

Architect Enterprise Applications with Java EE Eğitimi

Bu eğitim, kurumsal Java uygulamaları için nasıl güçlü mimariler geliştirileceğini ele alır. Katılımcılar, Java EE (Java Platform, Enterprise Edition) teknolojisinin nasıl kullanıldığı hakkında bilgi sahibi olurlar. Katılımcılar bu eğitime katılarak, mimariyi temel alarak geliştirilen kurumsal Java uygulamalarının hızlı değişime ve büyümeye nasıl uyum sağladığını öğrenirler.

Katılımcılar ayrıca, Java EE teknolojilerini uygularken en iyi şekilde çalışan ve uygulama planlarını oluşturmak için gerekli olan stratejileri de öğrenirler. Bu stratejiler; işlevsel olmayan niteliklerin (ölçeklenebilirlik ve esneklik gibi) kullanımıyla etkili karar almayı, Java EE teknoloji planlarını ve tasarım modellerini kapsar.

Bu eğitimde neler öğreneceksiniz?

- Kurumsal Mimarların rolleri, sorumlulukları ve çıktıları
- İşlevsel olmayan gereklilikler (NFRs), genel sorunlar ve çözümler
- İş gerekliliklerinin mimariye dönüştürülmesi
- İstemci, Web, iş , entegrasyon ve veri katmanlarının mimarisindeki seçimlerin dikkatli bir şekilde planlanması
- Mimari unsurları ve modelleri, araçları, sunucuları ve çerçeveleri seçerken çeşitli değerlendirme kriterlerini uygulama

Introducing Enterprise Architecture

- An Architect's Roles and Responsibilities
- What is Enterprise Architecture?

Introducing Fundamental Architectural Concepts

- Architectural Modeling Using UML
- What is an Enterprise Architecture Framework
- Architectural Deliverable Artifacts
- Architecture Workflow
- 4 + 1 View Model
- Architectural Patterns
- Distinguish between architecture and design

Developing a Security Architecture

- Analyzing the Impact of Security in Distributed Computing
- Understanding Web Services Security
- Examining Security in the Java EE Technology

Understanding Non-Functional Requirements

- Common Practices for Improving Qualities
- Prioritizing Quality-of-Service (QoS) Requirements
- Examining Non-Functional Requirements (NFRs)
- Inspecting QoS Requirements for Trade-offs

Defining Common Problems and Solutions: Risk Factors and System Flexibility

- Identifying Risk Factors
- Designing a Flexible Object Model

Defining Common Problems and Solutions: Network, Transaction and Capacity Planning

- Describing Network Communication Guidelines
- Justifying the Use of Transactions
- Planning System Capacity

Developing an Architecture for the Client Tier

- Discovering Reusability in the Client Tier
- Deployment Strategies for the User Interface
- Client Tier Development Roles
- Security Concerns in the Client Tier
- Selecting User Interface Devices and Technologies
- Information Architecture Client Concerns
- Testing

Developing an Architecture for the Web Tier

- Scaling the Web Tier
- Comparing Web Tier Frameworks
- Providing Security in the Web Tier
- Separation of Concerns
- Responsibilities of the Web Tier

Developing an Architecture for the Business Tier

- Business Tier Technologies
- Development Best Practices
- Architecting the Domain Model

Developing an Architecture for the Integration and Resource Tiers

- Examining Service-Oriented Architecture (SOA)
- Reviewing Java Integration Technologies
- Applying Integration Patterns
- Examining Enterprise Information System Integration

Evaluating the Software Architecture

- Evaluating Software Architectures
- Evaluating Java EE Technologies
- Creating System Prototypes
- Selecting Servers and Frameworks