

## NESNEYE YÖNELİK ÇÖZÜMLEME SÜRECİ

### NESNEYE YÖNELİK ÇÖZÜMLEMENİN TEMELLERİ

- Çözümleme (Analiz): Bir şeyi anlayabilmek için parçalarına ayırmak.
- Sistemi anlamaya yönelik çalışmalardan ve üst düzey planlama eylemlerinden oluşur.
  - Uygulama/problem alanının anlaşılması.
  - Kullanıcı gereksinimlerinin anlaşılması.
  - Koddaki sınıflar ve nesneler ile bunların arasındaki üst düzey etkileşimlerin belirlenmesi: Çözümleme modelinin oluşturulması.
- "Bir sorunu anlamadan çözemezsiniz."

### UYGULAMA ALANININ ÇÖZÜMLENMESİ (DOMAIN ANALYSIS)

- Amaç, uygulama alanını anlamak ve elde edilen bilgileri analiz modeline taşımaktır.
- Uygulama alanı hakkında bilgi edinilebilecek kaynaklar:
  - Teknik literatür
  - Mevcut uygulamalar
  - Müşteri anketleri
  - Uzman tavsiyeleri
  - Mevcut ve gelecekteki gereksinimler
- Problem alanı hakkında bilgi edinmeden "müşterinin dilinden konuşamazsınız".

## GEREKSİNİMLERİN BELİRLENMESİ

- Gereksinimler belgesi:
  - Müşterinin programdan beklentilerini anlatan, doğal konuşma dili ile yazılmış belge.
- Örnek gereksinimler belgesi:

### **NextGenPOS Perakende Satış Programı**

Eski yazılım ihtiyaçlarımızı karşılayamadığından, yenilenecek donanımla birlikte perakende satış programımızın da yenilenmesine gerek duyuyoruz. Program kasada yapılan alış-veriş işlemlerine yardımcı olmalıdır. Yapılan her işlem program tarafından saklanmalı; mali bilgiler harici bütçe sistemine, mal çıkış bilgileri ise harici envanter sistemine iletilmelidir. Saklanan işlemler üzerinde daha sonra raporlamalar ve analizler yapılabilmelidir. Sistem yapılan alış-verişler karşılığında müşteriye fiş vermelidir. Yapılan her satış için KDV de hesaplanarak ayrıca belirtilmelidir. Şirketimizin birden fazla şubesi olup tüm şubelerdeki işlemler merkezi sunucuya iletilmelidir.

- Doğal dille yazılmış gereksinimler belgesinden kullanım öykülerine geçiş yapılır.

## GEREKSİNİMLERİN BELİRLENMESİ

- Kullanım öyküleri:
  - Programın yapacağı işleri ayrıntılı adımlarla ve belli kurallara uyarak anlatan belgeler.
- Kullanım öykülerinin oluşturulmasındaki amaç:
  - Ürünün sağlaması beklenen işlevleri ve ürünün çalışma ortamını belirlemek,
  - Son kullanıcı ve yazılım ekibi arasında bir anlaşma zemini belirlemek,
  - Son kullanıcı ve sistemin birbirleri ile nasıl etkileşimde bulunacağını açık ve belirsizlikten uzak olarak tanımlamak,
  - Doğrulama testleri için bir zemin oluşturmak.
- Bir kullanım öyküsünün bölümleri:
  - Giriş bölümü: Sistemin neyi hangi koşullar ve sınırlar içerisinde yapması gerektiğini anlatır.
  - Ana senaryo / Ana başarılı akış: Her şeyin yolunda gitmesi halinde yürütülecek eylemler.
  - Alternatif senaryolar: Bir aksilik olması halinde yapılacak işlemler.

## KULLANIM ÖYKÜSÜ: Satış İşlemi

**Birincil Aktör:** Kasiyer.

**İlgililer ve İlgili Alanları:**

- Kasiyer: Doğru ve hızlı giriş ister, kasa açığı maaşından kesildiğinden ödeme hataları istemez
- Satıcı: Satış komisyonlarının güncellenmesini ister
- Müşteri: En az çaba ile hızlı hizmet ister. Ürün iadesinde kullanmak üzere fiş ister.
- ...

**Ön Koşullar:**

- Kasiyerin kimliği doğrulanır.

**Son Koşullar:**

- Ödeme tahsil edilir. Satış kaydedilir. Fiş yazılır.

- Dikkat: Kullanım öyküsünde yer alacak her şey, verilen ilgi alanlarına giren şeyler olmalıdır.
- Aktör: Sistem ile etkileşimde bulunan varlıklar.
  - İnsan
  - Yazılım veya donanım.

## KULLANIM ÖYKÜSÜ: Satış İşlemi

### Ana Öykü:

1. Müşteri kasaya alacağı ürünlerle gelir.
2. Kasiyer yeni bir satış işlemi başlatır.
3. Kasiyer ürünün barkodunu girer.
4. Sistem bir satış kanalı maddesi oluşturur. Bu maddede ürün tanımı, fiyatı ve toplam bedel (aynı maldan birden fazla alınmış olabilir) yer alır.
5. Kasiyer 3. ve 4. adımları müşterinin alacağı tüm ürünler için tekrarlar.
6. Sistem toplam bedeli KDV ile birlikte hesaplar.
7. Kasiyer müşteriye toplamı bildirir ve ödeme ister.
8. Müşteri ödemeyi yapar ve sistem ödemeyi tahsil eder.
9. Sistem tamamlanan işlemin kaydını tutmayı tamamlar ve harici envanter ile mali sistemlere gerekli bilgileri gönderir.
10. Sistem fiş verir.
11. Müşteri ürünlerle birlikte ayrılır.

## KULLANIM ÖYKÜSÜ: Satış İşlemi

### Alternatif Öyküler:

#### 3a. Geçersiz barkod

1. Sistem uyarı mesajı verir ve kayıt girişini reddeder.

#### 3-7a. Müşteri bir kalem malı alışverişten çıkartmak ister.

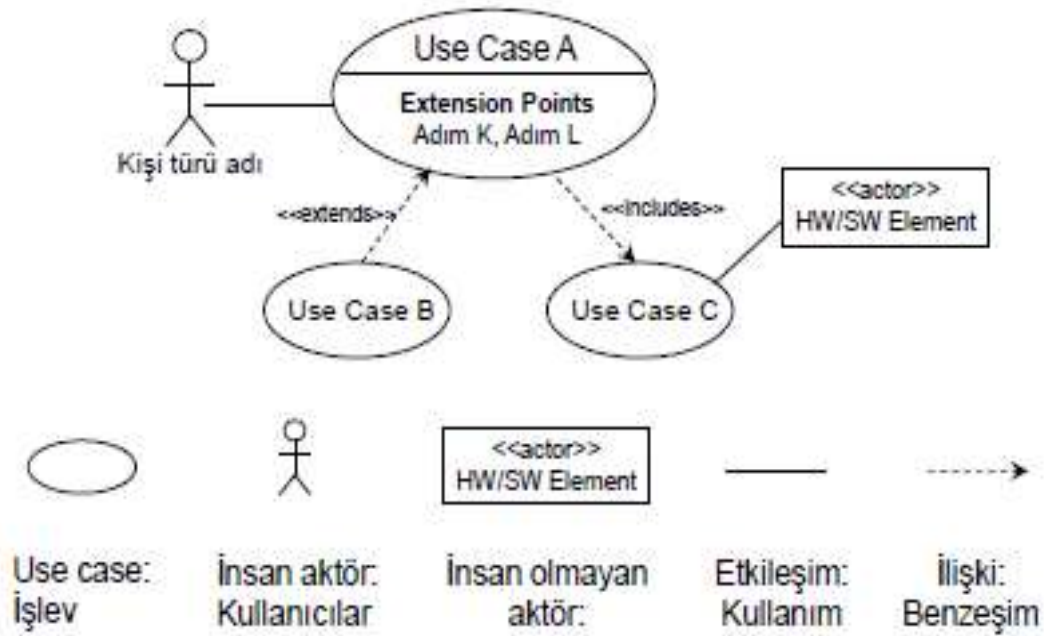
1. Kasiyer satıştan çıkarmak üzere ürünün barkodunu okutur.
2. Sistem güncel toplamı bildirir.

...

## KULLANIM ÖYKÜLERİNİN GRAFİK GÖSTERİMİ

- Kullanım öyküleri, ayrıntılı ve uzun belgelerdir.
- Yazılımın yapacağı işlerin özet gösterimi için kullanım şemaları çizilir (use-case diagrams).
- Çizim kurallarını verdikten sonra örnek öykünün şemasını çiz.

## ÇİZİM KURALLARI

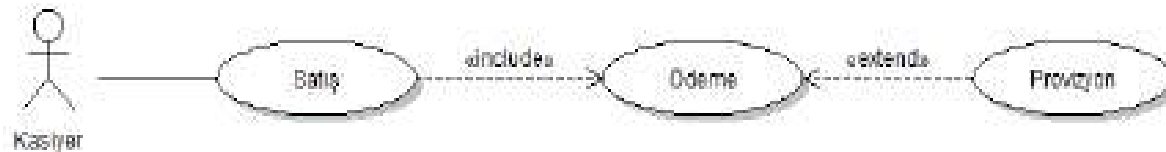




## ÇİZİM BİLGİLERİ

- Benzeşim ilişkileri:
  - Ok yönü aynı zamanda ilişkiyi okuma yönüdür.
  - UC-B extends UC-A : B işlevi, A işlevi yürütülürken oluşabilecek bir sapış anlamındadır.
    - A: Ana akış
    - B: Ana akıştaki bir seçenek, ana akıştan bir sapış, alt akış
  - UC-A includes UC-C: A işlevi, C işlevini içerir.
    - A : Ana akış, içeren akış
    - C: Alt akış, içerilen akış

## ÖRNEK ÇİZİM



- Bir POS yazılımının ödeme işlevini kasiyer kullanır.
- Satış işlevi, içerisinde ödeme yapma işlevini içerir.
  - Includes, çünkü: Her satış içerisinde mutlaka ödeme olur.
- Ödemenin kredi kartı ile olması halinde, provizyon alma işlemi yürütülür.
  - Extends, çünkü: Ödeme nakit ise provizyona gerek kalmaz.
  - Provizyon: Kredi kartının limitinin aşılp aşılmadığı, çalıntı olup olmadığı, vb. gibi bilgilerin sınanması anlamında bir bankacılık terimi.

## SINIFLARIN BELİRLENMESİ

- Kullanıcı gereksinimleri belgesinden ve kullanım senaryolarından sınıfların elde edilmesi.
  - İsimlerin taranarak aday sınıfların elde edilmesi.
  - Adaylar aşağıdaki kurallardan birini sağlamalıdır:
    1. Saklanan bilgi: Sistemin çalışması süresince bu varlığın durumu saklanmalıdır.
    2. Gereksinim duyulan hizmetler: Bu varlığın hizmetlerine ihtiyaç duyan başka varlıklar vardır.
    3. Gerekli varlıklar: Problemin çözümü ile ilgili bilgi üreten veya problemin çözümü için bilgi tüketen varlıklar.
  - Değinilen kurallardan birini sağlayamayan adayları, bir başka sınıfın üye alanı olarak değerlendirebiliriz.
- Örnek gereksinim belgesinden sınıfları oluştur.

## SINIFLARIN BELİRLENMESİ

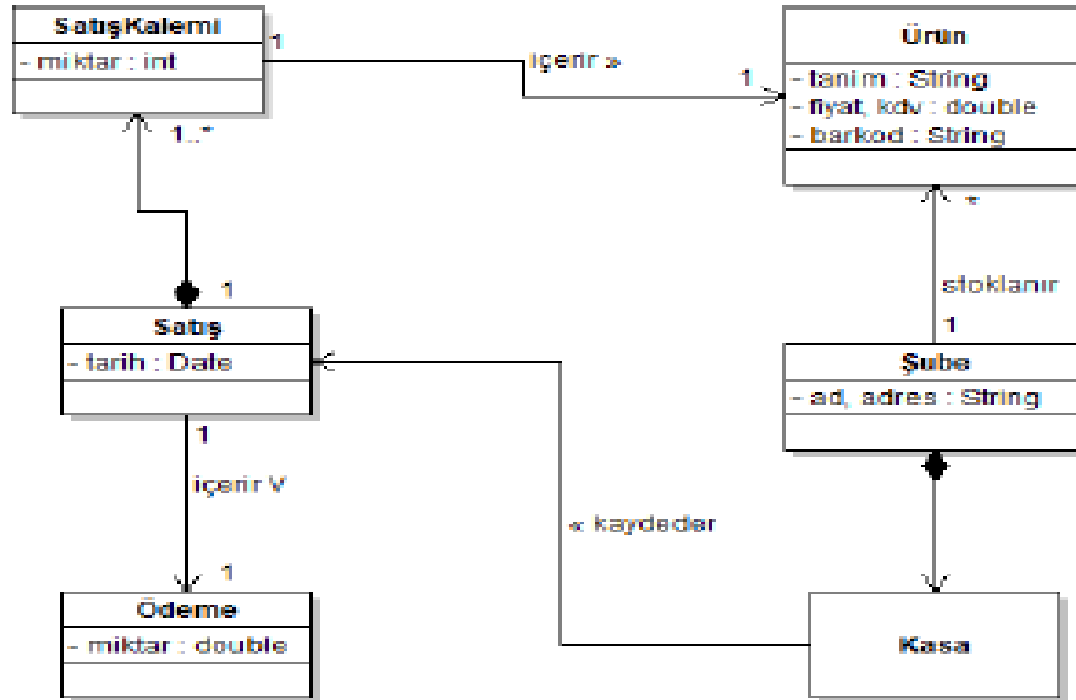
- Üyelerin belirlenmesi:
  - Sıfat ve eylemlerin taranması
  - Sorumlulukların belirlenmesi (CRC kartları)
- Sorumlulukların dağıtılması:
  - Sorumlulukların bir yerde yoğunlaşmaması
  - Sorumlulukların genelden özele doğru tanımlanması (kalıtım hiyerarşisinde genelden özele gidilmesi)
  - Bir bilgi ile ilgili davranışların, o bilgi ile aynı sınıfta yer alması (encapsulation)
  - Tek bir şey hakkındaki bilginin tek sınıfta yer alması
  - Gerekli sorumlulukların paylaşılması

## ETKİLEŞİMLERİN BELİRLENMESİ

- Etkileşim: Bir nesnenin üzerine düşen sorumluluğu yerine getirmek için diğer bir nesneye mesaj göndermesi.
- Nesneler arasındaki ilişkiler
  - Bağlantı, toplama, meydana gelme.
- Sınıflar arasındaki ilişkiler
  - Özelleşme/genelleşme
- Çözümleme aşamasında ne tür etkileşimlerin olabileceği düşünülür, etkileşimlerin nasıl olacağı düşünülmez.
- Bu konuların temeli "Nesneye Dayalı Kavramlar" dersinde atılmıştır.

## TEKNEKLE İÇEREN YÖNTEMLERİ ÖRNEK:

### ÖRNEK ALAN MODELİ



## ÇÖZÜMLEME SÜRECİNİN BELGELENDİRİLMESİ

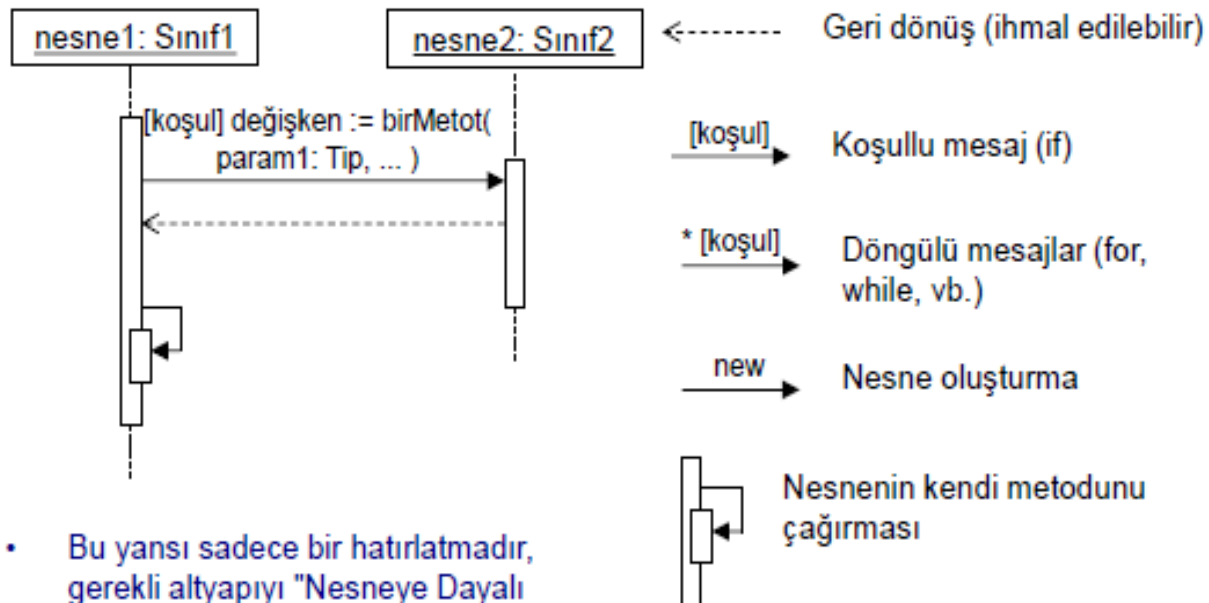
- Bir nesneye yönelik programın çözümleme sürecinin belgelendirilmesinde yer alan önemli belgeler:
  - UML Kullanım şemaları,
  - Kullanım senaryoları,
  - UML sınıf şemaları,
- Veritabanı işlemleri yapılacaksa bunlara ek olarak:
  - E-R diyagramı

# Nesne Yönelimli Tasarım



## SIRALAMA ŞEMALARI AYRINTILARI

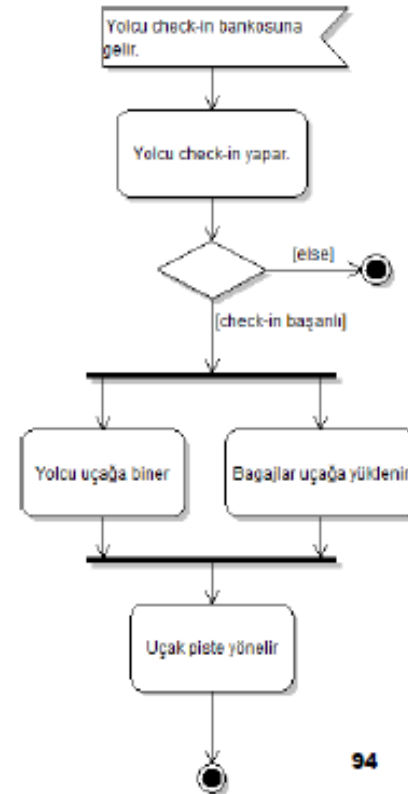
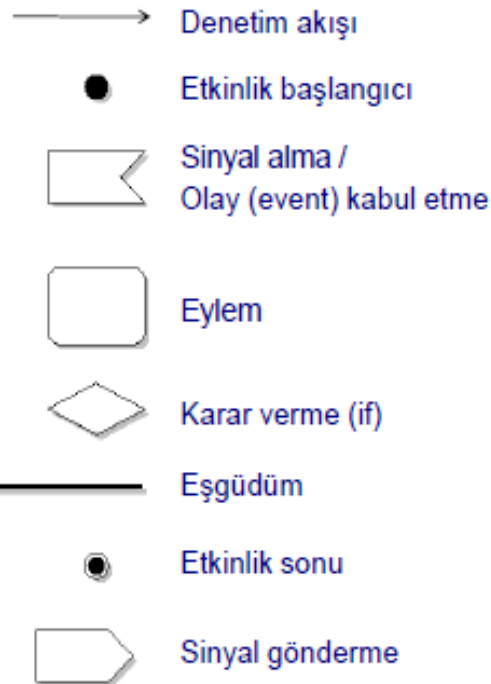
### SIRALAMA ŞEMALARI



- Bu yansı sadece bir hatırlatmadır, gerekli altyapıyı "Nesneye Dayalı Kavramlar" dersinde kazanmıştınız.

## ETKİNLİK ŞEMALARI – ACTIVITY DIAGRAMS

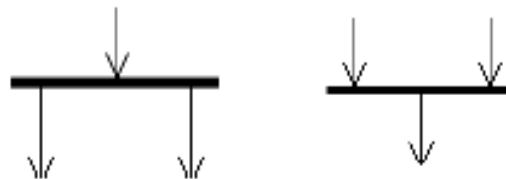
### ÇİZİM KURALLARI ve ÖRNEK ÇİZİM



## ETKİNLİK ŞEMALARI – ACTIVITY DIAGRAMS

### ÇİZİM BİLGİLERİ

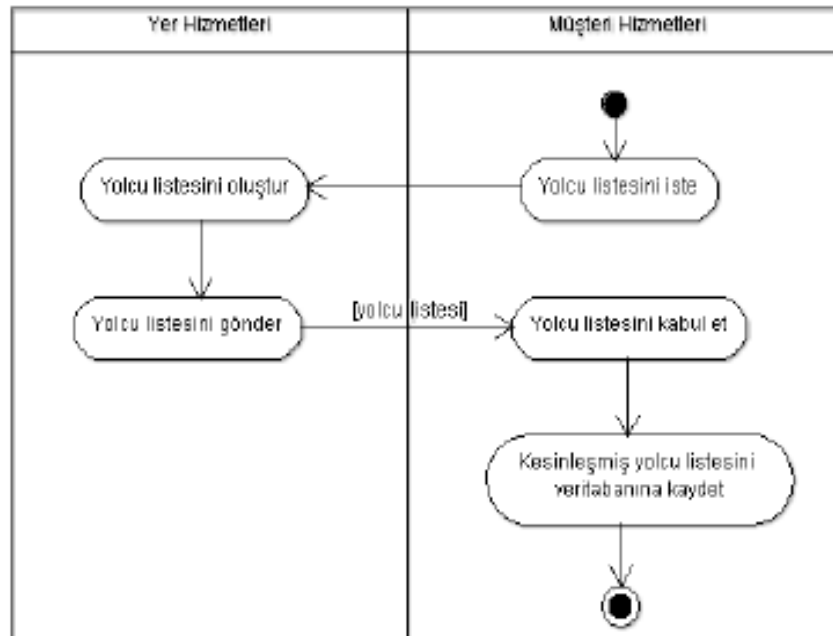
- Etkinlik şemaları başlangıç işareti veya sinyal alma işareti ile başlar.
- Sinyal alma: Beklemelidir.
  - Akış, bir sinyal alana kadar bekler.
  - Zamanlı olaylar da (timer) bununla gösterilebilir.
- Eşgüdüm: Beklemelidir.
  - Eşgüdüm çizgisine varan akış, çizgiyi geçmeden önce diğer akışların hepsini bekler.
    - fork
    - join



## ETKİNLİK ŞEMALARI – ACTIVITY DIAGRAMS

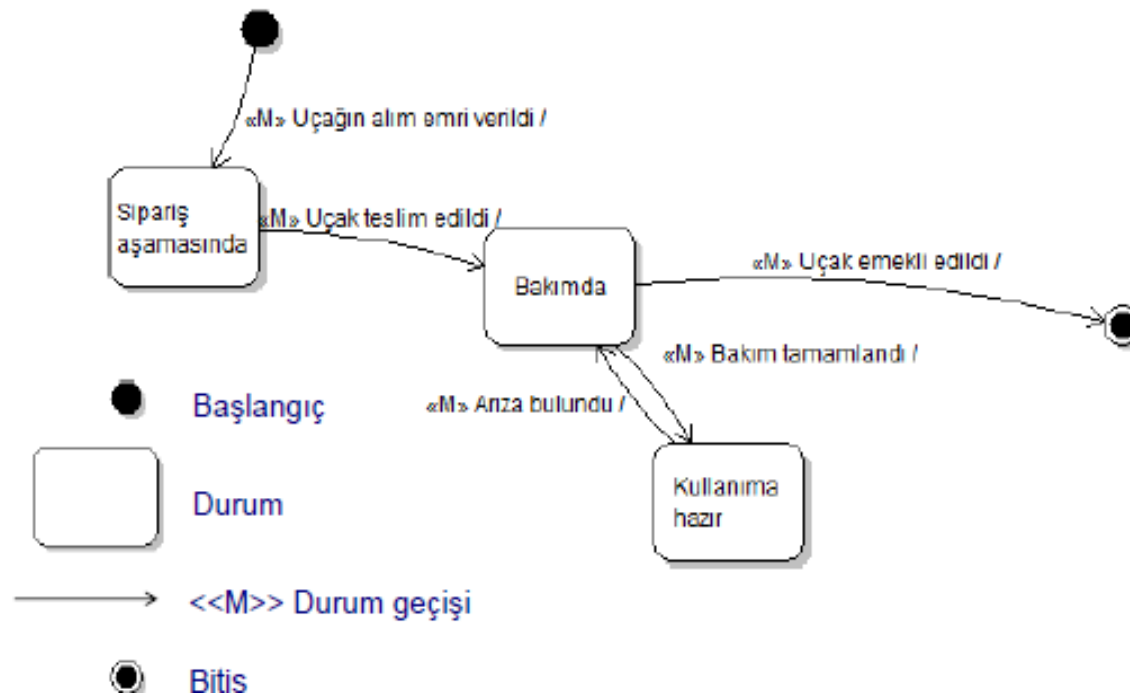
### ÖRNEK ÇİZİM

- Birden fazla aktörün ve aktörler arası bilgi akışının gösterilmesi:



## DURUM ŞEMALARI – STATE DIAGRAMS

### ÇİZİM KURALLARI ve ÖRNEK ÇİZİM



## DURUM ŞEMALARI – STATE DIAGRAMS

### ÖRNEK ÇİZİM (2)

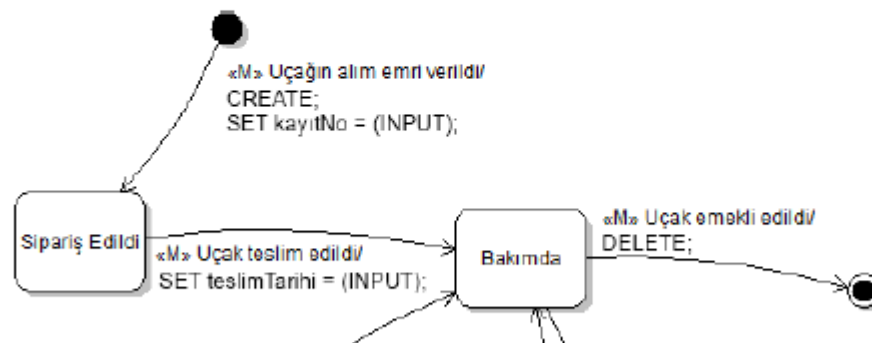


- Yorumlama: Kullanıma hazır bir uçak için bakım isteği gelmişse, uçak önce bakım sırasına alınır. Bu sırada önceden planlanmış uçuşları varsa onları yapar. Planlanan uçuşlar bitince uçak bakıma alınır.

## DURUM ŞEMALARI – STATE DIAGRAMS

### ÖRNEK ÇİZİM (3)

- Durum geçişi sırasında işlenen komutların şemada gösterilmesi:



# TASARIM MODELİ

## NextGenPOS Tasarım Modeli

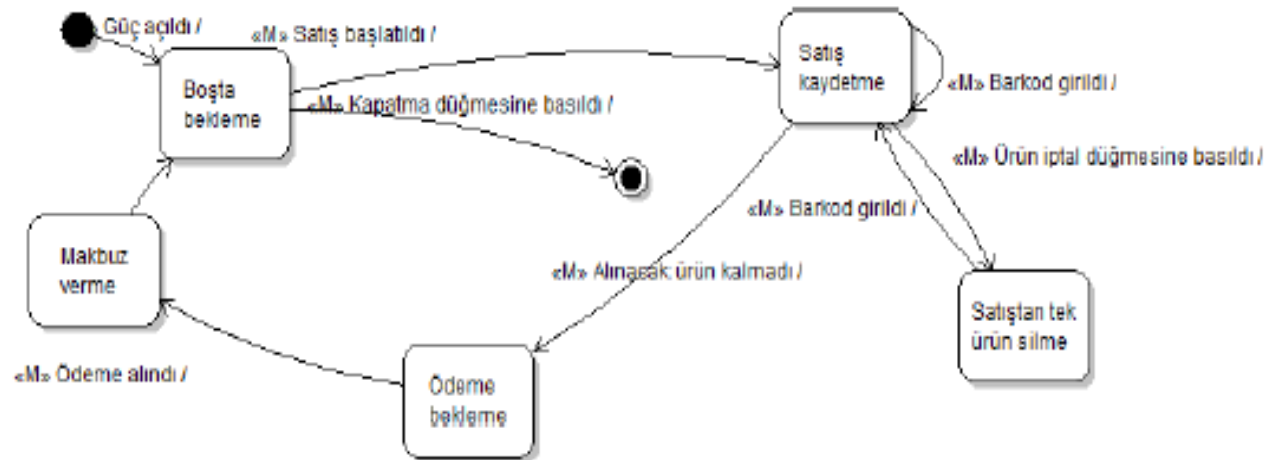
- Kullanım senaryosu metni ve alan modelinden yola çıkarak tasarım modelini oluşturalım.
  - Bu amaçla bir etkinlik, bir durum ve bir sınıf şeması çizelim.
  - Belki bu sırada keşfedeceğimiz yeni ayrıntılar olacaktır.
  - Tasarım modelindeki sınıf şemasının farkı, artık yönsüz ilişki bırakılmaması ve sınıfların metotlarının da eklenmesidir.



## TASARIM MODELİ

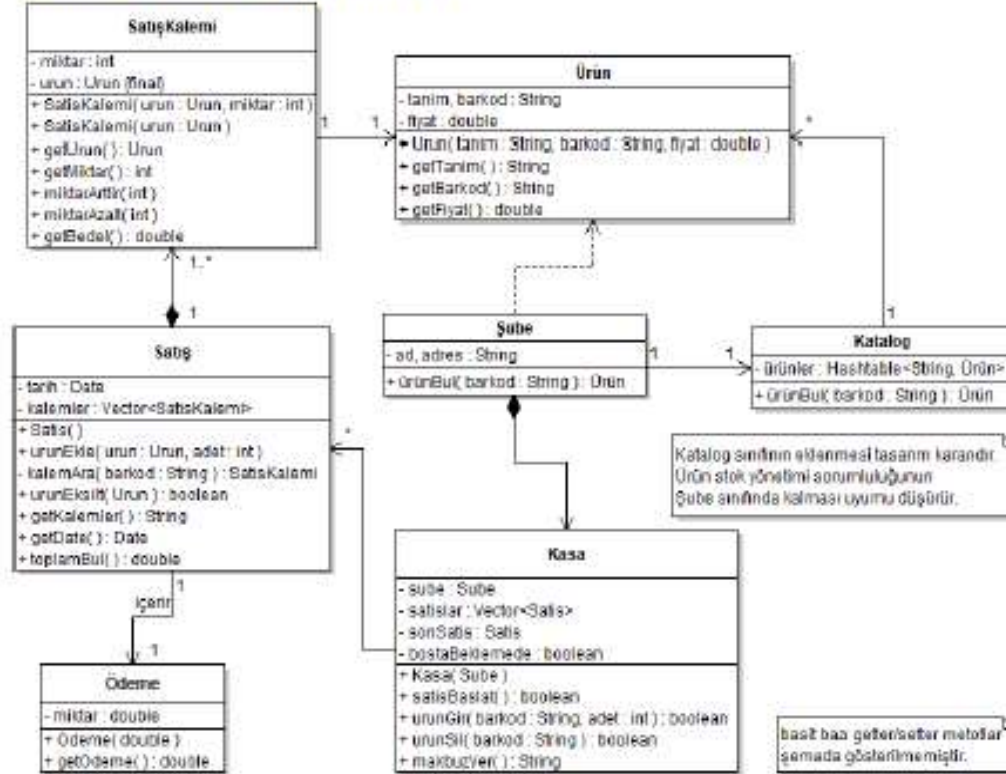
### NextGenPOS Tasarım Modeli

- Kasa ile ilgili durum şeması:



## TASARIM MODELİ

### NextGenPOS Tasarım Modeli



## YAZILIM KALİTESİ VE YAZILIM ÖLÇÜTLERİ

### ÖNERİLEN YAZILIM ÖLÇÜTLERİ

- Nesneye yönelik ölçütler:
  - Kaliteli bir yazılıma götüren tasarım ilkelerine yöneliktirler.
  - NYP'de çözümleme ve tasarım arasında kopukluk olmadığı için, aynı ölçütler çözümleme ve kodlama aşamalarında da kullanılabilir.
  - Böylece yazılım ekibi, 'kaliteli bir ürüne giden yolda' iz üstünde olup olmadıklarını anlayabilir.
  - Proje yöneticisi de, başka ölçütlerle birlikte, kestirimlerde bulunabilir.
- Chidamber ve Kemerer'in ölçütleri (CK metrics suite):
  - WMC: Sınıftaki ağırlıklı metot sayısı (Weighted Methods per Class).
  - DIT: Kalıtım ağacının derinliği (Depth of Inheritance Tree).
  - NOC: Alt sınıf sayısı (Number of Children)
  - RFC: Sınıfın yanıt kümesinin eleman sayısı (Response For a Class)
    - Good (RFC=[0,50]), regular (RFC=[51,100]), bad (RFC>100)
  - CBO: Sınıflar arası bağlaşım (Coupling Between Objects)
  - LCOM: Uyum eksikliği (Lack of COhesion in Methods)
    - Good (LCOM=0), regular (LCOM=[1-20]), bad (LCOM > 20).