



**Java ile Nesne Merkezli Programlamaya Giriş**

# **1. Bölüm**

## **Java'ya Giriş**

Akın Kaldıroğlu

**[www.javaturk.org](http://www.javaturk.org)**

Aralık 2016

# Küçük Ama Önemli Bir Konu

- Bu dosya ve beraberindeki tüm, dosya, kod, vb. eğitim malzemelerinin tüm hakları **Selsoft Yazılım, Danışmanlık, Eğitim ve Tic. Ltd. Şti.**'ne aittir.
- Bu eğitim malzemelerini kişisel bilgilenme ve gelişiminiz amacıyla kullanabilirsiniz ve isteyenleri <http://www.selsoft.academy> adresine yönlendirip, bu malzemelerin en güncel hallerini almalarını sağlayabilirsiniz.
- Yukarıda bahsedilen amaç dışında, bu eğitim malzemelerinin, ticari olsun/olmasın herhangi bir şekilde, toplu bir eğitim faaliyetinde kullanılması, bu amaca yönelik olsun/olmasın basılması, dağıtılması, gerçek ya da sanal/Internet ortamlarında yayınlanması yasaktır. Böyle bir ihtiyaç halinde lütfen benimle, **akin.kaldiroglu@selsoft.academy** adresinden iletişime geçin.
- Bu ve benzeri eğitim malzemelerine katkıda bulunmak ya da düzeltme ve eleştirilerinizi bana iletmek isterseniz çok sevinirim.
- **Böl Java'lı günler dilerim.**

# İçerik

- Bu bölümde şu konular ele alınacaktır:
  - Java'nın Kısa Bir Tarihi
  - Java Nedir?
  - Java'nın Diğer Özellikleri
  - Java Sertifikaları
  - Dünyada ve Türkiye'de Java
  - Sorulmaması Gereken Sorular

## Java'nın Kısa Bir Tarihi

## Java'nın Kısa Tarihi

- 1991'de Sun, Patrick Naughton, Mike Sheridan ve James Gosling önderliğindeki küçük bir takımla bir proje başlattı.
- 13 kişiden oluşan ve “**Green Team**” (**Yeşil Takım**) adlı bu grubun amacı ITC dünyasında geleceği tahmin edip ona yönelik AR-GE yapmaktı.
- Ve bu takım da geleceğin, kişisel ve ev elektronik cihazlarında olduğunu öngörüp bu alana yöneldi.



Green Project Team.

\*7

- 18 aylık hummalı bir çalışmadan sonra 1992 yazında “\*7”, “Star Seven” isimli ve dokunmalı ekrana sahip bir kontrol cihazı geliştirdirler.
- Bu cihaz TV, video oynatıcısı ve müzik seti gibi pek çok ev cihazını kontrol edebiliyordu.
- Cihazın bu yeteneği, platformdan bağımsız bir dile sahip olmasından kaynaklanıyordu.
- James Gosling bu dile “Oak” adını vermiştir.



# Java ve Internet

- Bu proje pek çok kablolu TV şirketine teklif edildi ama zamanının ilerisinde olduğundan kabul görmedi.
- Ama takımın ileri gelenleri o sırada, geliştirdikleri altyapının Internet'e çok uygun olduğunu farkettiler.
- Ve hemen yönlerini değiştirip, ismi daha sonra resmi olarak **HotJava** olan Java-tabanlı bir tarayıcı geliştirdiler: **WebRunner**



# İç Giyim Olarak Oak

➤ Gosling explains: "*We had already been developing the kind of 'underwear' to make content available at the same time the Web was being developed. Even though the Web had been around for 20 years or so, with FTP and telnet, it was difficult to use. Then Mosaic came out in 1993 as an easy-to-use front end to the Web, and that revolutionized people's perceptions. The Internet was being transformed into exactly the network that we had been trying to convince the cable companies they ought to be building. All the stuff we had wanted to do, in generalities, fit perfectly with the way applications were written, delivered, and used on the Internet. It was just an incredible accident. And it was patently obvious that the Internet and Java were a match made in heaven. So that's what we did.*"

## Oak'tan Java'ya

- Oak ismi bu sırada **Java** olarak değiştirildi,
- Ve **23 Mayıs 1995**'te Java resmi olarak piyasaya sunuldu.
- Ve o tarihlerde ne Java'yı geliştirenlerin ne de bir başkasının tahmin edemeyeceği yoğunlukta bir ilgiyle karşılaştı.
- 2010 yılında Oracle'ın Sun'ı satın almasıyla da Java el değiştirmiş oldu.



# Java Nedir?

# Java Nedir?

- Sun, 1995'te Java'yı sunarken yayınladığı ve "Java'nın babası" James Gosling'in kaleme aldığı bir yazında Java'yı şöyle tanıtıyordu:
  - **Java: Basit, nesne-merkezli, ağlarda yetenekli, yorumlanan, sağlam, güvenli, mimari olarak tarafsız, taşınabilir, yüksek performanslı/başarılı, çok kanallı, dinamik bir dil.**
  - **Java: A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language.**

# Java Basittir I

- Java'nın o kadar da basit değildir aslında,
  - Java, ataları olan C ve C++ dillerinden daha basittir.
- Çünkü bu dillerde olan ve karmaşıklık getiren özelliklerini ve yapılarını içermez:
  - Pointer aritmetiği,
  - Programcıya bağımlı bellek yönetimi (memory management),
  - İşlemci çoklu kullanımı (operator overloading),
  - Platform bağımlı yapılar.
- Java soyut dolayısıyla daha yüksek seviyeli bir dildir.

## Java Basittir II

- Java bu dillere göre aşağıdaki avantajlara da sahiptir:
  - Standartlar üzerine kurulmuş olmak,
  - Platforma bağlı olmayan yapılara (örneğin basit veri yapıları) sahip olmak,
  - Daha derli-toplu geliştirme ortamlarına sahip olmak,
  - Temel dil dışında da kurumsal ve gömülü uygulama yapılарına sahip olmak
- Başlarda Java, küçük, sade ve güçlü olmayı tercih etti,
  - Ama özellikle 5 sürümüyle başlayan eklemelerle Java bayağı büydü.
    - Generic yapıları,
    - 8 sürümüyle gelen fonksiyonel yapılar.

# Java Nesne Merkezlidir I

- Java nesne-merkezli bir dildir.
- Nesne-merkezli diller, daha çok aşina olunan prosedürel/yordamsal dillerden çok farklıdır.
- Yordamsal dillerde en önemli kavram/soyutlama yordam/prosedür/fonksiyondur ve bu yapılar, bir problemi alt problemlere bölüp, her birini adım adım (step-wise decomposition) tanımlamakta kullanılır.
- Fakat nesne-merkezli diller tamamen nesne kavramı üzerine otururlar.

# Nesne Nedir?

- Nesne, insan zihninin kendisine yöneldiği, özellik ve davranışlara sahip, fiziksel olan ya da olmayan herhangi bir olgudur.
- Yazılımda her tip nesne için, verisi ile davranışını bir paket haline getirip **sarmalayan** ya da **kapsülleyen** (**encapsulation**) ve adına genelde **Sınıf (class)** denen kalıplar (şablonlar) oluşturulur.

**Sınıf = Veri + Davranış**

- Nesneler, şablonlardan yani sınıflardan üretilmiş çalışma zamanı yapılarıdır.

# Nesne ve Varlıklar/Olgular

- Nesneler gerçek dünyadaki varlıkları ya da olguları ifade ederler
  - Nesnelerin özellikleri gerçek varlıkların/olguların özelliklerine,
  - Nesnelerin davranışları ise varlıkların/olguların birbirleriyle olan ilişşimine,
- karşılık geldiği söylenebilir.

## Java Nesne Merkezlidir II

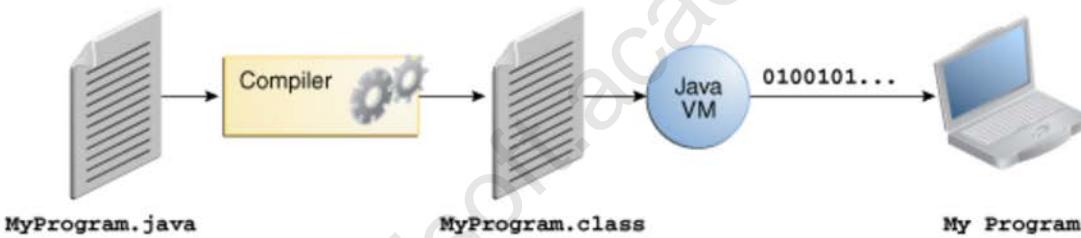
- Java'da basit veri tipleri hariç hersey, bütün tipler, bir nesnedir.
- Çok güçlü **sarmalama-kapsülleme (encapsulation)** ve **bilgi saklama (information hiding)** mekanizmaları vardır.
- **Arayüz (interface)** ile **gerçekleştirme (implementation)** arasında çok açık bir ayırım vardır.

# Java Ağlarda Yeteneklidir

- Tarihi olarak Java'nın ve Web devriminin doğumunu aynı döneme rast gelmiştir.
- Bu yüzden Java, ağ ve ilgili yapılar akılda tutularak tasarlanmış ilk dildir.
- Java ile bilgisayar için değil ağ için yazılım geliştirirsiniz.
  - HTTP ve HTTPS gibi en yaygın Web protokolleri
  - Web Servisleri
  - Uzak nesneler (remote objects)
  - Ve Telnet, FTP vs. gibi diğer bütün protokoller

# Java Yorumlanır

- Aslında Java hem **derlenen (compiled)** hem de **yorumlanan (interpreted)** bir dildir.
- Platformdan bağımsız olabilmesi için Java kaynak kodları doğrudan makina koduna derlenmez,
  - Java kaynak kodları (**.java dosyaları**), Java **bytecode**una (**.class dosyaları**) derlenir.
  - **Bytecode**, JVM'in komutlarıdır ve çalışma zamanında (run-time) JVM tarafından yorumlanır.
  - **JVM (Java Virtual Machine** ya da **Java Sanal Makinası**), donanım ve işletim sistemi üzerinde çalışan sanal bir platformdur.

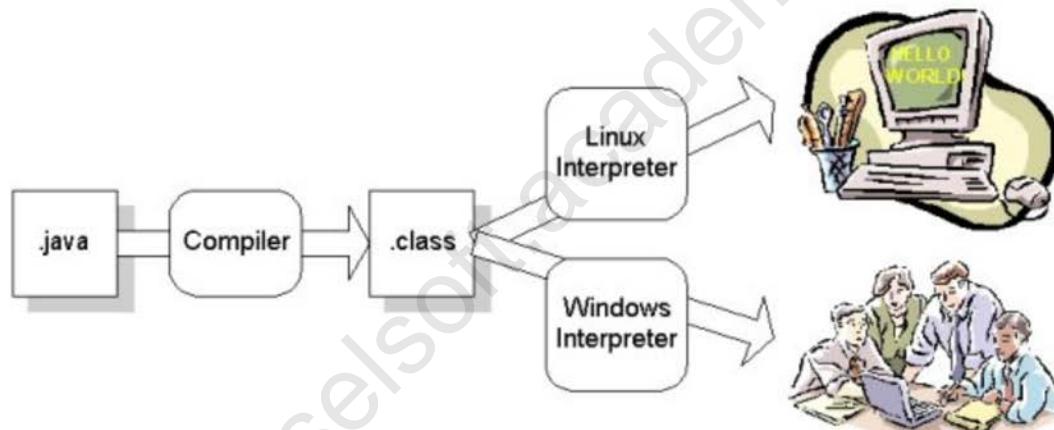


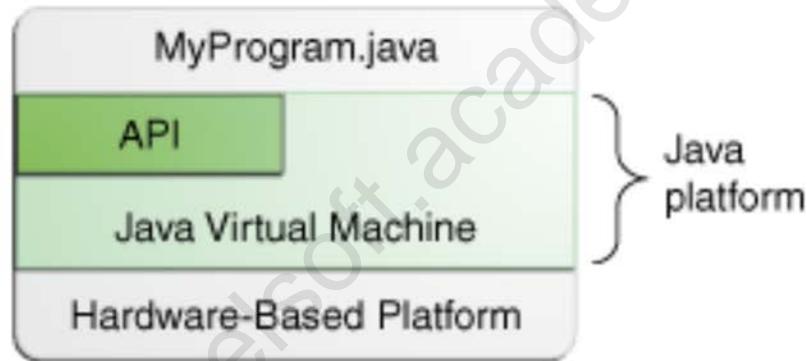
# Bytecode Nedir?

- Bytecode, JVM'in makina dilidir,
  - Bir byte uzunlığında olduğu için böyle adlandırılmıştır.
- Bytecode, herhangi bir işlemciye özel değildir ya da bir başka deyişle tamamen ve sadece JVM'e özeldir,
- Bytecodelar, JVM tarafından çalışma zamanında (run-time) platforma özel komutlara çevrilir:
  - Yorumlama
- Bu şekilde Java'nın platformdan bağımsız ya da taşınabilir olması sağlanır.

# JVM

- **JVM (Java Virtual Machine)** ya da **Java Sanal Makinası**, donanım ve işletim sistemi üzerinde çalışan sanal bir platformdur.
- Hemen her platform için en az bir JVM mevcuttur.
- Derlenmiş Java kodlarıyla platform arasında bir katman oluşturur ve aynı bytecode'ların farklı platformlarda değişmeden, aynı şekilde çalışmasını sağlar.





```
public class Selam {  
    private String world = "millet";  
  
    public String selamSoyle(String kime) {  
        String cumle;  
        if(kime != "")  
            cumle = "Selam " + kime + " :)";  
        else  
            cumle = "Selam " + world + " :)";  
  
        return cumle;  
    }  
}
```

```
public class SelamTest {  
  
    public static void main(String[] args) {  
        Selam nesne = new Selam();  
        String cevap = nesne.selamSoyle("Ali");  
        System.out.println(cevap);  
    }  
}
```

# Java Taşınabiliridir

- Java tanımlı **standartlar** ve **şartnameler (specifications)** üzerine kurulmuştur.
  - Bu açıdan Java'da gerçekleştirmeye (implementation) bağlı hiçbir noktanın olmaması amaçlanmıştır.
    - Örneğin int veri tipi her yerde 32 bittir.
- Ayrıca, Java dili ve JVM, platformlarla alakalı sadece en genel ön kabullere sahiptir.
- Bu durum Java'nın *platform bağımsızlığı* ve *taşınabilirliğinin* sağlayıcısıdır ve Java programcısına üzerinde çalıştığı ortamdan bağımsız davranışa gücünü verir.

## WORA

- Başından bu yana **bir kere yaz her yerde çalıştır (Write Once, Run Anywhere, WORA)** Java'nın taşınabilirlik konusundaki hedefidir.
- Tabi olarak bu prensibin geçerli olması için programcı da platforma özel kod yazmamalıdır.
- Java'yı uzunca süredir uygulayanlar, **bir kere yaz her yerde test et ↗ (Write Once, Test Anywhere, WOTA)** prensibinin daha doğru olduğu üzerinde hemfikirdirler.

# Java Sağlamdır

- **Sağlamlık (robustness)** çalışma zamanı güvenirliği olarak tanımlanır.
- Java, dayanıklı (reliable) ve sağlam yazılımlar üretmek için tasarlanmıştır.
  - Java güçlü bir tip sistemine sahiptir dolayısıyla hem derleme zamanında hem de çalışma zamanında tip uyumu ve çevrimi (cast) kontrol edilir,
  - Sıra dışı durumları ele almak için mekanizma oluşturulmuştur (exception handling),
  - Bellek yönetimi otomatiktir, programcıya bırakılmamıştır.
  - Ve daha pek çok derleme/çalışma zamanı kontrollerini içerir.

# Java Güvenlidir

- Java'nın ağ ortamlarında kullanılmak için tasarlanmış olduğu belirtilmiştir.
- Bu yüzden Java, ağ ortamlarında bulunmanın getirdiği tehlikelere karşı koyacak şekilde yapılandırılmıştır.
- Yabancı makinalardaki nesnelerin, yerel makinadaki nesnelere ve işletim sisteme erişimi kontrol altındadır.
  - Java'nın ilk yıllarda duyulan güvenlik açıklarına artık pek rastlanılmamaktadır.
  - Java'daki güvenlik API'ları sayesinde şifrelemeden (encryption) sayısal imzalara (digital signatures) kadar her türlü güvenlik kodu yazılabilir.

# Java Mimari Olarak Tarafsızdır

- Java, birbirlerinden farklı, pek çok değişik platformun bulunduğu ağ üzerinde çalışmak için tasarlandığından, herhangi bir platformu varsaymaz, platformlardaki ortak özellikler dışında herhangi bir ön kabulü yoktur.
- Bu durum Java'nın sanal bir makina üzerinde çalışması ve hiçbir platforma varsayımayan bir şekilde, standartlar üzerine inşa edilmesinin bir sonucudur.
- Java bytecode da herhangi bir platforma bağımlı değildir.

# Java Yüksek Başarımlıdır I

- Bu özelliği tabiki açıklanmaya muhtaçtır.
- Nesne-merkezli diller, tip hiyerarşileri ve polimorfizm gibi özelliklerinden dolayı zaten çalışma zamanında çok iş yaparlar.
  - Java, taşınabilir ve mimari olarak tarafsız olmak için sanal makina gibi bir ara katman üzerinde çalışmayı tercih etmiştir,
  - Ayrıca Java sağlam bir çalışma zamanı kurgulamak için çalışma zamanında daha pek çok kontrollere sahiptir.
  - Java'nın soyut olan yapısı da makina arasında daha fazla katmana sahip olmasına sebep olmaktadır.
  - Nesnelerin bellekten temizlenmesi (Garbage Collection, GC) de ciddi CPU gücü gerektiren bir işlemidir.

## Java Yüksek Başarımlıdır II

- Bu ve benzeri durumlar Java'nın çalışma zamanı performansını etkilemektedir.
- Fakat, Java'nın çalışma zamanı performansını artttırmak için çok mesafe katedilmiştir.
  - Etkin derleyiciler (Optimizing compiler) – HotSpot, JRockit, vd.
  - Etkin JIT ve benzeri teknolojiler
- Java'cılar da performanslı kod yazmak için hem hız hem de bellek tüketimi açısından, diğer dilleri kullananlara göre daha bilgili olmalıdır.

## Java Yüksek Başarımlıdır III

- Ama unutulmamalıdır ki projeler daha çok ihtiyaçların yanlış kavranması vs. gibi sebeplerden başarısız olmaktadır, performanstan değil.
- Ayrıca performans ve ölçeklenebilirlik (scalability) genel olarak mimarının bir özelliğidir ve mimariler bu konulardaki hedefler gözönüne alınarak tasarlanmalıdır.

# Java Çok Kanallıdır

- **Kanal (thread)**, bir işlem/süreç (process) içinde oluşturulan küçük bir işlem/süreçtir (lightweight process).
- Aynı işlem içinde çalışan farklı kontrol akışlarına *kanal* denir.
- Kanallar aynı işlem içinde çalışıklarından, birçok işi aynı anda ve daha hızlı ve etkin olarak yerine getirebilirler.
  - Böylece daha yetenekli ve hızlı cevap veren yazılımlar geliştirilebilir.
- CPU ve platformların çok hızlı hale gelmesi, dillerdeki çok kanallı özellikleri kullanmamızı gerektirmektedir.

# Java Dinamiktir

- **.class** dosyalarındaki Java tiplerinin daha program derlenirken hazır olmaları şart değildir, tipler JVM'e çalışma zamanında herhangi bir anda yüklenebilir.
- Dolayısıyla pek çok dilde var olan derleme zamanındaki statik bağlantı kurma (static linking) genel olarak Java'da yoktur.
- Bu da ağ üzerinde çalışan ve derleme sırasında erişilmesi mümkün olmayıp, çalışma zamanında ulaşılan nesnelerin dinamik olarak JVM'e yüklenebilmesini sağlar.

## Java'nın Diğer Özellikleri

# Java'nın Sahibi Kimdir?

- Excel'in sahibinin Microsoft olması anlamında Java'nın sahibi yoktur.
- 2010 yılına kadar Java'nın orijinal geliştiricisi olan Sun, Java'nın sahibi ve lisanslayıcısıydı.
- 2010 yılında Oracle Sun'ı satın alarak, Java'nın sahibi ve lisanslayıcısı oldu.

# Java'yi Kim Geliştirmektedir?

- Java, JCP (Java Community Process, <http://www.jcp.org>) tarafından geliştirilmektedir.
- JCP, kişi ve kuruluşların serbestçe katılabildiği ve kararların oy çoğunluğu ile verildiği milletlerarası bir ortamdır.
- Çok farklı bilgi ve tecrübe sahip binlerce kişi ve kurum JCP çatısın altında Java'nın gidişatını belirlerler.
- Her ne kadar Oracle (eskiden Sun), JCP'de çok etkin olsa ve bu yüzden eleştirilse de Java'nın geleceği çok daha demokratik yollarla belirlendiği söylenebilir.

## JSR I

- JCP'nin üyeleri yeni bir Java teknolojisini *Java Specification Request (JSR)* mekanizması ile teklif edebilirler.
- Her JSR'ın bir yürütücü heyeti ve teknik bir lideri vardır.
- JSR'lar, içinde halka açık incelemenin de (public review) bulunduğu bir dizi safhadan geçip demokratik bir oylama ile sonuçlandırılırlar.

## JSR II

- Kabul edilen her JSR aşağıdaki üç şeyi üretmek zorundadır:
  - API'ı da içeren bir şartname (specification)
  - Bir referans gerçekleme (reference implementation)
  - Teknoloji uyumluluk paketi (technology compatibility kit)
- Dolayısıyla her Java teknolojisi gerçekleştirmesi, bu açık ve üzerinde mutabakat olmuş tarifname ve API'lar üzerine bina edilirler.
- İsteyen her kişi ve kurum da bu tarifnameler üzerine ürün geliştirip, bedava ya da ücretiyle lisanslayabilir.

# Özgür Yazılım

- Java'nın bu açık ve standartlara dayalı yapısı özgür yazılımların gelişmesine de yardımcı olmaktadır.
- Java bugün açık kaynak kodlu (open source) yazılım felsefesine gönül verenlerin en fazla kullandıkları dil haline gelmiştir.
- Bu sayede hemen her Java teknolojisinin ticari olmayan, açık kaynak kodlu gerçeklestirmesi mevcuttur.

# Java'nın Tipleri

- Sun, tek bir tipin yeterli olmayacağılığını düşünerek, Java'nın üç farklı tipini oluşturmuştur:
  - Standart Java (Standard Edition, SE)
  - Mikro Java (Micro Edition, ME)
  - Kurumsal Java (Enterprise Edition, EE)
- Şu anda
  - Java SE'nin 8. sürümü vardır, 9.si 2017'ye planlanmıştır,
  - Java EE'nin 7. sürümü vardır, 8.si 2018'ye planlanmıştır,
  - Java ME'nin 8. sürümü vardır.
- [http://en.wikipedia.org/wiki/Java\\_version\\_history](http://en.wikipedia.org/wiki/Java_version_history)

# “Java” Derken Neyi Kastediyoruz?

- Java
  - Bir programlama dilidir,
  - Bir geliştirme ortamıdır,
  - Bir uygulama ortamıdır,
  - Bir kurulum ortamıdır.
- Bu anlamda Java, klasik programlama dillerinden farklı olarak, çok geniş ama yeknesak bir yazılım geliştirme eko-sistemi sunar.

# Java'ya Nasıl Başlıram?

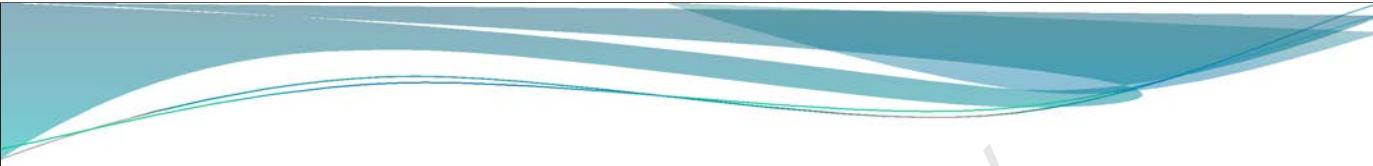
- Java'ya <http://www.oracle.com/technetwork/java/index.html> adresine gidip JDK'in en son sürümünü indirip makinaniza kurarak başlayabilirsiniz.
- JDK'i kurduğunuz yerdeki *bin* klasörünü *path*'e koyup, komut satırı (CMD, terminal, etc.) *javac* ile kodunuzu derleyip, *java* ile de çalıştırabilirsiniz.
- İsterseniz [Netbeans](http://www.netbeans.org) (<http://www.netbeans.org>) ya da [Eclipse](http://www.eclipse.org) (<http://www.eclipse.org>), [IntelliJIDEA](http://www.jetbrains.com/idea/) (<http://www.jetbrains.com/idea/>) gibi geliştirme ortamlarını da kullanabilirsiniz.

# Java'nın Tabiatı ve Zorlukları I

- Java *nesne-merkezli*dir, dolayısıyla artık prosedürel değil nesne merkezli düşünmeniz gereklidir.
- Java, sürükle-bırak merkezli programlama yaklaşımından ziyade, C/C++ geleneğine daha uygun olarak, ne yazdığını tam olarak bilmenizi ister.
- Java *şartname (specification)* temelliidir:
  - Dolayısıyla şartname ve onu gerçekleştiren farklı ürünlerin olduğunu anlamınız gereklidir.
  - Dolayısıyla aynı işi yapan, aynı API'ye sahip değişik ürünler vardır.

## Java'nın Tabiatı ve Zorlukları II

- Java, yazılım mimarilerine, tasarım şablonlarına (design patterns) önem verir ve onları kullanır.
  - Java'da düşünmek ve modellemek yazmaktan daha önemlidir
    - Küçük bir şey yapmak için bile hiç tahmin edemeyeceğiniz kadar düşünmeniz gerekebilir.
  - Dolayısıyla hızlıca sürükle-bırak şeklinde geliştirme yerine tasarımını üzerine düşünülmüş, sürekli-bırak kullanılsa bile arka planda ne olup bittiği iyice bilinen geliştirme modeli daha uygundur.
  - Amaç, örneğin, 2000 satırlık ASP ya da PHP sayfası yerine, çoğunlukla satır satır yazılmış ama çok daha modüler ve şablonlara uygun tasarlanmış 50-100 satırlık JSP sayfaları yazmaktır.
- Java özgür ve açık kaynak kodlu yazılımların en yoğun olduğu dildir:
  - Dolayısıyla aynı işi yapan açık kaynak kodlu pek çok farklı ürün vardır<sup>46</sup>.



## Java Sertifikaları

# Java Sertifikaları I

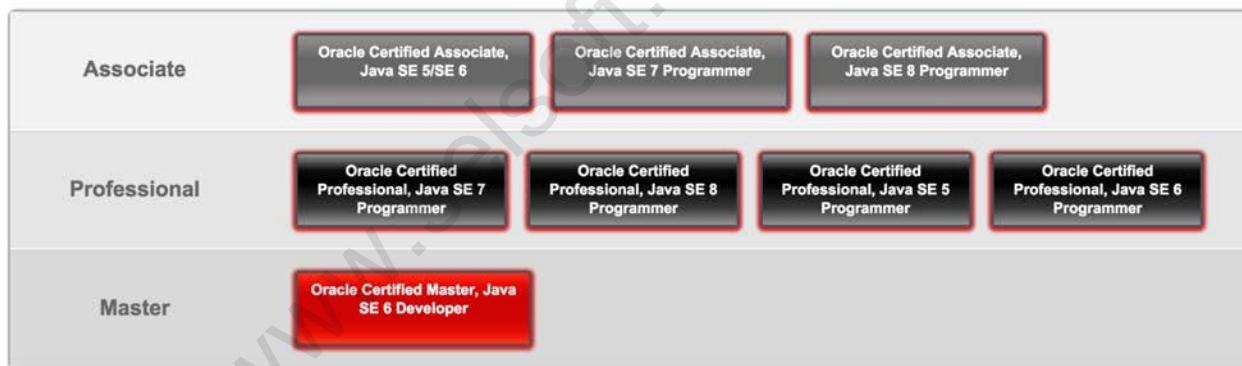
- Java'yi, sistemli bir şekilde, temellerinden başlayarak öğrenmenin bir yolu da Oracle'ın Java sertifikalarını almaktır.
- Oracle'ın Java sertifikaları, Java SE, Java EE ve Java ME üzerindedir.
- Sertifikalar, tecrübeiniz olmasa bile temel bilginizin sağlam olduğunu göstermenin en güzel yoludur.
  - *OCP, Java Programmer olmak "tecrübem yok ama Java'nın temellerini iyi biliyorum" demektir.*
  - Sertifikalı olmak yurt dışı için çok önemlidir ve ülkemizde de gittikçe daha fazla önem kazanmaktadır.

# Neden Sertifika?

- Sertifika alan bir Java’cı, şu iki şeyi göstermiş olur:
  - Kendi kariyeri ile ilgili bir hedefi vardır ve buna doğru ilerlemektedir,
  - Tecrübesi olmasa bile gerekli bilgi donanımına sahiptir.
    - Dolayısıyla, özellikle yeni mezunlar ya da Java’da yeni bir yapıyı öğrenmek isteyenler için önemlidir.
- Bu anlamda sertifika sahibi olmak saygın bir durumdur.
- Oracle’ın Java sertifikaları hakkında [http://education.oracle.com/pls/web\\_prod-plq-dad/ou\\_product\\_category.getFamilyPage?p\\_family\\_id=48](http://education.oracle.com/pls/web_prod-plq-dad/ou_product_category.getFamilyPage?p_family_id=48) sayfasından geniş bilgi alabilirsiniz.

# Java SE Sertifikaları I

- Java SE sertifikaları, girişten başlayarak, Associate (OCA), Professional (OCP) ve Master (OCM) adlarında 3 seviye olarak yapılandırılmıştır:



## Java SE Sertifikaları II

- OCA ve OCP, çoktan seçmeli sınavlardır.
- OCA, OCP için, OCP de OCM için zorunludur.
- OCA sınırlı ve giriş seviyede, OCP ise yerelleştirme, çok kanallı programlama ve veri tabanı programlaması gibi konuları da içeren, dolayısıyla Java SE bilgisini ölçmektedir.
- OCM ise bir proje ödevi ve bu ödevle ilgili yazılan bir makaleyi içermektedir.

# Java EE Sertifikaları

- Java EE sertifikaları temelde farklı Java EE teknolojilerine odaklanan ve onlardaki yetkinliği gösteren sertifikalardır
- Şu anda sadece Java EE 6 için ise OCE (Oracle Certified Expert) sertifikaları vardır.

## Certified Expert

Oracle Certified Expert,  
Java EE 6 JavaServer  
Faces Developer

Oracle Certified Expert,  
Java EE 6 Web Component  
Developer

Oracle Certified Expert,  
Java Platform, Enterprise  
Edition 6 Enterprise  
JavaBeans Developer

Oracle Certified Expert,  
Java EE 6 Web Services  
Developer

Oracle Certified Expert,  
Java EE 6 Java  
Persistence API Developer

## Daha Fazla Ayrıntı İçin

- Oracle'ın Java SE Sertifikaları: OCA, OCP ve OCM <http://www.javaturk.org/?p=2375>
- Oracle'ın Java EE 5 Sertifikaları: OCP ve OCM <http://www.javaturk.org/?p=2404>
- Oracle'ın Java EE 6 Sertifikaları: OCE ve OCM <http://www.javaturk.org/?p=2431>
- Oracle'ın Java sertifika sayfası [http://education.oracle.com/pls/web\\_prod-plq-dad/ou\\_product\\_category.getFamilyPage?p\\_family\\_id=48&p\\_mode=Certification](http://education.oracle.com/pls/web_prod-plq-dad/ou_product_category.getFamilyPage?p_family_id=48&p_mode=Certification)

## Dünyada ve Türkiye'de Java

# Dünyada Java - I

- Java, dünya çapında en büyük geliştirme toplumuna sahip olan dildir,
- TIOBE indexinde (<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>) Şubat 2015'de ikincidir.
- LangPop.com'da göre de ikincidir.
- Dünya çapında 7 ila 10 milyon civarında geliştirici sayısından bahsedilmektedir.
- Dünyadaki programcıların TIOBE'ye göre %17'si, Langpop'a göre ise %15'i Java kullanmaktadır.
- Muhtemelen ülkemizdeki durum daha değişiktir.

## Dünyada Java - II

- Sayısı 250'nin üzerinde JUG'lar son derece faal topluluklardır.
- Yıllık konferanslar, web üzerinden erişilebilen dergi, blog, forum vb. yazılı ve video kaynaklar ile çok zengin bir bilgilenme ortamı sunar.
- 24 Şubat 2015 akşamı itibarıyle Amazon'da Java için 24,405, C++ için 7,398, Python için 4,581, C# için ise 5,794 kitap sonucu gelmektedir.

## Ülkemizde Java

- Ülkemizde Java daha çok büyük ölçekli yazılımlar ile mobil ortamlarda kullanılmaktadır.
- Büyük ölçekli yazılımlar ise ezici olarak sunucu merkezli (server-centric) yazılımlardır.
- Sayı olarak daha fazla C#/.NET programcısının olduğu iddia edilebilir.
- Kesin sayıları söylemek mümkün değil.

# Üniversitelerde Java

- En son ACM yayınına göre ABD'de CS bölümlerinde, programlamaya giriş dersinde kullanılmada Python ilk sırada, Java ise ikinci sıradadır.
- <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>
- Bunda Python'un çok basit ve kısa sözdizimi, interpreted yapısı, basit shell arayüzü ve dinamik tip yapısı etkilidir.
- <http://javarevisited.blogspot.com.tr/2013/11/java-vs-python-which-programming-languange-to-learn-first.html>

# Türk Üniversitelerinde Java

- Çoğunlukla, C, C++, C#/.NET ve Java öğretiliyor.
- Hiç Java ile uğraşmadan mezun olanlarla sık sık karşılaşıyoruz.
- C#/.NET, muhtemelen dünya ortalamasının çok üzerinde bir oranda öğretiliyor.
- Python gibi diller ise azınlıktadır.

## Sorulmaması Gereken Sorular □

# Cesaret Etmeyin!

- Lütfen bu soruları bana sormayın ☺
  - Java zor mu?
  - Java ölüyor mu?
  - Java yavaş mı?
  - Javacılar ne kadar para kazanırlar?
  - Java mı daha iyidir yoksa .NET mi iyidir?
  - Java öğrenmek için matematik bilmeye ya da üniversite okumaya gerek var mıdır?

# Cesaret Etmeyin!

- Java, IQ'sü 70'in altındakiler, tembeller ve özellikle de kendine güvenmeyenler için zordur!
- Java ölmüyor, daha 20sine bile girmede ☺ Cobol 50'sini, C 40'ını devirmişken!
- PHP ve MySQL ile çalışan Facebook yavaş mı? Dolayısıyla Java yavaş değil ama Javacılar yavaş.
- Javacı olarak ben çok kazanıyorum, sizi bilemem ☺
- Mercedes mi daha iyidir yoksa BMW mu?
- Java bilmek ya da yazılımcı olmak için hiç bir eğitime gerek yoktur, okuma-yazma bilmek kafidir ☺ İşiniz doktora ya da avukata düştüğünde de aynı soruyu sormak kaydıyla ☺