



Java Spring Boot Bootcamp

HAZIRLAYAN

MEHMET ALTAN

Java Nedir?

- ▶ Java, Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış;
 - ▶ açık kaynak kodlu,
 - ▶ nesneye yönelik,
 - ▶ platform bağımsız,
 - ▶ yüksek verimli,
 - ▶ çok işlevli,
 - ▶ yüksek seviye,

hem yorumlanan hem de derlenen bir dildir.

Açık Kaynak

- ▶ "Open source" bir yazılım, donanım veya diğer bir ürünün tasarımı, kodu veya içeriği, geliştirici veya kullanıcıların telif hakkı sahibi veya sahipleri tarafından herkese açık bir şekilde paylaşıldığı bir yazılım lisans modelidir.
- ▶ Açık kaynaklı projelerin bazı temel özellikleri şunlardır:
 - ▶ Kaynak kod açıktır: Yazılımın tüm kaynak kodları genel olarak erişilebilir ve düzenlenebilir durumdadır.
 - ▶ Lisanslama: Açık kaynaklı yazılımlar, genellikle kullanıcılara, yazılımı kullanma, kopyalama, dağıtma ve değiştirme hakkı veren özgür bir lisansla sunulur.
 - ▶ Topluluk katılımı: Açık kaynak projeleri, genellikle küresel bir geliştirici ve kullanıcı topluluğu tarafından desteklenir ve katılıma açıktır.
 - ▶ Şeffaflık: Açık kaynaklı yazılımların geliştirilme süreçleri ve güvenlik açıkları gibi önemli bilgiler toplulukla paylaşılır.

Araştırma Ödevi

- ▶ Apache Lisansı, ve MIT Lisansını Lütfen araştırınız.

Tartışınız.

- ▶ Hangisi daha güvenli?
 - ▶ Open Source geliştirilen bir yazılım?
 - ▶ Kendi geliştirdiğimiz kapalı kaynak yazılım?

Nesneye Yönelik

- Bir programlama dilinin "nesneye yönelik" (Object-Oriented) olması, o dilin nesne tabanlı programlama paradigmalarını desteklediği anlamına gelir. Nesne tabanlı programlama, verileri ve bu veriler üzerinde işlem yapma yöntemlerini bir araya getiren bir programlama yaklaşımıdır. Temel olarak, gerçek dünyadaki nesnelerin (objelerin) davranışlarını ve ilişkilerini modellemek için kullanılır.

Nesne tabanlı programlamada temel kavramlar;

- ▶ Nesneler (Objects): Nesne, verileri (özellikler veya durum) ve bu verilere uygulanabilecek işlevleri (metodlar veya davranışlar) içeren bir programlama birimidir. Nesneler, gerçek dünyadaki varlıkların dijital temsilleridir.
- ▶ Sınıflar (Classes): Sınıf, nesnelerin taslağını veya şablonunu temsil eder. Nesneler, bir sınıftan türetilir ve sınıf, nesnelerin nasıl oluşturulacağı, ne tür özelliklere ve metodlara sahip olacakları gibi bilgileri içerir.
- ▶ Kalıtım (Inheritance): Kalıtım, bir sınıfın diğer bir sınıftan özelliklerini ve metodlarını miras almasıdır. Bu, kodun tekrar kullanılabilirliğini artırır ve kodun daha düzenli ve anlaşılır olmasını sağlar.
- ▶ Polimorfizm (Polymorphism): Polimorfizm, aynı işlemin farklı nesneler üzerinde farklı şekillerde çalışabilmesini ifade eder. Yani, bir metod farklı sınıflardaki nesnelerin türüne göre farklı davranabilir.
- ▶ Soyutlama (Abstraction): Soyutlama, bir nesnenin en önemli özelliklerine (ayırt edici) odaklanarak karmaşıklığı azaltmayı ve yalnızca ilgili ayrıntıları göstermeyi ifade eder.

Tartışınız;

- ▶ Nesne tabanlı programlama gerçek dünya örnekleriyle karşılaştırınız.

Platform Bağımsız

- ▶ Java, platform bağımsız bir programlama dili olarak tasarlanmıştır. Platform bağımsızlık, Java'nın yazılan bir programın herhangi bir işletim sistemi veya donanım mimarisinde çalışabilmesi anlamına gelir. Bu özelliği, Java'nın "Write Once, Run Anywhere" (Bir kere yaz, her yerde çalıştır) felsefesiyle ilişkilendirilir.

Platform bağımsızlığın arkasındaki ana nedenler şunlardır:

- ▶ Java Sanal Makinesi (Java Virtual Machine - JVM): Java programları, Java Sanal Makinesi (JVM) tarafından çalıştırılır. Java kaynak kodları önce JVM'ye çevrilir ve sonra JVM, uygun işletim sistemine ve donanım mimarisine uygun makine kodunu oluşturur. Bu sayede, aynı Java programı farklı işletim sistemlerinde JVM aracılığıyla çalışabilir.
- ▶ Bytecode (Bayt Kod): Java kaynak kodları, Java Derleyicisi tarafından makine diline derlenmeden önce bytecode adı verilen platform bağımsız ara birime çevrilir. Bu bytecode, JVM tarafından yorumlanır veya Just-In-Time (JIT) derleyici tarafından derlenir ve çalıştırılan platforma uygun makine koduna dönüştürülür.
- ▶ Standart Kütüphaneler: Java, platform bağımsızlığı sağlamak için zengin bir standart kütüphane setine sahiptir. Bu kütüphaneler, işletim sistemine özgü fonksiyonlar yerine Java koduyla yazılmış ve JVM tarafından yönetilen genel işlemleri içerir.

- Bu nedenlerle, Java uygulamaları yazılırken ve derlenirken, hedef platformun donanım ve işletim sisteminden bağımsızdır. Java, bu özelliği sayesinde bir kez yazılan bir uygulamayı Windows, Linux, macOS gibi farklı işletim sistemlerinde ve x86, ARM gibi farklı donanım mimarilerinde çalıştırabilirsiniz. Bu da Java'yı geniş bir yelpazede kullanılabilir ve popüler bir programlama dili haline getirir. Platform bağımsızlık, özellikle yazılımın farklı ortamlarda dağıtılması ve kullanılması gereken büyük ölçekli projelerde büyük bir avantajdır.

Araştırma Ödevi


- ▶ JIT (Just-In-Time) Derleme ve Bytecode Yorumlama kavramlarını araştırınız.

Tartışınız

- ▶ JRE ve JDK?

JRE ve JDK

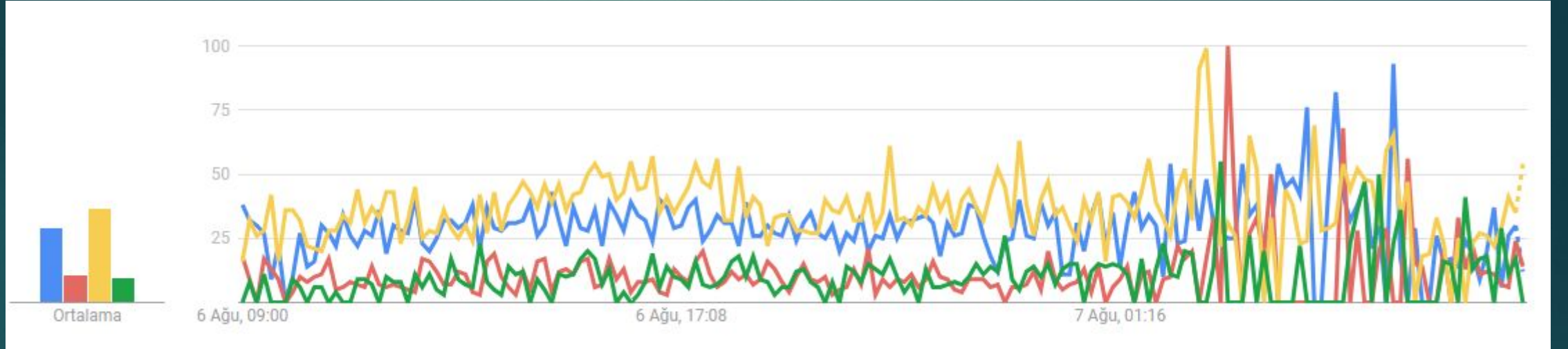
- ▶ JRE, Java programlarını çalıştırmak için gerekli olan minimum ortamı sağlayan bir yazılım paketidir. JRE, Java bytecode'ını doğrudan çalıştırmak ve Java programlarının çalışmasını sağlamak için gereken Java Sanal Makinesi'ni (JVM) içerir. Bu sayede, Java uygulamaları platform bağımsız olarak çalışabilir.

- 
- ▶ JDK, Java programlama dilini geliştirmek ve Java uygulamaları oluşturmak için kullanılan bir yazılım paketidir. JDK, Java derleyicisi, hata ayıklama araçları ve diğer yardımcı programlar gibi geliştirme işlemleri için gereken tüm araçları içerir. Aynı zamanda, Java Runtime Environment (JRE) ile birlikte gelir, bu nedenle JDK yüklediğinizde JRE de otomatik olarak sistemde bulunur.

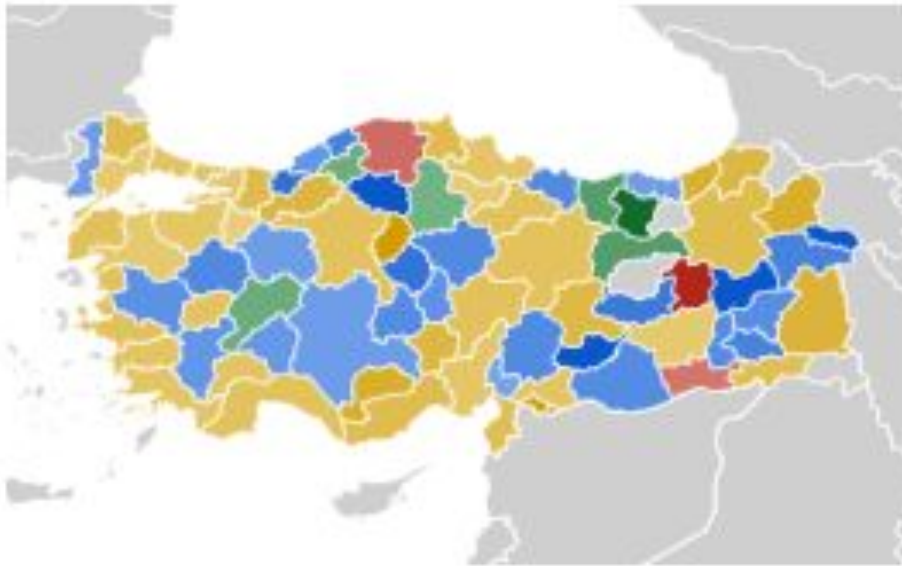
Tartışınız

- ▶ NEDEN JAVA?

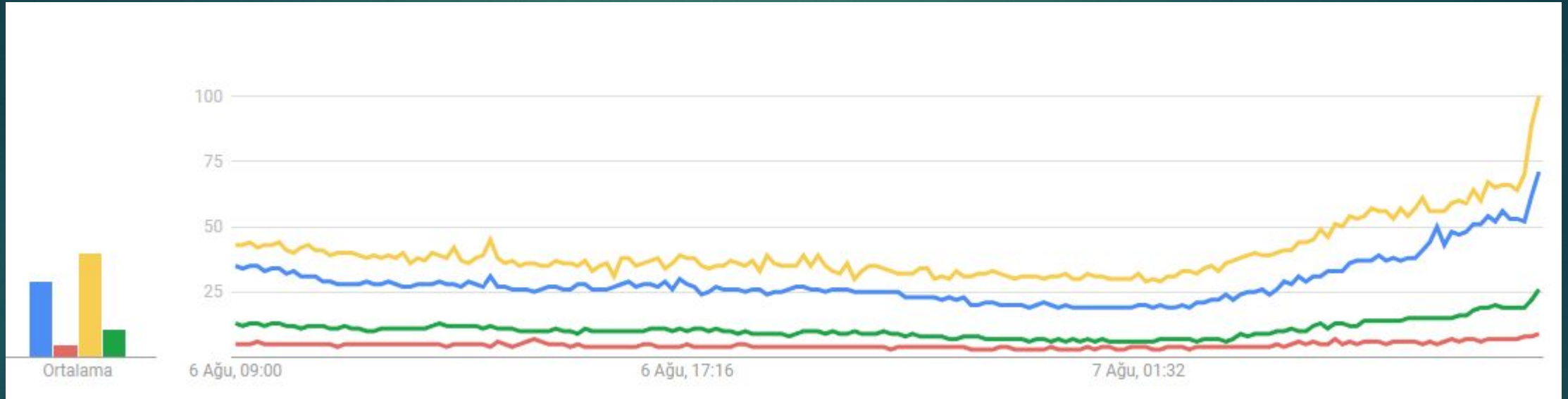
Java Karşılaştırma Türkiye

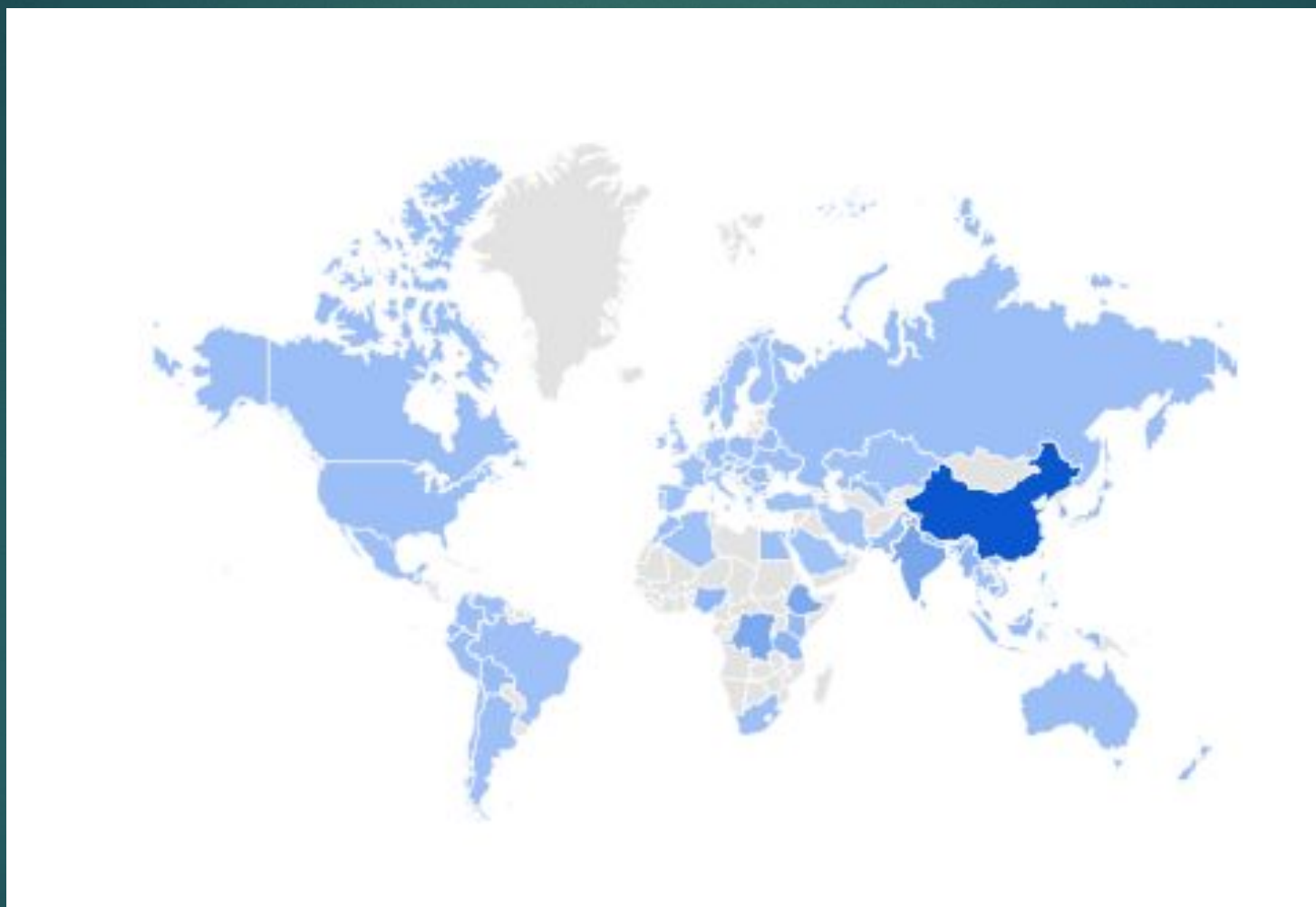


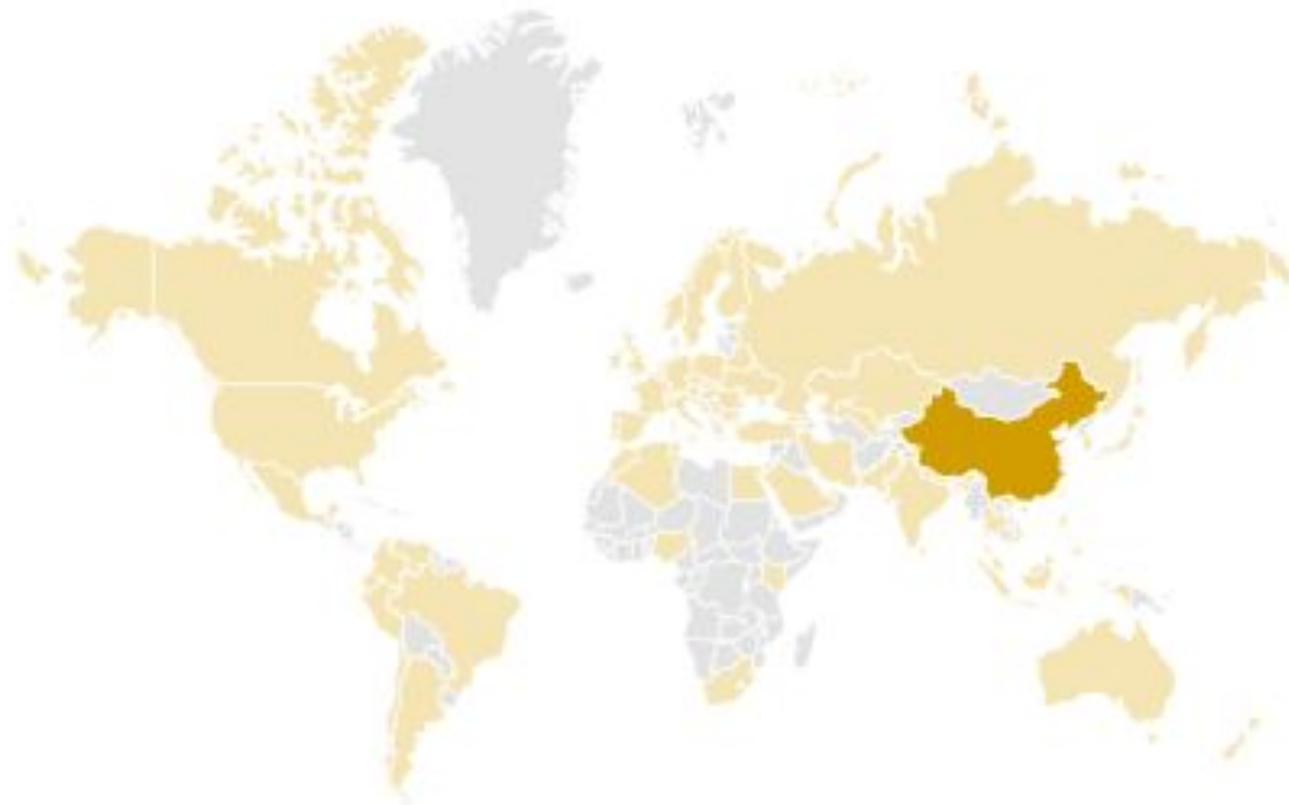
● Java ● .Net ● Python ● C++



Java Karşılaştırma Dünya







Maven Nedir?

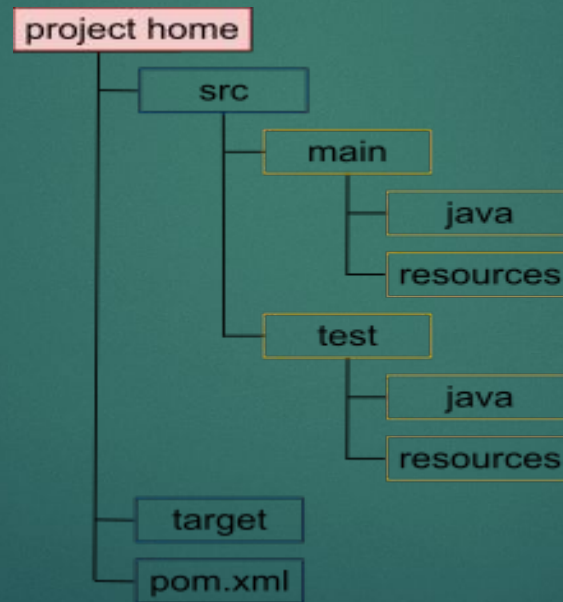
- ▶ Maven, Apache Yazılım Vakfı tarafından geliştirilen bir proje yönetim ve otomasyon aracıdır.
- ▶ Proje yapılandırmasını ve bağımlılıklarını kolayca yönetmeyi sağlar.

Maven'in Temel Kavramlar

- ▶ Proje: Maven tarafından yönetilen bir yazılım projesini temsil eder.
- ▶ POM (Project Object Model): Proje yapılandırmasını ve bağımlılıkları içeren XML dosyasıdır.
- ▶ Bağımlılık: Projenin dışarıdan kullandığı kütüphaneleri ifade eder.

Maven Proje Yapısı

- ▶ src/main, src/test gibi klasörler, ana kaynak kodları ve test kodları için ayrılmıştır.
- ▶ POM dosyası proje yapılandırmasını ve bağımlılıklarını tanımlar.



Bağımlılıkların Yönetimi

- ▶ Maven, merkezi bir depo üzerinde barındırılan kütüphaneleri kolayca yönetmeyi sağlar.
- ▶ Bağımlılıklar, POM dosyasında tanımlanır ve Maven, bu bağımlılıkları otomatik olarak indirir ve projeye ekler.

Temel Maven Komutları

- ▶ `mvn clean`: Proje derlemesi öncesinde oluşturulan dosyaları ve klasörleri temizler.
- ▶ `mvn compile`: Proje kaynak kodlarını derler.
- ▶ `mvn test`: Proje testlerini çalıştırır.
- ▶ `mvn package`: Proje paketlemesini (JAR, WAR, vs.) yapar.
- ▶ `mvn install`: Paketlenen proje dosyalarını yerel Maven deposuna kurar.