



# Java ile Nesne-Merkezi Programlamaya Giriş

*Ek 1 - Java Ortamı*



Eğitmen:

**Akın Kaldıroğlu**

Çevik Yazılım Geliştirme ve Java Uzmanı

# Konular



- Platformlar
- JDK, JRE ve JVM
- Makinamda Java Var mı?
  - MS Windows'da Java
  - MacOS'da Java
  - Linux'de Java
  - Java Bulunamazsa?
- Java Kurulumları
- JDK'nın Dizin yapısı
- Java Ayarları
  - Değişkenler
    - var ile Yerel Değişken Tanımlama
    - Kapsam
  - Tip Çevirimleri
  - Değişmezlik ve final





build better, deliver faster

# Platformlar



- **Platform:** Donanım ve yazılımdan oluşan sisteme denir.
- Yazılım dünyasında platform ile, bilgisayar donanımı ve üzerinde çalışan işletim sistemi (İS) (operating system (OS)) kastedilir.

$$\text{Platform} = \text{Donanım (HW)} + \text{İS (OS)}$$

- Örneğin, i7 gibi Intel CPU mimarisi (x86) üzerinde çalışan Windows (Wintel), Mac OS (MacIntel) ya da Linux işletim sistemleri farklı birer platformdurlar.



- Platformaların en ayırt edici özelliklerinden birisi 32-bit ya da 64-bit olmalarıdır.
- Bu ayırım, Java (ve aslen 32 ve 64-bitlik ayrı sürümleri olan diğer yazılımlar) ile ilgili kurulumlarda yanlış yapmamak için önemlidir.
- Dolayısıyla platformunuzun 32/64 bitlik olduğunu daima bilin.

# Bu Dersin Platformları



- Bu derste aşağıdaki 3 platformu kullanacağız ve gerektiğinde Java açısından aralarındaki farklılıkları belirteceğiz:
- Intel i7 CPU üzerinde Windows 10
- Intel i7 CPU üzerinde Mac OS 10.15 (Catalina)
- Intel i7 CPU üzerinde Linux Ubuntu 18.04 (Mac OS 10.15'da Oracle VMBox üzerinde)

# Farklı Sürümler ve Paketler



- Platformların sürümleri arasında Java açısından zaman zaman ufak-tefek farklılıklar olabileceğini daima göz önünde bulundurun.
- Bu derste böyle durumlar karşılaşıldıkça belirtilecektir.
- Ayrıca Linux Ubuntu için anlatılanların farklı Linux paketleri için geçerli olmama ihtimali de söz konusudur.
- Dolayısıyla platformunuz farklıysa, geçerli olan kurulum ve ayarları sizin bulmanız gereklidir.



- Bu derste sıklıkla **komut satırını** (**command line**) kullanacağız.
- Komut satırı ile temelde, Windows'da **CMD**, Mac OS ve Linux'ta ise **terminal** denen (bazen, iTerm, Xterm vb. araçlarla da karşılanan) programlar kastedilmektedir.





- MS Windows'da genel olarak **Control Panel**'den giderek platformunuzun bit bilgisini öğrenebilirsiniz.

**Control Panel => System and Security => System => System Type**

- Detaylı bilgi için:
  - <https://support.microsoft.com/en-us/help/827218/how-to-determine-whether-a-computer-is-running-a-32-bit-version-or-64>
  - <https://support.microsoft.com/en-us/help/15056/windows-7-32-64-bit-faq>

# Mac OS - I



- Mac OS'da, sol üst kösedeki Apple logosuna tıklayıp, **About This Mac** => **System Report** => **Hardware** => **Processor Name** bilgisini yandaki liste ile karşılaştırın:

İşlemci Adı	32 veya 64 bit
Intel Core Solo	32 bit
Intel Core Duo	32 bit
Intel Core 2 Duo	64 bit
Dört Çekirdekli Intel Xeon	64 bit
Çift Çekirdekli Intel Xeon	64 bit
Dört Çekirdekli Intel Xeon	64 bit
Core i3	64 bit
Core i5	64 bit
Core i7	64 bit



- 10.6 Snow Leopard, 32-bit olarak çalışabilen en son Mac OS sürümüdür, bundan sonraki sürümler 64-bittir.
- Ayrıca terminalde `getconf LONG_BIT` komutu da size 32 ya da 64 cevabını verecektir.
- Ya da yine terminalde `uname -m` komutu size `x86_64` cevabını verirse 64-bitlik sisteme sahipsiniz demektir.
- Detaylı bilgi için: <https://support.apple.com/tr-tr/HT201948>



- Linux Ubuntu'da `System Settings => System => Details` sayfasında sisteminizin kaç bitlik olduğu belirtilmiştir.
- Ayrıca Mac OS'taki gibi Ubuntu'da da
  - terminalde `getconf LONG_BIT` komutu size 32 ya da 64 cevabını verecektir.
  - Ve yine terminalde `uname -m` komutu size `x86_64` cevabını verirse 64-bitlik sisteme sahipsiniz demektir.



- Platformlarınızda kullandığınız patikalarda (path) Türkçe ve “%#<>” !” gibi özel karakterlerden daima kaçının.
- Bu tür karakterler sıklıkla erişim problemleri yaratmakta ve hataya sebep olmaktadır.
- Bu sebeple biz bu derste çok özel bir ihtiyaç olmadıkça, Java kodlarında hiç bir Türkçe veya özel karakter içeren isim kullanmayacağız.





build better, deliver faster

# JDK, JRE ve JVM



- **JDK, Java Development Kit**'in kısaltmasıdır.
  - Java SDK (Software Development Kit) de denir.
- Java ile kod geliştirmek için gerekli yazılım geliştirme kitidir.
  - Kit: Malzeme/alet takımı, çantası.
- Java kodlarını derlemek (compilation), debug etmek, çalıştırmak (run) vb. işler için gerekli araçları içerir.



- JDK'da bulunan çalıştırılabilirlerin (executables) yani komutların (commands) en önemlileri şunlardır:
- **javac**: Java derleyicisidir (compiler), Java kaynak kodlarını derler.
- **java**: Java yorumlayıcısıdır (interpreter), Java'nın çalışma zamanı ortamını (Java Runtime Environment, JRE) başlatır.
- **jdb**: Java debuggerdır.
- **jshell**: JShell komutudur.



- **JRE, Java Runtime Environment**'in kısaltmasıdır.
- JRE, Java'nın hem çalışma zamanı yapısını (kurulum, dosyalar vs.) hem de çalışma zamanının kendisini temsil eder,
- JRE JVM ve Java kütüphanelerini barındırır,
- JRE `java` komutuyla başlatılır.
- Java 11 sürümüyle birlikte artık Oracle ve OpenJDK ayrı JRE paketi sağlamamaktadır.



- **Server JRE**, JRE'nin gözlemleme (monitoring) gibi bazı sunucu (server) araçlarıyla genişletilmiş halidir.
- Oracle tarafından en son Java 8 ile sağlanmıştır.
- Sonrak sürümlerde sağlamamaktadır.





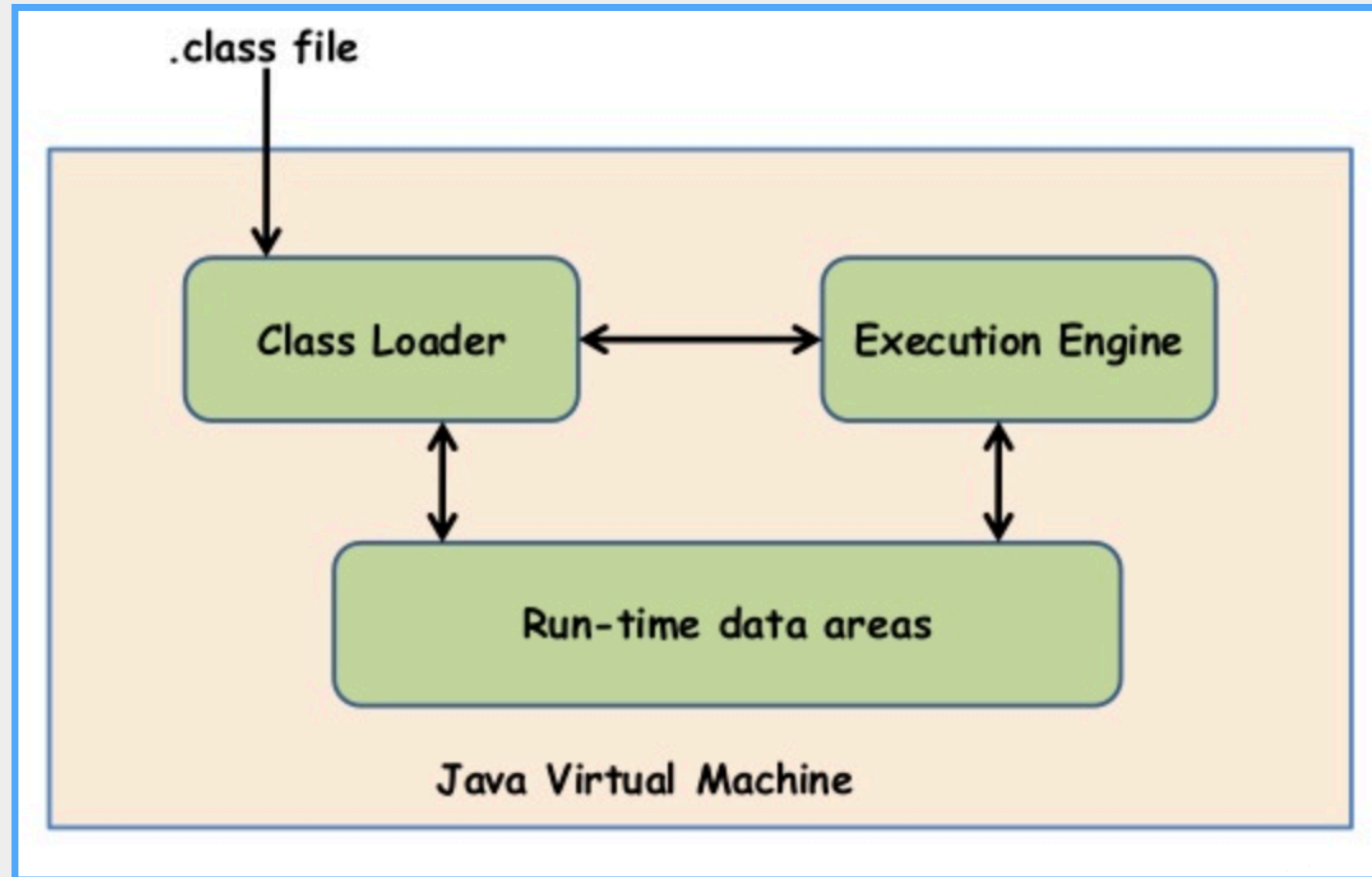
- **JVM: Java Virtual Machine**'in (**Java Sanal Makinası**) kısaltmasıdır ve derlenmiş Java kodlarının üzerinde çalıştığı yazılımdır, sanal bir makinadır.
- JVM, Java'nın çalışma zamanı işlemidir (process).
- JVM, Java geliştiricisini platformdan yalıtan, son derece karmaşık, sanal bir makinadır.
- JVM, İS üzerinden platform üzerinde çalışır.

# JVM'in Bileşenleri - I



- JVM'de şu üç yapı bulunur:
  - Sınıf yükleyici (class loader),
  - JIT (Just-In-Time Compiler) gibi çalıştırma yapıları ve
  - Stack, heap vb. veri alanlarını yöneten yapılar.

# JVM'in Bileşenleri - II





- JVM, tekrar kullanılabilen (reusable) bir Java kütüphanesi de içerir.
- Kütüphaneler, Java ile geliştirilmiş sınıf vb. yapılardan oluşur.
  - Bu yüzden sınıf kütüphaneleri (class libraries) da olarak adlandırılır.
- Java'da kod yazmayı hızlandırırlar.



- Java'da geliştirme yapmak için JDK, geliştirilmiş Java uygulamaların çalışması için JRE gereklidir.
- Yazılım geliştiricilerin makinalarında JDK, sunucu (server) makinalarında ve Java'nın üzerinde çalıştığı saat, telefon, araba vb. ortamlarda ise JRE bulunur.
- JDK, JRE'yi içerir ama tabi olarak JRE, JDK içermez.
- Makinanıza JDK 8 ve öncesi kurulumlarda, JDK içinde JRE için ayrı bir dizin olduğunu görebilirsiniz.



# JDK ve JRE - II



- Bir makinada JDK yok ama JRE varsa `“javac”` çalışmaz ama `“java”` çalışır.
- Ne JDK ne de JRE bir editör ya da IDE barındırmaz.
- Ayrıca her platform için ayrı JDK ve JRE gereklidir.
- Bundan sonra “Java’yı kurmaktan” ya da “bir ortamda Java olmasından” bahsettiğimizde aksini söylemedikçe daima JDK’yı kastedeceğiz.

# JDK ve JRE - IV



- JDK 8 ve öncesinde JDK, JRE'yi içeriyordu ama JRE JDK'in altında bir dizine kuruluyordu,
- JDK 9 ve 10, JDK JRE'yi içeriyordu ama JRE ayrı bir dizine kuruluyordu.

# JDK ve JRE - IV



- JDK ve JRE'nin ayrı kurulumlar olması ve JDK'in JRE'yi içermesi tarihi bir bilgidir, artık geçerli değildir.
- JDK 11'den itibaren Oracle ve OpenJDK ayrı bir JRE ya da Server JRE paketi sunmuyorlar.
- İsteyenler, `jlink` isimli bir JDK aracı yardımıyla, JDK'in daha küçük paketlerini, uygulamalarına özel JRElerini oluşturabilirler.
- Dolayısıyla JRE'yi artık fiziksel bir paket/kurulum yerine JDK'in çalışma zamanı yapısı olarak düşünmek daha doğru olacaktır.



# Makinamda Java Var mı?

# Makinamda Java Var mı? - I



- Bir makinada Java olup olmadığını nasıl anlarız?
- Bir makinada Java olması demek o makinada ya JDK ya JRE ya da ikisinin birlikte olması demektir.
- Geliştirici olarak bize JDK gereklidir (ama özellikle JDK 11 öncesinde makinalarda varsayılan durumda bazen sadece JRE olurdu).



# Makinamda Java Var mı? - II



- Bir makinada JDK olup olmadığını anlamanın en hızlı yolu komut satırını açıp `javac -version` komutunu çalıştırmaktır.
- JRE olup olmadığını anlamak için ise `java -version` komutu çalıştırılır.
- Bu komutlar bulunamazsa, o makinada JDK ya da JRE olmadığını söyleyemeyiz ama olsa bile en azından gerekli ayarların yapılmadığını, dolayısıyla bunlara komut satırından ulaşamadığını söyleyebiliriz.

# Makinamda Java Var mı? - III



- Örneğin yandaki bilgilere göre bu makinada JDK'in ve JRE'nin 1.8, 9. ve 10. sürümlerin olduğunu söyleyebiliriz.
- Ayrıca yandaki cevaplar JVMlerin, 64 bitlik HotSpot sunucu (server) JVM'i olduğunu da söylüyor.

```
1. bash
/Users/akin > java -version
java version "1.8.0_172"
Java(TM) SE Runtime Environment (build 1.8.0_172-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.172-b11, mixed mode)
/Users/akin > javac -version
javac 1.8.0_172
/Users/akin >
```

```
1. bash
/Users/akin > java -version
java version "9.0.4"
Java(TM) SE Runtime Environment (build 9.0.4+11)
Java HotSpot(TM) 64-Bit Server VM (build 9.0.4+11, mixed mode)
/Users/akin > javac -version
javac 9.0.4
/Users/akin >
```

```
1. bash
/Users/akin > java -version
java version "10.0.1" 2018-04-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.1+10)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.1+10, mixed mode)
/Users/akin > javac -version
javac 10.0.1
/Users/akin >
```

# Birden Fazla JDK ve JRE - I



- Şu durumlara da dikkat etmek gerekir:
  - Bir makinada birden fazla, aynı ya da farklı sürümlerde, JDK'lar ve JRE'ler olabilir.
  - Her JDK içinde bir JRE vardır ama JDK olmaksızın, farklı JRE'ler de kurulmuş olabilir.
  - Dolayısıyla komut satırından ulaşılan JDK ya da JRE, makinadaki tek kurulum olmayabilir.

# Birden Fazla JDK ve JRE - II



- Bir makinada kurulu birden fazla JDK ya da JRE'nin her birine bazen tam patika (full path) ile bazen de alias gibi küçük ayarlarla ulaşılabilir.
- IDE'lerde ise durum farklıdır:
  - Genelde IDE'ler ancak kendilerine tanıtılan JDK ve JRE'leri kullanabilirler.
  - Birden fazla JDK ya da JRE'nin tanıldığı durumlarda IDE'nin hangi projede hangisini kullanacağını ayarı yapılabilir.

# JDK ve JRE Nerededir? - I



- Eğer makinamda JDK(lar) ya da JRE(ler) varsa, nerededirler?
- Bu sorunun cevabı için şunlar göz önüne alınmalıdır:
  - JDK ve JRE'nin, varsayılan yani hiç bir değişiklik yapılmadan kurulduğundaki yeri, işletim sistemine bağlıdır, hatta aynı işletim sisteminin farklı sürümlerine göre de değişebilir.
  - Ayrıca kurulum sırasında varsayılan yer kullanılmayıp farklı bir yere kurulum yapılmış olabilir.

# JDK ve JRE Nerededir? - II



- Dolayısıyla, bir makinada hangi JDK ve JRE'lerin olduğunu bilmek için detaylı araştırma yapmak gerekir.
- Örneğin bazı programlar kendi JDK'leri (eskiden JRE'leri) ile gelirler ve kurarlar.
- Bazı tarayıcılar ve Java ile çalışan bazı masaüstü uygulamaları böyledir.





selsoft  
build better, deliver faster

# MS Windows'da Java

# Windows'da Java - I



- Microsoft Windows makinalarda JDK ve JRE'nin varsayılan yeri “C:\Program Files\Java”dır.
- Dolayısıyla bu dizinde bir ya da daha çok JDK ve JRE kurulum dizini bulunabilir, örneğin:

`C:\Program Files\Java\jdk14.0.1`

`C:\Program Files\Java\jdk1.8.0_144`

`C:\Program Files\Java\jre1.7.0_45`

# Windows'da Java - II



- Microsoft platformları 32-bit ya da 64-bit olabilir.
- Dolayısıyla bu durum, 32-bit Windows'a 32-bit, 64-bit Windows'a 64-bit JDK (JRE) kurulduğunda böyledir.
- Eğer, 64-bit Windows'a 32-bit JDK (JRE) kurulursa, varsayılan kurulum patikası `C:\Program Files (x86)\Java` olur.
- 32-bit Windows makinaya 64-bit JDK (JRE) kurulamaz!



build better, deliver faster

# MacOS'da Java



# Mac OS'da Java - I



- Mac OS HighSierra, 64-bittir ve sadece 64-bit JDK'sı (JRE'si) vardır.
- JDK'nın tipik kurulumlarında varsayılan dizin `"/Library/Java/JavaVirtualMachines/"` patikasıdır, örneğin:

`/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk`

`/Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk`

# Mac OS'da Java - II



- Mac OS'da , JRE'nin varsayılan patikası başka bir dizinedir: `/Library/Internet Plug-Ins/JavaAppletPlugin.plugin`
- Mac OS'da bu dizinde kurulu sadece bir tane JRE olabilir.
  - Her yeni JRE, yukarıdaki dizine, diğerini silerek kurulur.
- İstenirse farklı dizinlere farklı JRE'ler kurulabilir.



# Java Control Panel



- Bazı işletim sistemlerinde, sistemdeki Java kurulumları ile ilgili bilgi veren yapılar vardır.
- Örneğin, Mac OS'da “**Java Control Panel**”i vardır.
- Buna **System Preferences** => **Java** ile ulaşabilirsiniz.



build better, deliver faster

# Linux'de Java



- Linux platformları 32-bit ya da 64-bit olabilir.
- Bazı Linux dağıtımları Java kurulumuyla gelebilir.
- Örneğin Debian 9'da, OpenJDK 8 JRE vardır.
- Dolayısıyla, Debian 9'da JDK olmadığından `javac -version` çalışmaz ama `java -version` çalışır haldedir.
- Bu yüzden Java geliştirme için Debian 9'a JDK kurmak gereklidir.

# Ubuntu'da Java



- Ubuntu herhangi bir Java kurulumu ile gelmez.
- Kurulduğunda ise JDK'yı tipik olarak `/usr/lib/jvm` patikasına kurar.
- Ubuntu ayrıca bazı linkler de yaratır: `javac` ve `java` komutları `/usr/bin/javac` ve `/usr/bin/java` ile eşleştirilir.
- Dolayısıyla kurulumla beraber “`javac -version`” ve “`java -version`” çalışır gelir.



build better, deliver faster

# Java Bulunamazsa



# Standart Patikalarda Bulunamazsa



- Standart patikalarda JDK (JRE) bulunamazsa, geriye “**javac**” ya da “**java**” gibi çalıştırılabilir (executable) dosyaların tüm sistemde aranması kalır.
- Bunun için işletim sistemlerindeki arama (search) yapılarının kullanılması gereklidir.
- Dolayısıyla Java’yı varsayılan, patikaya kurmak yönetim açısından daha iyidir.



# Java Kurulumları

# Java Kurulumlarına Erişmek



- Bir makineye JDK ya da JRE kurulumlarına erişmek için en temelde iki yer vardır:
- Oracle'ın resmi Java sayfası: <http://www.oracle.com/technetwork/java/index.html>
- OpenJDK'in sayfası: <https://openjdk.java.net/>

# Oracle Resmi Java'nın Sayfası



- Oracle'ın resmi Java sayfasında <http://www.oracle.com/technetwork/java/index.html> Java ile ilgili Windows, MacOS, bir kaç Linux (eskiden Solaris) dağıtımı bulunmaktadır.
- Ayrıca burada Java ile ilgili tutorial, white paper, makaleler, Java Magazine vb. malzemeler de vardır.

# Open JDK - I



- **Open JDK**, Java SE'nin açık kaynak kodlu (open source) gerçekleştirmesidir: <https://openjdk.java.net/>
- 2006'da Sun'ın bağışladığı kod tabanı (code base) ile oluşturulmuş bir JDK gerçekleştirmesidir.

# Open JDK - II



- Windows, MacOS ve popüler Linux paketleri için OpenJDK kullanılabilir.
- Oracle ve pek çok farklı şirketin JDK'sı, Open JDK üzerine kurulmuştur.
- Open JDK kurulumunu birazdan ele alacağız.

# Farklı JDK/JVM Üreticileri - I



- Eğer bu iki yerde platformunuza uyan bir JDK kurulumu yok ise, platformunuzun üreticisinin sağladığı JDK’i bulup indirmeniz gereklidir.
- Örneğin IBM’in AIX ya da z/OS platformları için JDK kurulumları IBM’in ilgili sayfalarında bulunabilir <https://www.ibm.com/developerworks/java/jdk/>



# Farklı JDK/JVM Üreticileri - II



- Ayrıca JDK/JVM geliştiren farklı kurumlar da vardır:
- GraalVM <https://www.graalvm.org/>
- Amazon's OpenJDK dağıtımı: <https://aws.amazon.com/tr/corretto/>
- Wikipedia [https://en.wikipedia.org/wiki/List\\_of\\_Java\\_virtual\\_machines](https://en.wikipedia.org/wiki/List_of_Java_virtual_machines)



# JDK Kurulumu



- Yazılımların kurulmalarının yaygın iki şekli şunlardır:
  - İndirilen kurulum dosyasının otomatik olarak sistem tarafından “next/continue” vb. tıklamalarla kurulması:
    - Bu, Windows ve Mac OS için en yaygın durumdur.
    - Kurulum sırasında yer vb. seçenekler sorulabilir.
  - Ubuntu’da ise **apt-get** komutu ile bir listedeki yazılımların güncellenmesi (update), kurulması (install) vb. işlemleri yapılır.



- İkinci yöntem ise indirilen sıkıştırılmış dosyanın istenilen yere açılmasıdır.
- Bu durumda ilk yöntemde otomatik yapılan bazı ayarların elle yapılması söz konusu olabilir.

# JDK (JRE) Kurulumu



- Bu durum JDK (eskiden JRE) için de geçerlidir:
- Bazen platforma özel JDK kurulum dosyası (Windows için **exe**, Mac OS için **dmg**, bazı Linux dağıtımları için **rpm**) indirilir ve “next/continue” vb. tıklamalarla kurulur.
- Bazen de sadece sıkıştırılmış, **zip**, **gzip** ya da **tar** dosya indirilir ve açılarak istenilen yere konur.
- Sonrasında kısa yol vb. ayarlar yapılır.

# Hangi JDK?



- Şu anda Java SE'nin en son sürümü 14'tür.
- 9, 10, 12, 13, 14 hatta Eylül 2020'ye planlanlı 15 birer ara sürümdür.
  - Nitekim 9, 10, 12, 13 kurulumları Oracle/OpenJDK sayfalarında yoktur.
- Daha önceki iki ana sürüm 8 ve 11'dir.
- JDK 8 muhtemelen hala dünyadaki en yaygın kullanılan sürümdür.
- Dolayısıyla kalıcılık için mümkünse JDK 11 tercih edilmelidir.





build better, deliver faster

# MS Windows'da JDK Kurulumu

# JDK'yı Kurmak - Windows



- Windows için indirilen Oracle JDK, varsayılan ayarlarla, “next/continue” vb. tıklamalarla, kurulur.
- JDK ve JRE tipik olarak aşağıdaki dizine kurulur:

**C:\Program Files\Java\**

- 64 bitlik platforma 32 bitlik JDK/JRE kurulumları ise aşağıdaki dizine yapılır:

**C:\Program Files (x86)\Java\**

# JDK 8'i Kurmak - Windows



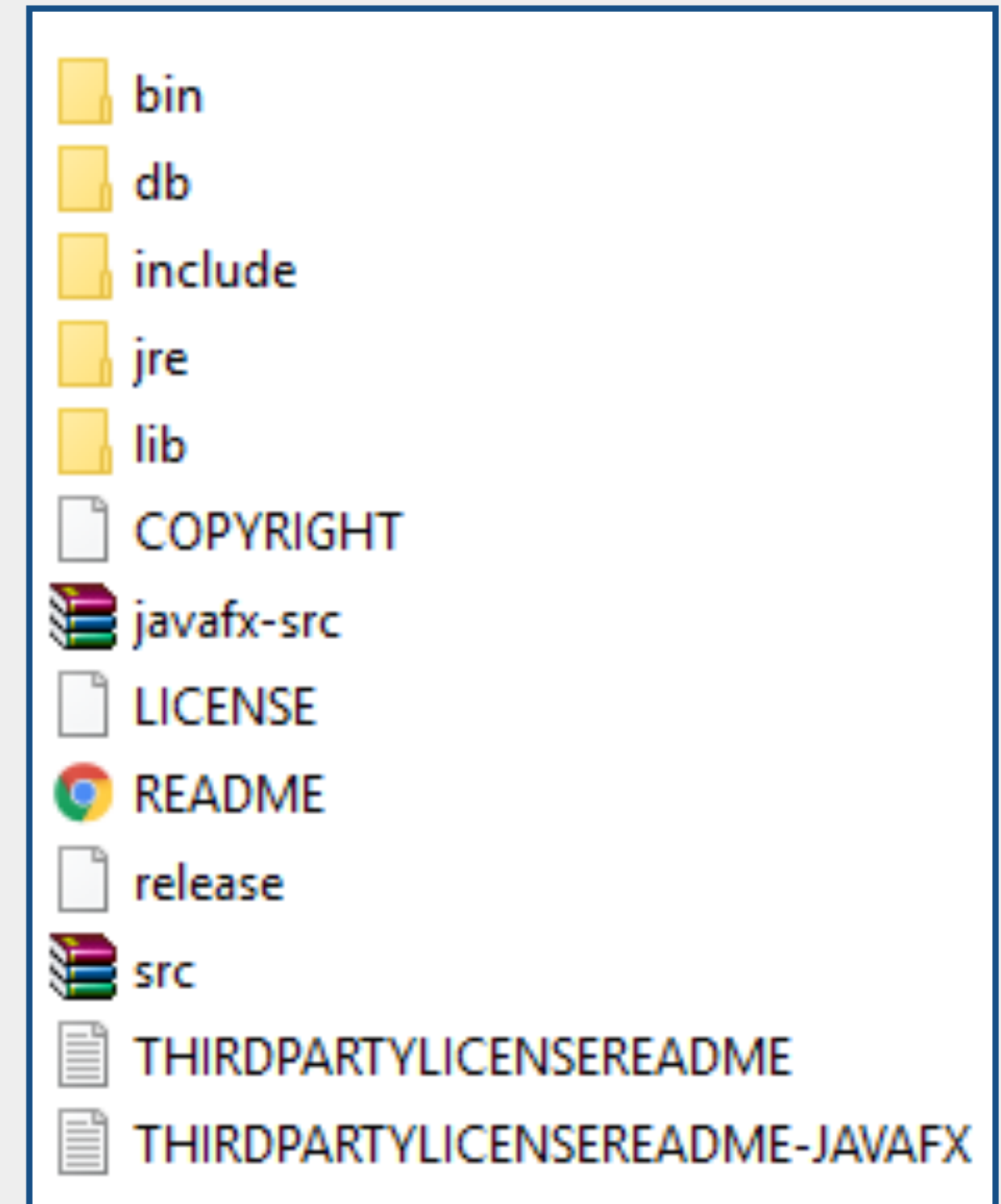
- Eğer JDK'nın sürümü 1.8.0.172 ise patikası

`C:\Program Files\Java\jdk1.8.0_172`

- JRE'nin patikası ise JDK'nın altında,

`C:\Program Files\ Java\jdk1.8.0_172\jre`

olur.



**JDK 8 Dizin Yapısı**

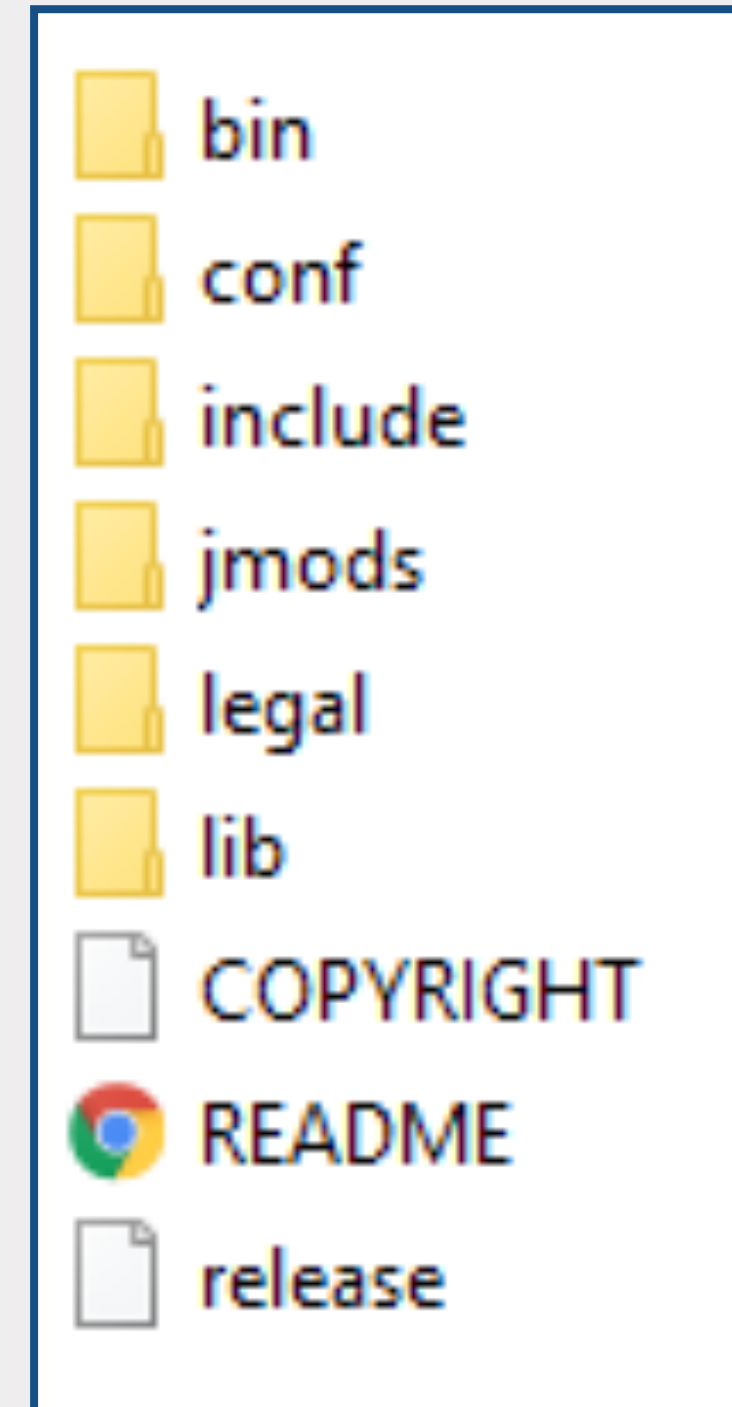
# JDK 11'u Kurmak - Windows



- Eğer JDK'nın sürümü 11.0.7 ise patikası şöyle olur.

**C:\Program Files\Java\jdk-11.0.7**

- JRE ise olmayacaktır.



JDK 10 Dizin Yapısı





build better, deliver faster

# MacOS'da JDK Kurulumu

# JDK 8'i Kurmak - Mac OS



- Mac OS için indirilen Oracle JDK, varsayılan ayarlarla, “next/continue” vb. tıklamalarla, kurulur.
- Eğer JDK'in sürümü 14.0.1 ise, JDK'nın patikası

**`/Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk`**

olacaktır.



# JRE Kurmak - Mac OS - I



- Mac OS'da, JRE otomatik olarak `/Library/Internet Plug-Ins/JavaAppletPlugin.plugin` dizinine kurulur
- Bu dizinde kurulu sadece bir tane JRE olabilir.
- Her yeni JRE, yukarıdaki dizine, diğerini silerek kurulur.

# JRE Kurmak - Mac OS - II



- Eğer kurulacak JRE, var olandan daha yeni bir sürüm ise, onun yerine kurulur,
- Eğer var olandan daha düşük bir sürüm kurulmaya çalışılırsa, örneğin JRE 10.0.1 varken JRE 1.8.172 kurulmak istenirse Mac OS uyarı verir.
- JRE sürümünü düşürecek şekilde bu dizine kurulum yapmayın!



# JDK/JRE Patikaları - Mac OS



- Mac OS'ta JDK/JRE patikaları biraz farklıdır.
- JDK dizinleri, JDK kurulumun altındaki “**Contents/Home**” dizinindedir.
- Yani örneğin JDK 10'un dosyaları aşağıdaki patikadadır:

**`/Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home/`**



# Linux'de JDK Kurulumu

# Ubuntu İçin JDK



- Linux platformlarına JDK kurmak için genelde iki seçeneğiniz vardır: **Oracle JDK** ve **OpenJDK**
- İkisinin de paketlenmiş kurulumları Java SE 8, 9 ve 10 için vardır.
- Ubuntu 18.04'te varsayılan durumda kurulu bir JDK yoktur.
  - Varsayılan kurulum indirildiğinde OpenJDK 10.0.1 kurulur.

# javac ve java İçin Sembolik Linkler



- Ubuntu'da JDK kurulduğunda `javac` ve `java` komutları için otomatik olarak sembolik linkler (symbolic links) oluşturulur.
- Kurulumun `javac` ve `java` komutları bu linklerle eşleştirilir:
  - `javac` için `/usr/bin/javac`
  - `java` için `/usr/bin/java`



# OpenJDK 10'u Kurmak - Ubuntu



- Ubuntu için varsayılan JDK, OpenJDK 11'dur.
- Bunu kurmak için aşağıdaki komutunu çalıştırın:

```
sudo apt install default-jdk
```

- Ubuntu JDK'yi `/usr/lib/jvm` patikasına kurar.
- Dolayısıyla JDK'in tam patikası şöyle olur:

```
/usr/lib/jvm/java-11-openjdk-amd64
```

# javac ve java



- Kurulumla `javac` ve `java` komutları `/usr/bin/javac` ve `/usr/bin/java` sembolik linklerle eşleştirilmiştir.
- Dolayısıyla “`javac -version`” ve “`java -version`” çalışır haldedir.

# OpenJDK 8'i Kurmak - Ubuntu



- Ubuntu için Open JDK 8'i kurmak için terminalde şu komutu girmeniz yeterlidir:

```
sudo apt-get install openjdk-8-jdk
```

- Ubuntu JDK'yi yine `/usr/lib/jvm` patikasına kurar.
- Dolayısıyla JDK'in tam patikası şöyle olur:

```
/usr/lib/jvm/java-8-openjdk-amd64
```



- İsterseniz OpenJDK paketlerini indirip kendiniz de kurabilirsiniz.
- Tavsiye edilen `apt-get` ile otomatik kurulumlardır.
- <http://openjdk.java.net/install/index.html> adresinde tüm Linux OpenJDK otomatik kurulum bilgileri vardır.

# Oracle JDK'yi Kurmak - Ubuntu - I



- Ubuntu'da Oracle JDK'yi kurmak için indirme listesinin bulunduğu yerden örneğin
- JDK 11 için `jdk-11.0.7_linux-x64_bin.tar.gz`
- JDK 8 için `jdk-8u251-linux-i586.tar.gz`
- dosyasını indirip açtıktan sonra oluşan `jdk-11.0.7` (`jdk-1.8.0_251`) dizinini istediğiniz yere, örneğin tavsiye edilen `/usr/local/java` dizinine, koyabilirsiniz.

# Oracle JDK'yi Kurmak - Ubuntu - II



- Bu durumda `/usr/bin`'deki `javac` ve `java` komutlarını, JDK 11 için `/usr/local/jdk11.0.7/bin/javac` ve `/usr/local/jdk11.0.7/bin/java` ile eşleştirebilirsiniz.
- Bu amaçla Ubuntu'da “`update-alternatives`” komutunu kullanabilirsiniz:

```
update-alternatives --install /usr/bin/java java /usr/local/java/jdk10.0.1/bin/java
```

- Bu durumda “`javac -version`” ve “`java -version`” JDK 10 için çalışır hale gelecektir.





# JDK 'nın Dizin Yapısı



- JDK kurulumundan sonra JDK patikasındaki dosya ve dizinlerin yapısını ve işlevlerini bilmek önemlidir.
- JDK dizinin yapısı uzun süredir pek değişmiyordu ama getirdiği modülerlik özelliklerinden dolayı Java 9 ile bu yapı ciddi bir şekilde değişti.
- Dolayısıyla şimdi 8 ve 11 sürümleri için JDK'nın organizasyonunu ele alacağız.
- JDK 8'in dizin yapısı JDK 7 ve 6 için de geçerlidir.

# JAR, WAR ve EAR



- **JAR**, Java **AR**chive, sıkıştırılmış dosyalardan oluşan arşiv dosyasının kısaltmasıdır.
- Ayrıca `jar` bu tipten dosyaların ekidir.
  - `rt.jar` gibi.
- Java'da JAR yanında **Web AR**chive olarak **WAR** ve **Enterprise AR**chive olarak **EAR** sıkıştırılmış dosyalar vardır.

# Properties Dosyaları



- Java'da *anahtar-değer çiftleri* (*key-value pairs*) şeklindeki ayar bilgileri tutan dosyalara **özellikler dosyası** (**properties file**) denir.
- Özellikler dosyaları **properties** uzantılı olurlar:
  - `captions.properties`
  - `db.properties`

# JDK 8 Dizin Yapısı - I



- JDK 8 kurulumundaki en önemli dizinler şunlardır:
- **bin:** `javac`, `java`, `javap` gibi çalıştırılabilir (executable) komutları içerir.
- **lib:** JDK gerçekleştirmesinin JAR dosyaları vardır.
- **jre:** JRE'nin yeridir.

```
jdk1.8.0
  bin
    java*
    javac*
    javap*
    javah*
    javadoc*
  lib
    tools.jar
    dt.jar
  jre
    bin
      java*
    lib
      applet
      ext
        jfxrt.jar
        localdata.jar
      fonts
      security
      sparc
        server
        client
      rt.jar
      charsets.jar
```

# JDK 8 Dizin Yapısı - II



- Diğer dizin ve dosyalar şunlardır:
- **src.zip**: Kütüphanelerin ve diğer araçların C ve Java kaynak kodlarıdır.
- **javafx-src.zip**: JavaFX'in kaynak kodlarıdır.

▶	bin
▶	db
▶	include
▶	jre
▶	lib
▶	man
	COPYRIGHT
	javafx-src.zip
	LICENSE
	README.html
	release
	src.zip
	THIRDPARTYLICENSEREADME-JAVAFX.txt
	THIRDPARTYLICENSEREADME.txt

**JDK 8 Dizin Yapısı**



# JDK 8 Dizin Yapısı - III

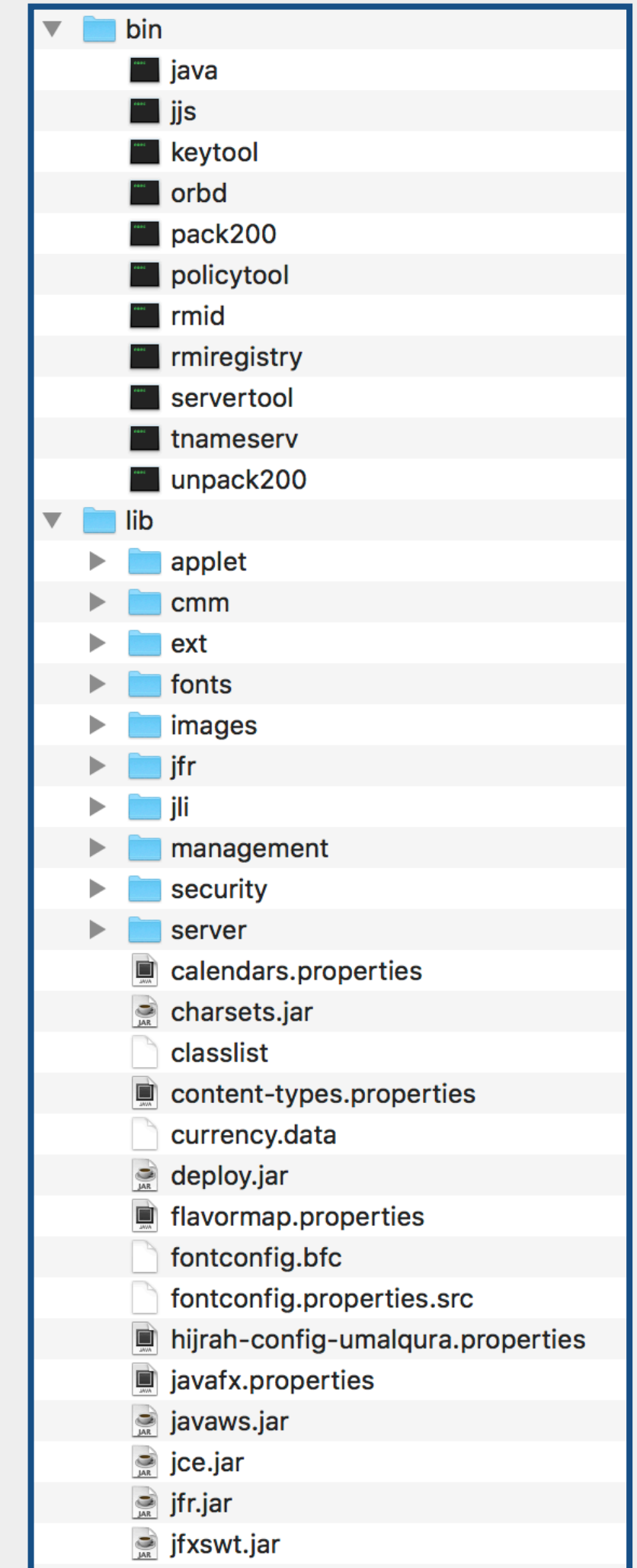


- **db**: Apache Derby (JavaDB) veri tabanı vardır.
- Derby, Java ile yazılmış ilişkisel (relational )bir veri tabanıdır.
- Java 7'den itibaren, JDK 8 dahil, JDK ile gelmektedir.
- Java 9'da JDK'dan kaldırılmıştır.
- **man**: JDK araçları için manual sayfaları vardır.
- **include**: C header dosyalarını içerir.

# JRE 8 Dizin Yapısı



- **jre:** İçinde JRE vardır. Burada da şu iki dizin bulunur:
- **bin:** `java` gibi çalıştırılabilir (executable) komutlar vardır.
- **lib:** `rt.jar` gibi JDK gerçekleştirmesinin jar dosyaları vardır.
- Ayrıca pek çok özellik (properties) dosyası vardır.



JRE 8 Dizin Yapısı

# JDK ve JRE'nin bin Dizinleri



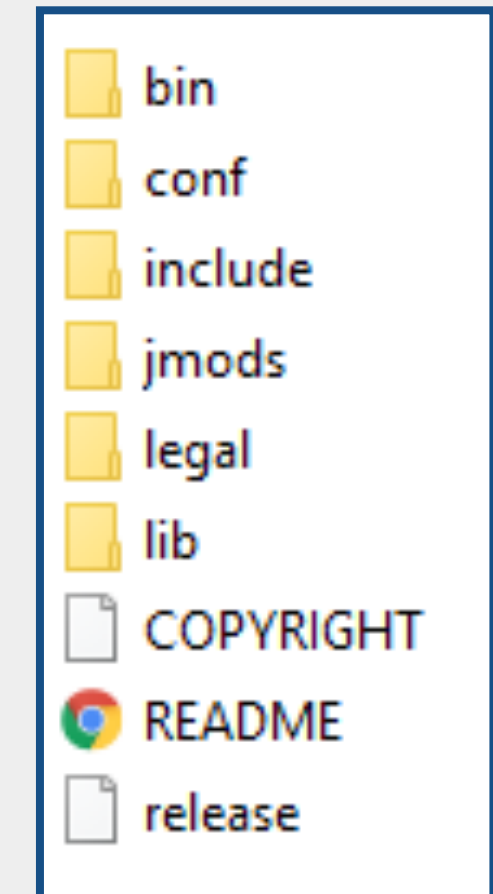
- `jre`'nin `bin` dizinindeki tüm çalıştırılabilirler, JDK'nın `bin` dizininde de vardır.
- Bunlar sadece çalışma zamanı ile ilgilidirler.
- JDK'nın içindeki `bin` dizininde, `java` gibi çalışma zamanı ile ilgili çalıştırılabilirler dışında `javac` derleyicisi, `javap` class dosya ayrıştırıcı (class file dissassembler), `javadoc` API dokümantasyon aracı vb. araçlar vardır.

# JDK 11 Dizin Yapısı - I

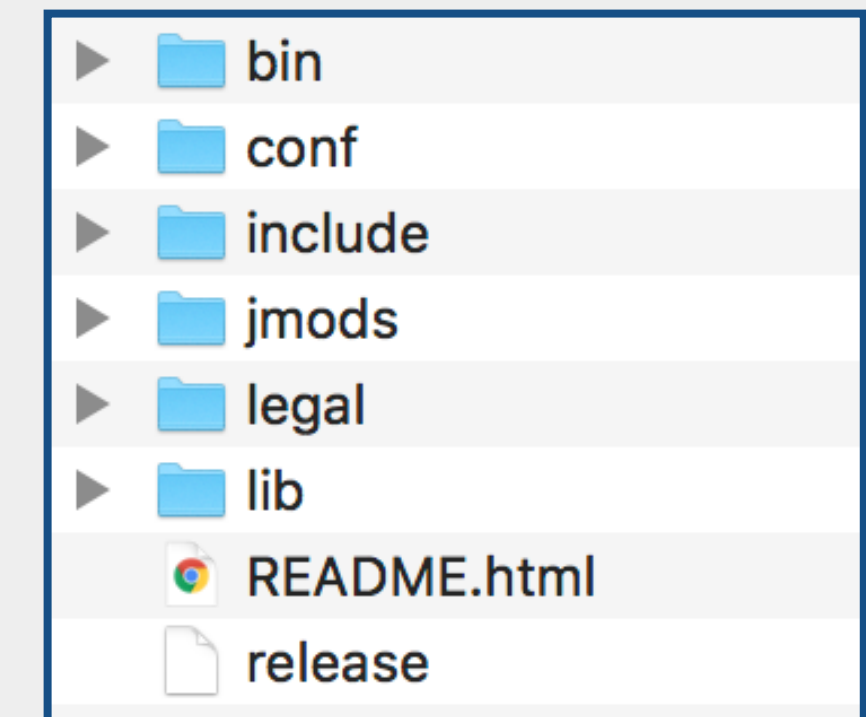


- Java 8'de JDK, JRE dizinini içermesine rağmen Java 11'de JRE yoktur..

## Windows



## Mac OS



## JDK 11 Dizin Yapısı

# JDK 11 Dizin Yapısı - II



- Dizinler:
  - **bin**: Çalıştırılabilirler (executable) vardır.
  - **lib**: JDK gerçekleştirmesinin jar dosyaları vardır.
  - **conf**: Geliştirici tarafından değiştirilebilecek bazı `.properties` vb. ayar dosyaları vardır.
  - **jmods**: Modül tanımlarını içerir.

# JDK 9 ve 10 Dizin Yapısı - II



- **legal:** Her bir modul için haklar (copyright) ve lisans bilgileri vardır.
- **include:** C header dosyalarını içerir.



# Java Ayarları



- JDK kurulduğunda yapılacak en temel iki şey, **JAVA\_HOME** ortam değişkenini (environment variable) tanımlamak ve **PATH** ortam değişkenine gerekli eklemeyi yapmaktır.
- Platformlarda her halükarda varsayılan bir **PATH** ortam değişken değeri vardır.
- Ama **JAVA\_HOME** ayarı kurulumla otomatik olarak yapılmaz, elle yapılmalıdır.

# JAVA\_HOME Ortam Değişkeni



- JDK'in patikasına **JAVA\_HOME** denir ve bu ayarla JDK kurulumunun yeri ifade edilir.
- Bazı Java yazılımları, örneğin Tomcat, **JAVA\_HOME** ile Java'ya ulaşırlar.
- Dolayısıyla **JAVA\_HOME** bir ortam değişkeni olarak tanımlanmazsa ileride problem olabilir.

# PATH Ortam Değişkeni - I



- Yapılacak diğer şey ise işletim sistemlerinde çalıştırılabilir program ve komutlara ulaşımı kolaylaştırmak amacıyla tanımlanmış **PATH** ortam değişkenine JDK'in **bin** dizinini eklemektir.
- Aksi taktirde örneğin JDK'in **bin** dizinindeki çalıştırılabilir komutlara, komut satırından ulaşılamayacaktır.

# PATH Ortam Değişkeni - II



- **PATH** ayarı yapılmazsa, örneğin platformlarda JDK sürümü için kullanılan “**javac -version**” da, JRE sürümü için kullanılan “**java -version**” de “komut bulunamadı” hatası verecektir.
- Dolayısıyla JDK’in **bin** dizini **PATH** ortam değişkenine eklenmelidir ki **javac** vb. komutlara komut satırından ulaşılabilсин.





**selsoft**  
build better, deliver faster

# MS Windows'da Java Ayarları



# Windows'da Java Ayarları



- Windows'da Java ayarları Ortam Değişkenleri (Environmental Variables) ile yapılır.
- Ortam Değişkenleri'ne şöyle ulaşılır:

Kontrol Panel (Control Panel)  $\Rightarrow$  Sistem ve Güvenlik (System and Security)  $\Rightarrow$  Sistem (System)  $\Rightarrow$  Gelişmiş Sistem Ayarları (Advanced System Settings)  $\Rightarrow$  Ortam Değişkenleri (Environment Variables)

# Windows'da JAVA\_HOME - I



- Windows'da `JAVA_HOME` şöyle tanımlanır:
  - Ortam Değişkenleri (Environment Variables) sayfasında Yeni (New) tıklanarak açılan pencerede değişken adı olarak `JAVA_HOME`, değeri olarak da JDK'nın patikası, örneğin "`C:\Program Files\Java\jdk-10.0.1`" girilir.
- Eğer tanımlı bir `JAVA_HOME` varsa değeri değiştirilebilir.

# Windows'da JAVA\_HOME - II



- Bu değişikliği uyguladıktan sonra açılan komut satırında `javac -version` ve `java -version` komutları doğru sonucu verecektir.
- Komut satırı zaten açık ise, değişikliğin etkisini görmek için kapatılıp yenisinin açılması gereklidir.
- Ayrıca komut satırında `set JAVA_HOME` komutu da size `JAVA_HOME` değişkeninin değerini verecektir.

# Windows'da PATH - I



- Windows'da **PATH** şöyle tanımlanır:
  - Ortam Değişkenleri (Environment Variables) sayfasında **PATH** değişkeninin değerine örneğin JDK 11 için "**C:\Program Files\Java\jdk-11.0.7\bin**" eklenir.
  - **PATH** değişkenindeki değerler arasın ";" koymayı unutmayın.
- Eğer **PATH** değişkeni tanımlı değilse, **Yeni (New)** tıklanarak açılan pencerede **PATH** değişkeni ve değeri girilir.

# Windows'da PATH - II



- Komut satırı zaten açık ise, değişikliğin etkisini görmek için kapatılıp yenisinin açılması gereklidir.
- Ayrıca komut satırında “`set PATH`” komutu da size `PATH` değişkeninin değerini verecektir.



# MacOS'da Java Ayarları



# Mac OS'da JAVA\_HOME - I



- Mac OS'da `JAVA_HOME` için kullanıcı dizinindeki “`.bash_profile`” dosyasında aşağıdaki gibi tanım yapılması gereklidir:

```
export JAVA_HOME = /Library/Java/JavaVirtualMachines/  
jdk-10.0.1.jdk/Contents/Home
```

- JDK patikası için, “`/Contents/Home`” eklemeyi unutmayın.

# Mac OS'da JAVA\_HOME - II



- Bu değişikliği uyguladıktan sonra açılan terminalde “`javac -version`” ve “`java -version`” komutları doğru sonucu verecektir.
- Terminal zaten açık ise, değişikliğin etkisi için kapatılıp yenisinin açılması ya da terminalde, aynı dizinde “`source .bash_profile`” çalıştırılması gereklidir.
- Benzer şekilde terminalde “`echo $JAVA_HOME`” komutu da size `JAVA_HOME` değişkeninin değerini verecektir.

# Mac OS'da Bazı Komutlar - I



- Mac OS'da `/usr/bin/javac` ve `/usr/bin/java` komutları `javac` ve `java` komutlarıyla eşleştirilmiştir.
- Benzer şekilde `/usr/libexec/java_home` komutu da `JAVA_HOME` değişkeniyle eşleştirilmiştir.
- Ayrıca `/usr/libexec/java_home -v` komutu size sistemde kurulu tüm JDK ve JRE'leri listeler.
- `/usr/libexec/java_home -x` komutu da sistemde kurulu `x` sürümlü JDK ve JRE'leri listeler.

# Mac OS'da Bazı Komutlar - II



```
echo export "JAVA_HOME=$(/usr/libexec/java_home -v X) >> ~/.bash_profile
```

- komutu da **x** sürümlü JDK kurulumunun patikasını **JAVA\_HOME** olarak atayacaktır:

```
export JAVA_HOME=$(/usr/libexec/java_home -v 1.8) >> ~/.bash_profile
```

ya da

```
export JAVA_HOME=$(/usr/libexec/java_home -v 11) >> ~/.bash_profile
```

# Mac OS'da PATH



- Mac OS'da `PATH` için kullanıcı dizinindeki “`.bash_profile`” dosyasında aşağıdaki gibi tanım yapılması gereklidir:

```
export PATH = $PATH:/Library/Java/JavaVirtualMachines/  
jdk-11.0.7.jdk/Contents/Home/bin
```

- Zaten tanımlı bir `PATH` varsa JDK'nın bin dizinini sonuna ekleyin.
- `PATH`'in değerlerinin arasına “:” koymayı unutmayın.



# Linux'de Java Ayarları



# Ubuntu'da Java Ayarları



- Ubuntu'da Java ayarları, kullanıcı dizinindeki `.bashrc` dosyasında yapılır.
- Bu amaçla komut satırında (terminalde), *vim* gibi bir kelime işlemci ile `.bashrc` dosyasında değişiklik yapmak gerekir.
- Değişikliğin etkisini görmek için açık olan terminalin kapatılıp tekrar açılması ya da terminalde aynı dizinde “`source .bashrc`” çalıştırılması gereklidir.

# Ubuntu'da JAVA\_HOME



- Ubuntu'da `JAVA_HOME` değişkenini `.bashrc` dosyasında aşağıdaki gibi tanımlayabilirsiniz:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

```
export JAVA_HOME=/usr/local/java/java-1.8.251-openjdk-amd64
```

# Ubuntu'da PATH - I



- Ubuntu'da JDK kurulumuyla `javac`, `java`, `javap`, ve `javadoc` gibi pek çok komut `/usr/bin/` dizinde tanımlanmış ve `/etc/alternatives` dizinindeki linkler üzerinden JDK'in `bin` dizinindeki çalıştırılabilir komutlarla eşleştirilmiştir.
- Ubuntu'da `/usr/bin/` dizini de zaten `PATH`'dedir.
- Dolayısıyla Ubuntu'da `PATH` ayarı yapmanıza gerek yoktur.

# Ubuntu'da PATH - II



- Yine de Ubuntu'da **PATH** ayarı yapmak isterseniz **.bashrc** dosyasına makinenizdeki kurulumlara göre aşağıdaki şekilde eklemeler yapabilirsiniz:

```
export PATH=$PATH:/usr/lib/java/jdk-11.0.1/bin
```

```
export PATH=$PATH:/usr/local/java/java-1.8.251-openjdk-amd64/bin
```

- **PATH**'in değerlerinin arasına “:” koymayı unutmayın.



selsoft  
build better, deliver faster

Uyari

# Bir Uyarı



- Bazen `PATH` ortam değişkeninde birden fazla JDK ya da JRE'nin `bin` dizini olabilir.
- Bu durumda sistem ilk bulunan komutu çalıştıracaktır, yani önce gelen JDK ya da JRE aktif olacaktır.
- Dolayısıyla `PATH`'e sadece bir tane JDK (JRE) koymayı ya da öncelik verdiğiniz en önde tutmayı tercih etmelisiniz.



# Kod Geliştirme Ortamları



- Şu ana kadar gördüğümüz gibi Java kodu geliştirmek, debug etmek ve çalıştırmak için temelde JDK ile bir kelime işlemciye (editör) ihtiyaç vardır.
- Fakat projelerdeki pek ok farklı sayıda dosyayı yönetmek, komut satırından kurtulmak ve faydalı pek çok ek araç-gereçten (plug-in) faydalanmak için genelde IDE denilen karmaşık araçlar kullanılır.

# Integrated Development Environment (IDE)



- **IDE, Integrated Development Environment'in (Birleşik/Tümleşik Geliştirme Ortamı) kısaltmasıdır.**
- Yazılım geliştirme sürecinde, kod geliştirme için kullanılan araçtır.
- Kodun yazılması, derlenmesi, çalıştırılması, gerektiğinde debug edilmesi vb. işler IDE'de yapılır.
- Dolayısıyla IDE'ler, dillere özel yetkinliklere sahip kelime işlemci (editor) yanında pek çok farklı araca da sahiptir.



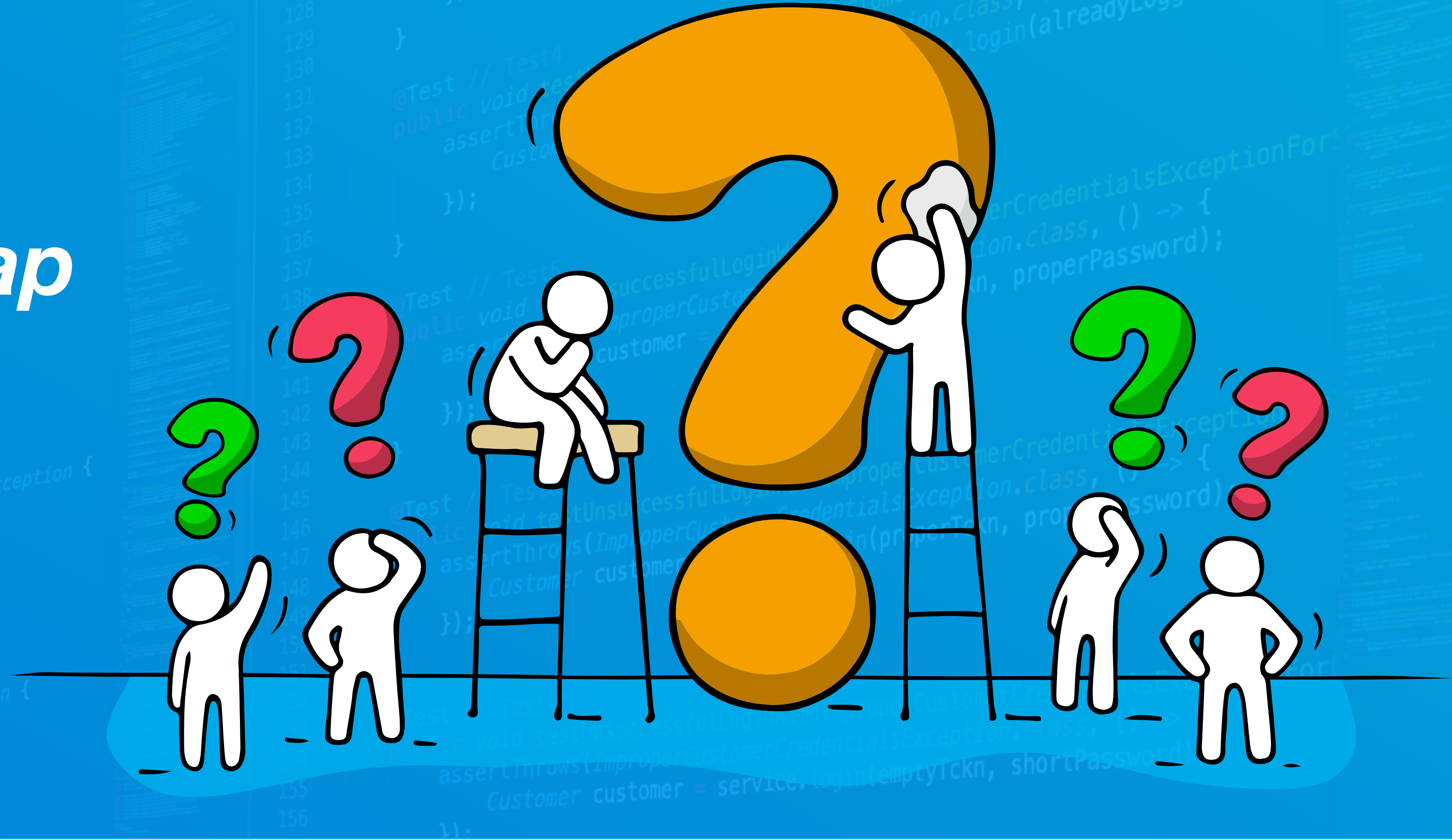
- Java kodu geliştirmek için ücretli veya ücretsiz pek çok IDE vardır. En bilinenleri:
- Eclipse [www.eclipse.org](http://www.eclipse.org)
- Netbeans <https://netbeans.apache.org/>
- IntelliJ Idea [www.jetbrains.com/idea](http://www.jetbrains.com/idea)
- Oracle JDeveloper <http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html>



- Bu derste IDE olarak **Eclipse** kullanılacaktır.
- Sadece bir kaç özel durumda bir IDE olmaksızın, basit bir kelime işlemci (editor) ile kod yazıp, komut satırını kullanarak kodu derleyip çalıştıracacağız.



# Soru ve Cevap Zamanı!







build better, deliver faster

# Ödevler

# Ödevler - I



1. Makinanızdaki JDK kurulumlarını inceleyin.
2. Gerekli `JAVA_HOME` ve `PATH` ayarlarını yapın.

# Bölüm Sonu

## Soru ve Cevap Zamanı!

