

# Java Giriş

---

Java günümüz yazılım dünyasının en popüler dillerinden birisidir hatta kurumsal yazılım dünyası olarak düşünerek olursak en popüleride diyebiliriz. Peki böyle bir platforma neden ihtiyaç duyuldu ?

Aslında Java' nın çıkış noktası platform bağımsızlığı. 90 lı yılların başlarında James Gosling liderliğindeki Sun Microsystems mühendisleri Green Project kodlu bir projeyle tüm platformlarda çalışacak bir mimari yaratmayı amaçladılar. Bu anlamda ilk olarak OAK isminde bir platform ortaya oluşturuldu. Aslında bu Java' nın ilk versiyonuda diyebiliriz. Buradaki amaç çamaşır makinesi, buzdolabı, televizyon gibi yani nesil akıllı ortamlarda platform bağımsız çalışacak uygulamalar yaratmaktı. Ancak daha sonralarında, Windows, Linux gibi sistemlerin popüleritesini artırması ve internet in iyice yaygınlaşması ve ekibi Java platformunu gerçekleştirmeye itti.

Java' nın ilk versiyonu yani Java 1.0 1994 senesinde yayınlanmıştır. Bundan sonraki süreçlerde bu gün 1.7 versiyonuna kadar gelinmiştir.

Java' nın gerçek gücü internete dayanıyordu. Özellikle ilk yıllarda popüler olan applet mimarisi internet sayesinde her ortamda her platformda çalışan browser tabanlı uygulamalar getiriyordu. Bu uygulama yapısını her ortamda çalışabilen kurulum gerektirmeyen hale getiriyordu.

Bunlar dışında Java' nın asli özelliklerini sayacak olursak basit, güçlü, platform bağımsız, güvenli, dinamik, yeniden kullanılabilir, çoklu hizmet verebilen ve yorumlanan bir platformdur diyebiliriz.

## Java dili mi ? Java platformu mu ?

Burada dil demek kadar doğru diye düşünebilirsiniz. Java aslında hem dil hemde platform ismi olarak kullanılmaktadır. Örneğin .net bir platformken c# veya vb.net bir dil ismidir. Java da hem platforma hemde dile Java ismi verilmektedir.

## Java Mimarisi

Java ile geliştirdiğimiz uygulamalar direkt olarak makine koduna derlenmemektedir. Bunun yerine Java Sanal Makinesi dediğimiz JVM (Java Virtual Machine) ' nin anlayabileceği byte kod şeklinde derlenmektedir. Bu sayede JVM olan tüm platformlarda Java uygulamaları çalışabilmektedir. Bu Java' nın önemli noktalarından biridir. Biz bu sayede platform bağımsız uygulama geliştirme imkanına sahip olmuş oluyoruz. Örneğin cep telefonları için yazdığımız bir uygulama Java destekli tüm cep telefonları için çalışabiliyor. Buradaki "Java destekli cep telefonu" kelimesi o telefonda bir sanal makine olduğunu kastediyor. Bu durumda sanal makine olan tüm telefonlarda Java uygulamalarımızı çalıştırabilme imkanına sahip oluyoruz.



Java Uygulaması

Sanal Makine

İşletim Sistemi

Bunun haricinde Java’ daki bir başka önemli konu bellek kullanımı. C++ ta ki önemli zorluklardan biri bellek üzerindeki adres alanlarını bizişm yönetiyor olmamızdı. Mesela bellek üzerinde var olan bir alan artık kullanılmıyorsa bizim bu alanı serbest bırakmamız gerekiyordu. Bu Java tarafında Garbage Collector (çöp toplayıcısı) mimarisi ile yönetiliyor. Eğer bellek üzerinde işaret edilmeyen bir alan varsa bu alanlar Garbage Collector tarafından otomatik temizleniyor.

## Java Platformları

**Java SE (Java Standard Edition)** : Java’ nın standart sürümünü ifade eder. Masaüstü, tek başına çalışabilen (Sandalone) gibi uygulamaları kapsar Java dilini taşır. Temel sürümdür de diyebiliriz.

**Java ME (Java Micro Edition)** : Java’ nın küçük cihazlar için çıkarılmış sürümüdür. Kusutlanmış ve özel kütüphanelere sahiptir. Örneğin cep telefonları gibi ortamlar için kamera veya bluetooth gibi özel apiler içerir. Bunun yanında bu tarz cihazlar işlemci ve bellek açısından daha kısıtlı olduğundan Java SE’ ye göre bazı kütüphaneleri kısıtlanmıştır.

**Java EE (Java Enterprise Edition)** : Java’ nın kurumsal uygulamalar için olan sürümünü ifade eder. Özellikle çok katmanlı kurumsal mimarileri hedefleyen bir çok bileşen içerir. Bunlar JSF, JSP gibi web mimarileri olmasının yanında EJB, JMS, Web Servisleri gibi özel bileşen mimarileri de olabilir. Java EE uygulama suları EJB gibi bu standartlardaki bileşenlerin koşturulmasını sağlar.

**Java Card** : Java’ nın akıllı kartlar için çıkarılmış sürümüdür. Bir akıllı kartın Java ile programlanabilmesi için Java Card desteğinin olması gerekmektedir. Buna örnek olarak cep telefonları için kullanılan sim card lar veya çipli kredi kartları verilebilir.

## Java’ nın Çalışma Alanları

JVM (Java Virtual Machine – Java Sanal Makinesi) sayesinde Java; Windows, Linux, Unix, Mac OS gibi bir çok platformda çalışabilir. Daha doğrusu Java ile geliştirdiğimiz uygulamalar bu platformlarda çalışabilir. Çünkü temel prensip olarak Java’ nın platform bağımsızlığını sağlayan JVM mimarisinin olduğu her Java uygulamalarımızı çalıştırabiliriz.

Bunlara örnek olarak aşağıdaki platformları sayabiliriz.

Masaüstü Uygulamalar	Cep Telefonları	Akıllı Cihalar	Akıllı Kartlar (Java Card)
Web Uygulamaları	Lego Oyuncakları	Televizyonlar (Java TV API)	

## Java Sürümleri

Java sürümleri aşağıdaki gibi ilerlemektedir.

<b>JDK 1.1</b>	<b>1997</b>		AWT, JDBC, Inner Class, Java Beans vs...
<b>J2SE 1.2</b>	<b>1998</b>		Reflection, Collections Framework, IDL, Swing ve JIT
<b>J2SE 1.3</b>	<b>2000</b>		HotSpot, JavaSound, JNDI, JPDA
<b>J2SE 1.4</b>	<b>2002</b>	<b>JSR-59</b>	Regular expression, XML ve XSLT için JAXP, Java Web Start
<b>J2SE 5.0</b>	<b>2004</b>	<b>JSR-176</b>	For-each, generic ler, annotation, autoboxing, varargs
<b>Java SE 6</b>	<b>2006</b>	<b>JSR-269</b>	Yeni annotation lar, GUI kütüphanesinde değişiklikler
<b>Java SE 7</b>	<b>2011</b>	<b>JSR-336</b>	Temel dil değişiklikleri, SCTP protokolü, IO üzerinde değişiklikler

Burada dikkat çeken bir kaç nokta var.

Öncelikle isimlendirmelerde 1.2 versiyonu ile birlikte J2SE olarak bir değişiklik var. Java' nın 1.2 versiyonunda ciddi değişiklikler olmasından dolayı isimlendirme J2SE olarak yapılmaya başlandı. Benzer değişiklikler 1.5 versiyonunda da olmasından dolayı J3SE demek yerine isimlendirmeler 5.0 olarak devam etti. Ancak 6 ile birlikte artık isimlendirme standardı Java SE 6 olarak son şeklini almıştır. Bu aslında orjinal versiyon olarak 1.6 olarak ta isimlendirilmektedir. Özellikle eski Java cılar 1.6 da demeye devam etmekte. Ancak doğru isimlendirme Java SE 6 şeklindedir.

Bunun haricinde görünen başka bir şey daha var 1.4 sürümüyle birlikte JSR kodları geliyor. Java 1.4 ten itibaren standartlar JCP (Java Community Process) tarafından planlanmaktadır. JCP yani Java Community Process bağımsız bir gruptur diyebiliriz. Bu community içerisinde Oracle, IBM gibi bir çok kurum üyedir ve JCP içerisinde yeni konular hakkında JSR' lar (Java Specification Request) açılıp bu JSR' lar oylanır ve bir belirtim (Specification) ortaya çıkarılır. Bu aynı zamanda Java hakkındaki konuların ortak karar verebilmesini sağlamaktadır. Yani aslına bakarsanız Java' nın lisans hakları Oracle da olmasına karşın karar mercii halen JCP' dir.

JSR içerisinde bir çok önemli konu vardır örnek olarak JSR 003 JMX' i ifade ederken JSR 314 JSF 2.0 i ifade etmektedir.

## Java Açık Kaynak mı ?

2006 yılındaki Java One konferansında Sun Microsystems Java' nın açık kaynak olacağını açıkladı. 13 Kasım 2006 yılından itibaren ise Java GPL (General Public License – Genel Kamu Lisansı) daha doğusu GNU GPL (Kısıtlı Genel Kamu Lisansı) ile açık kaynak olarak yayınlanmaya başlandı. Tabi burada bazı bileşenler ve özel kapsam dahilindedir.

Tüm hakları Melih Sakarya' ya aittir izinsiz dağıtımı ve kullanımı yasaktır.

Java' nın kaynak kodları bu şekilde lisanslansa da kullanım olarak farklı lisanslara da sahiptir. Örneğin JVM üzerinde yapacağınız şeyler Java' nın lisans kurallarına aykırı olmamalı. Bunlar biride platform bağımsızlıktır. Java üzerinde yaptığınız optimizasyonlar platform bağımsızlığı engellerse bu durumda Oracle tarafından açılacak bir tazminat davası ile karşı karşıya kalabilirsiniz.

Sun içerisinde Sun's Binary Code License (BCL) isimli özel bir lisanslama modeli vardır. Bu model içerisinde örneğin embedded mimariler için ayrı başlıklar bulunur. Tabi bu bağlayıcılıklar son kullanıcıları değil üretici ve gerçekleştirimi sağlayanları etkilemektedir.

Tüm bunlar dışında Java tabanlı bir çok açık kaynak ürüne sahip şirket ve grup bulunmaktadır. Bunlar arasında Apache ve JBoss gibi kurumlar örnek verilebilir.

## Java Paketleme Standartları

Java ile geliştirdiğimiz uygulamalar belirli standartlarda paketlenilmektedir. Buna örnek verilecek olursa windows ortamında exe yada dll gibi paketler ile karşılaşılıyor olabiliriz işte bunların Java tarafındaki karşılıkları jar, war ve ear' dır. Bu paketler standartları Java tarafında otomatik olarak algılanırlar.

**JAR (Java Archive)** : Java' nın standart paketlemesidir. Bir çok sınıf veya başka kaynağı tek bir paket haline getirmeye yarar. Zip tabanlı bir sıkıştırma format içerir. Jar dosyaları bir kütüphane veya tek başına çalıştırabilir uygulamalar olabilir. Eğer direkt çalıştırılabilir uygulamalar olması isteniyorsa çalıştırılma anında manifest tanım dosyasına bakılır. Manifest dosyası jar içerisinde META-INF/MANIFEST.MF altında olmalıdır. Bu dosya içerisinde aşağıdaki gibi jar paketine ait ana sınıfın ismi bulunmalıdır. Bu sınıf jar çalıştırıldığı anda ilk çalışacak sınıfı ifade etmektedir.

Main-Class: paketAdi.AnaSınıf

Bir jar paketini çalıştırmak için ise aşağıdaki komut kullanılabilir.

```
java -jar uygulama.jar
```

Eğer paket ten çıkarmak istiyorsak aşağıdaki komutu kullanabilirsiniz.

```
jar -xf uygulama.jar
```

**WAR (Web Application Archive)** : Java' web uygulamaları için paketleme standartıdır. Yine bir sıkıştırma formatı içerir. Java Web Container' lar (Örneğin Tomcat veya Jetty gibi) war formatını otomatik olarak tanıyıp container üzerinde koşturulmasını sağlarlar. War içerisinde WEB-INF özel bir klasördür. Bu klasöre dışarıdan erişim kısıtlanmış olup web.xml adında bir tanım dosyası içerir. Bu tanım dosyası Java web uygulamaları için ana tanım dosyasıdır. War içerisinde kütüphane şeklinde jar paketleri yer alabilir bu jar paketleri WEB-INF/lib/ içerisinde yer alır.

**EAR (Enterprise Archive)** : Java EE için kullanılan paketleme standartıdır. İçeriğinde war ve jar gibi paketler yer alabilir. Bu paketler kütüphaneler olabileceği gibi birer modül olarak ta tanımlanabilir.

Tüm hakları Melih Sakarya' ya aittir izinsiz dağıtımı ve kullanımı yasaktır.

Örnek verilecek olursa war bir ear paketi içerisinde web modülü olarak geçebilir. Bunun haricinde EJB (Enterprise Java Beans) paketi bir jar olarak modül tanımında yer alabilir. Ear içerisinde META-INF altında application.xml adında bir deployment descriptor yani ear için bir dağıtım tanım dosyası yer alır. Bu dosya içerisinde modül tanımları ve ear standartlarındaki diğer tanımlar yer alabilir.

### Java Kullanan ve Destekleyen Başlıca Şirketler

Oracle	Siemens
IBM	Google
SAP	Yahoo
EMC	RedHat
HP	Samsung
Adobe	Motorola
Alcatel Lucent	RIM
Accenture	Ericsson
Huawei	Nokia
Intel	Fujitsu
AT&T	Sony