

13. Hafta

YMÜ225-YMÜ321

Yazılım Gereksinim Analizi

Dr. Öğr. Üyesi Feyza Altunbey Özbay

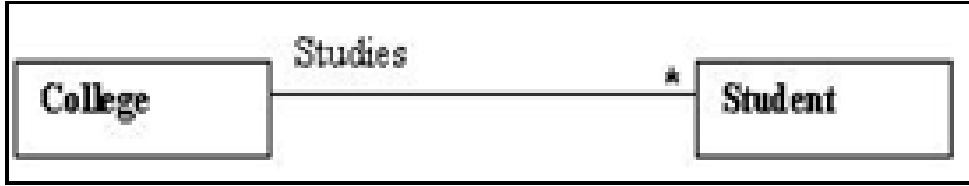
İçerik

- Nesne (Object) Diyagramları
- Ardıl Etkileşim (Sequence) Diyagramları
- İşbirliği (Collaboration) Diyagramları
- Durum (State) Diyagramları

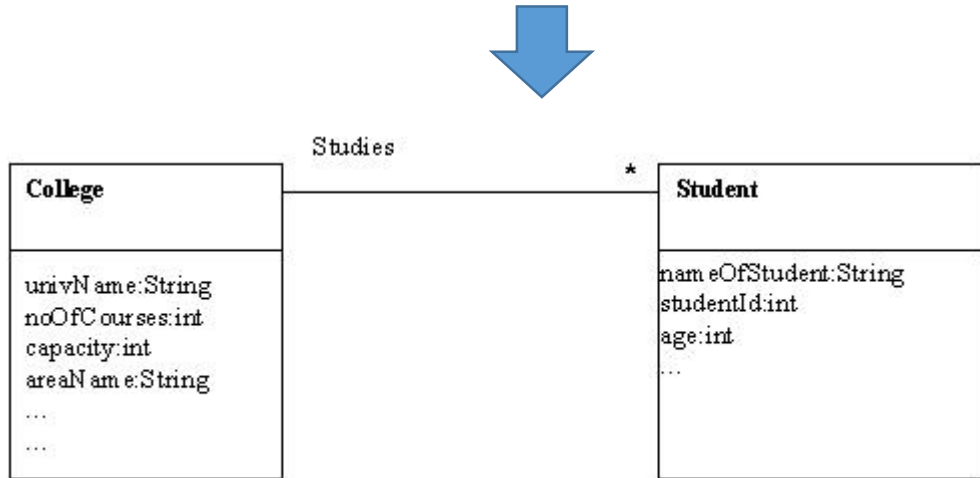
Nesne (Object) Diyagramları

- Nesne diyagramları sınıf diyagramlarından türetilir, böylece nesne diyagramları sınıf diyagramlarına bağımlıdır.
- Nesne diyagramları, bir sınıf diyagramının bir örneğini temsil eder. Temel kavramlar sınıf diyagramları ve nesne diyagramları için benzerdir.
- Nesne diyagramları aynı zamanda bir sistemin statik görünümünü de temsil eder, ancak bu statik görünüm belirli bir anda sistemin anlık görüntüsüdür.
- Nesne diyagramları, bir nesne kümesini ve ilişkilerini bir örnek olarak işlemek için kullanılır.

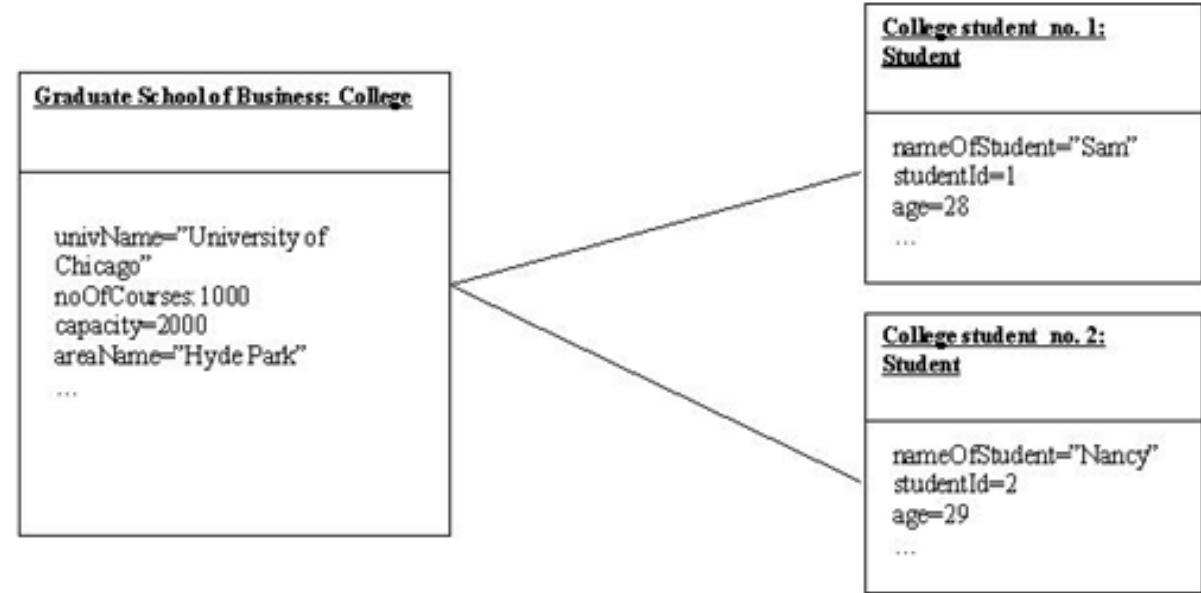
Nesne (Object) Diyagramları



Birden fazla öğrenci ile okul arasındaki ilişkiyi gösteren sınıf diyagramı



Sınıf diyagramının, College ve Student sınıflarına öznitelik eklenmiş hali

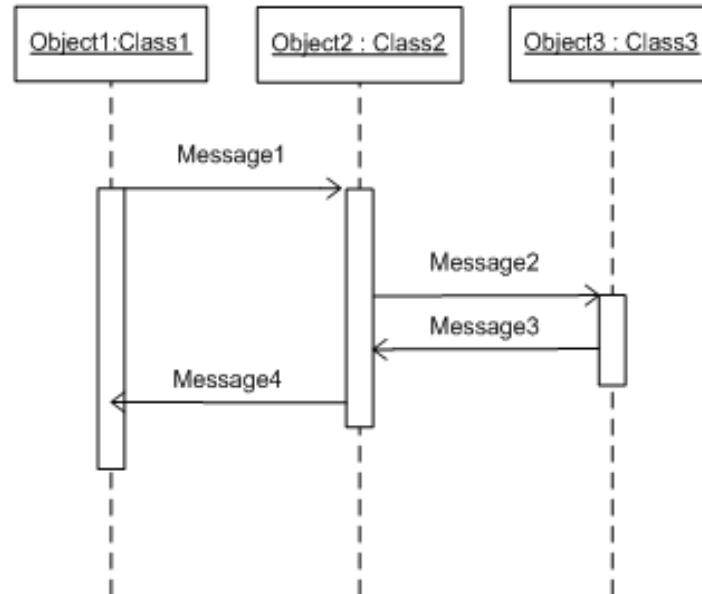


Object diyagramı, College ve Student sınıflarının örnekleri özniteliklere başlangıç değeri atanarak yaratılır.

Ardıl Etkileşim (Sequence) Diyagramları

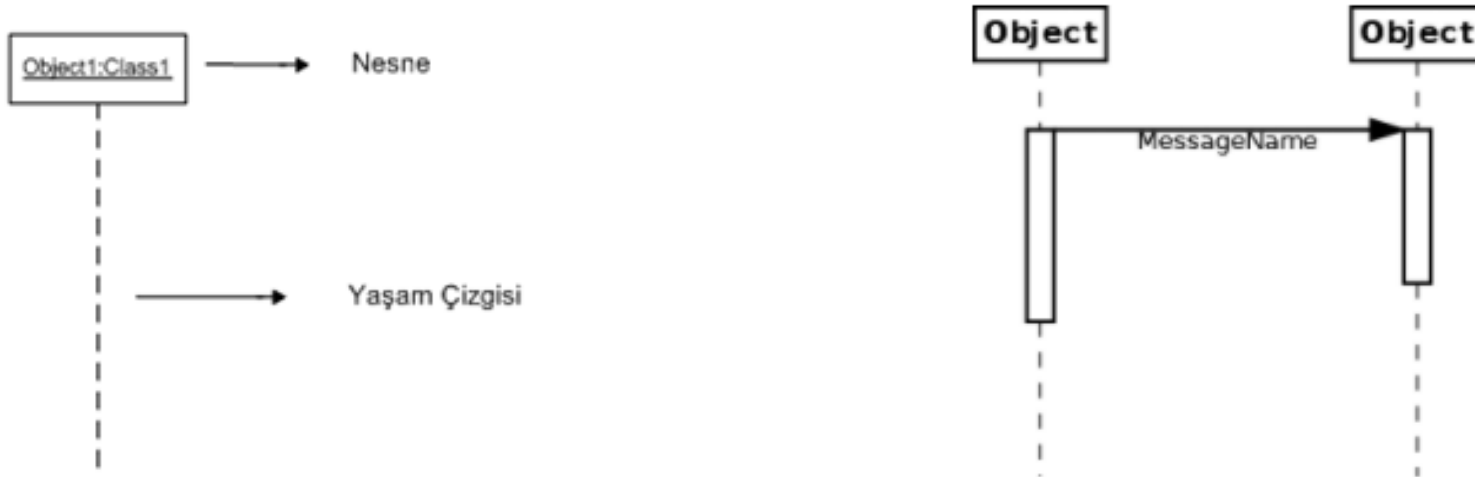
Sequence diyagramları belli bir kullanım senaryosunda (*use case*) rol alan nesneleri ve bu nesneler arasında geçen mesajları gösteren bir diyagramdır.

- Sequence diyagramı iki boyutludur:
Dikey boyut: Mesajların/olayların sırasını oluşma zamanı sıralarına göre gösterir.
- Yatay boyut: Mesajın gönderildiği nesne örneklerini (object instances) gösterir.



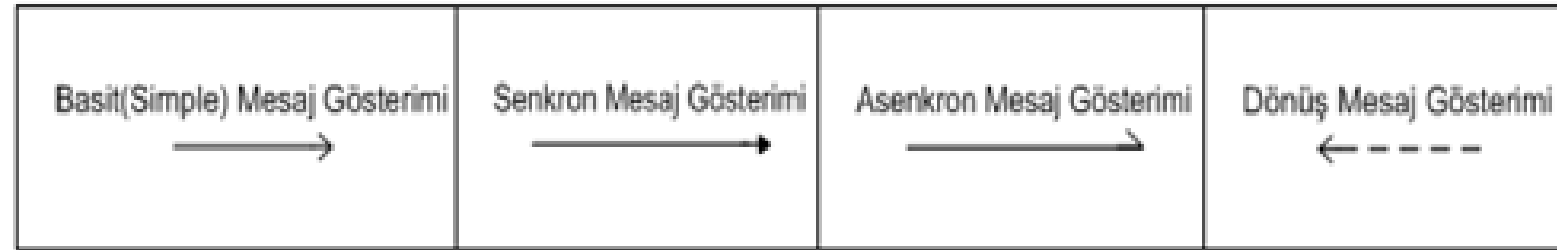
Ardıl Etkileşim (Sequence) Diyagramları

Nesnelerin altına çizilen dikey kesikli çizgilere **Lifeline (Yaşam Çizgisi)** adı verilir. Lifeline üzerinde nesnenin hangi anlarda ve ne kadar süre ile aktif olduğunu gösteren **Activation Bar (Aktivasyon Çubuğu)** adı verilen dikdörtgensel çubuklar yer alır. İki nesne arasındaki etkileşime **mesaj (message)** adı verilir ve oklarla temsil edilir. Okların yönü mesajı gönderenden alana doğrudur.



Ardıl Etkileşim (Sequence) Diyagramları

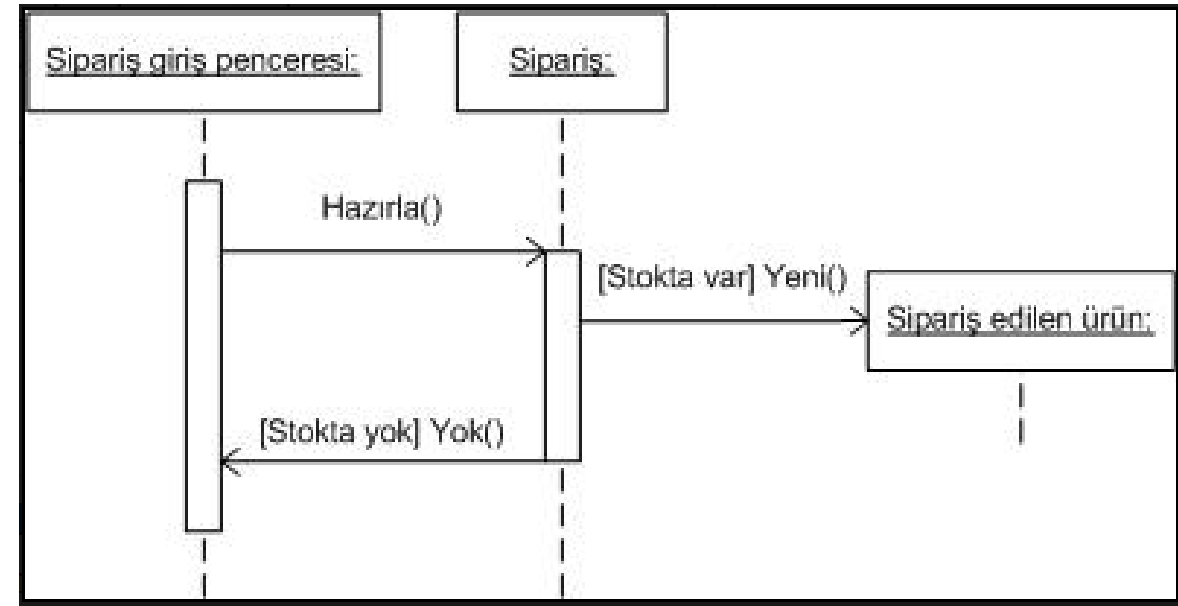
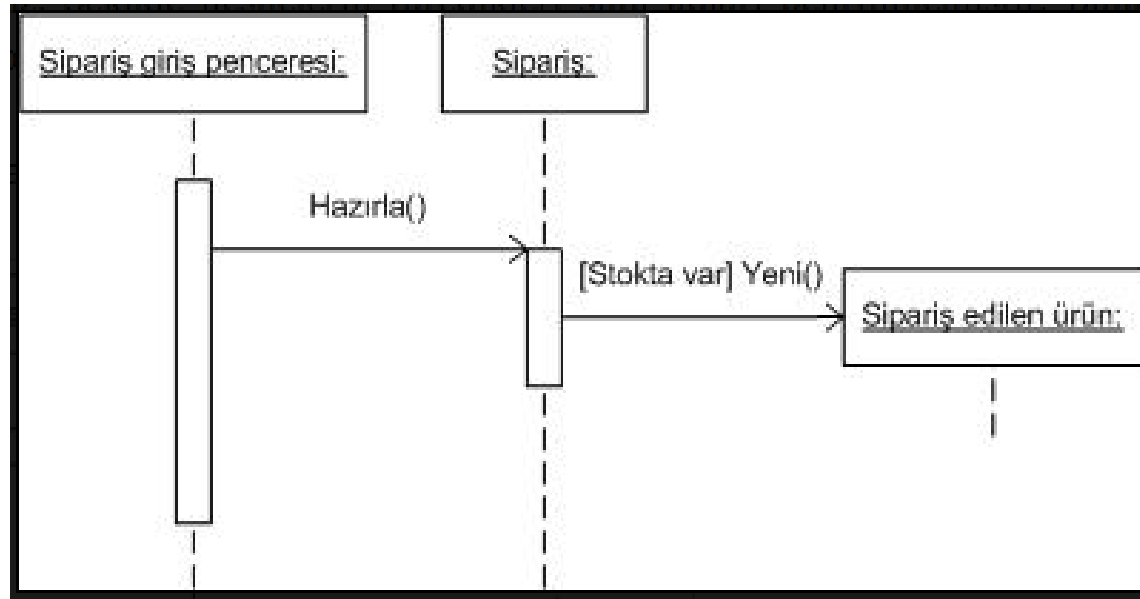
Mesaj Tipleri



- **Basit(Simple) Mesaj Tipi:** Nesneler arasındaki akış kontrolünün iletimini göstermek için kullanılır.
- **Senkron (Synchronous)/ Çağrı yapan(Call) Mesaj Tipi:** Nesne mesajı alıcı nesneye gönderir ve onun işlemini bitirmesini bekler, bu durumda senkron mesaj tipi kullanılır.
- **Asenkron Mesaj Tipi:** Senkron mesajların tersine, asenkron mesajlar nesneye mesaj gönderdikten sonra cevap beklemeden işleme devam etmesinin gösteriminde kullanılır.
- **Dönüş(Return) Mesaj Tipi:** Senkron mesajlarda alıcı nesnenin işleminin bitimini, gönderen nesneye bildirmesinde kullanılır.

Ardıl Etkileşim (Sequence) Diyagramları

“Sipariş Giriş Penceresi” adında bir nesne yaratılmakta ve siparişin hazırlanması için Sipariş nesnesine mesaj göndermektedir. Daha sonra Sipariş nesnesi ürünün stokta olup olmadığına bakmakta ve eğer [StoktaVar] koşulu sağlanırsa “Sipariş Edilen Ürün” nesnesine mesaj göndermektedir. Bir sonraki diyagramda ise Sipariş nesnesine koşullu mesaj eklenmiştir. Eğer ürün [StoktaYok] koşulunu sağlarsa “Sipariş Giriş Penceresi” nesnesine ürünün kalmadığını belirten bir mesaj gönderir.



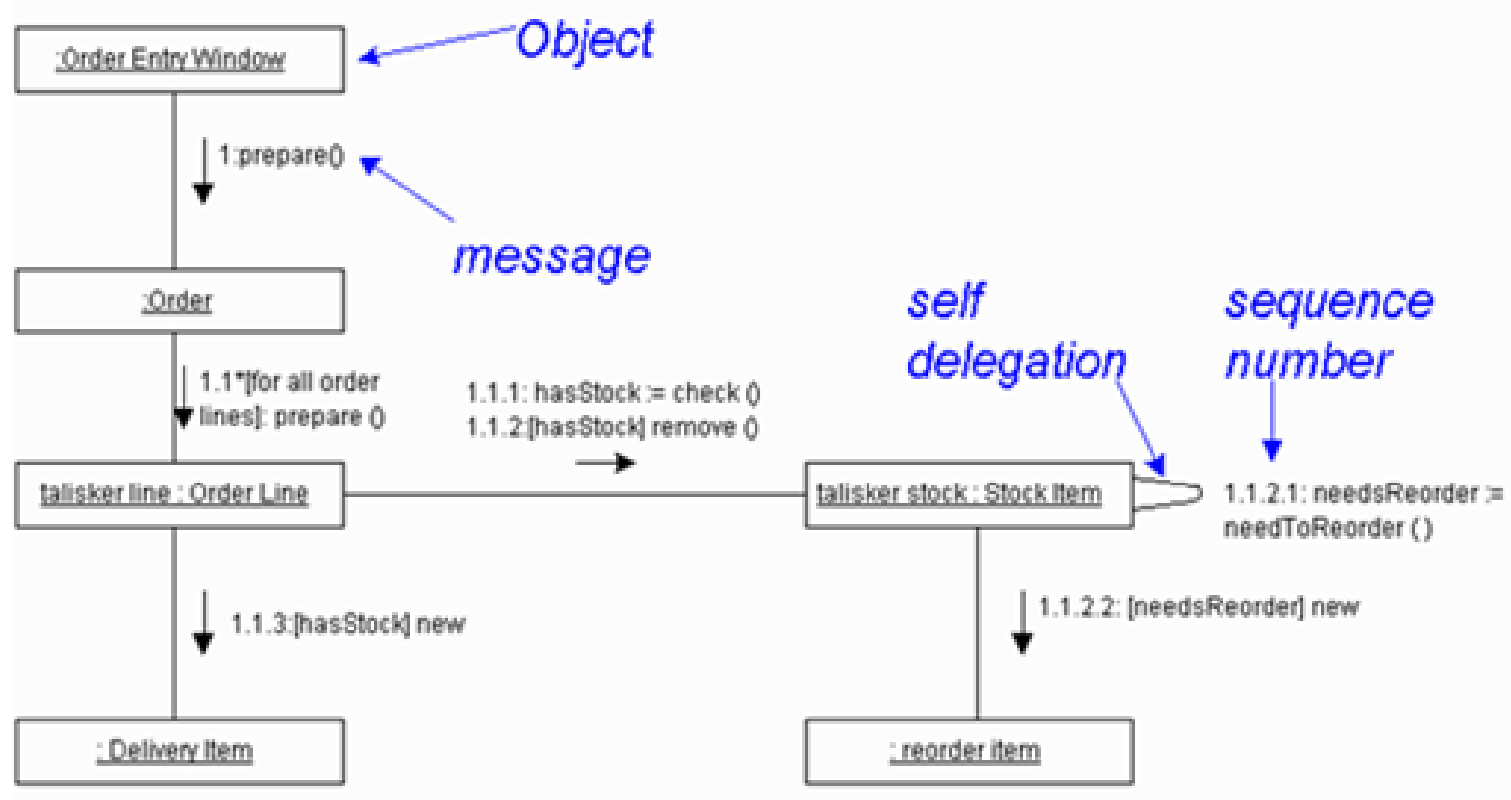
İşbirliği (Collaboration) Diyagramları

Nesneler arasındaki mesajlaşmaların numaralandırılmış akış sırası ile ifade eden diyagramlardır.

Sequence diyagramları ile aynı bilgileri taşırlar, fakat sequence diyagramlara göre nesneler arasındaki modelleme farklıdır.

Collaboration diyagramlarında nesnelerin iki farklı karakteri olduğundan söz edilir, şayet bir nesne (örneğin A) diğer bir nesnenin (B'nin) fonksiyonunu çağırıyorsa; A *istemci (client)* B ise *tedarikçi (supplier)* terimleri ile bu nesneler adlandırılır. A ---> B.

İşbirliği (Collaboration) Diyagramları



Sequence ve Collaboration Farkları

Sequence Diyagramlar	Collaboration Diyagramlar
Zamana bağımlı mesajlaşmayı gösterir	Zamandan bağımsızdır.
Dinamik yapıyı temsil eder.	Dinamik yapıyı ve nesnelerin statik ilişkisini temsil eder.
Okuması kolaydır.	Daha karmaşık sistemlerin modellemesi için kullanılır. Okunması sequence diyagramlara göre daha zordur. .
Senaryoya ait tüm resmi net olarak göstermeyebilir.	Senaryoya ait tüm resmin net olarak gösterilmesini sağlar
Bir bağlantı, bir nesneden diğerine giden mesajı gösterir.	Bir bağlantı nesneler arasındaki 1 veya daha fazla mesajı gösterebilir.
Nesnelerin yaratılışı ve sonlandırılışını kolay şekilde gösterebilmenize imkan vermesidir. Zaman boyutunda "yeni ()" mesajı ile nesneyi yaratabilir, X işareti ile de nesnenin sonlandırıldığını gösterebilirsiniz.	Nesnelerin yaratılışı ve sonlandırılışını belirtmek sequence diyagramlardaki kadar basit değildir. Bu durumu gerçekleştirmek için "oluştur()" ve "sonlandır()" mesajlarını modele katmanız gerekir.

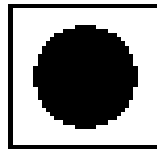
Durum (State) Diyagramları

- Sistem davranışlarını gösteren diyagramlardır.
- Her diyagram tek bir sınıfın nesnelerini ve sistem içerisinde nesnelerin durumlarını (state), geçişlerini (transition), olaylarını (Events) içermektedir.
- Sistemdeki nesnenin davranışı, kendisine gönderilen iletilerin yanı sıra x anı içerisinde değişiklik gösteriyor ise state diyagram kullanılır.
- Durum değiştirmeyen nesneler için state diyagram modellemesinin kullanılması uygun değildir.

Durum (State) Diyagramları

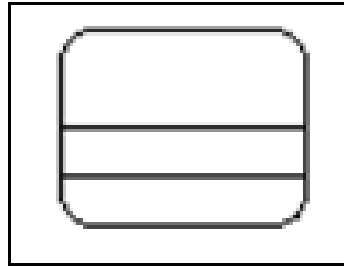
Temel Kavramlar:

Initial State (İlk Durum): Yaşam döngüsünün ilk eylemi ya da başlama noktasını ifade eden elemandır. İçi dolu yuvarlak ile gösterilir. Sözde durum (pseudo state) olarak da adlandırılır. Sözde durum denilmesinin sebebi değişkeni veya herhangi bir eyleminin olmayışıdır.



Durum (State) Diyagramları

- **State (Durum):** Nesnenin ya da sistemin x anındaki durumunu ifade etmek için kullanılır. Köşeleri yuvarlatılmış dikdörtgenler ile gösterilir.



Durum (State) Diyagramları

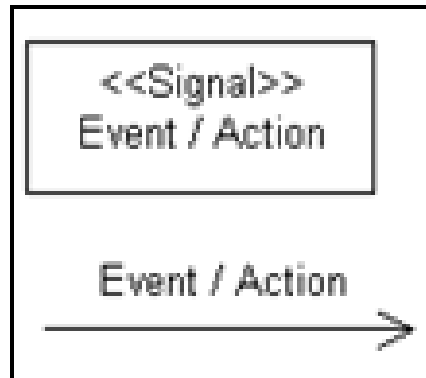
- **Transition (Geçiş):** Nesnenin bir durumdan diğer duruma geçişini ifade eder. Ok sembolü ile gösterilir. 4 yapı vardır:
 - Hedef (target)
 - Kaynak (source)
 - Olay tetikleyicisi (Event trigger): Kaynak durumdan hedef duruma geçişi tetikleyen.
 - Action: Durum geçişi sonrasında nesnenin ne şekilde davranacağı.

Durum (State) Diyagramları

- **History States (Geçmiş Durumlar):** Bir akışta, bir nesne, bekleme durumuna geçebilir. Belirli bir olayın gerçekleşme süresinde bekleme durumuna girdiği zaman son aktif durumuna geri dönmek istenebilir. Daire içerisinde H harfiyle gösterilir.
- **Event and Action (Olay ve Eylem):** Durumlar arası geçişi tetikleyiciler olay (event) olarak adlandırılır. Eylem (action) ise yaşanan durum değişimi sonrasında yaptığı işlerdir.

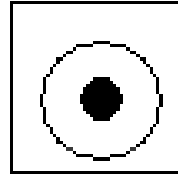
Durum (State) Diyagramları

Signal (Sinyal): Duruma bir mesaj ya da tetikleyici gönderimi olduğunda geçiş oluşur ve mesaj event (olay) ile gönderildiğinde sinyal olarak adlandırılır. << Sinyal>> şeklinde gösterilir.



Durum (State) Diyagramları

- **Final State (Son Durum):** Sistemin yaşam döngüsü içerisindeki son durumunu ifade eder.



- **State Machine (Durum Makinesi):** Bir nesne ya da sisteme ait bütün durumları şema halinde gösteren yapıdır.

Durum (State) Diyagramları

