

# KİM KORKAR JAVADAN EĞİTİM KATALOĐU

Alper AKALIN

İletişim

Tel: 0 212 318 0179

Fax: 0 212 275 7535

email: [egitim@kimkorkarjavadan.com](mailto:egitim@kimkorkarjavadan.com)

bilgi: <http://kimkorkarjavadan.com/>

---

# İçindekiler

1. JAVA 8 ile NESNE YÖNELİMLİ PROGRAMLAMA .....	1
1.1. Java ve Nesneye Yönelim .....	1
1.2. Java Syntax Kuralları .....	2
1.2.1. Eclipse IDE .....	2
1.2.2. İlk Program .....	2
1.2.3. Temel Veri Tipleri, Primitif Tipler .....	2
1.2.4. Tip Dönüşümleri, Type Casting .....	3
1.2.5. Java Operatörler .....	3
1.2.6. Kod Blokları .....	3
1.2.7. Kontrol Yapıları .....	3
1.2.8. Klavyeden Basit I/O Giriş-Çıkış İşlemleri .....	4
1.2.9. Algoritma .....	4
1.2.10. Metotlar .....	4
1.2.11. Diziler .....	5
1.2.12. String İşlemler .....	5
1.3. Java ile Nesne Yönelimli Programlama .....	6
1.3.1. Java ile Nesne Yönelimli Programlamaya Giriş .....	6
1.3.2. Inheritance - Kalıtım .....	6
1.3.3. Polymorphism - Çok Biçimlilik .....	7
1.3.4. Abstract Classes - Soyut Sınıflar .....	7
1.3.5. Interface - Arabirimler .....	7
1.3.6. Inner Class - İç Sınıflar .....	7
1.4. Exceptions - İstisnalar - Hata Ayıklama .....	7
1.5. Generics - Jenerik Sınıflar .....	8
1.6. Collections - Koleksiyonlar .....	8
1.7. Java 8 ve Getirdikleri .....	9
1.7.1. Java 8 Tarih Saat İşlemleri .....	9
1.7.2. Java 8 Arayüzler .....	9
1.7.3. Java 8 Lambda Expressions .....	9
1.7.4. Java 8 Stream API .....	9
1.8. Multithreading .....	9
1.9. Dosya İşlemleri .....	9
1.10. JDBC .....	10
2. DESIGN PATTERNS ve EFFECTIVE JAVA .....	11

---

2.1. Advanced Java ve OOP .....	12
2.2. Yaratımsal Tasarım Şablonları .....	12
2.3. Yapısal Tasarım Şablonları .....	12
2.4. Davranışsal Tasarım Şablonları .....	13
2.5. Mimarisel Tasarım Şablonları .....	13
2.6. More Java Best Practises .....	13
3. ANDROID 7 NOUGAT ile MOBİL PROGRAMLAMA .....	15
3.1. Android Ortam Kurulumu .....	15
3.2. Android "Hello World" Uygulaması .....	16
3.3. Button, Layout, Activites, Activity Lifecycle, Java Classes.....	16
3.3.1. Button Sayaç Uygulama Örneği .....	16
3.4. Layout Details, Designer Studio Details, Listeners... ..	17
3.4.1. Hesap Makinası Uygulama Örneği .....	17
3.5. Async Task, Permissions, Processing XML, ListView, Menus, Adapter, ArrayAdapters, RelativeLayout... ..	18
3.5.1. Download Mekanizması Örneği .....	18
3.6. Third Party Library Kullanımı... ..	18
3.6.1. YouTube Implementation Örneği .....	18
3.7. JSON, Rest,Search, Validation, RecyclerView, Data Download, Gesture Detector, SearchView... ..	19
3.7.1. Flickr Implementation Örneği .....	19
3.8. Andorid ile Veri Tabanı Erişimi... ..	20
3.8.1. Arkadaş Listesi Oluşturma Örneği .....	20
4. JAVA WEB SERVICES .....	23
4.1. SOA – Service Oriented Architecture .....	24
4.2. XML Architecture .....	24
4.3. XSD Architecture .....	25
4.4. SOAP Architecture .....	25
4.5. WSDL Architecture .....	26
4.6. JAX-WS .....	26
4.7. Web Services Design Best Practises .....	27
4.8. Client Creation .....	27
5. JAVA RESTFUL SERVICES .....	29
5.1. Restful Elements .....	30
5.2. HTTP Methods .....	31
5.3. Advanced Rest Features .....	31

5.4. Restful Service Design Best Practises .....	32
5.5. Client Creation .....	32
6. HIBERNATE & JPA .....	33
6.1. Basics .....	34
6.2. Hibernate .....	34
6.3. JPA – Java Persistence API .....	35
6.4. Persistence Context .....	35
6.5. Object Access .....	36
6.6. Relations .....	36
6.7. DAO – Database Access Object .....	37
7. SPRING FRAMEWORK .....	39
7.1. Dependency Injection .....	40
7.2. Spring Framework Basics .....	41
7.3. Spring CORE .....	41
7.4. Spring AOP .....	42
7.5. Spring JDBC .....	43
7.6. Spring ORM .....	44
8. SPRING MVC ile JAVA WEB .....	45
8.1. Applicability Real Web Project Basics .....	46
8.2. JSP – Java Server Pages .....	47
8.3. MVC Pattern .....	47
8.4. Bootstrap API .....	47
8.5. Spring MVC .....	48

# 1

## JAVA 8 ile NESNE YÖNELİMLİ PROGRAMLAMA

---

### Eğitimin Amacı

Nesne Yönelimli Programlama dillerinin öncülerinden ve Android Mobil Programlama platformunun temelini oluşturan JAVA bire bir çalışan kod örnekleriyle incelenmektedir. Bu eğitimde katılımcıların Nesne Yönelimli Programlama mantığına ve Java Programlama diline tam hakim olması hedeflenmektedir.

### Kimler Katılabilir

Yazılım Dünyası ile ilgilenen ya da Kurumsal ve Mobil Yazılım Dünyasında yer edinmek isteyen herkes katılabilir. Bu eğitim katılımcıları Android ile Mobil Programlama, Bankacılık ve Telekomünikasyon gibi büyük kurumsal projelerde çalışabilmek adına eşsiz bir başlangıç yapmış olacaklardır.

- **Süre: 72 saat**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi: <http://kimkorkarjavadan.com/>**

### 1.1. Java ve Nesneye Yönelim

- Modülerlik
- Bilgi Saklama
- Kodu Tekrar Kullanabilme

- Hata Ayıklama Kolaylığı

## **1.2. Java Syntax Kuralları**

- Derleyici
- Yorumlayıcı
- Paket Kavramı
- Sınıf Kavramı
- Nesne Kavramı
- Metot Kavramı
- Main Metot Kavramı

### **1.2.1. Eclipse IDE**

- JDK8 Kurulumu
- Java Ortamı Kurulumu
- Eclipse IDE Kurulum ve Kullanımı

### **1.2.2. İlk Program**

- String İşlemler
- + Operatörü

### **1.2.3. Temel Veri Tipleri, Primitif Tipler**

- Tam Sayılar
- Ondalıklı Sayılar
- Başlangıç Değerleri
- Casting, Tip Belirleme
- Karakter Literaller
  - Değer Atama - Karakter Atama incelemesi
- Wrapper'lar
- JavaSE7 ile Gelen Yenilikler

### **1.2.4. Tip Dönüşümleri, Type Casting**

- Primitif Veri Tipleri ve Karakter Katar Dönüşümleri
- Primitif Sayısal Veri Tiplerinin String Veri Tipine Dönüştürülmesi

### **1.2.5. Java Operatörler**

- Aritmetik Operatörler
- Aritmetik Artırma - Eksiltme Operatörleri
- Aritmetik Atama Operatörleri
- İlişkisel - Karşılaştırma Operatörleri
  - İlişkisel Operatörleri if Deyimi Üzerinden İnceleme
  - Mantıksal Operatörler
  - Bitwise Operatörler
  - Operatör Öncelikleri

### **1.2.6. Kod Blokları**

- Kod blok yapılarının incelenmesi

### **1.2.7. Kontrol Yapıları**

- Karşılaştırma İfadeleri
  - if Deyimi
  - Birden Fazla Önermeyi Bir if Cümlecğinde Birleştirme
  - if-else deyimi, Çift Seçimli Yapı
  - else-if deyimi
  - if-then-else
  - Switch-Case Yapısı - Çok Seçimli Yapı
  - String İfadeler ve Switch
- Döngü İfadeleri - Tekrarlı Yapılar
  - While Döngüsü

- do-while döngüsü
- For Döngüsü
- Dallanma İfadeleri
  - Break Deyimi
  - Continue Deyimi
  - Return Deyimi

### **1.2.8. Klavyeden Basit I/O Giriş-Çıkış İşlemleri**

- Paket Kavramı
- Import Kavramı
- Joker Kavram .\*
- Scanner Kavramı
- Bazı Scanner Yöntemleri
- Bilgi Giriş-Çıkışı

### **1.2.9. Algoritma**

- Algoritma Hazırlamanın Faydaları
- Algoritma İfade Şekilleri
- Uygulamalar
  - Sayaç Mantığı
  - Math Sınıfı
  - Rastgele Sayı Üretimi

### **1.2.10. Metotlar**

- Metotlar
  - Parametresiz Metotlar
  - Parametrelili Metodlar
- Metotların Oluşturulması



- Signatur
- Metotların Çağırılması
- Dönüş Değeri Olmayan Metotların Çağırılması
- Yapılandırıcı Metotlar - Constructors
- Method Overloading - Metot Aşırı Yükleme
- Değişkenlerin Faaliyet Alanları

### **1.2.11. Diziler**

- Referans Tipler
- Dizi Oluşturma
- Dizi Boyutunun Değiştirilmesi
- Dizi Boyutunun Belirlenmesi
  - Dizi Elemanlarına Değer Atama
- Geliştirilmiş For Döngüsü
- Dizilerin Metotlara Argüman Olarak Atanması
- Parametre Sayısı Değişken Metotlar
- İki Boyutlu Diziler - Matrisler
  - İki Boyutlu Diziler
  - Çok Boyutlu Diziler
- Dizilerin Kopyalanması
- Array Sınıfı
- ArrayList Sınıfı ve Dinamik Dizi Tanımlama

### **1.2.12. String İşlemler**

- Stringlerin Oluşturulması
- String Metodlar
- String'lerin Karşılaştırılması

## **1.3. Java ile Nesne Yönelimli Programlama**

### **1.3.1. Java ile Nesne Yönelimli Programlamaya Giriş**

- İlk Sınıf
- New Operatörü
- Nokta Operatörü
- Çöp Toplama - Garbage Collection
- Yapılandırıcılar - Constructors
- Erişim Belirleyiciler
  - Public (Açık)
  - Private (Özel)
  - Protected (Korumalı)
  - Encapsulation – Kapsülleme
  - Default-Friendly
- Static Alanlar
- This Kelimesi
- Final Alanlar
- Metotlara Argüman Olarak Nesne Gönderme
  - Pass By Reference

### **1.3.2. Inheritance - Kalıtım**

- Kalıtımın Kontrolü
  - Kalıtımın Faydaları
- Metot Overriding
- Object Sınıfı
  - toString Metodu
  - super() Deyimi
- Kompozisyon

### **1.3.3. Polymorphism - Çok Biçimlilik**

- Çok Biçimlilik
- Geç Bağlama - Late Binding
- instanceof Metodu
- Genişletilebilme - Extensibility

### **1.3.4. Abstract Classes - Soyut Sınıflar**

- Soyut Sınıf Kısıtları
- Final Sınıf ve Metodlar İçin Uyarı

### **1.3.5. Interface - Arabirimler**

- Arabirimlerde Çok Biçimlilikten Faydalanma

### **1.3.6. Inner Class - İç Sınıflar**

- Dahili Üye Sınıflar
  - Dahili Üye Sınıflarda Erişim
  - İç İçte Dahili Üye Sınıflar
  - Dahili Üye Sınıflar ve Türeme-Türetilme Durumları
- Static Dahili Sınıflar
- Dahili Yerel Sınıflar
- Anonim Dahili Sınıflar

## **1.4. Exceptions - İstisnalar - Hata Ayıklama**

- Try - Catch Blokları
- JavaSE7 ve Try Catch Blokları
- İstisna Sınıfları
  - Kontrollü İstisnalar
  - Kontrolsüz İstisnalar

- İstisna Tanımlama
- İstisna Sınıflarında Ortak Kullanılan Metotlar
- Finally Bloğu
- JavaSE7 Kaynak Kapama

## 1.5. Generics - Jenerik Sınıflar

- Generic Sınıf Oluşturma
- Generics – Jenerik Metodlar
- Generic Metot Oluşturma

## 1.6. Collections - Koleksiyonlar

- List Arabirimi
- Set Arabirimi
- Map Arabirimi
- Set Arabirimi
  - HashSet Sınıfı
  - LinkedHashSet Sınıfı
  - TreeSet Sınıfı
- List Arabirimi
  - ArrayList Sınıfı
  - LinkedList Sınıfı
- Vektör Sınıfı
- Stack Sınıfı
- Queue (kuyruk) Sınıfı
- Map Arabirimi
  - SortedMap Arabirimi
  - Map Arabiriminin Somut Sınıfları
- JavaSE7 Diamond Inference ve Generic Tipler

## **1.7. Java 8 ve Getirdikleri**

### **1.7.1. Java 8 Tarih Saat İşlemleri**

- LocalDate Sınıfı
- LocalTime Sınıfı
- LocalDateTime Sınıfı
- Tarih Formatlama

### **1.7.2. Java 8 Arayüzler**

- Default Metod
- Static Metod

### **1.7.3. Java 8 Lambda Expressions**

- Functional Interfaces
- Lambda Expressions

### **1.7.4. Java 8 Stream API**

- StreamAPI Özellikleri
- Koleksiyonlar ve Streamler
- Stream'ların Doğası
- Intermediate ve Terminal Operasyonlar

## **1.8. Multithreading**

- Thread Sınıfının Metodları
- Synchronization - Eş Zamanlama
- Monitör

## **1.9. Dosya İşlemleri**

- Data Hiyerarşisi

- Dosya - Stream (Akım)
- Java'daki Dosya İşlemleri Sınıfları
- File Sınıfı
  - File Sınıfı Yapılandırıcıları
  - File Sınıfı Metotları
- Serialization - Serileştirme

## 1.10. JDBC

- Bağlantı Adresi
- Bağlantı Kurma
- SQL Tipleri ve Java Karşılıkları
- JDBC Sorgu - Query
- Statement-Demeç
- Prepared Statement
- Callable Statement
- Result Set - Sonuç Kümesi
- Kaydırılabilir ResultSet
- Metadata - Üstveri
- ResultSetMetaData
- Transaction

# 2

## DESIGN PATTERNS ve EFFECTIVE JAVA

---

### Eđitimin Amacı

Design Patterns ve Effective Java Eđitiminde geliřtirilen uygulamaların sahip olması gereken belirli best practise 'ler incelenmektedir. Eđitimin Hedefi, yazılımcıların belirli kurallara sahip kaliteli yazılım geliřtirmeleri ve de yazılımda ortak bir dil oluřturmalarına yardımcı olmaktır. Yani belirli standartlara sahip yazılımlar geliřtirmeleridir.

Eđitimde iřlenecek her konu/pattern

- Senaryo
  - Çözüm
    - Örnek

řeklinde ilerleyen bir workshop mantıđında ele alınmaktadır.

### Kimler Katılabilir

Enterprise Java Projelerinde, Nesne Yönelimli Programlama felsefesini tam anlamıyla uygulamak isteyen ve ileri seviye Java Framework'lerini daha iyi kavrayıp daha etkin kullanmak isteyen tüm yazılım geliřtiricileri katılabilir/katılmalıdır.

- **Süre: 3 Gün**

- **Eğitmen: Alper AKALIN**
  - **Bilgi:** <http://kimkorkarjavadan.com/>

## 2.1. Advanced Java ve OOP

- Encapsulation
- Polymorphism
- Prefer Interfaces to Abstract Classes
- Composition vs Inheritance
- Static Factory Method
- Static Block
- Reflection API
- Design patterns with Java

## 2.2. Yaratımsal Tasarım Şablonları

- Creational Design Patterns
  - The Abstract Factory Pattern
  - The Factory Method Pattern
  - The Singleton Pattern
  - The Builder Pattern
  - The Prototype Pattern

## 2.3. Yapısal Tasarım Şablonları

- Structural Design Patterns
  - The Adapter Pattern
  - The Bridge Pattern
  - The Facade Pattern
  - The Decorator Pattern
  - The Composite Pattern



- The Proxy Pattern
- The FlyWeight Pattern

## 2.4. Davranışsal Tasarım Şablonları

- Behavioral Design Patterns
  - The Command Pattern
  - The Iterator Pattern
  - The Memento Pattern
  - The State Pattern
  - The Observer Pattern
  - The Strategy Pattern
  - The Chain of Responsibility
  - The Mediator Pattern
  - The Visitor Pattern
  - The Template Pattern

## 2.5. Mimarisel Tasarım Şablonları

- Dependency Injection & Inversion Of Control
- Java ile Dependency Injection örneği
- JavaDI vs SpringDI
- Java ile AOP örneği
- JavaAOP vs SpringAOP
- DAO Şablonu
- MVC Şablonu

## 2.6. More Java Best Practises

- Çoklu parametre yerine Builder Pattern kullanımı
- Gereksiz nesne yaratmadan kaçınmanın önemi
- Autoboxing ve Unboxing

- Erişilmeyen nesne referanslarını kaldırma
- Interface ile refere etmenin önemi
- Reflection vs Interface
- toString metodunun önemi
- double vs bigDecimal
- Refactoring ...
  - Extracting methods
  - Extracting supertypes
  - Replace Nested Conditional with Guard Clauses
  - Replace Conditional with Polymorphism
- Exceptions

# 3

## ANDROID 7 NOUGAT ile MOBİL PROGRAMLAMA

---

### Eğitimin Amacı

Mobil platformlardan Android en güncel versiyonu Android 7-Nougat, Android Studio ile birlikte incelenmektedir. Katılımcıların Android Mobil İşletim Sistemine tam hakim olabilmeleri için gerekli tüm bileşenler, 7 tane çalışan Mobil Proje içerisinde katılımcılarla birlikte adım adım incelenmektedir. (BootCamp) Böylece Android üzerindeki hakimiyetin mümkün olduğunca hızlı sağlanması hedeflenmektedir.

### Kimler Katılabilir

Temel düzeyde Java Programlama dilini bilen ve Mobil Yazılım Dünyasında yer edinmek isteyen herkes katılabilir. Android ile Mobil Programlama Eğitimi katılımcıları Bankacılık, Telekomünikasyon, eTicaret gibi büyük kurumsal firma projelerinde çalışma fırsatı yakalama adına eşsiz bir başlangıç yapmış olacaklardır.

- **Süre: 80 saat**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi:** <http://kimkorkarjavadan.com/>

### 3.1. Android Ortam Kurulumu

- Windows için Android Studio Kurulumu

- Android Studio Konfigürasyonu
- VT-X Ayarları

## **3.2. Android "Hello World" Uygulaması**

- Android Studio Templates
- İlk Uygulama Örneği
- Android Studio İncelemesi
- Android Virtual Device Kurulumu
  - "Hello World" Uygulamasının Sanal Makinada Çalıştırılması
- "Hello World" Uygulamasının Android Üzerinde Çalıştırılması

## **3.3. Button, Layout, Activites, Activity Lifecycle, Java Classes...**

### **3.3.1. Button Sayaç Uygulama Örneği**

- Designer Studio
- Görüntü Boyutlama
- Constraint
- Content Düzenleme
- Densities
- Autoconnect
- Baseline Constraint
- Scrolling
- Infer Constraints
- Layout Oluşturma
- IDs & XML code
- Activites
- Bundles and Classes
- Java Classes & Fields

- Click & onClickListener
- TextView
- EditText
- Null Kontrolü
- IDs Kullanımı
- Save Instance State
- Activity Lifecycle
- Method Override
- Logging

### **3.4. Layout Details, Designer Studio Details, Listeners...**

#### ***3.4.1. Hesap Makinası Uygulama Örneği***

- Calculator Layout Oluşturma
- Layouts Detaylı İnceleme
- Input Type & Focus Changes
- Landscape Layout Ekleme
- Comparing Code and Layouts
- Comparing Directories
- Flowchart & onClickListeners
- Operation Button Listeners
- Implement Operations
- Decimal Point Bug Fix
- Bug Fix Challenge
- Add Neg Button
- Neg Button Functionality
- Portrait Layout

## **3.5. Async Task, Permissions, Processing XML, ListView, Menus, Adapter, ArrayAdapters, RelativeLayout...**

### **3.5.1. Download Mekanizması Örneği**

- Async Task
- Custom Async Task
- doInBackground & downloadXML
- Exceptions & Buffered Reader
- Security Exception
- Stack Trace & The Logcat
- Android Permissions
- Processing XML Data
- XML Parsing Factory Classes
- Main XML Parsing Loop
- ListView & ArrayAdapters
- Array Adapter
- Adapter Layout
- Custom Adapter
- Menus
- Menu Item Groups

## **3.6. Third Party Library Kullanımı...**

### **3.6.1. YouTube Implementation Örneği**

- YouTube API Setup
- İkincil Activity Ekleme
- Dinamik Layout Ekleme
- Google API Key
- Hatalı onInitialization
- onInitialization

- Standalone Activity
- Intents & Manifest

## **3.7. JSON, Rest,Search, Validation, RecyclerView, Data Download, Gesture Detector, SearchView...**

### **3.7.1. Flickr Implementation Örneği**

- Flickr API
- JSON Api
- Validation
- Async Task
- Project Diagram
- Photo Class Code
- Flickr JSON Data Class Code
- Create URI
- JSON Parsing
- Call Back
- RecyclerView
- Search
- Photo Detail Activities
- Nesting Linear Layouts
- RecyclerView Adapter
- Picasso
- Gesture Detector
- Touch Events
- Touch Methods
- BaseActivity
- Serializable
- Material Design
- Backwards Compatibility

- Style Sheets
- Custom Style Sheet
- API Levels
- String Resources
- Search Menu Code
- SearchView
- SearchManager
- Search Messages

## **3.8. Andorid ile Veri Tabanı Erişimi...**

### **3.8.1. Arkadaş Listesi Oluşturma Örneği**

- Database Terminology
- Command Line
- Windows Path
- SQLite
- Querying
- Order By, Joins
- Wild Cards
- Views
- SQL in Android
- Debugging SQL in Code
- Cursor
- Navigation
- Android File System
- Content Providers
- Contract Provider
- ContractsContract.Contracts
- ContentResolver
- Permissions API



# 4

## JAVA WEB SERVICES

---

### Eđitimin Amacı

Dijitalleşen dünyada neredeyse her elektronik aygıtın bir şekilde birbirine bağlandığı görölmektedir. Büyük ya da küçük farketmeksizin hemen hemen her uygulama bir şekilde diđer yani yabancı uygulamalarla konuşmaktadır.

SOA ve Web Servisler, özellikle kurumsal taraftaki bu bağlantıların çok büyük bir kısmını yönetmektedirler. Bu nedenle yazılım dünyasında yer edinmek isteyen her developer web servis mimarisine hakim olmalıdır.

Her şeyin xml'in icadıyla başladığı düşünülürse xml ile başlayan ve wsdl ile biten bir serüven incelenmektedir. Java tarafında web servisler için üretilen Jax-WS apiyi kullanarak eldeki veri modellerinin web servisler ile dış dünyaya açılması incelenmektedir.

SOA nedir?

- Soap ile Soa aynı sistematiđi mi tanımlar?
- SOA yönelimli geliřtirmelerin kazanımları nelerdir?
- Web servis nedir?

Web servisler, uygulamaların ürettikleri verileri diđer yani yabancı uygulamalar ile paylařtıkları metodolojidir. Bu paylařım için yapılacak geliřtirmeler web servis interface'e yani wsdl'a yönelik yapılmaktadır. Bu eğitimde bir web servis interface nasıl yaratılır ve yönetilir birlikte inceleyeceđiz.

Web servis interface yaratılmasında kullanılan Contract First & Contract Last yaklaşımlarını derinlemesine irdelenecektir. Böylece bu eğitimde iyi tasarlanmış bir web servisin nasıl ortaya konulacağı incelenmektedir

### **Kimler Katılabilir**

Enterprise denilen ileri seviye Java projesi geliştirmek isteyen ya da Web Servisler kullanan bir Enterprise Java projesine sonradan dahil olan her developer katılabilir. Büyük ya da küçük farketmeksizin hemen hemen her uygulama bir şekilde diğer yani yabancı uygulamalarla konuşmaktadır. SOA ve Web Servisler bu bağlantıların çok büyük bir kısmını yönetmektedirler. Bu nedenle yazılım dünyasında yer edinmek isteyen her geliştirici katılmalıdır.

- **Süre: 3 Gün**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi:** <http://kimkorkarjavadan.com/>

## **4.1. SOA – Service Oriented Architecture**

- SOA – Service Oriented Architecture
- Benefits of SOA
- Web Services
- Web Services and SOA
- Soap
- Rest

## **4.2. XML Architecture**

- Xml Name Conflicts
- Tag
- CData
- Attribute
- Child Element
- Empty Element

- Well Formed – Valid
- Xsl
- Namespace
- Prefix

### **4.3. XSD Architecture**

- Xml Schema Definition
- Element Definitions
- Schema Root Element
- Attribute Definitions
- Reference
- Element and Attribute References
- Type Definitions
- Maxoccurs – Minoccurs
- Group Definitions
- Complex Element Definitions
- Simple & Complex Element

### **4.4. SOAP Architecture**

- Soap Web Services
- Provider
- Consumer
- Interface
- WSDL
- Request
- Response
- Xml
- Operations

## 4.5. WSDL Architecture

- Definitions
- Types
- Interface
- Port Type
- Operations
- Port
- Binding
- Service
- Documentation
- Import

## 4.6. JAX-WS

- Java Web Services
- First Jax-WS project
- Application Server
- Glassfish AS
- Definitions
- Service
- Binding
- Port Type
- Messages
- Types
- Document Type
- Crud Operations
- @WebService
- Xml Formatting
- @WebMethod
- @WebResult

- Exception Handling
- Soap UI

#### **4.7. Web Services Design Best Practises**

- Web Service Interface
- Contract First
- Contract Last
- Interface Design Projects
- Service Endpoint
- Web Service Interface
- Java Interface vs Web Service Interface

#### **4.8. Client Creation**

- Wsimport
- Client creation
- Test

---

# 5

## JAVA RESTFUL SERVICES

---

### Eđitimin Amacı

Web teknolojilerinin yaygınlaşması ile birlikte bu teknolojiler birbirleri ile sürekli iletişim halindedirler. Kurumsal uygulamalarda kullanılan soap yani xml tabanlı iletişim yolları pahalı olduğundan daha az kaynak tüketen yollar geliştirilmiştir.

Bu bağlamda web teknolojilerinde kullanılan http metodlar baz alınarak Restful Web Servisler geliştirilmiştir. Restful servisler, Soap servislere nazaran daha efektif üç özelliđe sahiptirler.

- Doküman tipi zorunluluđu yoktur. Yani istenilen herhangi bir formatta veri iletebilirler.
  - Veri alış veriřleri daha hızlıdır.
    - Kaynak tüketimleri düşüktür.

Günümüzde her bilginin web ortamına taşındığı düşünülürse web projelerinin ve bu projelerinin önemi artmaktadır. Buradaki yıldızlı nokta web uygulamalarının arasındaki iletişimin sağlıklı ve en az kaynak tüketecek şekilde kurgulanmış olması zorunluluđudur.

Restful bir servisin tasarımı geliştirilmesinden daha zordur. Eđitim boyunca Resource ve servis tanımlarının nasıl tasarlanabileceđi tek tek örnekler üzerinden incelenmektedir.

- Response ve ResponseBuilder gibi yapılarla yerinde response oluşturmanın ve Response'ta daha fazla bilgi taşımanın değişik yöntemleri örneklenmektedir.
- Statü kodları ile kullanıcı ya da client uygulamanın servis hakkında daha fazla bilgi sahibi olmasını sağlanacak. Hata ayıklama mekanizmaları ile olası hata durumlarında kullanıcı ya da client uygulamanın hata hakkında max düzeyde bilgi sahibi olmaları sağlanmaktadır.

### **Kimler Katılabilir**

Enterprise Java Web projesi geliştirmek isteyen ya da geliştirilmiş bir Enterprise Web Java projesine sonradan dahil olan her developer katılabilir. Özellikle Mobil Yazılım Geliştirme ile ilgilenen yazılımcılar bu eğitime katılmalıdırlar.

- **Süre: 3 Gün**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi:** <http://kimkorkarjavadan.com/>

### **5.1. Restful Elements**

- Http
- Html
- Resources
- Resource Locations
- Http Methods
- Status Code
- Message Format
- Resource Tasarımı
- Jax-Rs
  - Jax-Rs Annotations
- Response
  - Response Document Type

- Json

## 5.2. HTTP Methods

- Get
- Post
- Put
- Delete
- Idempotency
- After Idempotency
  - Get
  - Delete
  - Put
  - Post

## 5.3. Advanced Rest Features

- Filtering
- Pagination
  - @QueryParam
- Bean Filtering
  - @BeanParam
- Sub Resources
- Status Codes
- Location Headers
- Response
  - Response Builder
- Generic Entity
- Exceptions



- Error Handling
- Exception Mapper
- Web Application Exception

## **5.4. Restful Service Design Best Practises**

- Hateoas
- Richardson Maturity Model
- PoX
- Resources
- Http Verbs
- Hypermedia Controls

## **5.5. Client Creation**

- Server
- Client
- Client & Server Architecture

# 6

## HIBERNATE & JPA

---

### Eğitimin Amacı

Veri tabanı işlemlerinin Object Relational Mapping konsepti altında incelendiği eğitimde,

- Object Relational Mapping nedir?
- Nesnelerin persist edilmesi için hangi durumlarda ORM araçları kullanılabilir ya da kullanılmamalıdır.
- Veri tabanı veya Uygulama seviyesinde işlem yapılması ne demektir? Bu seviyelerdeki işlemlerde kazanım ya da kayıplar nelerdir?

ORM konseptinde Hibernate ile ilk çerçeve çizildikten sonra JPA ile ilerlenerek eğitimde oldukça farklı bilgiler paylaşılmaktadır. Bu bağlamda, Hibernate üzerindeki ileri seviye konular bambaşka bir bakış açısı ile ele alınmaktadır.

- Örneğin Hibernate içerisindeki en karmaşık konuların başında Lazy Loading yapısı gelmektedir. Bu yapının sadece @Lazy anotasyonundan ibaret olmadığını altında Java Proxy Objects ya da Object States gibi bir çok dinamiğin olduğunu katılımcılarla birlikte gözlemlenmektedir.

### Kimler Katılabilir

Enterprise seviye Java projesi geliştirmek isteyen ya da ORM kullanan bir Enterprise Java projesine sonradan dahil olan her developer katılabilir. Hibernate'in ülkemizde yaygın kullanımı göz önüne alındığında Hibernate konusunda ileri bilgiye sahip yazılım geliştiriciler avantajlı konuma sahip olacaklardır.

- **Süre: 3 gün**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi:** <http://kimkorkarjavadan.com/>

## 6.1. Basics

- Persistence
- Database Level Process
- Application Level Process
- Object Persistence
- Object Relational Pradigms
- ORM - Object Relational Mapping
- Why ORM Frameworks?
- ORM Frameworks
- POJOs

## 6.2. Hibernate

- Hibernate Architecture
- SessionFactory
- Entity Type Objects
- Simple Hibernate Project
- Hibernate Xml Mapping
  - Xml Mapping
- Hibernate Annotation Mapping
- Transaction
- CRUD Operations
  - Update
  - SaveOrUpdate
  - Merge methods

- Getter Read

### **6.3. JPA – Java Persistence API**

- Java Persistence Api
- Hibernate & JPA
- JPA Architecture
  - Persistence Context
  - Entity Manager
- Simple JPA Project
- Annotation Mapping
  - Basic Annotations
- Transaction
- CRUD Operations
  - Merge method
- Secure Parameter Passing
- Named Queries

### **6.4. Persistence Context**

- Persistence Context
- Session
- Entity Lifecycle
- Object States
  - Transient State
  - Managed – Persistent State
  - Detached State
- States of CRUD operations
- JPA more annotations

- @Embeddable
- @Embedded
- Value Type Objects
- Filtering
- Collections

## 6.5. Object Access

- Hibernate Object Access
- Object Levels
  - First Level Objects
  - Second Level Objects
- Object Types
  - Concrete Objects
  - Proxy Objects
- Fetch Types
- Lazy Fetching
- Eager Fetching
- Lazy Loading
  - Advanced Lazy Loading Review

## 6.6. Relations

- Object Relations
- Table Relations
- Embedded
- ElementCollection
- OneToOne
  - Unidirectional

- Bidirectional
- OneToMany
- ManyToOne
- ManyToMany
- Cascade Types
- Inheritance
- Single Table
- Separate Table

## **6.7. DAO – Database Access Object**

- DAO
- DAO Architecture
- Connection Pooling
- C3P0 Usage

# SPRING FRAMEWORK

---

## Eđitimin Amacı

Spring, yazılım geliřtiricilerin belirli tasarımlar/standartlar altında yapması gereken geliřtirmeleri hazır olarak sunmaktadır. Bu eđitimde Spring'i oluřturan bileřenleri incelenerek Spring ile yapılacak geliřtirmeler için temel oluřturulmaktadır.

- Dependency Injection
- Java ile klasik nesne yaratmak iřlemleri
- Factory Tasarım řablonu, Inversion of Control
- Java ile Proxy Nesne Yaratma
- Java ile AOP geręeklemeleri

incelenerek Spring'in temellerini oluřturan bileřenler ele alınmaktadır.

Unutmamak gerekir ki Spring bir Java Framework'üdür. Yani gücünü Java'dan almaktadır. Bizler de gücümüzü bu framework altında yatan Java'dan alırsak Spring hakimiyetimiz daha kalıcı ve kaliteli olacaktır. Bu nedenle, eđitime doğurdan Spring ile bařlamak yerine Spring'te řokca kullanılan bileřenleri inceleyerek bařlamaktayız.

Spring, nesne yaratma ve yönetme süreçlerini tamamen üstlendiđinden **Spring Yařam Döngüsü** üzerinde fazlaca durulmaktadır. Spring Yařam Döngüsünün incelenmesinde Dependency Injection ięerisinde incelenen Java bileřen tanımlarından yararlanılmaktadır. Örneđin

- Sterotype'lar için proxy nesnelerin ne olduđu,

- Bean initialization veya Bean destruction için Aop mantalitesinin bilinmesi vb. konuya hakimiyet açısından oldukça yararlı olacaktır.

Aslında ne oluyor? ya da Bu teknolojinin altında ne yatıyor? sorusu sorulmadan hiç bir teknolojiye tam hakim olmak mümkün değildir.

Spring Core, Spring AOP, Spring Jdbc veya Spring ORM hakkında bilgilendirme yapılırken bu yapıların altında yatan Java mekanizmalarına sık sık atıfta bulunmaktadır.

### **Kimler Katılabilir**

Enterprise seviye Java projesi geliştirmek isteyen ya da Spring kullanan bir Enterprise Java projesine sonradan dahil olan her yazılım geliştirici katılabilir.

Spring Framework, JavaEE yapısına alternatif oluşturmaktadır. Bu nedenle Java dünyasındaki yeri diğer frameworklerin ötesindedir. Spring sadece bir frameworkten öte bir frameworkler platformudur. Bu platformda bir Enterprise Java projesinde gerekli olabilecek hemen hemen her bileşen bulunmaktadır. Bu nedenle Java dünyasında yer edinmek veya farklı bir yerde olmak isteyen her yazılım geliştirici katılmalıdır.

- **Süre: 5 Gün**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi:** <http://kimkorkarjavadan.com/>

## **7.1. Dependency Injection**

- Dependency
- Dependency Injection
  - Classic Injection
  - Polymorphism
  - Method Parameter
  - Composition
  - Factory Design Pattern



- Dependency Inversion
- Java Dynamic Proxy
  - Proxy Pattern
- Aspect Oriented Programming
  - AOP with Java Proxy

## **7.2. Spring Framework Basics**

- What is Spring Framework
- Inversion of Control
- Spring & J2EE
- Spring Philosophy
- Spring Architecture
- Spring Modules
- Why use Spring

## **7.3. Spring CORE**

- Setup Spring
- Bean Definitions
- Bean Factory
- Spring Containers
- Setter Injection
- Constructor Injection
- Data Types
- Object Injection
- Spring Life Cycle
- Setter vs. Constructor Injection
- Collections
- Inner Beans
- Alias & Name

- Autowiring
- Bean Scopes
- Inheritance
- Bean Initialization
- Bean Destruction
- BeanPostProcessor
- Spring Life Cycle
- Placeholder
- Interface
- Spring Annotations
- Stereotypes
- Message Source
- Event Handling
- JSR-330 Java Annotations
- Java Base Annotations

## 7.4. Spring AOP

- Aspect Oriented Programming
  - Separation of Concerns
  - Code Scattering
  - Code Tangling
  - Crosscutting Concerns
- AOP Terminology
- AOP Types
  - Static – Compile Time Weaving
  - Dynamic – Load Time Weaving
  - AspectJ
  - Spring AOP

- Spring AOP Limitations
- Advice
  - Before Advice
  - After Advice
  - Pointcut
  - JoinPoint
  - Passing Parameters
  - AfterReturning
  - AfterThrowing
  - Around
  - Custom Advice
- Java Proxy Objects
  - AOP with Java Proxy
- Spring AOP Life Cycle

## **7.5. Spring JDBC**

- Spring Jdbc
- Spring Jdbc Template
- Row Mapper
- Crud Operations
- Named Parameter
- Dao Support
- Batch
- Transaction
- ACID
- Spring Transaction Basics
- Spring Exception Handling
- Spring Transaction

- DAO

## 7.6. Spring ORM

- Spring ORM
- Hibernate
- JPA
- @PersistenceContext
- Application Managed EntityManager
- Container Managed EntityManager
- DAO
- Splitting applicationContext to Multiple Files
  - beans.xml
  - datasource.xml
  - applicationContext.xml

# SPRING MVC ile JAVA WEB

---

## Eđitim Amacı

Yazılım ve özellikle Java meraklılarından en çok aldığım sorulan başında “hocam çalışan bir proje örneđi verebilir misiniz” gelmektedir. Bu eğitimde katılımcılarla hem bir online kitap mağazası yapılmakta hem de Java Web ve Spring MVC'nin derinliklerine inilmektedir.

İlk olarak geliştirilecek online kitap mağazasının tasarımı ve bileşenleri ele alınmaktadır. Böylece eğitimde ele alınan teknolojilerin gerçek hayattaki kullanım yerleri netleştirilmektedir.

- JSP ile Java Web dünyasına giriş yapılarak bir Java web projesinin yapısı ve ortamın kurulumu açıklığa kavuşturulmaktadır.
- MVC tasarım şablonu incelenerek Jsp/Java ile bu şablonun nasıl uygulanabileceđi örneklenmektedir.

Bu akış ile MVC tasarım şablonunu kullanan ya da hedefleyen framework veya api'lerin aslında ne yaptığı ve nasıl çalıştıkları incelenmektedir

- Spring MVC ile bir Java web projesinin nasıl daha efektif geliştirilebileceđi incelenmektedir.
- Web projelerinde kullanılacak Spring MVC bileşenleri incelenerek deđişik senaryolar tartışılmaktadır.

**Bootstrap** ile geliştirilen örnek projenin görselliđi zenginleştirilerek kullanıcı dostu bir uygulama ortaya konulmaktadır.

- Bootstrap, Twitter tarafından geliştirilmiş açık kaynak kodlu bir Html5, Css, Js framework'üdür.

İşte o çok sorulan çalışan proje örneği bu eğitimde bu aşamalarla ince ince geliştirilerek örneklenmektedir.

### **Kimler Katılabilir**

Spring MVC ile ileri seviye Java Web Projesi geliştirmek isteyen ya da Spring MVC kullanan bir Enterprise Java projesine sonradan dahil olan her developer katılabilir. Bu eğitimdeki incelenen teknolojilerin zenginliği göz önüne alındığında katılımcılara güncel hayatta büyük avantajlar sağlayacaktır.

- **Süre: 5 Gün**
  - **Eğitmen: Alper AKALIN**
    - **Bilgi:** <http://kimkorkarjavadan.com/>

## **8.1. Applicability Real Web Project Basics**

- Book Store Project
- Developing Domain Model
- Product Model
- Product List
- Using Bootstrap Table Components
- Creating Dynamic Page
- Product CRUD Functions
- Validation
- Session
- Security
- Registration
- Developing Shopping Chart
- Developing Admin Panel
- Developing Rest Api

## 8.2. JSP – Java Server Pages

- Setting up a Web Project
- Understanding the Servlet
- Servlet XML Configuration
- POST, Passing Parameters
- GET and POST
- Request, Session, Context
- Init, Service, ServletConfig
- Understanding the JSP
- Scopes
- PageContext Object
- JspInit and InitParams
- MVC Pattern First Look
- Writing an MVC Application
- Model Object
- RequestDispatcher
- JSTL, SetProperty Tag

## 8.3. MVC Pattern

- Mvc Pattern
- Model View Controller
- Model
- View
- Controller
- Why MVC

## 8.4. Bootstrap API

- Html5
- CSS

- JavaScript
- Responsive web sites
- Mobile first projects
- Templates
- Created by Twitter

## 8.5. Spring MVC

- DispatcherServlet
- Handler Mapping
- Controller, Model
- Spring Tool Suite
- Setting up Spring Mvc
- Writing Controlles
- View Resolvers
- Resource Url's
- Writing Form
- @RequestMapping
- @PathVariable
- Form Validation
- Playing Data
- ModelAndView
- @ModelAttribute
- Custom Validation
- @InitBinder
- Exception Handling
  - Global Exception Handling
- Session
- Interceptors
- Internationalization



- Spring Restful Services
  - `@ResponseBody`