



# NESNE TABANLI PROGRAMLAMA

**MARDİN ARTUKLU  
ÜNİVERSİTESİ**

**HAZIRLAYAN: Öğr.Gör. Hüseyin AHMETOĞLU**

2

# 1.HAFTA

## JAVA'ya Giriş

# Java'nın Özellikleri

3

- Java Basittir.
- Java Nesneye Yöneliktir.
- Java Dağıtıktır.
- Java Sağlamdır.
- Java Güvenlidir.
- Java Mimarilerden Bağımsızdır.
- Java Taşınabilirdir.
- Java Yorumlanır.
- Java Yüksek Başarımlıdır.
- Java Çok Kanallıdır.
- Java Dinamiktir.

# Java Basittir.

- Java model olarak aldığı C++'tan çok daha kolaydır.
- Java gereksiz çok fazla özellik içermez.
- Java'da bir özelliğin öğrenilmesi ve kullanılması çok kolaydır.
- Java programlamanın her dalında iddia sahibidir. Her geçen gün Java yeni bir alanda söz sahibi olmaktadır. Bu bağlamda '**basit**' demek, 'çok gelişmiş işlemlerin basit bir şekilde yapılabilmesini sağlayan' demektir.

# Java Nesneye Yöneliktir.

- Java tamamen (bazı kişilere göre %99) **nesneye yönelik**'tir. Diğer bazı diller gibi nesneye yöneliklik sonradan dile eklenmemiştir, başından beri Java'da bulunmaktadır. Bir Java programında olabilecek her şey ya nesnedir ya da bir nesnenin parçasıdır. Java nesneye yönelik programlamayı sadece mümkün değil aynı zamanda kolay kılmıştır. Java'da bir nesnenin yapılması, kullanılması, geliştirilmesi, başka bir yere aktarılması çok kolaydır.

# Java Dağıtıktır.

- 'Dağıtık' birden fazla bilgisayarda çalışan programların bir biriyle uyumlu çalışabilmesidir. Bir yazılım parçasının bir kısmının bir makinede diğerinin başka makinede aynı anda çalışması mümkündür. Bu yüzden Internet'in dilinin Java olduğu söylenmektedir. Günümüzde tek bir makine üzerinde çalışan, başka hiç bir yerle bağlantısı olmayan uygulama kalmamış gibidir. Bir çok işletme Interenet üzerinden iş yapmaktadır. Bir çok işletmede birden fazla makine birbirine bağlı olarak çalışmaktadır.

# Java Sağlamdır.

- Programlamadaki hataların çoğu daha yazılma aşamasında anlaşılabilir. Yazılma aşamasında anlaşılmayanlar, programın çalışması esnasında yakalanabilmektedir. Bir '**exception**'la, programda hatanın ne olduğu, nerede olduğu ve hangi işlemi yaparken olduğu bile belirtilebilmektedir. Başka dillerin aksine çalışma esnasında bir Java programı "Bir hata oldu!" deyip çökmez. Bir çok durumda hataya rağmen program çökmeden çalışmaya devam eder. Hata olması durumunda da programı çalıştıranlar hatayı ayrıntılarıyla öğrenme olanağına sahip olur.

# Java Güvenlidir.

- Java diliyle virüs yapılamaz. Hiç bir virüs bir Java programına bulaşamaz. Bir Java programının yaptığı her hareket takip edilir. Kötü niyetli bir program, bir işlemi eğer izin verilmediyse yapamaz. Bu özellik, Internet gibi herkesin başkasının makinesine erişebildiği bir ortamda çok önemlidir. Java'yı **güvenlik** gereksinimi yüksek bir çok firma bu yüzden tercih etmektedir.



# Java Mimarilerden Bağımsızdır.

- Java'da yazılan bir program hemen hemen bütün işletim sistemlerinde hiç değiştirmeye gerek duymaksızın çalışır. Diğer dillerde bu özellik yoktur. Hatta Windows'un bir versiyonunda çalışan program diğer bir versiyonda çalışmayabilmekte, bazen makinenin çökmesi gibi büyük sonuçlar bile doğurabilmektedir. Windows'u üreten Microsoft firmasının yazdığı programlar bile kendilerinin işletim sistemlerinin bazılarında çalışmamaktadır. Kendilerinin yazdığı belgelerde hangi işletim sistemlerinin hangi versiyonlarında hangi programlarının çalışmadığını belirtilmektedir. Bu durum bütün işletim sistemi ve bütün programlar için geçerlidir. Ürettiği yazılımların her platformda çalışmasını isteyen firmalar daha çok Java'ya yönelmektedir.

# Java Taşınabilirdir.

- Java programları her ortamda aynı veya benzer bir şekilde çalışır. Her ortam/makine için ayrı bir program yazmaya gerek kalmaz. Programcı programın çalışacağı makinenin durumunu göz önüne almak zorunda kalmaz. Bir programın görüntüsü çalıştığı hey yerde hemen hemen aynıdır. C programları da hemen hemen her işletim sisteminde yeniden derlenmek suretiyle çalışabilir. Ama programcı bunun için hazırlık yapmalıdır. Bir işletim siteminde tamsayı  $-2^{15}$  ile  $2^{15}$  arasında değer alırken diğerinde  $-2^{31}$  ile  $+2^{31}$  arasında olabilir. Ama Java bu farkları programcıya şeffaf kılar. Programcı her işletim sistemi ve her sürüm için ayrı bir program yazmak zorunda kalmaz.

# Java Yorumlanır.

- Java '**yorumlamalı**' bir dildir. Yani bir Java programının komutları, çalışırken makinenin anlayacağı formata çevrilir. Java'da bu **Java Virtual Machine (JVM)** tarafından yapılır. Bunun avantajı bir programın kullandığı standart kütüphanelerin programla birlikte taşınması zorunluluğunu ortadan kaldırmasıdır. Bir yorumlayıcı herhangi bir ortamda varsa, bir dildeki standart her özellik o ortamda var demektir. Programla birlikte bu kütüphanelerin de taşınması gerekmez. Bu da bir Java programının bir makineden başka makineye indirilmesini çok hızlandırır. Çünkü sadece programcının yazdığı nesnelere yolculuk yapar.

# Java Yüksek Başarımlıdır.

- Java'nın ilk versiyonlarında çalışan programlar diğer dillerde yazılan eşdeğerlerine göre elbette yavaştır. Ancak Java'nın gelişmesiyle birlikte Java bu farkı, üstün özelliklerinde vazgeçmek zorunda kalmaksızın kapatmaktadır. En son Java sürümüyle birlikte **JIT** (Just-In Time-Tam Zamanında Derleme) teknolojisi devreye girmiş bulunmaktadır. Bununla birlikte Java programları, diğer dillerde en iyi yazılmış programların hızını hemen hemen yakalayabilmektedir. Java yapılması 'imkansız' gibi görüneni çok da fazla performans kaybına yol açmadan yapar.

# Java Çok Kanallıdır.

- Java dili başından 'çok kanallıdır' (**multi-threaded**'dir). Çok kanallılık, bir programın aynı anda birden fazla işlemi yürütebilmesi demektir. Bir program herhangi bir şeyi beklerken arada başka bir işlemi gerçekleştirebilir. Beklenen olay gerçekleşince ilk işlem kaldığı yerden aynen devam ettirilir. Üstelik bunun için programcının fazla bir şey yapması da gerekmemektedir. Çok kanallılık bir çok dilde hiç yoktur. C++ gibi dillerde de dilin özelliği değil, ona sonradan eklenmiş kütüphanelerle kullanılabilmektedir. Ama Java'nın kendisi doğuştan çok kanallıdır.

# Java Dinamiktir.

- Java'da bir programla kullandığı birimlerin (kütüphaneler, modüller veya sınıfların) birbirine bağlanması çalıştırma anında yapılır. Buna 'sonradan bağlama' (**late binding**) denir. Kullanılan birimlerin iç yapısı değiştirildiğinde, bu birimleri kullanan programın değişmesi gerekmez. Yeter ki birimlerin dışarıdan çağırma şekilleri değişmesin. Oysa C++ gibi dillerde herhangi bir değişiklikte (bu bir modülün iç yapısında çağırıcıları ilgilendirmeyen çok ufak bir değişiklik bile olsa) her şeyin yeniden işleme tabi tutulması gerekir. Buna da 'erken bağlama' (early binding) denir. Java'da late-binding bu özellik olmasaydı, Java kütüphanelerindeki her hata düzeltme ve değişiklikte dünyadaki bütün programları yeniden işleme tabi tutmamız gerekirdi ki bu mümkün değildir.

# Java Platformu

- Java sadece bir programlama dili değildir. Aynı zamanda bir yazılım platformudur. **Java Platform'u** işletim sistemi üzerinde çalışan, uygulamalara çok çeşitli servisler veren, bunu her işletim sistemi için standart bir biçimde başarabilen bir platformdur. **Java Programlama Dili** ise bu platform üzerinde yazılım geliştirmek için kullanılan programlama dilidir. O yüzden Java'yı öğrenmek sadece Java dilini öğrenmek demek değildir. Java platformu üzerinde çalışan farklı bir dil de üretilebilir.

# Derleme (Compilation) ve Yorumlama (Interpretation)

- Programalama dili, insanla makine (bilgisayar) arasında iletişim kurmak için kullanılan dildir. Başka bir deyişle programcı bilgisayara, özellikle de işlemciye istediklerini yaptırmak için bir dil kullanır. Bu dil konuşma diline yakın bir dil olmalıdır ki insanlar rahatça kullanabilsin. Bilgisayar işlemcisinin dili ise sadece sayılardan oluşur.
- Bilgisayarlar sayısal (dijital) aygıtlardır. İşlemciler sadece belli sayılarla ifade edilen makine kodlarından anlar. Bu kodlardan oluşan dile '**makine dili**' denir, üretilen koda da '**makine kodu**' (machine code) denir.



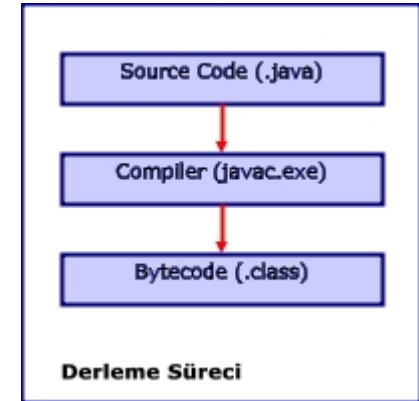
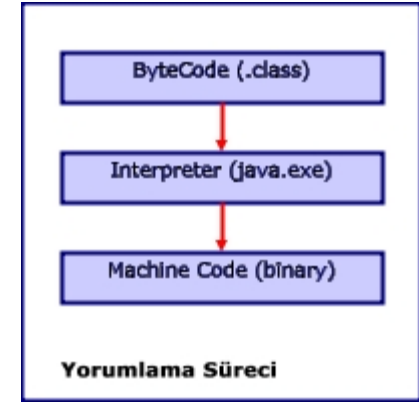
# Derleme (Compilation) ve Yorumlama (Interpretation)

- Bilgisayarın ilk dönemlerinde insanlar programlarını sayı kodları kullanarak yazıyorlardı. Bu kodlar her işlemci türü için değişik olduğundan, yazılan programlar sadece belli bir işlemci ile çalışabiliyordu. Hatta bir işlemci sürümü için yazılan kod bir önceki sürümde çalışmayabiliyordu. Bu yüzden '**programlama dili**' kavramı ortaya çıkmıştır. Bu yapıda, programcı bilgisayarın yapması gerekenleri ona, konuşma diline yakın bir dille söyler. Bu dilde yazılanlar makine diline çevrilir. Böylece programcılar, hem işlemciminin kendi komut kümesini öğrenmek zorunda kalmazlar; hem de yazdıkları programları farklı işlemciyle çalışabilir.

# Derleme (Compilation) ve Yorumlama (Interpretation)

18

- Programlama diliyle düz metin şeklinde olan programa '**kaynak kodu**' (source code) denir. Makine dilinde olan koda da "ikili kod" (binary code) denir. Programlama dilinden makine diline çevirme işi iki şekilde yapılabilir. Biri 'anında', ikincisi 'önceden'. Anında çevirmek, programa dilindeki ifadeleri bir yandan okuyup bir yandan makine diline çevirmek demektir. Buna '**yorumlama**' (interpretation) denir. Önceden çevirme ise, programlama dilindeki ifadelerin çalıştırılmadan önce makine diline çevrilmesi, sonra da çalıştırılması anlamına gelir. Buna da '**derleme**' (compilation) denir.



# Derleme (Compilation) ve Yorumlama (Interpretation)

- Derleme'nin faydası daha hızlı olmasıdır. Çünkü makine diline çevirme sadece ve sadece bir kere yapılmaktadır. Yorumlama da ise her çalıştırmada çevirme işlemi yapıldığından daha yavaştır. Derlemenin zararı da, programdaki her değişiklikte önce derlemeyi sonra çalıştırmayı gerekli kılmasıdır. Sık değişiklik yapılan durumlarda bu, programcı için ciddi bir sorundur. Oysa yorumlama anında olduğu için, değişiklik de anında etkili olmaktadır. Perl, Basic gibi diller yorumlamalı, C ve Pascal gibi diller de derlemeli dildir.

# JVM - Java Virtual Machine (Java Sanal Makinesi)

20

- Java'daki durum biraz daha karışıktır. Java hem 'derlemeli' bir dildir, hem de 'yorumlamalı'. Java programlama diliyle yazılmış kaynak kodu, sanal bir işlemcinin anlayabileceği makine koduna çevrilir. Bu kod gerçek bir makine olmadığından ona makine kodu denmez. Ama programla dili gibi konuşma dilinde de olmadığından kaynak kodu da değildir. Bu ara dille yazılmış koda '**bayt kod**' (bytecode) denir. Çünkü makine kodu gibi baytlardan oluşur. Yani kaynak kodu bayt koduna 'derlenir'.

# JVM - Java Virtual Machine (Java Sanal Makinesi)

21

- Sanal işlemci bu bayt kodlarının makine koduna programın çalışması esnasında çevirir. Her işletim sistemi ve işlemci için ayrı bir sanal makine vardır. Java kaynak kodları Java derleyicisi tarafından bayt koda çevrilir. Üretilen bu kod bütün işlemciler için aynıdır. Her işletim sistemi ve işlemciye göre sanal makine, bu bayt kodu ilgili işlemcinin makine diline anında çevirir yani yorumlar. Bu sanal makineye **Java Sanal Makinesi - Java Virtual Machine (JVM)** denir.

# JVM - Java Virtual Machine (Java Sanal Makinesi)

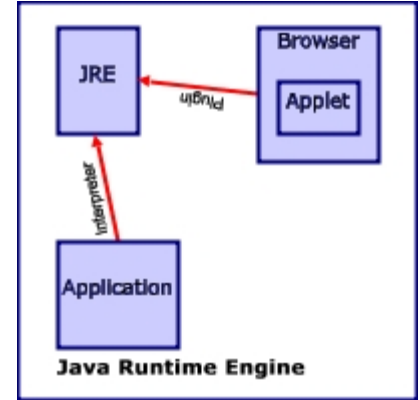
22

- Java platformunun programlama dili 'Java' dır. Bu dilde yazılmış kaynak kodları '**.java**' uzantılı dosyalarda saklanır. "javac" adlı bir derleyici ile ikili koda çevrilir. İkili kod'lar '**.class**' uzantılı dosyalarda saklanır. Bu dosyalar '**java.exe**' adlı yorumlayıcı tarafından çalıştırılır. Java kaynak kodu herhangi bir metin düzenleyici ile yazılabilir. Java'nın derleyicisi ücretsiz olarak dağıtılmaktadır. Java'nın yorumlayıcısı (JVM)'de plug-in olarak yüklenebilmektedir. Bu plug-in de ücretsizdir. Ayrıca bilinen bütün tarayıcılar'ın bir JVM'i bulunmaktadır.

# JRE (Java Runtime Engine - Java Çalışma Zamanı Motoru)

23

- Bir makinede Java'yla yapılmış bir programın çalışması için orada mutlaka Java interpreter'ı olması gereklidir. Applet'ler browser içindeki Java'yla çalışırlar. Java kodlarını çalıştıran sisteme **JRE** (Java Runtime Engine- Java Çalışma Zamanı Motoru) denir. Java'da yapılmış bir program herhangi bir makineye kurulurken (eğer yoksa) öncelikle JRE kurulmalıdır. Ancak bu işlem bir makine için sadece bir kere yapıldığından, üzerinde JRE kurulmuş bir makineye herhangi bir program yüklenirken tekrar JRE kurulmasına gerek yoktur. JRE bazı işletim sistemlerinde kendiliğinden kurulu olup, bazılarında da tercihe bağlı olarak kurulabilmektedir.



# JRE (Java Runtime Engine - Java Çalışma Zamanı Motoru)

24

- Elbette yazılan program, Java'nın o makinede kurulu sürümünden daha ileri bir sürümü gerekli kılıyorsa JRE'nin upgrade edilmesi gerekir. Java'da yeni bir sürüm yüklenince eski sürümle çalışan programlar çalışmaya devam eder. Bütün Java sürümleri 'geriyle uyumlu' (backward compatible)'dirler. Yıllar önce yapılmış Java'nın ilk sürümüne göre çalışan uygulamalar, son sürümle de çalışmaktadır.



# SDK (Software Development Kit - Yazılım Geliştirme Kiti)

- Java'da yazılmış programların çalışması için JRE gerekmektedir. Ancak JRE program geliştirmek için kullanılamaz. JRE sadece Java programlarını çalıştırmak için gerekli araçları içerdiğinden derleyici içermez. Bir programı çalıştırmak isteyen bir kullanıcının zaten bir program yapıp derlemesi gibi bir gereksinmesi yoktur. O yüzden program yapmak isteyenlerin Java **SDK** (Software Development Kit - Yazılım Geliştirme Kiti)'ni kurmaları gerekmektedir.

# SDK (Software Development Kit - Yazılım Geliştirme Kiti)

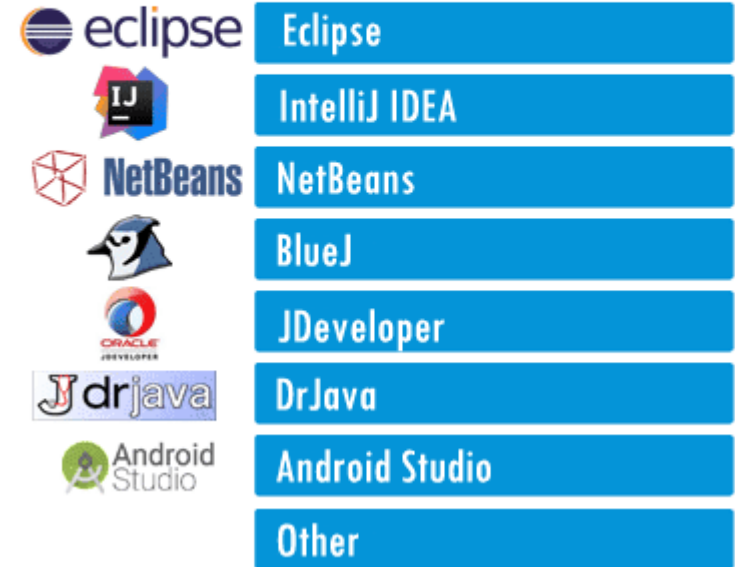
- SDK JRE'yi de içermektedir. Programcının derlediği programı elbette çalıştırması da gerekir. SDK belli bir klasöre kurulur ve hemen hemen bütün araçlar bu klasörün altındadır. Bu yüzden bir makineden diğerine kopyalandığında bile çalışabilir. Sistemde çok fazla değişiklik yapmadığından sistemin bütününe veya başka programları olumsuz yönde etkileyecek hiç bir işlem yapmaz. O yüzden de kurulumu riskli değildir. Ayrıca sisteme çok fazla bağlı olmadığından, sistemdeki değişikliklerden etkilenmez.

# SDK (Software Development Kit - Yazılım Geliştirme Kiti)

27

- Java'yı derlemek ve çalıştırmak için standart tek bir editör (IDE, RAD) yoktur. SDK sadece derleme ve çalıştırmak için kullanılır. Java kodu yazmak için sadece bir metin düzenleyici yeterlidir. Ancak daha hızlı kod yazabilmek için çeşitli şirketlerin Java geliştirme ortamları vardır.

What's your favorite IDE for Java Development?



# LİNKLER

28

JDK

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

İntelliJ IDEA

<https://www.jetbrains.com/idea/download>