



Chapter 1

Java'ya

Java How to Program, 11/e

- ▶ 1.1 Tanıtım
- ▶ 1.2 Donanım ve Yazılım
 - 1.2.1 Moore Kanunu
 - 1.2.2 Bilgisayar Organizasyonu
- ▶ 1.3 Veri Hiyerarşisi
- ▶ 1.4 Makina Dilleri, Assembly ve Yüksek seviyeli diller
- ▶ 1.5 Nesne (Object) Teknolojisine Giriş
 - 1.5.1 Bir araba nesnesi
 - 1.5.2 Metodlar ve Classlar (Sınıflar)
 - 1.5.3 Instantiation (İklendirme)



- 1.5.4 Yeniden Kullanım (Reuse)
- 1.5.5 Mesajlar ve Metod çağırımları (Method calls)
- 1.5.6 Attributeler ve Instance Variablelar (Örnek Değişkenleri)
- 1.5.7 Encapsulation ve Information Hiding (Veri Saklama)
- 1.5.8 Inheritance (Kalıtım)
- 1.5.9 Interfaceler
- ▶ 1.5.10 Object Oriented Analysis and Design
- ▶ The UML (Unified Modelling Language)

1.1 Tanıtım

- ▶ Java dünyada en çok kullanılan programlama dillerinden biridir.
- ▶ Java ile günümüzün popüler programlama metodolojilerinden biri olan nesneye dayalı (object oriented programlama metodolojisini öğreneceksiniz.
- ▶ Bir çok yazılım nesnesi üreterek bunlar üzerinde çalışacaksınız.

1.1 Introduction (Cont.)

- ▶ Çoğu kurum, Java dilini tercih etmektedir.
- ▶ Java aynı zamanda Internet tabanlı uygulamalarda ve ağ üzerinden haberleşen cihazların yazılımlarında çok yaygın bir şekilde kullanılmaktadır.
- ▶ Oracle'a göre kurumsal bilgisayarların %97'sinde, kişisel bilgisayarların %89'unda Java kullanılmakta ve 9 milyondan fazla Java developer'ı bulunmaktadır.

Devices

Airplane systems	ATMs	Automobile infotainment systems
Blu-ray Disc™ players	Cable boxes	Copiers
Credit cards	CT scanners	Desktop computers
e-Readers	Game consoles	GPS navigation systems
Home appliances	Home security systems	Light switches
Lottery terminals	Medical devices	Mobile phones
MRIs	Parking payment stations	Printers
Transportation passes	Robots	Routers
Smart cards	Smart meters	Smartpens
Smartphones	Tablets	Televisions
TV set-top boxes	Thermostats	Vehicle diagnostic systems

Fig. 1.1 | Some devices that use Java.

1.1 Giriş (Dvm.)

Java Standard Edition (11)

- ▶ **Java Standard Edition** masaüstü ve sunucu uygulamalarını üstlenmektedir.
- ▶ Java SE 8'den önce Java üç farklı programlama paradigması (türü) desteklemekteydi: yordamsal (procedural) programlama, nesneye dayalı programlama ve generic programlama. Java SE 8 ile fonksiyonel programlama da eklenmiştir.



1.1 Giriş (Dvm.)

Java Enterprise Edition

- ▶ The **Java Enterprise Edition (Java EE)** geniş ölçekli, dağıtık, ağ uygulamalarını ve web tabanlı uygulamaları desteklemektedir.



1.1 Giriş(Dvm.)

- ▶ Java Micro Edition (Java ME)
 - Kısıtlı gömülü sistemlerde çalışma için oluşturulmuştur.
 - Akıllı saatler
 - MP3 oynatıcılar
 - Akıllı sayaçlar
 - v.b.

1.2 Donanım ve Yazılım

- ▶ Bilgisayarlar insanlardan çok daha hızlı bir şekilde hesaplamalar gerçekleştirebilmekte ve mantıksal işlemler yapabilmektedir.
- ▶ Günümüz kişisel bilgisayarları bir saniye içerisinde milyarlarca işlem gerçekleştirebilmektedir (Bir insanın hayatı boyunca yapabileceğinden bile fazla ☺).
- ▶ *Superbilgisayarlar şimdiden saniyede binlerce trilyon (katrilyon) işlem gerçekleştirebilmektedir!*
- ▶ Bilgisayarlar verileri bilgisayar programı adı verilen programlar yoluyla işleyebilmektedir.



1.2 Donanım ve Yazılım (Dvm.)

- ▶ Bu programları bildiğiniz gibi programcılar geliştirmektedir.
- ▶ Bu ders kapsamında programcının üretimini arttıran dolayısıyla yazılım gelişim sürecindeki maliyeti düşüren nesneye dayalı programlama kavramını öğreneceksiniz.



1.2 Donanım ve Yazılım (Dvm.)

- ▶ Bir bilgisayar donanım olarak tanımlanan çeşitli aygıtlardan oluşmaktadır (klavye, fare, hard disk, memory, CPU)
- ▶ Hesaplama maliyetleri yazılım ve donanımdaki hızlı gelişmeyle birlikte çok büyük miktarda düşmüştür.
- ▶ Eskiden bilgisayarlar büyük bir oda büyüklüğünde ve fiyatı milyar dolar cinsindendi. Silikon-çip teknolojisiyle artık bilgisayar üretimi çok az maliyete gerçekleştirilebilmektedir.

1.2 .1 Moore Kanunu

- ▶ Her bir ya da iki yılda bir bilgisayarların kapasitesi iki katına çıkacaktır. Bu kanun Intelin kurucu ortaklarından Gordon Moore tarafından ortaya atılmıştır.
- ▶ Moore'ın kanunu bilgisayarın belleği, depolama kapasitesi ve hızı ile ilgilidir.
- ▶ Benzer bir devrim iletişim alanında da gerçekleştirilmiş olup bu gelişmeler bilgi devrimini hızlandırmışlardır.



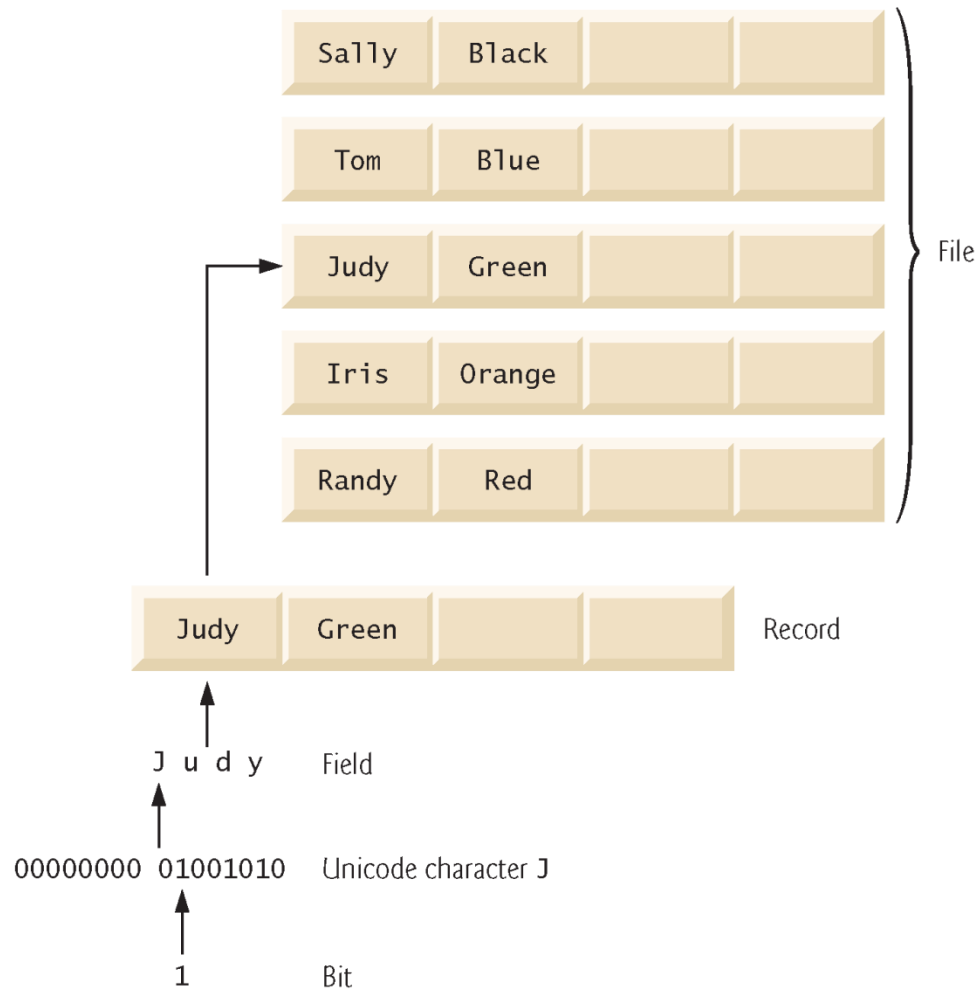
1.2.2 Bilgisayar Organizasyonu (Computer Organization)

- ▶ Bilgisayarlar çeşitli mantıksal birimler olarak tasavvur edilebilir.
- ▶ Giriş birimleri, Çıkış birimleri, Bellek birimi, Aritmetik ve mantıksal birim, ikincil bellek birimi.



1.3 Veri Hiyerarşisi (Data Hierarchy)

- Bilgisayarlar tarafından işlenen veri elemanları en basitinden (bitlerden) çok daha karmaşığa doğru gitmektedir (karakterler ve alanlar).





1.3 Veri Hiyerarşisi (Dvm.)

- ▶ Dosyalar
- ▶ Bir dosya, birbiriyle ilgili kayıt grubudur.
- ▶ Daha genel olarak, bir dosya rasgele biçimlerdeki keyfi verileri içerir.
- ▶ Bazı işletim sistemlerinde, bir dosya basitçe bir bayt dizisi olarak görülür; dosyaları bir bayt dizisinde herhangi bir kayıta düzenleme gibi, uygulama programcısı tarafından oluşturulan bir görünümdür.

1.3 Veri Hiyerarşisi (Dvm.)

Database (Veri Tabanı)

- ▶ **Veritabanı**, kolay erişim ve manipülasyon için düzenlenmiş bir veri koleksiyonudur.
- ▶ En popüler veritabanı modeli, verilerin basit tablolarda saklandığı ilişkisel veritabanıdır.
- ▶ Bir tablo kayıtları ve alanları
 - ▶ Örneğin, bir öğrenci tablosu, ad, soyad, büyük, yıl, öğrenci kimlik numarası ve not ortalamaları alanlarını içerebilir.
 - ▶ Her bir öğrencinin verileri bir kayıttır ve her kayıttaki tek tek bilgi alanlarıdır.
 - ▶ Verileri, birden çok tablo veya veri tabanıyla olan ilişkisine dayanarak, arama, sıralama ve başka şekilde manipüle edebilirsiniz.

1.3 Veri Hiyerarşisi (Dvm.)

Big Data (Büyük Veri)

- ▶ Dünya genelinde üretilen veri miktarı muazzam bir şekilde artmaktadır.
- ▶ IBM'e göre günlük olarak yaklaşık 2,5 quintillion bayt (2,5 exabytes) veri oluşturuluyor ve dünya verisinin % 90'ı son iki yılda oluşturuldu! (www.ibm.com/software/data/bigdata/)
- ▶ **Big Data** uygulamaları büyük miktarda veri ile uğraşıyor ve bu alan hızlı bir şekilde büyüyor ve yazılım geliştiricileri için çok fazla fırsat yaratıyor.



Unit	Bytes	Which is approximately
1 kilobyte (KB)	1024 bytes	10^3 (1024 bytes exactly)
1 megabyte (MB)	1024 kilobytes	10^6 (1,000,000 bytes)
1 gigabyte (GB)	1024 megabytes	10^9 (1,000,000,000 bytes)
1 terabyte (TB)	1024 gigabytes	10^{12} (1,000,000,000,000 bytes)
1 petabyte (PB)	1024 terabytes	10^{15} (1,000,000,000,000,000 bytes)
1 exabyte (EB)	1024 petabytes	10^{18} (1,000,000,000,000,000,000 bytes)
1 zettabyte (ZB)	1024 exabytes	10^{21} (1,000,000,000,000,000,000,000 bytes)

Fig. 1.4 | Byte measurements.



1.5 Nesne Teknolojisine Giriş (Introduction to Object Technology)

- ▶ Nesneler, veya daha doğrusu, nesnelerin geldiği sınıflar, aslında yeniden kullanılabilir yazılım bileşenleridir.
- ▶ Tarih nesneleri, zaman nesneleri, ses nesneleri, video nesneleri, otomobil nesneleri, insan nesneleri vb. vardır.
- ▶ Neredeyse herhangi bir isim, nitelikleri (ör. Ad, renk ve boyut) ve davranışları (örn. Hesaplama, taşıma ve iletişim) açısından bir yazılım nesnesi olarak makul bir şekilde temsil edilebilir.
- ▶ Yazılım geliştirme grupları, "yapısal programlama" gibi daha popüler tekniklerden ziyade genellikle anlaşılması, düzeltilmesi ve değiştirilmesi daha kolay olan ve daha çok üretken olmak için nesne yönelimli modüler bir tasarım ve uygulama yaklaşımı kullanabilir.

1.5.1 Bir nesne (Object) olarak otomobil

- ▶ Bir nesne olarak otomobil
 - Basit bir benzerlik ile başlayalım.
 - Bir araba kullanmak ve gaz pedalına basarak daha hızlı gitmesini istediğinizi varsayalım.
 - Araba sürmeden önce, birisi bunu tasarlamalıdır.
 - Bir araba tipik olarak bir evin tasarımını tasvir eden planlara benzer mühendislik çizimleri olarak başlar.
 - Bu çizimler, bir gaz pedalı tasarımını içerir
 - Frenpedalı, arabayı yavaşlatan mekanizmaları sürücünden gizler, gaz pedalına basıldığında arabayı daha hızlı hale getiren karmaşık mekanizmaları da gizler ve direksiyon ise, otomobili döndüren mekanizmaları sürücünden gizler.

1.5.1 Bir nesne (Object) olarak otomobil (Dvm.)



- Bu pedallar motorların, frenleme ve direksiyon mekanizmalarının nasıl çalıştığını az bilen veya hiç bilmeyenler için bir otomobili kolayca kullanmaya olanak tanır.
- Bir araba kullanmadan önce, onu tarif eden mühendislik çizimlerinden yapılmalıdır.
- Üretilmiş bir otomobil, daha hızlı gitmesini sağlamak için gerçek bir gaz pedalına sahiptir, ancak bu bile yeterli değildir - araba kendi kendine hızlanmayacaktır (umarım!), sürücü hızlandırmak için pedala basmalıdır.



1.5.2 Methods (Metotlar) and Classes (Sınıflar)

- ▶ Bir programda bir görevi yerine getirmek için bir **metod** gerekir.
- ▶ **Metod** aslında görevlerini yerine getiren program ifadelerini içerir.
- ▶ Bu ifadeleri kullanıcıdan gizler, tıpkı bir arabanın gaz pedalından arabadan daha hızlı gitmesini sağlayan mekanizmalar sürücüden gizlenir.
- ▶ Java'da, bir sınıf oluşturduk aslında. Sınıfın görevlerini yerine getiren metod setini barındıracak bir sınıf olarak adlandırılan bir program birimi oluşturduk.

1.5.3 Instantiation (İklendirme)

- ▶ Tıpkı bir arabayı sürmeden önce mühendislik çizimlerine sadık kalınarak somut bir arabanın oluşturulması gerektiği gibi, bir programın sınıfın yöntemlerinin tanımladığı görevleri yerine getirebilen için bir sınıf nesnesi oluşturmanız gerekir.
- ▶ Bir nesne (object) sınıfının bir örneği olarak anılır.

1.5.4 Reuse (Yeniden Kullanım)

- ▶ Tıpkı bir otomobilin mühendislik çizimleri birçok otomobil üretmek için birçok kez tekrar kullanılabilir gibi, birçok nesneyi oluşturmak için bir sınıfı birçok kez tekrar kullanabilirsiniz.
- ▶ Yeni sınıflar ve programlar oluştururken mevcut sınıfların yeniden kullanımı zaman ve çaba tasarrufu sağlar.
- ▶ Mevcut sınıflar ve bileşenler çoğu zaman kapsamlı test, hata ayıklama ve performans ayarlama işlemlerine tabi tutulduğundan, yeniden kullanım, daha güvenilir ve etkili sistemler oluşturmanıza yardımcı olur.
- ▶ Değiştirilebilir parçalar fikri Endüstriyel Devrim açısından çok önemli olduğu gibi, nesne teknolojisi tarafından harekete geçirilen yazılım devrimi için yeniden kullanılabilen sınıflar çok önemlidir.



Software Engineering Observation 1.1

Use a building-block approach to creating your programs. Avoid reinventing the wheel—use existing high-quality pieces wherever possible. This software reuse is a key benefit of object-oriented programming.



1.5.5 Mesajlar ve Method Çağırımları (Message Calls)

- ▶ Bir otomobil kullandığınızda, gaz pedalına basmak, bir görevi yerine getirmek için arabaya bir mesaj gönderir. Benzer şekilde, bir nesneye mesajlar gönderiyorsunuz.
- ▶ Her ileti, nesnenin bir methoduna görevi yerine getirmesini söyleyen bir method çağrısı olarak uygulanır.

1.5.6 Attributeler ve Instance Variables

- ▶ Bir arabanın özellikleri (attributeleri)
- ▶ Renk, kapı sayısı, tankındaki gaz miktarı, mevcut hızı ve toplam mil kaydını (yani kilometre sayacı okuması).
- ▶ Aracın özellikleri, mühendislik şemalarında tasarımının bir parçası olarak gösterilir.
- ▶ Her araba kendi özelliklerini koruyor.
- ▶ Her araba, kendi benzin deposunda ne kadar gaz olduğunu bilir, ancak diğer araçların depolarında ne kadar olduğu bilinmez.



1.5.6 Attributeler ve Instance Variables (Dvm.)

- Bir nesne, normalde taşıdığı niteliklere programda da sahiptir.
- Nesnenin bir parçası olarak belirtilir. Örneğin, bir banka hesabı nesnesinin, hesaptaki para miktarını temsil eden bir bakiye özelliği vardır.
- Ancak, her banka hesabı nesnesi, temsil ettiği hesaptaki bakiyeyi biliyor ancak bankadaki diğer hesapların bakiyelerini bilmiyor.
- Özellikler, sınıfın örnek değişkenleri (instance variable) tarafından belirtilir.



► Encapsulation ve Inheritance

1.5.7 The UML (Unified Modeling Language)

- ▶ The Unified Modeling Language (UML), nesneye dayalı sistemleri modellemek için en yaygın kullanılan grafik şemadır.

