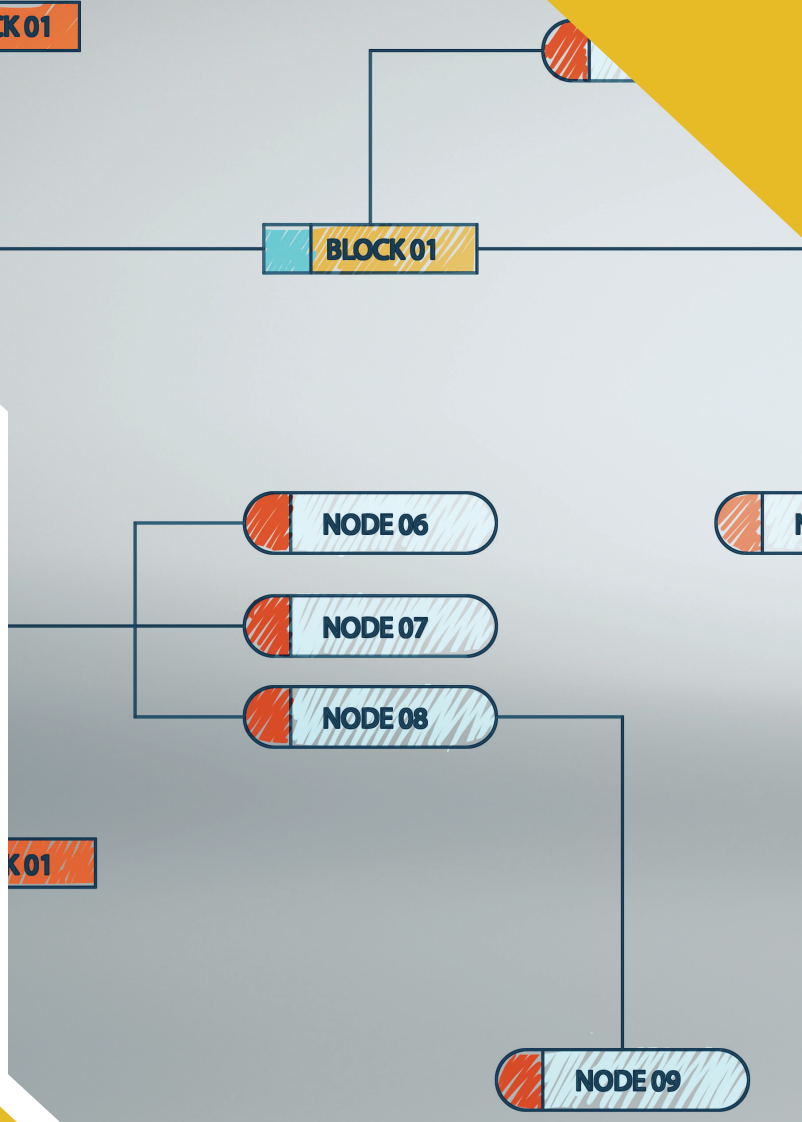


BÖLÜM 7

NESNE YÖNELİMLİ MODELLEME

ANAHTAR KAVRAMLAR

Nesne, Sınıf, Katılım, Çok Bizimlilik,
Kapsülleme, UML

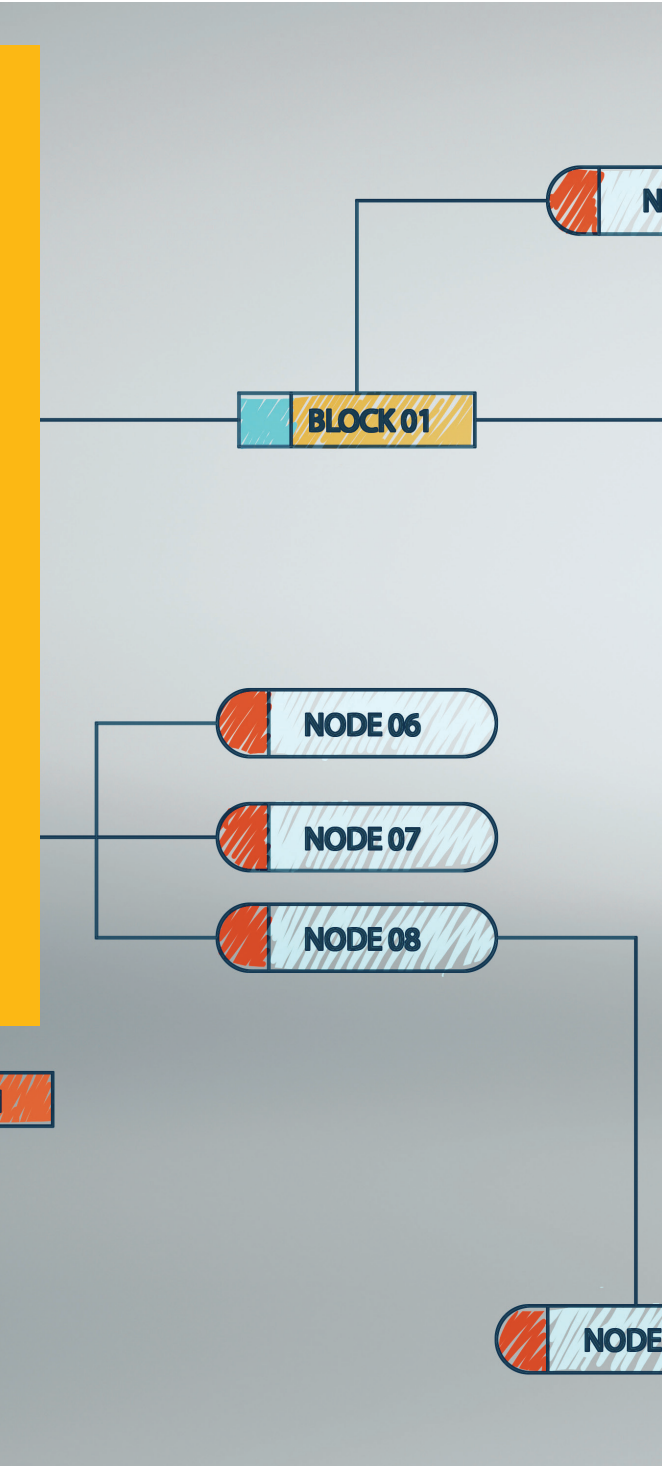


ÖĞRENME HEDEFLERİ

-  1 Nesne tabanlı modelleme kavramını açıklayabilecek
-  2 Nesne tabanlı modelleme temel kavramlarını açıklayabilecek
-  3 UML diyagramlarını ve çeşitlerini tanımlayabilecek
-  4 Nesne tabanlı modelleme sürecini açıklayabilecek

GİRİŞ

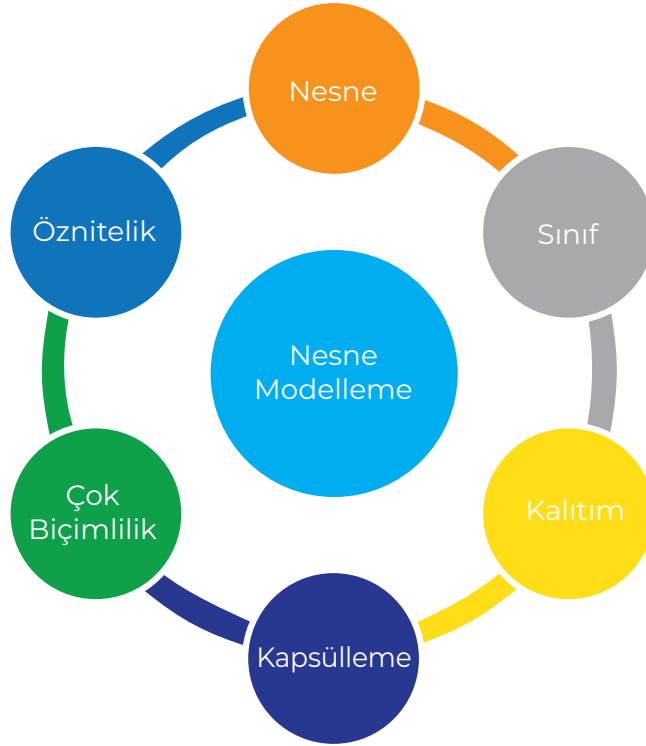
Dijital ortamlar ve uygulamalar hayatımızın vazgeçilmez bir parçası olmuştur. Teknolojideki hızlı gelişim ile birlikte kurum-kuruluşlar, özel şirketler, sivil toplum kuruluşları gibi birçok yapı için bahsi geçen ortamları ya da uygulamaları kullanma zorunluluğu oluşmuştur. Dolayısıyla sistem ve yazılım geliştirme süreçleri çok önemli bir hale gelmiştir. Nesne tabanlı modelleme de yazılım geliştirme ve sistem tasarımı yaklaşımlarının temelini oluşturan bir kavramdır. Nesne yönelimli modelleme, nesne ve sınıflarının tanımlanmasıyla karakterize edilen bir görselleştirme sürecidir. Başka bir ifade ile nesne ve sınıfları temel alan bir yaklaşımdır. Bahsi geçen yaklaşımın daha hızlı yürütülmesi, daha kolay bir yapı sağlaması, tasarlanacak sistemin bakımının, güncellenmesinin ya da hata ayıklamasını kolaylaştırması, daha kısa sürede kullanılabilir uygulamalar geliştirmeye olanak sunması en önemli avantajlarından. Bu avantajlardan dolayı geleneksel modelleme yaklaşımlarına alternatif bir seçenek olmuştur.



1. NESNE MODELLEMeye İLİŞKİN TEMEL KAVRAMLAR

Nesne tabanlı tasarım ve modelleme süreçlerini gerçekleştirmek için bazı temel kavramları öğrenmek gerekmektedir. Bu kavramların bilinmesi, ilgili sürecin anlaşılması için önemlidir. Nesne tabanlı modelleme geleneksel yöntemle-

rin aksine, davranış modelleri oluşturmaktadır. Davranış modellerini oluşturmak adına gerekli temel kavramlar Şekil 1'de verilmiş olup takip eden bölümde tanımlamaları yapılmıştır.



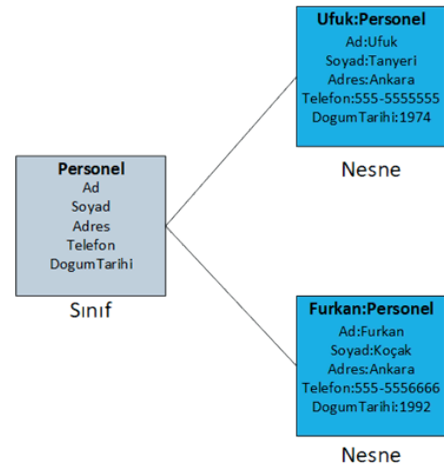
Şekil 1. Nesne tabanlı modelleme temel kavramları

Sınıf:

Nesne yönelimli tasarımın temelini sınıflar oluşturmaktadır. Kendi veri üyelerini ve üye fonksiyonları olan bir veri türüdür. Sınıf, bir nesnenin planı gibidir. Bir başka ifade ile sınıf, veri üyelerine ve üye işlevlerine sahip olan veri türüdür. Sınıflar genel olarak nesnelerin genel özelliklerini tanımlarlar.

Nesne:

Nesneye yönelik tasarımlarda, nesneler sınıflardan türetilirler. Nesne, bir sınıfın örneğidir. Sınıfın tüm veri üyelerine ve üye işlevlerine nesneler yardımıyla erişilebilir. Aynı sınıftan türetilmiş nesnelerin sahip olduğu özellikler benzerdir. Şekil 2'de nesne ve sınıf kavramlarının ilişkileri verilmiştir.



Şekil 2. Sınıf ve Nesne ilişkisi

Söz konusu şekil incelendiğinde Personel adında bir sınıf tasarımı yapılmıştır. Personel sınıfı genel olarak bir şablon tasarımıdır. Ad, soyad

vb. gibi değişkenler Personel sınıfının veri üyelerini oluşturmaktadırlar. Ufuk ve Furkan yapıları ise Personel sınıfından türetilen nesneleri temsil etmektedirler. Her iki nesnede sınıfa ait özellikleri taşımaktadırlar. Dolayısıyla nesneler sınıfların şablon tasarımını kullanan yapılardır. Sınıflar soyut bir veri tipi iken, nesneler ise onun somutlaşan halidir. Nesneler ancak sınıflar türetildikten sonra oluşturulurlar.

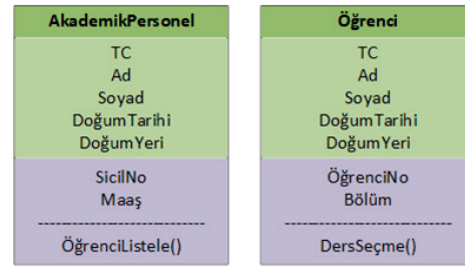
Öznitelik:

Sınıflardan oluşturulan nesneleri tanımlamaya yarayan özelliklere öznitelik denir. Şekil 2’de verilen Furkan ve Ufuk nesnelerinin öznitelikleri ad, soyad, adres, telefon ve doğum tarihidir.

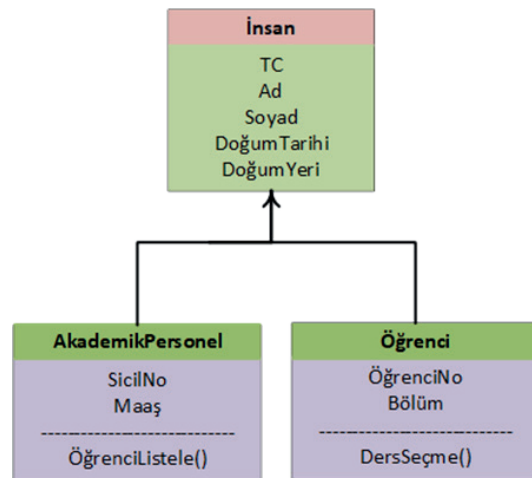
Kalıtım:

Nesne yönelimli modellemede kalıtım, bir nesneyi veya sınıfı, başka bir nesnenin veya sınıfın özelliklerini kullanarak alt sınıf ya da alt nesneler oluşturma mekanizması olarak tanımlanmaktadır. Temel sınıf gibi mevcut sınıflardan yeni sınıflar-alt sınıflar türetmek ve ardından bunları bir sınıf hiyerarşisi halinde oluşturmak olarak da tanımlanır. Kalıtım yoluyla oluşturulan bir sınıf, temel sınıfın tüm özelliklerini ve davranışlarını alır. Kalıtımın ana avantajı, sistemin yeniden kullanılabilirliğine yardımcı olmasıdır. Sınıflar yalnızca bir kez tanımlanır ve birden çok kez kullanılabilir. Kalıtım yoluyla zaman tasarrufu sağlanmaktadır ve okunabilirlik yapısı geliştirilmektedir. Kalıtım ile oluşturulan sınıfların ortak veri üyeleri temel sınıf ya da üst sınıfta bulunmaktadır. Bu yapıya genelleşme denir. Alt sınıflar ayrıca temel sınıfların veri üyelerinin dışında farklı öznitelik ya da fonksiyonlara sahip olabilmektedir. Bu durum ise özelleşme olarak tanımlanmaktadır. Örneğin bir İnsan temel sınıfından türetilmiş Öğrenci ve AkademikPersonel isimlerinde alt sınıflar düşünebiliriz. Öğrenci ve AkademikPersonel alt sınıfları İnsan temel sınıflarının özelliklerini taşımaktadır. Bu durum ge-

nelleşme kavramına örnek teşkil ederken, bahsi geçen Öğrenci ve AkademikPersonel alt sınıflarının temel sınıftan farklı veri üyeleri ve fonksiyonlara sahip olabilmesi özelleşmeye örnektir. Şekil 3 AkademikPersonel ve Öğrenci sınıflarını göstermektedir. Her iki sınıfta ortak özellikleri bulunmaktadır. Şekil 4 ise kalıtım yapısını temsil etmektedir. Şekil 4’te AkademikPersonel ve Öğrenci sınıfları alt sınıfları gösterirken, İnsan sınıfı ise temel sınıfı temsil etmektedir. Alt sınıfların ortak özellikleri temel sınıf altında toplanmıştır. Kalıtım yoluyla ise alt sınıflara farklı özellikler ve fonksiyonlar eklenmiştir. Şekillerden de anlaşılacağı üzere ortak öznitelikleri bir defa tanımlamak yeterli olacaktır. İlaveten İnsan temel sınıfı üzerinde yapılacak herhangi bir güncelleme işlemi her iki alt sınıfta da uygulanacaktır. Dolayısıyla bu işlem zaman ve hesaplama maliyetini azaltan bir işlemdir.

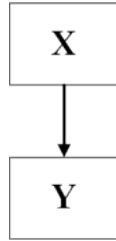


Şekil 3. Kalıtım kullanılmadan oluşturulan sınıflar

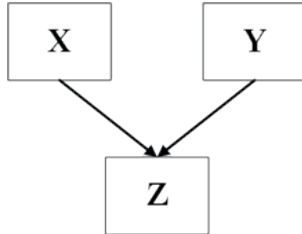


Şekil 4. Kalıtım yoluyla oluşturulan sınıflar

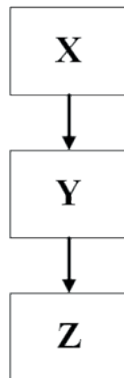
Kalıtım işlemi, **tekli kalıtım**, **çoklu kalıtım**, **çok seviyeli kalıtım**, hiyerarşik kalıtım ve hibrit kalıtım olmak üzere 5 çeşittir. Tekli kalıtımda alt sınıf tek bir üst sınıfın özelliklerini taşıırken, çoklu kalıtımda birden fazla üst sınıfın özelliklerini taşımaktadır. Bir diğer kalıtım çeşidi olan çok seviyelide ise bir sınıfın bir alt sınıfı oluşturulduktan sonra, bu alt sınıftan başka bir alt sınıf oluşturulur. Hiyerarşik kalıtım ise bir sınıfın birden fazla alt sınıfa temel sınıf yapmasıdır. Son olarak hibrid kalıtım ise yukarıda bahsi geçen kalıtım mimarilerinden iki ya da daha fazlasının kullanılmasıdır. Şekil 5, Şekil 6, Şekil 7 ve Şekil 8 sırasıyla tekli, çoklu, çok seviyeli ve hiyerarşik kalıtım çeşitlerini simgelemektedir.



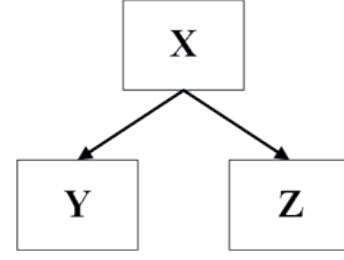
Şekil 5. Tekli kalıtım



Şekil 6. Çoklu kalıtım



Şekil 7 Çok seviyeli kalıtım



Şekil 8 Hiyerarşik kalıtım

Kapsülleme

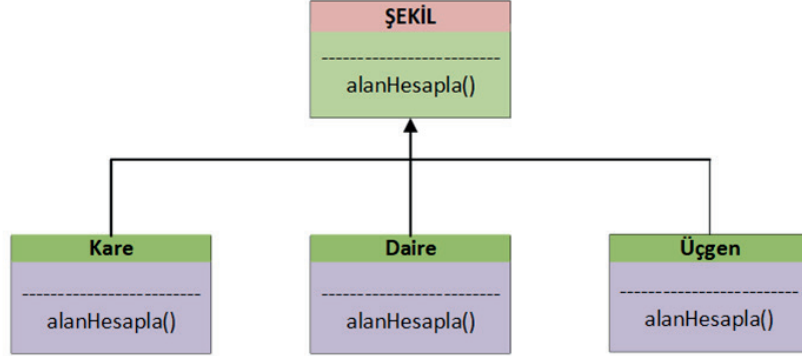
Nesne tabanlı modellemenin temel kavramlarından biri olan kapsülleme, kullanıcıların sınıf, metot verilerin ne kadarına erişebileceğini belirler. Başka bir ifade ile sınıf, metot ya da verilere doğrudan erişimlerin kısıtlanması ve izin verilmesi gibi işlemleri gerçekleştirmek için kullanılır. Erişim izinleri erişim belirleyici anahtar kavramları ile oluşturulurlar. Bu kavramlar **public**, **private**, **protected**, **internal** olarak isimlendirilir. Public olarak tanımlanan bir nesne modelleme bileşeninin (sınıf, metot ya da veri) erişim kısıtlaması yoktur ve her yerden erişilebilirler. Bir üye private olarak tanımlanmış ise; o üyeye sadece kendi sınıfı içerisinde ulaşılır. Protected erişim belirleyicisi model içerisindeki bir üyeye bulunduğu sınıftan veya o sınıftan türetilen diğer sınıflar içinden erişebileceğini gösterir. Internal ise bir üyeye aynı program içinden erişime izin verirken, başka programdan erişimi kısıtlaması için başvurulmuş bir erişim belirleyicisidir.

Çok Biçimlilik:

Çok biçimlilik herhangi bir verinin birden fazla biçimde kullanılmasıdır. Çok biçimlilik, nesne yönelimli modellemede en önemli kavramlarından biridir. İlgili kavram, herhangi bir mesajın birden çok sınıftaki nesnelere gönderilebildiği bir özelliktir ve her nesne, sınıf özelliklerine bağlı olarak uygun bir şekilde yanıt verme eğilimindedir. Yani bir fonksiyonun birçok türde ve birçok biçimde temsil edilebilmesidir. Çok

biçimliliğin gerçek hayattan bir örneğinde, herhangi bir kişinin aynı anda farklı rollere sahip olmasıdır. Bir birey, aynı zamanda bir ebeveyn, bir eş, bir çalışan ve bir evlattır. Yani aynı kişi birçok

özellığe sahip olabilir ama her birini duruma ve koşula göre uygular. Şekil 9 çok biçimlilik kavramını temsil etmektedir.



Şekil 9 Çok biçimlilik

Şekil 9'da ŞEKİL isimli temel sınıftan Kare, Daire ve Üçgen olmak üzere 3 sınıf türetilmiştir. Temel sınıftaki alanHesapla fonksiyonu her üç alt sınıfta da bulunmaktadır. Ancak ŞEKİL sınıfı bir Kare sınıfı nesnesinin yerine geçti ise alanHesapla fonksiyonu karenin alanını, Daire sınıfının

nesnesinin yerine geçti ise daire alanını hesaplayacaktır. Çok biçimlilik ile bir kez tasarlanan, test edilen ve uygulanan sınıfların yeniden yazılmasına gerek kalmaz ve dolayısıyla nesne yönelimli modellemede işlemlerinde zaman kazandırır.

2. UML DİYAGRAMLARI

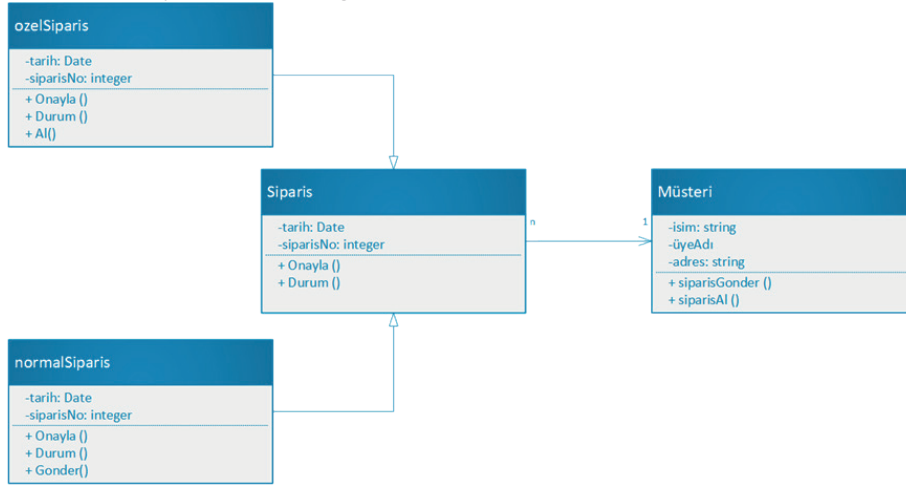
Birleşik modelleme dili (UML: Unified Modeling Language), yazılım sistemlerini, görselleştirmek, oluşturmak ve belgelemek için kullanılan standart bir dildir. Yazılım sistemlerinin geliştirilmesi genel olarak analiz, dizayn, kodlama ve uygulama aşamalarından oluşmaktadır. Özellikle analiz ve dizayn adımlarında UML diyagramları sıklıkla tercih edilmektedir. Geliştirilecek olan sistemin görselleştirilmesi kodlama ve uygulama aşamalarında hata riskini azaltmaktadır. UML Java, C#, Python gibi programlama dillerinden farklıdır. UML, yazılım sistemlerinin tasarımlarını oluşturmak adına kullanılan resimsel bir dildir. UML genellikle sistemleri modellemek için kullanılmaktadır. UML nesne yönelimli analiz ve tasarımın vazgeçilmez bir unsurudur. UML yapısının amacı tasarlanacak sistemi kolay anlaşılır ve basit hale getirmektir. İlaveten tüm kullanıcılar ya da tasarımcılar için genel amaçlı modelleme oluşturmaktır.

UML diyagramları yapısal ve davranışsal olmak üzere genel olarak 2 kategoride toplanmaktadır. Yapısal model, sistemin çerçevesini temsil eder. Yapısal model bileşenleri verileri ve onları bir araya getiren mekanizmayı temsil etmektedir. İlgili model sistemin dinamik davranışından ziyade statik davranışlarını tanımlamaktadır. Sınıf, nesne, paket, dağılım, bileşen, birleşik gibi diyagramlar yapısal modeli oluşturan UML diyagramlarıdır. Sınıf diyagramı ise en yaygın kullanılan yapısal diyagramdır. Davranışsal model ise sistemdeki etkileşimi tanımlamaktadır. Davranışsal modelleme, sistemin dinamik doğasını gösterir. Etkinlik diyagramları, etkileşim diyagramları, kullanım-durum diyagramları dinamik modele örnektir. İlerleyen bölümlerde nesne tabanlı modelleme için en çok kullanılan UML diyagramları tanımlanacaktır.

2.1. Sınıf Diyagramları (Class Diagram)

Sınıf diyagramı statik bir diyagramdır. Bir sistemin statik görselini temsil eder. Sınıf diyagramı, aynı zamanda yazılım uygulamasının çalıştırılabilir kodunu oluşturmak için de kullanılmaktadır. İlgili diyagram ile bir sınıfın özniteliklerini, fonksiyonlarını ve ayrıca varsa eğer sisteme uygulanan kısıtlamalar ifade edilebilir. Sınıf diyagramları, doğrudan nesne yönelimli programla-

ma dilleriyle örtüşen tek UML diyagramları olduklarından, nesne yönelimli sistemlerin modellenmesinde yaygın olarak kullanılmaktadır. Sınıf diyagramı, sınıfların, ara yüzlerin, ilişkilerin, işbirliklerinin ve kısıtlamaların bir koleksiyonunu gösterir. Yapısal diyagram olarak da bilinir. Örnek bir sınıf diyagramı Şekil 10 ile gösterilmiştir.

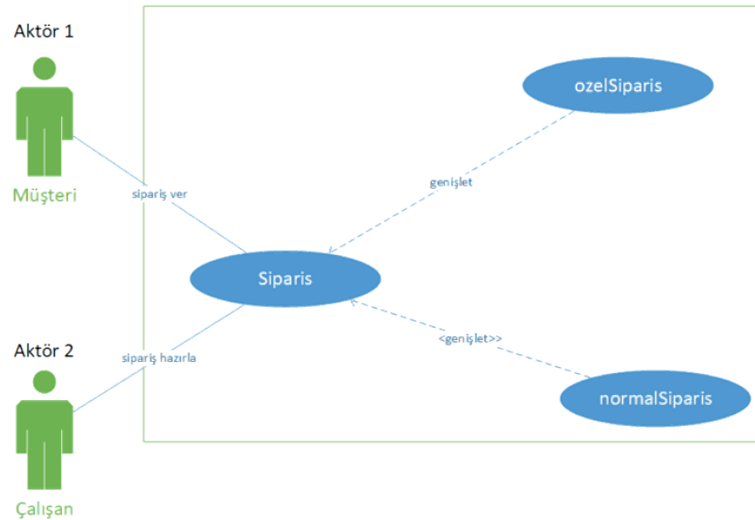


Şekil 10. Sınıf diyagramı

2.2. Kullanım Durum Diyagramları (Use Case Diagram)

Kullanım durum diyagramının temel amacı, bir sistemin dinamik yönünü simgelemektir. İlgili diyagramlar, iç ve dış etmenler de dâhil olmak üzere bir sistemin gereksinimlerini toplamak için kullanılır. Bu gereksinimler genellikle tasarım gereksinimleridir. Bu nedenle, kullanım

senaryoları hazırlanır ve aktörler belirlenir. Aktörler, sistemle etkileşime giren kullanıcıların rolleri, dâhili veya harici uygulamaları olarak tanımlanmaktadır. Kullanım durum diyagramını Şekil 11 özetlemektedir.

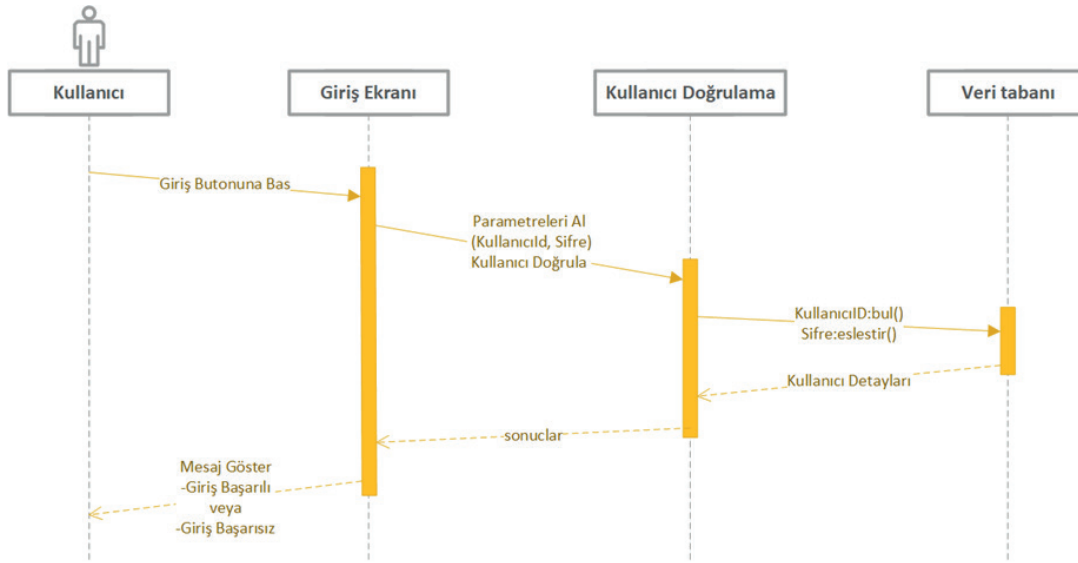


Şekil 11 Kullanım durum diyagramı

2.3. Dizge Diyagramları (Sequence Diagram)

Dizge diyagramları, nesneler arasındaki etkileşimi sıralı bir şekilde gösteren diyagramlardır. Dizge diyagramları, bir sistemdeki nesnelerin ve olayların nasıl ve hangi sırayla gerçekleştiğini açıklar. Bu bağlamda dinamik bir bakış açısı sağlamaktadır. Bu diyagramlar sistem ya da yazılım geliştiriciler tarafından yeni ve mevcut sistemlerin gereksinimlerini belgelemek ve ifade etmek adına yaygın olarak kullanılmaktadır. Zaman ve sıra dizge diyagramlarında önemli kavramlardır.

Aktörler diyagramların en tepesine yerleştirilir ve aşağı indikçe zaman geçmektedir. Yani bir sistemdeki gerçekleşecek işlemler yukarıdan aşağı sırasıyla yazılır. Şekil 12 bir kullanıcının herhangi bir sisteme giriş işleminden başlayan süreci gösteren bir dizge diyagramını göstermektedir. Söz konusu şekilde giriş butonuna basılmasıyla süreç başlamaktadır. Sonraki her bir adım sistemin içerisindeki işlemleri tanımlamaktadır ve belirli bir sıraya göre ilerleme gerçekleşmektedir.

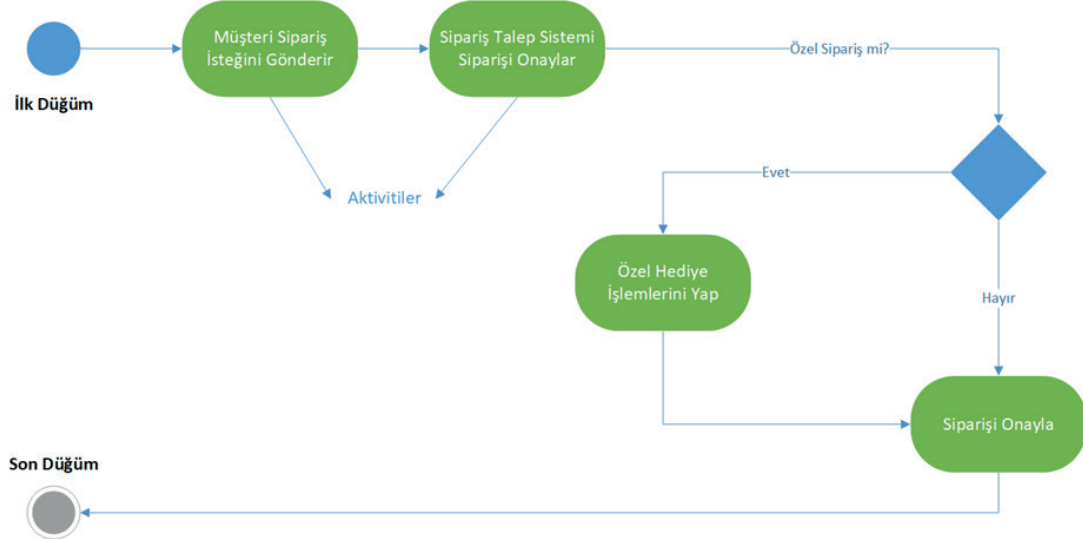


Şekil 12. Dizge diyagramı

2.4. Etkinlik Diyagramları (Activity Diagram):

Etkinlik diyagramlarının temel amacı tasarım yapılacak olan sistemin dinamik davranışlarını yakalamasıdır. Dizge diyagramları bir nesneden diğerine mesaj akışını göstermek için kullanılırken, etkinlik diyagramı bir etkinlikten diğerine mesaj akışını göstermek için kullanılır. Etkinlik, sistemin içerisinde belirli bir işlemdir. Etkinlik diyagramları sadece bir sistemin dinamik

doğasını görselleştirmek için kullanılmaz, aynı zamanda ileri ve tersine mühendislik teknikleri kullanılarak yürütülebilir sistemi oluşturmak için de kullanılır. Bir etkinlik diyagramı, bir sistemin etkinlik akışını görselleştirmek, etkinliklerin sırasını açıklamak, sistemin paralel, dallanmış ve eşzamanlı akışını tanımlamaktadır. Şekil 13 örnek bir etkinlik diyagramını ifade etmektedir.



Şekil 13. Etkinlik diyagramı

3. NESNE YÖNELİMLİ MODELLEME

Sistemlerin karmaşıklığının artması, proje maliyetlerinin ve iş gücünün önemi modelleme tekniklerinin önemini arttırmaktadır. Nesne yönelimli modelleme, gerçek dünyanın kavramları ile organize edilen bir çözümü görselleştirme tekniğidir. Söz konusu modelleme yaklaşımı, problemleri ifade etmeye, sistemlerin çalışma süreçlerini ifade etmeye ya da genel olarak tasarlamaya yardımcı olmaktadır. Bir sistem mekanizmasını geliştirmeden önce bir model geliştirmek çok önemlidir. Çünkü yapılacak hataları indirgeyerek zaman ve maliyet tasarrufu sağlar. Dolayısıyla nesne yönelimli modelleme popüler bir yaklaşımdır. Nesne yönelimli modeller diyagramlarla temsil edilir. Optimum düzeyde hazırlanmış bir model, ekipler arasındaki iletişim ve mimari olarak iyi bir yapılanmaya olanak sağlar.

Nesne Yönelimli modellemenin hedefi, tasarım sırasında sistemin detaylarını içeren bir uygulama modeli oluşturmaktır. Bu süreçte geliştiricinin, sistemin doğasında bulunan kavramları (Nesne, sınıf, özellik, fonksiyon vb.) tanımlaması gerekmektedir. İlk adımda, geliştirici sistemi net olarak organize eder. Devamında veri yapısı ve fonksiyonların detaylarını etkin bir şekilde tespitini yapar.

Nesne yönelimli modelleme aşağıda belirtilen 3 aşamadan meydana gelir:

- Sistem Analizi
- Sistem Dizaynı
- Nesne Dizaynı

İlk aşama olan sistem analizinde problem net bir şekilde belirlenir ve veriler toplanır. Analiz aşamasından sonra, istenen sistemin nasıl geliştirilmesi gerektiğine dair anlaşılır ve kesin bir durum ortaya çıkar. İkinci aşamada, sistem mimarisi tasarlanır. İlgili aşama, hem sistem analiz modeline hem de sistemin önerilen mimarisine dayalı olarak tüm sistemin alt sistemlere ayrıldığı aşamadır. Nesne tasarımı aşamasında, analiz modelinde verilen sistem detaylarının aracılığıyla sistemdeki bileşenlerin (sınıfların) yapılarına karar verilir.

Nesne yönelimli modellemenin yukarıda bahsi geçen aşamalarını tasarlamak için nesne, dinamik ve fonksiyon modelleme kullanılmaktadır. Nesne modelleme, sisteminin statik yapısını nesneler açısından tasarlar. Nesneleri, nesnelerin sınıflarını ve nesneler arasındaki ilişkileri tanımlar. Ayrıca, her bir sınıfı özelleştiren temel niteliklerini ve işlemlerini tanımlar. Nesne modelleme aşamaları aşağıdaki gibidir.

- ✓ Nesneleri tanımlanması ve sınıflara ayrılması
- ✓ Sınıflar arasındaki ilişkilerin belirlemesi
- ✓ Kullanıcı durum modeli diyagramı oluşturulması
- ✓ Kullanıcı nesne özniteliklerini tanımlanması

Sistemin statik tavrı analiz edildikten sonra dinamik davranışlarının incelenmesi gerekir. Dinamik modelleme, bir nesnenin, diğer nesneler tarafından tetiklenen sistem içi olaylar veya sistem dışından tetiklenen harici olaylar gibi durumlarda nasıl tepki göstereceğini açıklar. Dinamik modelleme süreci:

- ✓ Olayların tanımlanması
- ✓ Etkinlik diyagramları ile olayların ifade edilmesi
- ✓ Her durumun nesne özniteliklerinin tanımlanması

aşamalarından oluşmaktadır. Son olarak fonksiyonel modelleme, nesne yönelimli analizin son aşamasıdır. Fonksiyonel model, gerçekleşen işlemler ve yöntemler verilerin nasıl değiştiğini gösterir. Nesne modelleme işlemlerinin ve dinamik modelleme eylemlerinin sonuçlarını ifade eder. Fonksiyonel model, geleneksel analizin veri akış diyagramına benzemektedir.

BÖLÜM ÖZETİ

Dijital ortamlar ve uygulamalar hayatımızın vazgeçilmez bir parçası olmuşlardır. Teknolojideki hızlı gelişim ile birlikte kurum-kuruluşlar, özel şirketler, sivil toplum kuruluşları gibi birçok yapı için bahsi geçen ortamları ya da uygulamaları kullanma zorunluluğu oluşmuştur. Dolayısıyla sistem ve yazılım geliştirme süreçleri çok önemli bir hâle gelmiştir. Nesne tabanlı modelleme de yazılım geliştirme ve sistem tasarımı yaklaşımlarının temelini oluşturan bir kavramdır. Nesne yönelimli modelleme, nesne ve sınıflarının tanımlanmasıyla karakterize edilen bir görselleştirme sürecidir. Başka bir ifade ile nesne ve sınıfları temel alan bir yaklaşımdır. Bahsi geçen yaklaşımın daha hızlı yürütülmesi, daha kolay bir yapı sağlaması, tasarlanacak sistemin bakımının, güncellenmesinin ya da hata ayıklamasını kolaylaştırması, daha kısa sürede kullanılabilir uygulamalar geliştirmeye olanak sunması en önemli avantajlarındanır.

Nesne tabanlı tasarım ve modelleme süreçlerini gerçekleştirmek için bazı temel kavramları öğrenmek gerekmektedir. Nesne yönelimli tasarımın temelini sınıflar oluşturmaktadır. Kendi veri üyeleri ve üye fonksiyonları olan bir veri türüdür. Sınıf, bir nesnenin planı gibidir. Bir başka ifade ile sınıf, veri üyelerine ve üye işlevlerine sahip olan veri türüdür. Nesneye yönelik tasarımlarda, nesneler sınıflardan türetilirler. Nesne, bir sınıfın örneğidir. Sınıfın tüm veri üyelerine ve üye işlevlerine nesneler yardımıyla erişilebilir. Sınıflardan oluşturulan nesneleri tanımlamaya yarayan özelliklere öznitelik denir. Nesne yönelimli modellemede kalıtım, bir nesneyi veya sınıfı, başka bir nesnenin veya sınıfın özelliklerini kullanarak alt sınıf ya da alt nesneler oluşturma mekanizması olarak tanımlanmaktadır. Nesne tabanlı modellemenin temel kavramlarından biri olan kapsülleme, kullanıcıların sınıf, metot verilerin ne kadarına erişebileceğini belirler. Başka bir ifade ile sınıf, metot ya da verilere doğrudan erişimlerin kısıtlanması ve izin verilmesi gibi işlemleri gerçekleştirmek için kullanılır. Çok biçimlilik herhangi bir verinin birden fazla biçimde kullanılmasıdır. Çok biçimlilik, nesne yönelimli modellemede en önemli kavramlardan biridir. İlgili kavram, herhangi bir mesajın birden çok sınıftaki nesnelere gönderilebildiği bir özelliktir ve her nesne, sınıf özelliklerine bağlı olarak uygun bir şekilde yanıt verme eğilimindedir.

Nesne yönelimli modelleme, gerçek dünyanın kavramları ile organize edilen bir çözümü görselleştirme tekniğidir. Söz konusu modelleme yaklaşımı, problemleri ifade etmeye, sistemlerin çalışma süreçlerini ifade etmeye ya da genel olarak tasarlamaya yardımcı olmaktadır. Nesne Yönelimli modellemenin hedefi, tasarım sırasında sistemin detaylarını içeren bir uygulama modeli oluşturmaktır.

Nesne yönelimli modelleme 3 aşamadan meydana gelir. Bunlar; Sistem Analizi, Sistem Dizaynı ve Nesne Dizaynıdır. İlk aşama olan sistem analizinde problem net bir şekilde belirlenir ve veriler toplanır. Analiz aşamasından sonra, istenen sistemin nasıl geliştirilmesi gerektiğine dair anlaşılır ve kesin bir durum ortaya çıkar. İkinci aşamada, sistem mimarisi tasarlanır. İlgili aşama, hem sistem analiz modeline hem de sistemin önerilen mimarisine dayalı olarak tüm sistemin alt sistemlere ayrıldığı aşamadır. Nesne tasarımı aşamasında, analiz modelinde verilen sistem detaylarının aracılığıyla sistemdeki bileşenlerin (sınıfların) yapılarına karar verilir.

GÖZDEN GEÇİRELİM

1. Kitap ödünç yazılımı için bir sistem tasarlayınız ve bu sistemin sınıflarını, nesnelerini, aktörlerini belirleyiniz.
2. Sınıf diyagramları dinamik bir süreci modellemek için kullanılır (D/Y)
3. diyagramı bir nesneden diğerine mesaj akışını göstermek için kullanılırken, diyagramı bir etkinlikten diğerine mesaj akışını göstermek için kullanılır.
4. Kullanım Durum diyagramı özellikleri hakkında bilgi veriniz.
5. Erişim kısıtlaması olmayan ve her yerden erişilebilen bir nesne modelleme bileşeninin erişim belirleyicisi nedir ?

Kaynakça

- Roth, R.M., Dennis, A., and Wixom, B.H., (2013). Systems Analysis and Design, 5th Edition, John Wiley&Sons Inc.
- Tilley, S. (2019).Systems analysis and design. Cengage Learning
- <https://www.geeksforgeeks.org/software-engineering-object-modeling-technique-omt/>
- Kenneth E. Kendall Julie E. Kendall, (2010). Systems Analysis and Design,Pearson Education
- Robert Lafore, (2009). Nesne Yönelimli Programlama Kılavuzu, 4. Baskı, ALFA yayıncılık

