Java Design Patterns Eğitimi

Süre: **4 Gün**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

Overview

Java dili ve popüler Java-tabanlı framework'ler her bir ana sürüm ile programlama arayüzlerine daha kanıtlanmış geliştirme pratikleri dahil etmektedir. Tasarım örüntüleri olarak da geçen bu pratikler well-known names, kod uygulama ve yeniden yapılandırma tekniklerini ve bunların kullanımıyla ilgili risk ve ödünleşimleri belgelemektedir. Tasarım örüntülerinde, her bir bileşenin sorumluluğu rol olarak belirlenir. Tasarım örüntüsü dokümantasyonunun teamülleri, geliştirme ekiplerinin programlama niyetlerini iletmesini kolaylaştırır ve tüm Java geliştirme topluluğu için bir referans noktası oluşturur.

Java Patterns eğitimi, Java SDK ve EE geliştirmeye özel ortak ve yeni ortaya çıkan örüntüleri değerlendirmektedir. Öğrenciler, özellikle Java EE 6 teamüllerine yoğunlaşılarak, Java'da örüntü-tabanlı tekniklerin derin noktalarını ve evrimini öğrenirler.

Bilgisayarlı uygulamalar öğrencilere seçilmiş örüntüleri NetBeans ya da Eclipse IDE ve the GlassFish Application Server v3 kullanarak nasıl belirleyeceklerini, uygulayacaklarını ve koda doğru yeniden düzenleyeceklerini öğretir. Öğrenciler ayrıca kod yerine tasarımla iletişim kurmayı hızlandırmak için bir UML notasyon altkümesini öğrenirler.

Prerequisites

Herhangi bir ön koşul yoktur.

Who Should Attend

- Uygulama Geliştiricileri
- Mimar
- J2EE Geliştiricisi
- Java Geliştiricileri
- Java EE Geliştiricileri

What You Will Learn

- Java EE 5 ve Java EE 6 örüntü tabanlı özellikleri ayırt etme
- Java EE ortamının her bir katmanında ilgili örüntüleri uygulama
- Katmanlar arası iletişimleri iyileştirmek için kod refaktör edilmesi
- Örüntü-tabanlı geliştirmeyi bir uygulama/yürütme mimarisiyle ilişkilendirme
- Nesne yönelimli prensipler ve tasarım yönlendirici ilkelerini uygulama
- Java'ya özgü kod problemlerine iyi bilinen örüntüleri uygulama

Outline

Java Design Patterns Eğitimi

Süre: **4 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Reviewing Object-Oriented Principles in Java

- Describe how OO concepts apply to Java
- Describe how OO principles apply to Java
- List the goals of an OO language
- Interpret Unified Modeling Language (UML) notation and create UML diagrams
- Identify selected design patterns

Reviewing Gang of Four Patterns

- List key behavioral, creational and structural patterns
- Apply the Facade pattern
- Apply the Strategy pattern
- Apply the Observer pattern
- Apply the Composite pattern
- Review the Model-View-Controller (MVC) patterns

Implementing Patterns in Java

- Use implementation patterns designed for Java
- List forces affecting class, state, and behavioral patterns
- Describe how patterns, idioms and refactoring differ from each other

Exploring Changes in Java EE Technology

- Describe the design goals of the Java EE model
- Describe improvements in the Java EE 6 model

Implementing Integration Patterns

- Describe design patterns for the integration tier
- Review Java EE integration changes that apply design patterns
- Identify use cases for applying integration tier patterns

Implementing Patterns in Business Components

- Describe the role of an enterprise bean
- Describe design patterns for the business tier

Implementing Infrastructural Patterns in Java EE

- Describe the role of infrastructural Java EE patterns
- Describe the Service Starter pattern
- Describe the Singleton pattern
- Describe the Bean Locator pattern
- Describe the Resource Binder pattern

Implementing More Infrastructure Patterns

- Describe how Java EE interceptors work
- Describe the Dependency Injection Extender pattern
- Describe the Payload Extractor pattern
- Describe the Context Holder pattern
- Describe the Thread Tracker pattern

Java Design Patterns Eğitimi

Süre: **4 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Exploring Anti-Patterns

- Describe the Law of Leaky Abstractions
- Define AntiPatterns
- Describe Integration Tier AntiPatterns
- Describe Business Tier AntiPatterns
- Describe Presentation Tier AntiPatterns

Selecting Patterns for Architecture

- Define the roles of architect, designer, and developer
- Describe the relationship between design patterns and architecture
- List guidelines for applying patterns to an architectural solution