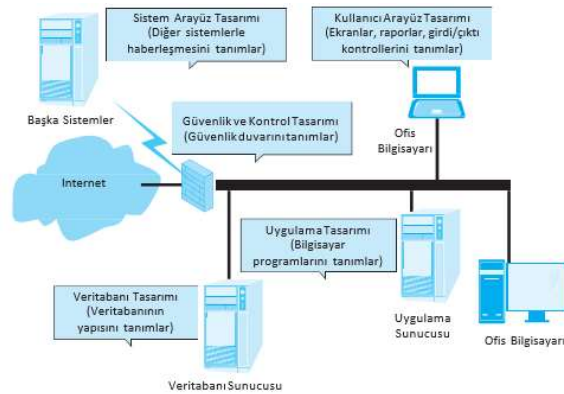
## Bu Derste

- Sistem Tasarımı
  - Altyapı Belirleme
  - Ön (Genel) Tasarım
  - Ayrıntılı Tasarım
  - Tasarım Modelinin Oluşturulması

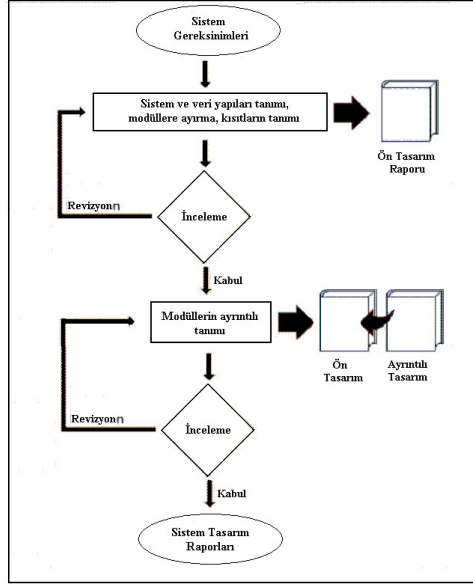
# Sistem Tasarımı

- Ön Tasarım
  - Alt yapı belirleme
  - Modül mimarisinin oluşturulması
- Ayrıntılı Tasarım
  - Sistem Etkileşimi tasarımı
    - Çıktı tasarımı
    - Girdi tasarımı
    - Arabirim tasarımı
  - Program tasarımı
  - Veritabanı tasarımı

## Sistem Tasarımı Bileşenleri



## Tasarım Aşamaları



## Sistem Tasarım Yaklaşımları

- Model Güdümlü Yaklaşımlar: Geliştirilen mantıksal modeller kullanılarak model güdümlü tasarım modelleri elde edilir. Bu modeller yeni kurulum ve gerçekleştirme için tasarılır.
  - Modern Yapısal Tasarım
  - Bilgi Mühendisliği
  - Prototipleme
- Hızlı Uygulama Geliştirme: Model hazırla, prototip yap, model hazırla, prototip yap, vb. döngü. Ortak uygulama geliştirme oturumları
- Ortak Uygulama Geliştirme: Tasarım tüm paydaşların katıldığı atölye çalışmaları ile gerçekleşir.

## Ön (Genel) Tasarım

- Sistem nasıl temin edilecek?
  - Sistem sıfırdan oluşturulabilir: Kurum içi geliştirme +/-?
  - Satın alınıp ihtiyaca göre özelleştirilebilir +/-?
  - Dışarıdan (hizmet olarak) temin edilebilir +/-?
- İşlevsel olmayan gereksinimler mimari tasarımı etkiler
  - Sistem ne kadar hızlı çalışacak? Kapasitesi ne olacak? Şifreleme ve virüs kontrol ihtiyacı var mı? Vb.
  - Yeni sistemi desteklemek üzere alınacak donanım ve yazılımlara yönelik mimari kararlar

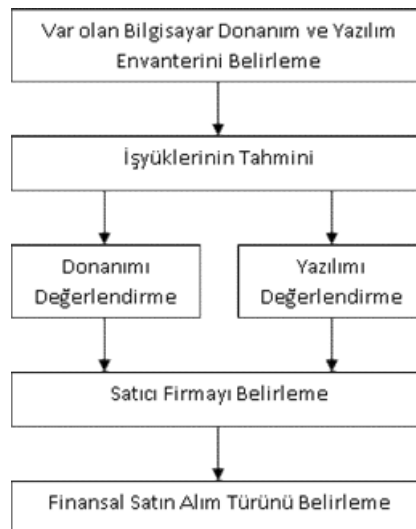
## Sistem Mimari Modellemesi

- Bilgi sistemi, merkezi mi yoksa dağıtık mı olacak? (Birçok sistem, ağ üzerinden dağıtık çalışmaktadır.)
- Bir ağ üzerinden depolanan verinin dağıtımı nasıl olacak? (Birçok modern veritabanı, dağıtık ya da ağ üzerinde çok kopyalı olarak bulunabilmektedir.)
- Geliştirilecek yazılım için uygulama teknolojileri ne olacak? Hangi programlama dili ve araçları kullanılacaktır?
- Satın alınabilen ticari yazılımların bütünleştirilmesi nasıl olacak? (Ticari yazılımın gereksinimlere göre düzenlenme ihtiyacı vardır.)
- Kullanıcı arayüz uygulamasında kullanılacak teknolojiler neler olacaktır?
- Diğer sistemlerle arayüz oluşturmak için kullanılacak teknolojiler neler olacaktır?
- ...

## Mimari Tasarım Bileşenleri

- Yazılım
  - Veri Depolama
  - Veri Erişim Mantığı
  - Uygulama Mantığı
  - Sunum Mantığı
- Donanım
  - İstemci bilgisayarlar
  - Sunucu bilgisayarlar
  - Ağ yapısı

## Sistem Altyapısını Belirleme



## Envanter Belirleme

- Sistem analisti, var olan sistemin alt yapısının durumunu görmek için; teçhizatın
  - modelini ve üreticisini,
  - durumunu (kullanılmıyor, bakıma ihtiyacı var, çalışır durumda vb.),
  - yaşını,
  - planlanan yaşını,
  - işletme içindeki fiziksel yerini,
  - sorumlu çalışanı veya bölümü,
  - finansal durumunu (işletmenin kendi malı, kiralık ve leasing yapılmış vb. şeklinde) belirler.
- Mevcut donanım, gerekli altyapıyla kıyaslanır.

## İş Yüklerinin Tahmini

- Donanım altyapı kapasitesinin yeterliliğini belirlemek için yapılır.
- Zaman-maliyet kıyaslaması yapılır, gereksiz sistem kurulumu engellenmeye çalışılır.

	Mevcut Sistem	Önerilen Sistem
İş	Bayi satışlarının aylık raporu	Bayi satışlarının aylık raporu
Yöntem	Dosya takibi	Bilgisayar tabanlı takip
Personel	Bayi yöneticisi	Bilgisayar uzmanı
Maliyet / Saat	30 TL	15 TL
Ne Zaman? Nasıl?	Günlük : Her bayi için faturaların dosyalanması Aylık : Hesap makinası kullanarak günlük kayıtların özetlenmesi ve rapor hazırlanması	Günlük : Fatura bilgilerinin girilip toplamın alınacağı programın çalıştırılması Aylık : Raporlamayı yapıp yazdıran programın çalıştırılması
Gerekli Olan İnsan Zamanı	Günlük : 20 dakika Aylık : 8 saat	Günlük : 4 dakika Aylık : 20 dakika
Gerekli Olan Bilgisayar Zamanı	Yok	Günlük : 4 dakika Aylık : 20 dakika

## Donanım Değerlendirme

- İş yükü ve mevcut envantere göre proje ihtiyaçları belirlenir.
- İş yüklerinin farklı sistemlerde simülasyonu yapılarak karşılaştırılır.
- Karşılaştırma kriterleri:
  - İşlemin gerçekleşmesi için geçen ortalama süre (Bu süreye giriş verisinin sisteme girilmesi ve çıkışı elde etme de dahildir.)
  - Sistemin toplam kapasitesi (Herhangi bir problem oluşmadan aynı zamanda kaç işlem gerçekleştirildiği.)
  - CPU'nun atıl zamanı
  - Önerilen belleğin büyüklüğü

## Yazılım Değerlendirme

Yazılım Gereksinimleri	Yazılım Özellikleri	Yazılım Gereksinimleri	Yazılım Özellikleri
Performans Etkinliği	<ul style="list-style-type: none"><li>▪ İhtiyaç duyulan tüm işleri yerine getirme</li><li>▪ Tüm işleri istenildiği şekilde yerine getirme</li><li>▪ İyi tasarlanmış ekran çıktıları</li><li>▪ Yeterli kapasite</li></ul>	Esneklik	<ul style="list-style-type: none"><li>▪ Girdi seçeneği</li><li>▪ Çıktı seçeneği</li><li>▪ Diğer yazılımlarla birlikte kullanılma</li></ul>
Performans Verimliliği	<ul style="list-style-type: none"><li>▪ Hızlı cevap zamanı</li><li>▪ Verimli giriş ve çıkış</li><li>▪ Verimli veri saklama</li><li>▪ Verimli yedekleme</li></ul>	Dökümantasyon Kalitesi	<ul style="list-style-type: none"><li>▪ İyi organizasyon</li><li>▪ Yeterli çevrim içi kaynak</li><li>▪ Web sitesinde sık sorulan soruların(FAQ) bulunması</li></ul>
Kullanım Kolaylığı	<ul style="list-style-type: none"><li>▪ İyi tasarlanmış kullanıcı arabirimi</li><li>▪ Yardım menüleri</li><li>▪ Esnek arabirim</li><li>▪ Yeterli geri besleme</li><li>▪ Hata düzeltme</li></ul>	Üretici Desteği	<ul style="list-style-type: none"><li>▪ Teknik destek</li><li>▪ Ürün güncellemesinin Web sitesi üzerinden yapılabilmesi</li></ul>

## Satıcı Firmayı Belirleme

Temel Satıcı Hizmetleri	Satıcının Sunduğu Hizmetler
Donanım Desteği	<ul style="list-style-type: none"><li>Ürün kalitesi</li><li>Garanti</li></ul>
Yazılım Desteği	<ul style="list-style-type: none"><li>Toplam yazılım ihtiyacı</li><li>Özel programlama gereksinimi</li><li>Garanti</li></ul>
Kuruluş ve Eğitim	<ul style="list-style-type: none"><li>Zaman programına uyma</li><li>Eğitim</li><li>Müşteri hizmetleri</li></ul>
Bakım	<ul style="list-style-type: none"><li>Rutin bakım işlemleri</li><li>Acil durumdaki cevap verme süresi</li></ul>

## Satın Alma Şeklini Belirleme

	Avantajları	Dezavantajları
Satın Alma	<ul style="list-style-type: none"><li>Uzun dönemde leasing ve kiralamadan ucuz olması</li><li>Sistemin değiştirilebilmesi</li><li>Vergi indirimi sağlanması</li><li>Tam kontrol</li></ul>	<ul style="list-style-type: none"><li>İlk ödemenin yüksek olması</li><li>Eskime riski</li><li>Seçimin yanlış yapılması sonucu yatırımın batma riski</li><li>Tam sorumluluk</li></ul>
Leasing	<ul style="list-style-type: none"><li>Sermayenin bağlanmaması</li><li>Finanslamaya ihtiyaç olmaması</li><li>Kira bedelinden düşük olması</li></ul>	<ul style="list-style-type: none"><li>Sözleşme bittiğinde işletmenin sisteme sahip olamaması</li><li>Ödemenin gecikmesi durumunda yüksek faiz oranı</li><li>Satın almadan daha pahalı olması</li></ul>
Kiralama	<ul style="list-style-type: none"><li>Sermayenin bağlanmaması</li><li>Finanslamaya ihtiyaç duyulmaması</li><li>Sistemin kolayca değiştirilmesi</li><li>Genelde bakım ve garantinin fiyat kapsamında olması</li></ul>	<ul style="list-style-type: none"><li>İşletmenin sistemin gerçek sahibi olmaması</li><li>Satıcı kendi riskini üstlendiği için kira bedelinin çok yüksek olması</li></ul>



## Ayrıntılı Tasarım

- Çıktı Tasarımı
- Girdi Tasarımı
- Arabirim Tasarımı
- Uygulama (Program) Tasarımı
- Veritabanı Tasarımı

## Çıktı Tasarımı

- Çıktı: Sistemin Kullanıcılara verdiği bilgi, üretilen raporlar
- Amaçlar:
  - Belirlenen amaca hizmet etme
  - Kullanıcı için anlamlı olma
  - Uygun sayıda olma
  - Hangi kullanıcılara dağıtılacağına doğru belirlenmesi
  - Zamanında sağlanma (günlük, aylık, yıllık veya koşula bağlı raporlar)
  - Doğru çıktı yönteminin (ortamının) seçilmesi
- Kullanıcıyı etkileyecek yönlendirme:
  - Bilgilerin belirli kriterlere göre sıralanması
  - Sınırların Belirlenmesi
  - Grafik tipi – rengi – ölçeğinin belirlenmesi

## Girdi Tasarımı

- Kaliteli çıktı için girdi kalitesi önemli
- Amaçlar
  - Etkinlik: Form ve ekran görüntülerinin belli bir amacının olması
  - Doğruluk: Analizde tanımlanan tüm işlemleri yerine getirmesi
  - Kullanım kolaylığı: Bilgi girişi kullanıcılarının fazla zamanını almaması ve ergonomik olması
  - Uyumluluk: Bir formda diğerine ya da ekran görüntüsüne geçişte düzenin değişmemesi
  - Basitlik: Gereksiz ayrıntıya yer verilmemesi ve karmaşık olmaması
  - Çekicilik: Ekran ve form yapılarının güzel görünmesi

## Girdi Tasarımı Prensipleri

- Çevrimiçi işleme ve toplu işleme girdileri uygun olarak kullanılır
- Veri, kaynağında tutulur
- Klavye tuşlamaları azaltılır
- Doğrulama ve geçerleme yapılır
  - Tamlık, Biçim, Aralık, Tutarlılık

## Girdi Tasarımı - Ekranlar

- **Kolay kullanım ve basitlik:** Gereksiz bilgi bulunmaması, pencereler içinde girilecek bilgilerle ilgili açıklamalar olması
- **Uyumluluk:** Bilgi toplama formları ve diğer ekran görünümüleri arası
- **Hareket kolaylığı:** Ekrandan ekrana geçme ve başlık kolonunu sabit tutarak diğer kolonları kaydırma vs.
- **Çekici ekran tasarımı:** Tüm ekranların belli bir düzene uygun hazırlanması ve imleç yapısı, font tipinin seçimi

## Kullanıcı Arabirimi Tasarımı

- Arabirim: Sistemin kullanıcıyla iletişime giren elemanı
- Tipleri:
  - Doğal dil arabirimleri
  - Soru-cevap sorgulamalar
  - Menüler
  - Girdi-çıkı formları
  - Komut dili
  - Sistem bildirimleri
- Amaç:
  - Etkinlik: Kullanıcıların gereksinimlerine uygun olarak sisteme erişmelerini sağlama
  - Verimlilik: Hataları azaltma, veri giriş hızını arttırma
  - Kullanıcıların görüşlerinin alınabilmesi
  - Ergonomik olması

## Kullanıcı Arabirimi Tasarımı

- Yerleşim planı: Ekran iç ana alana bölünür.
  - En üst alan, sistemde gezinim sağlar
  - Orta alan kullanıcı çalışmalarına ayrılır
  - En alt alan, yapılanlarla ilgili durum bilgisi verir

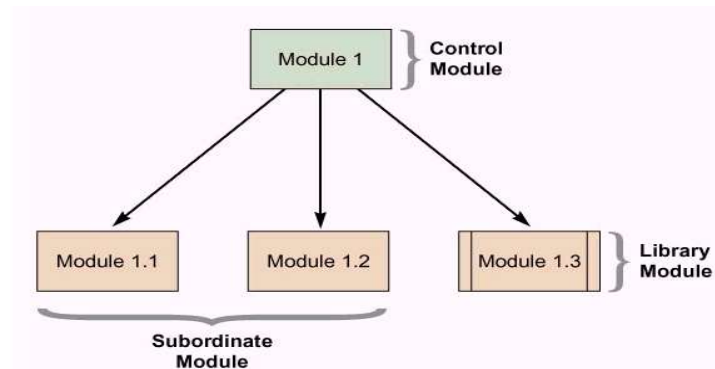
## Arabirim Tasarım Problemleri

- Bilginin düzensiz görünümde yığınlar şeklinde olması
- Kullanıcının yürütmesi gereken çeşitli görevler arasındaki geçişin uyumlu olmaması
- Ekran üzerindeki komutlarda kullanılan terminolojinin karmaşık olması
- Sistem tarafından verilen hata mesajlarının açıklayıcı olmaması
- El kitaplarının anlaşılmaz derecede karmaşık olması
- ...

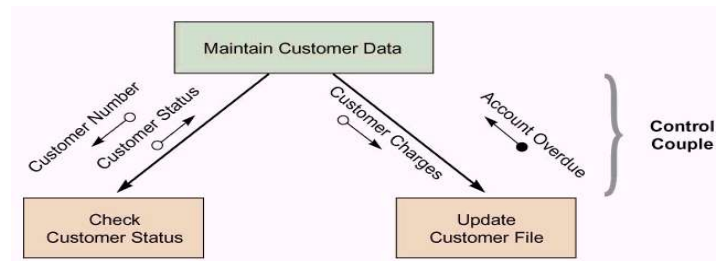
## Sistem(Uygulama) Mimarisini Belirleme

- Modüler program yapısı geliştirilir.
- Aralarındaki ilişkilerin denetimi belirlenir.
- Veri yapısı ve program birleştirilir, ara birimler tanımlanır.
- VAD ve veri sözlüğüne dayanan yapı diyagramı ile gösterilir.
  - Modüller: Anlaşılması ve bakımı kolay olan, mantıksal program işlem birimleri
  - Veri iletişimi: Modüller arası iletilen veri
  - Kontrol mesajları: Modüller arasında bir durumu ya da hareketi başka modüle aktarma mesajı (örneğin: dosya sonu)
  - Durumlar: Kontrol Modülünün hangi alt modülü çağıracağını gösterimi
  - Döngüler: Bir veya daha fazla tekrar eden alt modül işleminin gösterimi

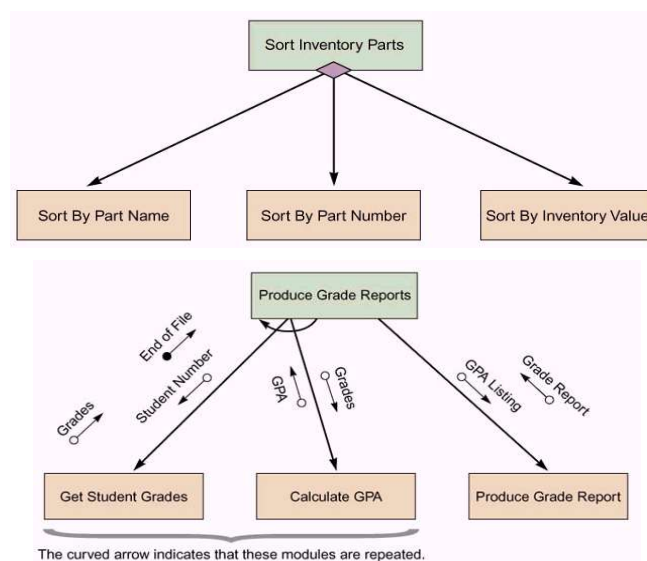
## Ana Modül – Alt Modüller



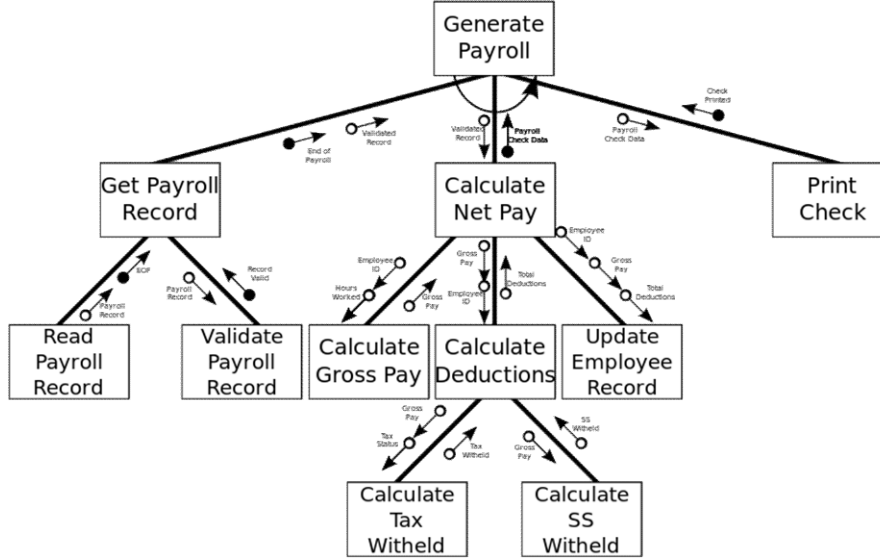
## Veri İletişimi – Kontrol Mesajları



## Durumlar - Döngüler



## Yapı Diyagramı



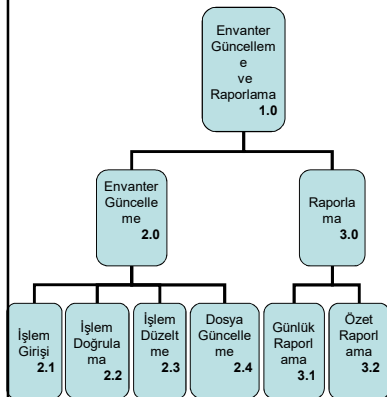
## Tasarım Spesifikasyonları

- Ön tasarım spesifikasyonları: Yazılım sisteminin genel özellikleri ve ilişkileri
- Mimari tasarım spesifikasyonları: Sistemin yapısı ve kuruluşu
- Ayrıntılı tasarım spesifikasyonları: Modüller içerisindeki kontrol akışı, veri gösterimi ve diğer algoritmik ayrıntılar

# Hierarchy & Input-Process-Output

- HIPO
- Sistemdeki her fonksiyonun birbiriyle olan ilişkisini belirler.
- Temel fonksiyonlar belirlendikten sonra alt fonksiyonlar tanımlanarak fonksiyon hiyerarşisi oluşturulur.
- Diyagramlar:
  - Görsel içerik tablosu (Visual Table of Contents, VTOC)
  - Genel IPO ( Input Process Output ) Diyagramı
  - Ayrıntılı IPO Diyagramı

## Görsel İçerik Tablosu



### 1.0 Envanter Güncelleme ve Raporlama

Diğer program modüllerin bir arada çalışma koordinasyonunu sağlar.

### 2.0 Envanter Güncelleme

Tüm işlemleri yürütür ve envanter dosyasını günceller.

#### 2.1 İşlem Girişi

Eklenen, satılan, bozulan veya kaybolan tüm işlemlerin girişini sağlar.

#### 2.2 İşlem Doğrulama

Yapılan işlemin tipini ve bölüm numarasını doğrular.

#### 2.3 İşlem Düzeltme

Yapılan işlemlerin tipini ve bölüm numarasını düzeltir.

#### 2.4 Dosya Güncelleme

Envanter dosyasındaki nesneleri ve birimleri silerek veya düzelterek değiştirir.

### 3.0 Raporlama

Günlük ve özet rapor modüllerini çalıştırır.

#### 3.1 Günlük Raporlama

Envanterin günlük raporunu oluşturup yazdırır.

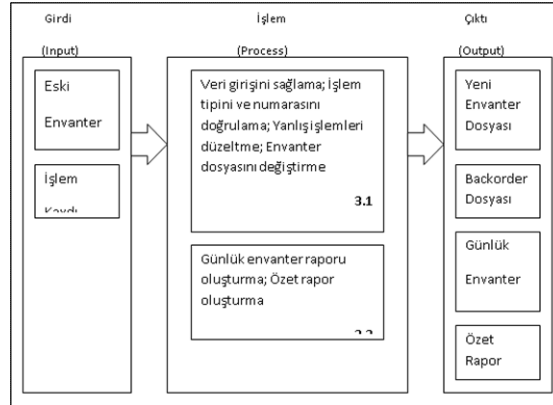
#### 3.2 Özet Raporlama

Özet raporu oluşturup yazdırır.

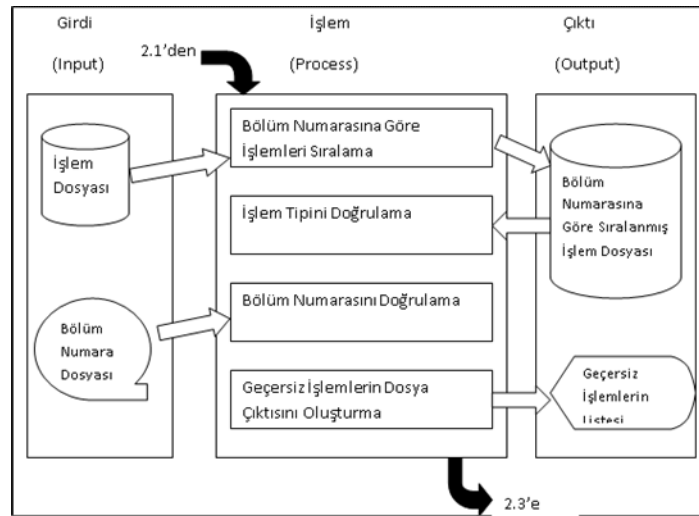


## Genel IPO

- VTOC diyagramındaki her fonksiyon için IPO detaylı olarak tanımlanır.



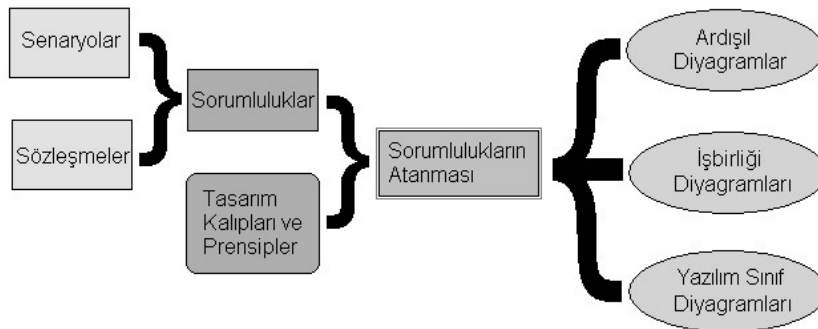
## Ayrıntılı IPO



## Nesneye Dayalı Tasarım Modelinin Oluřturulması

- Problemin mantıksal çözümü oluşturulur.
- YAZILIM SINIFLARI ve aralarındaki İŐBİRLİŐİ (etkileřim) belirlenir.
  - Tasarım Sınıf Diyagramı
  - Etkileřim Diyagramları
- Etkileřim: sınıfların davranıřlarının belirlenmesi → sorumlulukların atanması

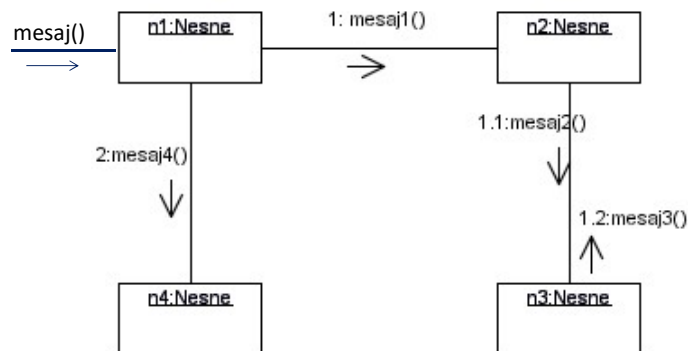
## Nasıl Yapılır?



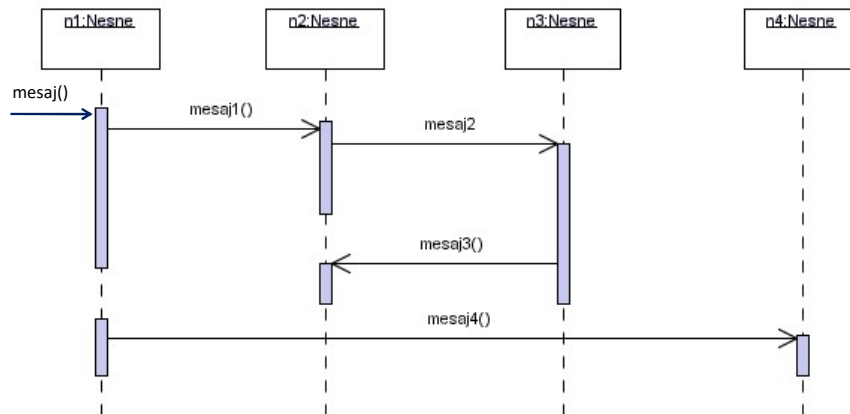
## Etkileşim Diyagramları

- İşbirliği Diyagramları
  - Az yer kaplar
  - Dallanma paralellik ve iterasyonlar kolay gösterilir
  - Mesaj sırasını anlamak zor
- Ardışıl Diyagramlar
  - Fazla yer kaplar
  - Mesaj sırasını anlamak kolaydır

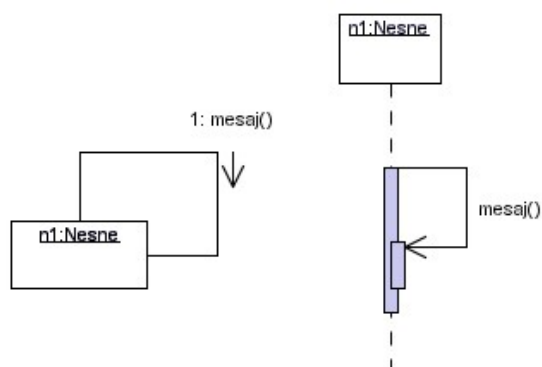
## Mesaj Sıra Numaraları



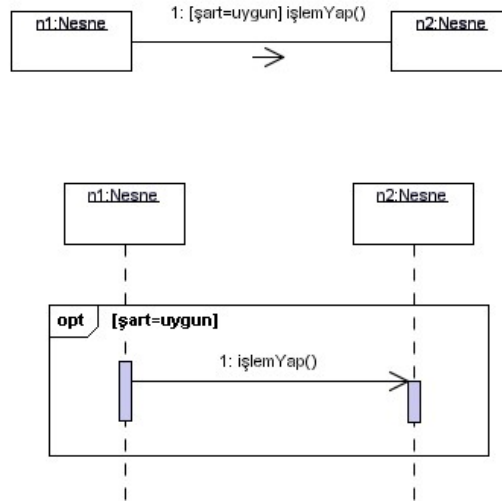
## Mesaj Sıra Numaraları



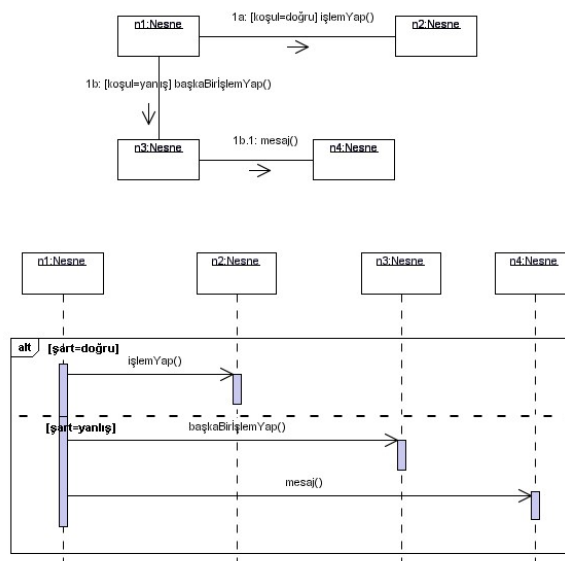
## Kendine mesaj



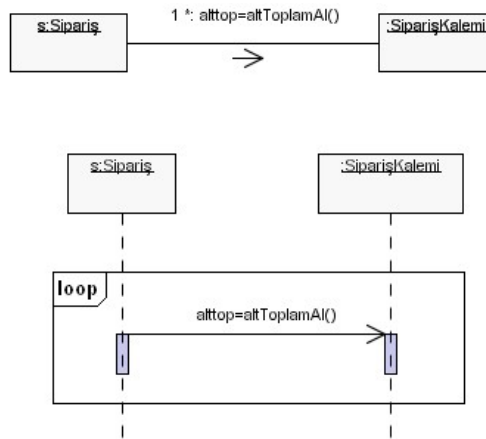
## Koşullu mesajlar



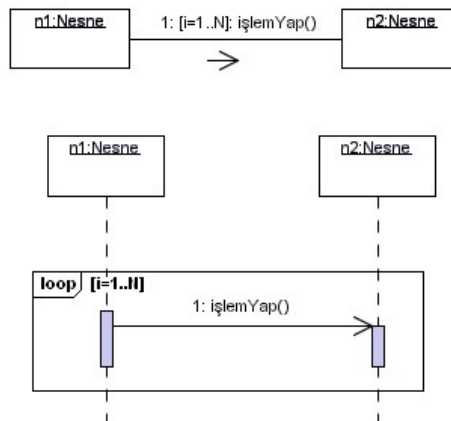
## Karşılıklı dışlamalı mesajlar



## Döngüler



## Döngüler



## Kullanım senaryolarının gerçekleşmesi

- Senaryodaki durumların modellenerek gerçekleşmesi
- Tasarım: yazılım sınıflarına metotların eklenmesi ve istekleri yerine getirmek üzere nesneler arası mesajların belirlenmesi.
- Sorumluluklar:
  - Bilinmesi gerekenler
    - Kendi özel verileri
    - İlgili diğer nesneler
    - Üzerinde hesap yapabileceği, hesapla elde edebileceği veriler
  - Yapılması gerekenler
    - Hesap yapma, nesne yaratma yok etme
    - Başka nesneleri harekete geçirme
    - Başka nesnelerin hareketlerini denetleme

## Senaryoların gerçekleşmesi

- Sorumlulukları yerine getirmek için metotlar oluşturulur
- Bir sorumluluğu yerine getirmek için bir metot başka metotlarla işbirliği yapabilir
- İlk iterasyonda senaryoların ana akışları gerçekleştirilir
- İkinci iterasyonda alternatif akışlar ele alınır ve gerçekleşir.
- Büyük senaryolar birkaç iterasyon sürebilir. Daha küçükleri bir iterasyonda bitebilir.
- Tasarımın sonunda tersine gidilerek, problem domeninin diyagramı çıkarılır. Bu sayede analiz diyagramlarının son hali de elde edilmiş olur.

## Örnek senaryo – Ana Akış

**Satışın toplam bedelinin hesaplanması:**

**Main Success Scenario (or Basic Flow):**

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules. *Cashier repeats steps 3-4 until indicates done.*
5. System presents **total with taxes calculated.**

Senaryoya göre satışın toplam bedelinin hesaplanması gerekiyor.

## “Yabancılarla Konuşma” Prensipleri

- Don't Talk To Strangers.
- Bir nesne ancak tanıdık (sınırlı sayıda) bir hedefe mesaj göndermelidir.
  - Kendisi (this)
  - Metodun parametresi olan nesne
  - Nesnenin üyesi (özellikleri) olan nesne
  - Nesnenin üyesi olan bir grubun (liste, vektör vs) elemanı olan nesne
  - Metodun içinde yaratılan nesne
- Dolaylı (tanıdığın tanıdığı) nesneler ise yabancı nesnelerdir.
- Metod içinden yabancı nesnelere mesaj gönderilmesi, bağımlılık yaratır. Görülmesi zor, olmaması gereken bir bağımlılıktır. Bu yüzden tercih edilmez.
- Örnek: Money amount= sale.getPayment().getTenderedAmount(); kötü
- Money amount= sale.getTenderedAmountOfPayment(); iyi
- Gerçekten gerekliyse, tanıdık nesneye sorumluluk olarak atamak gerekir.



## Senaryoların gereklenmesi

- Anlatılan kalıplara uygun olarak
  - Belirlenen tm senaryolar
  - Tm szleřmeler

Gerekleřtirilir.

- Kavramsal sınıflardan yola ıkılarak yazılım sınıfları oluřturulur.
- Sorumluluklar, metotlarla gerekleřtirilir.
- Her bir iřbirlięi, tasarım sınıf diyagramında bir baęlantı olarak gsterilir.

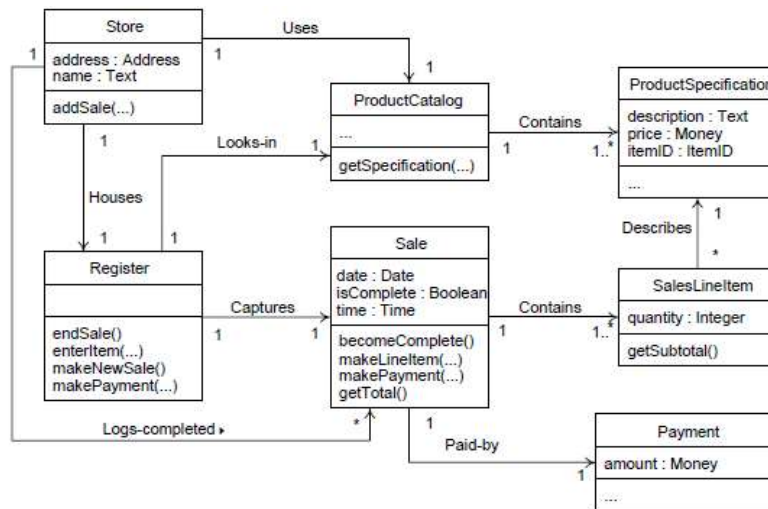
## Başlangı İřlemleri

- Sistem ilk alıřmaya bařladıęında yapılacaklar ayrı bir senaryo grubu olarak yazılabilir.
- Bu iřleri tasarımın en son ařamasında belirlemek uygundur.
- Bir başlangı nesnesi (initial domain object) belirlenir.
- Program alıřmaya bařlayınca bu nesne yaratılır.
- Doęrudan ierdięi dięer nesneleri yaratma ve aralarındaki baęlantıyı saęlama sorumluluęuna sahiptir.
- Tm programın alıřmasını denetleyen temel bir nesne gibidir. Ancak bu denetim ana programda veya arayz nesnesinde de olabilir.
- Başlangıta arayz nesnelere denetinin referansı da gnderilir.

## Tasarım Sınıf Diyagramı

- Senaryoları gerçeklerken çizilen etkileşim diyagramlarına paralel olarak, yazılım sınıflarından oluşan diyagram(lar) da çizilir.
- Yazılım sınıfları, özellikleri ve tipleri, metotların parametreleri ve erişim hakları büyük ölçüde belirtilir.
- İlişkiler ve bağımlılıklar yönlü olarak gösterilir. (Yeni ilişkiler de bulunabilir)

## Örnek



## Gelecek Ders

- Veri Modelleme
- Veri Yapısı ve Veri Tabanı Tasarımı