

# Java Nedir?

**Akın Kaldırođlu**

**[www.javaturk.org](http://www.javaturk.org)**

Şubat 2009

# Akın Kaldırođlu

- Akın Kaldırođlu 1993 yılından bu yana yazılım geliřtirmektedir.
- Mesleđe C ve C++ ile bařladı, 1996'da ilk sürümüyle birlikte Java'ya geđti ve daha sonra onun üzerine bařka bir gül koklamadı 😊.
- 1993 yılından bu yana ABD ve Türkiye'de pek çok projede deđiřik rollerle yer almaktadır.
- Ayrıca, Java, nesne-merkezli programlama, tasarım řablonları vb. pek çok konuda ders vermekte ve Türkiye'de sađlıklı bir yazılım kùltürü oluřmasını arzu etmektedir.
- Yakınlarda <http://www.javaturk.org> adresinde günlük yazmaya bařlamıř olup, katkıda bulunacaklara řimdiden müteřekkirdir.
- Kendisine günlüđünden, [akin@javaturk.org](mailto:akin@javaturk.org) adresinden ve diđer Facebook/LinkedIn gibi alanlardan eriřilebilir.

# Gündem

**Java'nın Kısa Tarihi**

**Java Nedir?**

**Java Nasıl Gelişmektedir?**

**Java'ya Başlama ve Sertifikasyon**

➤ Bu sunumda 48 sayfa vardır.

**Java'nın Kısa Tarihi**

**Java Nedir?**

**Java Nasıl Gelişmektedir?**

**Java'ya Başlama ve Sertifikasyon**

# Java'nın Kısa Tarihi

- 1991 yılında Sun, kapalı kapılar ardında ve Patrick Naughton, Mike Sheridan, and James Gosling önderliğindeki küçük bir takımla bir proje başlattı.
- 13 kişiden oluşan ve “Green Team” (Yeşil Takım) adı verilen bu grubun amacı ITC dünyasında geleceği tahmin edip ona göre ARGE yapmaktı.
- Ve bu takım da geleceğin, kişisel ve ev elektronik cihazlarında olduğunu öngörüp bu alana yöneldi.

# \*7

- 18 aylık hummalı bir çalışmadan sonra 1992 yazında “\*7”, “Star Seven” isimli ve dokunmalı ekrana sahip bir kontrol cihazı geliştirdiler.
- Bu cihaz TV, video oynatıcısı ve müzik seti gibi pek çok ev cihazını kontrol edebiliyordu.
- Cihazın bu yeteneği, platformdan bağımsız bir dile sahip olmasından kaynaklanıyordu.
- James Gosling bu dile “Oak” adını vermişti.



# Java ve Internet

- Bu proje pek çok kablolu TV şirketine teklif edildi ama zamanının biraz ilerisinde olduğundan kabul görmedi.
- Ama takımın ileri gelenleri o sırada, geliştirdikleri altyapının Internet'e çok uygun olduğunu farkettiler.
- Ve hemen yönlerini değiştirip, ismi daha sonra resmi olarak HotJava olan Java-tabanlı bir tarayıcı geliştirdiler: WebRunner.



# İçgiyim Olarak Oak

- Gosling explains: "We had already been developing the kind of `underwear' to make content available at the same time the Web was being developed. Even though the Web had been around for 20 years or so, with FTP and telnet, it was difficult to use. Then Mosaic came out in 1993 as an easy-to-use front end to the Web, and that revolutionized people's perceptions. The Internet was being transformed into exactly the network that we had been trying to convince the cable companies they ought to be building. All the stuff we had wanted to do, in generalities, fit perfectly with the way applications were written, delivered, and used on the Internet. It was just an incredible accident. And it was patently obvious that the Internet and Java were a match made in heaven. So that's what we did."



# Oak'tan Java'ya

- Oak ismi bu sırada Java olarak değiştirildi
- Ve 23 Mayıs 1995'te Java resmi olarak piyasaya sunuldu
- Ve o tarihlerde ne Java'yı geliştirenlerin ne de bir başkasının tahmin edemeyeceği yoğunlukta bir ilgiyle karşılaştı

**Java'nın Kısa Tarihi**

**Java Nedir?**

**Java Nasıl Gelişmektedir?**

**Java'ya Başlama ve Sertifikasyon**

# Java Nedir?

- Sun 1995'te Java'yı sunarken yayınladığı bir yazıda Java'yı şöyle tanıtıyordu:
  - Java: Basit, nesne merkezli, ağlarda yetenekli, yorumlanan, sağlam, güvenli, mimari olarak tarafsız, taşınabilir, yüksek performanslı/başarımlı, çok kanallı, dinamik bir dil.
  - Java: A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language

# Java Basittir I

- Java'nın o kadar da basit değildir aslında
  - Java, ataları olan C ve C++ dillerinden daha basittir
- Çünkü bu dillerde olan ve karmaşıklık getiren özellikleri ve yapıları içermez:
  - Pointer aritmetiği
  - Bellek yönetimi (Memory management)
  - İşlem Yükleme (Operator Overloading)
- Dolayısıyla Java daha yüksek seviyeli bir dildir

# Java Basittir II

- Java bu dillere göre aşağıdaki avantajlara sahiptir
  - Platforma bağlı olmayan basit veri yapıları
  - Daha derli toplu geliştirme ortamları
- Daha küçük ve sade bir dil ile daha güçlü olmayı tercih eder
  - Ama son yapılan eklemelerle Java nispeten büyüdü

# Java Nesne Merkezlidir

- Java nesne merkezli bir dildir.
- Nesne merkezli diller, daha çok aşına olunan prosedürel/yordamsal dillerden çok farklıdırlar.
- Yordamsal dillerde en önemli kavram/soyutlama yordam/prosedür/fonksiyondur ve bu yapılar, bir problemi alt problemlere bölüp, her birini adım adım (step-wise decomposition) tanımlamakta kullanılır
- Fakat nesne merkezli diller nesne kavramı üzerine otururlar

# Nesne Nedir?

- Nesne, insan zihninin kendisine yöneldiği, özellik ve davranışlara sahip, fiziksel olan ya da olmayan herhangi bir varlıktır.
- Yazılımda her tip nesne için, verisi ile davranışını bir paket haline getirip sarmalayan (encapsulation) ve adına genelde sınıf (class) denen şablonlar oluşturulur.
  - Nesnelere, şablonlardan üretilir.
- Nesnelerin çoğu zaman gerçek dünyadaki varlıkları ifade ettikleri düşünüldüğünde, nesnelerin özellikleri gerçek varlıkların özelliklerine, davranışları ise varlıkların birbirleriyle olan iletişimine karşılık gelir

# Alan Key'in 5 Kuralı

- Herşey bir nesnedir
- Program birbirlerine mesaj (message) göndererek ne yapmalarını belirten nesnelere oluşur
- Her nesne diğer nesnelere (association/composition) oluşur ve hafızada yer kaplar
- Her nesnenin bir tipi (type) vardır
- Aynı tipten olan nesnelere aynı mesajları alabilirler



# Java Ağlarda Yeteneklidir

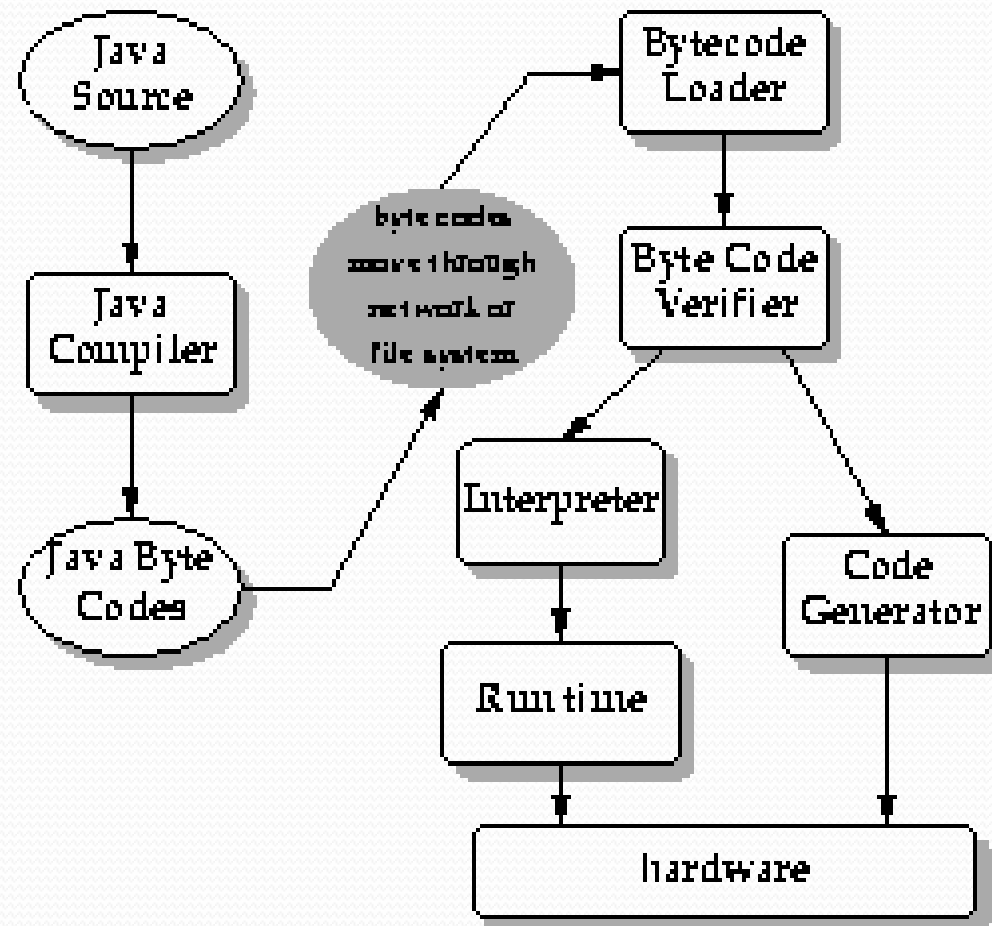
- Tarihi olarak Java'nın ve Internet devriminin doğumu aynı döneme rast gelmiştir.
- Bu yüzden Java, ağ ve ilgili yapılar akılda tutularak tasarlanmış ilk dildir.
- Java ile bilgisayar için değil ağ için yazılım geliştirirsiniz.
  - HTTP ve HTTPs gibi en yaygın Web protokolleri
  - Web Servisleri
  - Uzak nesnelere (remote objects)
  - Ve Telnet, FTP vs. gibi diğer bütün protokoller

# Java Yorumlanır

- Aslında Java hem derlenen (compiled) hem de yorumlanan (interpreted) bir dildir.
- Java kaynak kodları (.java dosyaları) Java bytecodelarına (.class dosyaları) derlenir.
- Bytecodelar JVM'in komutlarıdır ve çalıştırma zamanında (run-time) JVM tarafından yorumlanır.
- Dolayısıyla Java'da kod yazarken, nihai çalışma ortamını bilmeniz gerekmez.

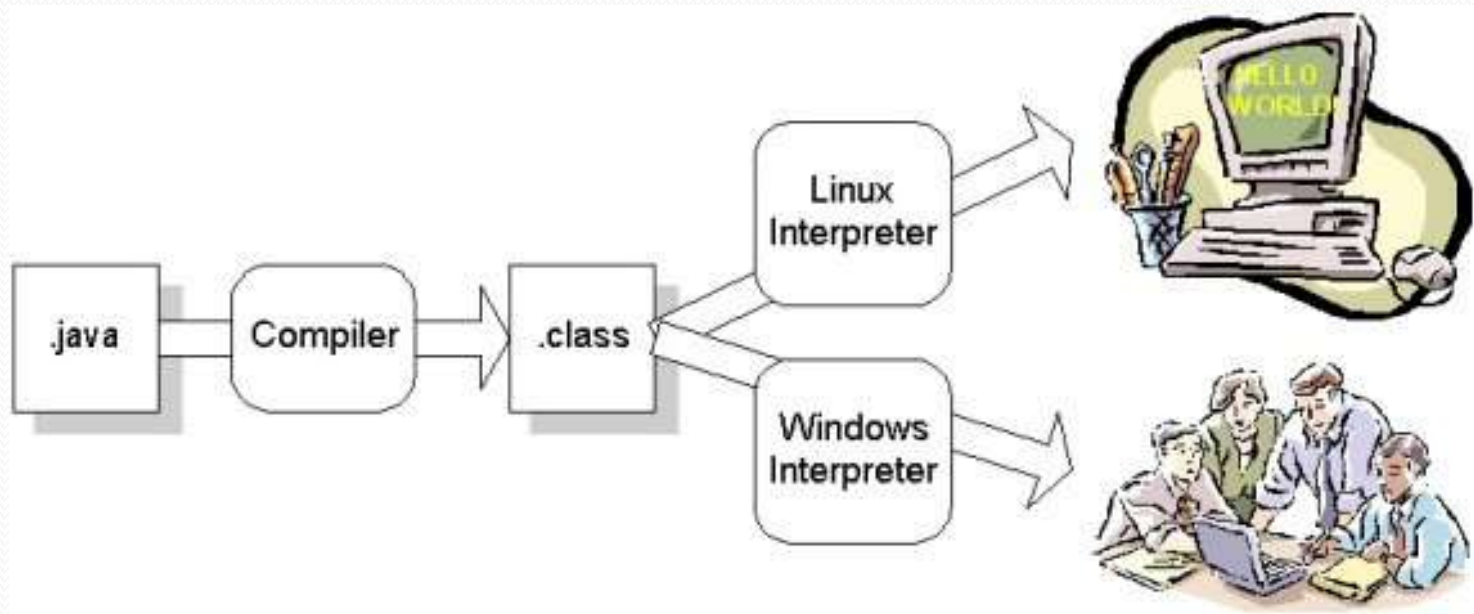
## Compile Time

## Run Time



# JVM

- JVM (Java Virtual Machine ya da Java Sanal Makinası), donanım ve işletim sistemi üzerinde çalışan sanal bir platformdur.
- Her platform için en az bir JVM mevcuttur.
- Derlenmiş Java kodlarıyla platform arasında bir katman oluşturur ve aynı bytcodeoların farklı platformlarda değişmeden, aynı şekilde çalışmasını sağlar.



```
// Selam.java
```

```
public class Selam{  
  
    public static void main(String[] args){  
  
        Selam object = new Selam();  
        object.saySelam();  
  
    }  
  
    public void saySelam(){  
        System.out.println("Selam!");  
    }  
}
```

**Sonuç**  
Selam!

# Java Taşınabilirdir

- Java tanımlı standartlar/teknik şartnameler (specifications) üzerine kurulmuştur.
  - Bu açıdan Java'da gerçekleştirmeye (implementation) bağlı hiçbir noktanın olmaması amaçlanmıştır.
    - Örneğin int veri tipi her yerde 32 bittir.
- Ayrıca, Java dili ve JVM, platformlarla alakalı sadece en genel önkabullere sahiptir.
- Bu durum Java'nın platform bağımsızlığı ve taşınabilirliğinin sağlayıcısıdır ve Java programcısına üzerinde çalıştığı ortamdan bağımsız davranma gücünü verir.

# WORA

- Başından bu yana *bir kere yaz her yerde çalıştır* (Write Once, Run Anywhere, WORA) Java'nın taşınabilirlik konusundaki hedefidir.
- Tabii olarak bu prensipin geçerli olması için programcı da herhangi bir platforma özel kod yazmamalıdır.
- Java'yı uzunca süredir uygulayanlar, *bir kere yaz her yerde test et* (WOTA) prensibinin daha doğru olduğu üzerinde hemfikirdirler.



# Java Sağlamdır

- Sağlamlık (robustness) çalışma zamanı güvenilirliği olarak tanımlanır.
- Java, dayanıklı (reliable) ve sağlam yazılımlar üretmek için tasarlanmıştır.
  - Java güçlü bir tip sistemine sahiptir dolayısıyla hem derleme zamanında hem de çalıştırma zamanında tip uyumu ve çevrimi (cast) kontrol edilir.
  - Sıradışı durumları ele almak için mekanizma oluşturulmuştur (exception handling).
  - Hafıza işletimi otomatiktir, programcıya bırakılmamıştır.
  - Ve daha pek çok derleme/çalışma zamanı kontrollerini içerir.

# Java Güvenlidir

- Java'nın ağ ortamlarında kullanılmak için tasarlanmış olduğu belirtilmişti.
- Bu yüzden Java, ağ ortamlarında bulunmanın getirdiği tehlikelere karşı koyacak şekilde yapılandırılmıştır.
- Yabancı makinelerdeki nesnelerin, yerel makinedeki nesnelere ve işletim sistemine erişimi kontrol altındadır.
  - Java'nın ilk yıllarında duyulan güvenlik açıklarına artık pek rastlanılmamaktadır.
- Java'daki güvenlik API'ları sayesinde şifrelemeden (encryption) sayısal imzalara (digital signatures) kadar her türlü güvenlik kodu yazılabilir.

# Java Mimari Olarak Tarafsızdır

- Java, birbirlerinden farklı, pek çok değişik platformun bulunduğu ağ üzerinde çalışmak için tasarlandığından, herhangi bir platformu varsaymaz, platformlardaki ortak özellikler dışında herhangi bir ön kabulü yoktur.
- Bu durum Java'nın sanal bir makina üzerinde çalışması ve hiç bir platforma varsaymayan bir şekilde, standartlar üzerine inşa edilmesinin bir sonucudur.

# Java Yüksek Başarımlıdır I

- Bu özelliği tabiki açıklanmaya muhtaçtır.
- Nesne-merkezli diller, tip hiyerarşileri ve polimorfizm gibi özelliklerinden dolayı zaten çalışma zamanında çok iş yaparlar.
  - Java, taşınabilir ve mimari olarak tarafsız olmak için sanal makina gibi bir ara katman üzerinde çalışmayı tercih etmiştir.
  - Ayrıca Java sağlam bir mimari kurgulamak için daha pek çok çalışma zamanı kontrollerine sahiptir.
- Bu durum Java'yı tabii olarak çalışma zamanında daha yavaş kılmaktadır.

# Java Yüksek Başarımlıdır II

- Java'nın çalışma zamanı performansını arttırmak için çok mesafe katedilmiştir.
  - Etkin derleyici (Optimized compiler)
  - Etkin JVM, JIT ve HotSpot teknolojileri
- Java'cılar da performanslı kod yazmak için hem hız hem de bellek tüketimi açısından, diğer dilleri kullananlara göre daha bilgili olmalıdırlar.

# Java Yüksek Başarımlıdır III

- Ama unutulmamalıdır ki projeler daha çok ihtiyaçların yanlış kavranması vs. gibi sebeplerden başarısız olmaktadır, performanstan değil.
- Ayrıca performans ve ölçeklenebilirlik (scalability) genel olarak mimarinin bir özelliğidir ve mimariler bu konulardaki hedefler gözönüne alınarak tasarlanmalıdır.

# Java Çok Kanalıdır

- Kanal (thread), bir işlem/süreç (process) içinde oluşturulan hafif hafif bir işlem/süreçtir (lightweight process).
- Aynı işlem içinde çalışan farklı kontrol akışlarına kanal denir.
- Kanallar aynı işlem içinde çalıştıklarından, birçok işi aynı anda ve daha hızlı ve etkin olarak yerine getirebilirler
- Böylece daha yetenekli ve hızlı cevap veren yazılımlar geliştirilebilir.
- CPU ve platformların çok hızlı hale gelmesi, dillerdeki çok kanallı özellikleri kullanmamızı zorlamaktadır.

# Java Dinamiktir

- .class dosyalarındaki Java tiplerinin daha program derlenirken hazır olmaları şart değildir, JVM'e çalıştırma zamanında her hangi bir anda yüklenebilir.
- Dolayısıyla pek çok dilde var olan derleme zamanındaki statik bağlantı kurma (static linking) Java'da yoktur.
- Bu da ağ üzerinde çalışan ve derleme sırasında erişilmesi mümkün olmayıp, çalıştırma zamanında ulaşılan nesnelerin dinamik olarak JVM'e yüklenebilmesini sağlar.



**Java'nın Kısa Tarihi**

**Java Nedir?**

**Java Nasıl Gelişmektedir?**

**Java'ya Başlama ve Sertifikasyon**

# Java'nın Sahibi Kimdir?

- Excel'in sahibinin Microsoft olması anlamında Java'nın sahibi yoktur.
- Java'nın orijinal geliştiricisi olan Sun, Java'nın lisanslayıcısıdır.

# Java'yi Kim Geliştirmektedir?

- Java, JCP (Java Community Process, <http://www.jcp.org> ) tarafından geliştirilmektedir.
- JCP, kişi ve kuruluşların serbestçe katılabildiği ve kararların oy çoğunluğu ile verildiği milletlerarası bir ortamdır.
- Çok farklı bilgi ve tecrübeye sahip binlerce kişi ve kurum JCP çatısının altında Java'nın gelişimini belirlerler.

# JSR I

- JCP'nin üyeleri yeni bir Java teknolojisini Java Specification Request (JSR) mekanizması ile teklif edebilirler.
- Her JSR'ın bir yürütücü heyeti ve teknik bir lideri vardır.
- JSR'lar, içinde halka açık incelemenin de (public review) bulunduğu bir dizi sayfadan geçip tamamen demokratik bir oylama ile sonuçlandırılırlar.
- Her ne kadar Sun, JCP'de çok etkin olsa ve bu yüzden eleştirilse de java'nın geleceği çok daha demokratik yollarla belirlenmektedir.

# JSR II

- Kabul edilen her JSR aşağıdaki üç şeyi üretmek zorundadır:
  - API'ı da içeren bir tarifname (specification)
  - Bir referans olan gerçekleştirme (reference implementation)
  - Teknoloji uyumluluk paketi (technology compatibility kit)
- Dolayısıyla her Java teknolojisi gerçekleştirilmesi, bu açık ve üzerinde mutabakat oluşmuş tarifname ve API'lar üzerine bina edilirler
- İsteyen her kişi ve kurum da bu tarifnameler üzerine ürün geliştirip, bedava ya da ücretiyle lisanslayabilir.

# Özgür Yazılım

- Java'nın bu açık ve standartlara dayalı yapısı özgür yazılımların gelişmesine de yardımcı olmaktadır.
- Java bugün açık kaynak kodlu (open source) yazılım felsefesine gönül verenlerin en fazla kullandıkları dil haline gelmiştir.
- Bu sayede hemen her Java teknolojisinin ticari olmayan, açık kaynak kodlu gerçekleştirilmesi mevcuttur.

# Java'nın Tipleri

- Sun, tek bir tipin yeterli olmayacağını düşünerek, Java'nın üç farklı tipini oluşturmuştur:
  - Standart Java (Standard Edition, SE)
  - Mikro Java (Micro Edition, ME)
  - Kurumsal Java (Enterprise Edition, EE)
- Şu anda
  - Java SE'nin 6. sürümü vardır, 7.si çıkmak üzeredir,
  - Java EE'nin 6. sürümü Aralık 2009'da çıkmıştır
  - Java ME'nin 3.0 sürümü vardır.

**Java'nın Kısa Tarihi**

**Java Nedir?**

**Java Nasıl Gelişmektedir?**

**Java'ya Başlama ve Sertifikasyon**



# Java'ya Nasıl Başlarım?

- Java'ya <http://java.sun.com> adresine gidip JDK'in en son sürümünü indirip makinalarınıza kurarak başlayabilirsiniz.
- JDK'i kurduğunuz yerdeki *bin* klasörünü *path*'e koyup, *javac* ile kodunuzu derleyip, *java* ile de çalıştırabilirsiniz.
- İsterseniz Netbeans (<http://www.netbeans.org>) ya da Eclipse (<http://www.eclipse.org>) gibi geliştirme ortamlarını da kullanabilirsiniz.

# Java'nın Tabiatı ve Zorlukları I

- Java nesne-merkezlidir:
  - Artık prosedürel değil nesne merkezli düşünmeniz gereklidir.
- Java tarifname temellidir:
  - Dolayısıyla tarifname ve onu gerçekleştiren farklı ürünlerin olduğunu anlamanız gereklidir.
  - Dolayısıyla aynı işe yapan değişik ürünler vardır.
- Java özgür ve açık kaynak kodlu yazılımların en yoğun olduğu dildir:
  - Dolayısıyla aynı işe yapan açık kaynak kodlu pek çok farklı ürün vardır.

# Java'nın Tabiatı ve Zorlukları II

- Java, yazılım mimarilerine, tasarım şablonlarına (design patterns) önem verir ve onları kullanır.
  - Java'da düşünmek ve modellemek yazmaktan daha önemlidir
    - Küçük bir şey yapmak için bile hiç tahmin edemeyeceğiniz kadar düşünmek zorunda olursunuz.
- Dolayısıyla örneğin sürükle-bırak yoğun, 2000 satırlık ASP ya da PHP sayfası yerine, çoğunlukla satır satır yazılmış ama çok daha modüler ve şablonlara uygun tasarlanmış 50-100 satırlık JSP sayfaları yazmak gereklidir.

# Java Sertifika Sınavları

- *Sun Certified Java Programmer* olmak “*tecrübem yok ama temellerini iyi biliyorum*” demektir.
- Java sertifikaları hakkında <http://www.sun.com/training/certification/java/index.xml> sayfasından geniş bilgi alabilirsiniz.
- Uzun süredir Java sertifikalarına yönelik Sun’ın standart eğitimlerini vermekteyim.

# Sun'in Java Sertifikalari



# Sun'ın Java Sertifikaları Sınavları

- Sınavlarda çok yoğun kitabi bilgi ve uygulama sorulmaktadır.
- Dolayısıyla sadece teorik bir altyapıya sahip olmadan Java'yı uyguluyor olmak ya da hiçbir Java uygulama tecrübesi olmadan sadece kitaba çalışarak bu sınavları geçmek pek de mümkün değildir.
  - SCJD'da programlama projesi ve makale yazımı ,
  - SCEA'de ise sınav, programlama projesi ve makale yazımı vardır.
  - Diğer sınavlar çoktan seçmelidir
- Ayrıca SCJA dışındaki sınavların hepsi için SCJP sınavını başarmış olmak bir ön koşuldur.

# Sun Certified Programmer

- Sun Certified Programmer for the Java Platform, Standard Edition 6 (CX-310-065) sınavını başarmış olmak, SCJA dışındaki sınavların hepsi için bir ön koşuldur.
- Bu sınavda 60 soru sorulmakta ve 180 dakika verilmektedir.
- Geçer not 35 soruya doğru cevap yani %58,33'tür.

Dinlediđiniz için teŝekkür ederim.

Sorular?