

Java'ya Giriş

Tarihçe

- C ve C++ üzerine kuruldu.
 - C'den syntax (c-like syntax)
 - C++'dan nesne tabanlı programlama
- Fortran
 - Formula Translator.
 - Bilimsel uygulamalar için uygun, ancak sistem programlama için değil
- Basic
 - Kolay ancak büyük uygulamalar için etkili değil.
- Assembly
 - Hızlı, ancak yazım zor ve multi-platform değil.
- Bu diller aynı zamanda GOTO kullandığı için spagetti kod oluşturuyordu.

C

- Bell Labs'da **Dennis Ritchie** tarafından yaratıldı.
 - Atası BCPL'dir.
 - BCPL'den B çıktı (**Ken Thompson**)
 - B'den de C çıktı.
- C, tam olarak programcılara uygun bir dildi. Diğer diller bilim insanları ya da basit uygulamalar için yapılıyordu.
- Sistem programlama ve işletim sistemleri için en uygun dildir.
 - İpucu: Garbage collection yok.

C++

- C, eskisine göre daha kompleks programlar yazmaya olanak sağladı.
- Buna çözüm olarak NTP(OOP) geliştirildi.
 - C++, C'ye NTP eklenmiş dildir.
 - Tam olarak *yeni bir programlama dili* olsun diye geliştirilmemiştir.
- Yine Bell Labs'da **Bjarne Stroustrup** tarafından geliştirildi.
 - C with Classes

Java

- Java, Sun Microsystems'da **James Gosling** tarafından yaratılmıştır (1991).
 - James Gosling, Patrick Naughton, Chris Warth, Ed Frank, Mike Sheridan
- İlk ismi Oak. 1995'de adı Java oldu.
- Çıkış amacı, yazılan programların **platform-bağımsız** olmasıydı.
 - Yani yalnızca işletim sistemleriyle sınırlı kalmadılar. Elektronik cihazlar üzerinde de çalışmasını istediler aynı dilin.
 - Mikrodalga fırın, kumanda, vb.
- Diğer diller **derleniyor** ancak bu derleme işlemi bildiğiniz gibi işletim sistemine göre değişir.

Java

- İnternetin gelişmesiyle birlikte Java'nın popularitesi arttı çünkü bu alanda kendini gösterme şansı buldu.
 - Java Applets
 - Tarayıcı üstünde rahatlıkla çalıştırabiliyorduk. (Artık çok popüler değil.)
- C ve C++'A benzemesi de programcılar arasında kabul edilmesini hızlandırdı.
 - C, C++'nın gelişmiş bir versiyonu olarak düşünülemez. Farklı dillerdir ve amaçları aynı değildir.

Bytecode

- Derleyiciler kodu derlediğinde, o işletim sistemi üzerinde çalışabilecek *executable* bir çıktı verirler.
- Java öyle değil. Farklı bir çıktı verir, buna da bytecode deriz.
 - Taşınabilir kod da deniyor.
- Bu, hem yorumlayıcı (interpreter) hem de makine diline derlenebilir durumdaki komut setine verilen isimdir.
- Bytecode, **JVM** üzerinde çalışır.

JVM

- Java, ilk olarak yorumlanan (interpreted) bir dil olarak tasarlanmıştır.
- Derleyici, bu dili bir *ara form* haline getirir ve bu da bir sanal makine (VM) tarafından yorumlanır (JVM).
 - Bu, derlenen dillere göre bir hız dezavantajı verebilir. Ancak bytecode tasarımı çok iyi olduğu için Java yine de hızlıdır.
 - Özellikle diğer yorumlanan dillere göre hızı göze çarpar.
- JVM, aynı zamanda **güvenlik** de sağlar çünkü bir nevi **sandbox** görevi görür.
- Java kodunun çalışması için, JVM kurulu olması yeterlidir.
 - JRE: Java Runtime Environment

JDK, JRE, JVM

- **JDK**
 - Java Development Kit
 - Java programları geliştirmek ve çalıştırmak için gereken şeyleri kapsar.
- **JRE**
 - Java Runtime Environment
 - Daha önce derlenmiş Java programlarını çalıştırmak için gereken şeyleri kapsar.
- **JVM**
 - Java Virtual Machine
 - Java derleyicisinin oluşturduğu *bytecode*'u yorumlayan sanal makine.

Java

- Güvenli
 - JVM, sandbox görevi görüyor.
- Kolay
 - Pointer yok.
- Nesne tabanlı
- Her ortamda çalışabilir

IDE

- Java için birden fazla IDE kullanılabilir.
 - Integrated Development Environment
 - Eclipse, NetBeans, IntelliJ, vb.
- Yazdığınız kodlar `.java` uzantısı ile kaydedilir ve derlendiğinde `.class` haline gelirler.
- `javac` komutu ile derlenir ve `java` komutuyla çalıştırılır.
 - Command line
 - `javac dosya_adi.java`
 - `java dosya_adi`

