

# Enterprise Softwarearchitekturen in Java

**Dauer: 5 Tage**

**1. Tag: Vorbereitungstag** ..... 2

Der erste Tag richtet sich an alle, die bislang wenig Praxiserfahrung mit der Programmiersprache Java haben. Die Teilnehmer erhalten einen Einblick in die universelle Java-entwicklungsumgebung „Eclipse“ und lernen die Grundlagen der Java Architektur und Programmierung. Ein sehr wichtiger Bestandteil des Workshops ist das beispielhafte Erarbeiten einiger Klassen der umfangreichen Java API. Das „Schmankerl“ zum Schluss ist die Programmierung einer kleinen Java Benutzeroberfläche. ... 2

**Voraussetzungen:** Kenntnisse einer Programmiersprache (z.B. C++) ..... 2

**Inhalte:** ..... 2

**2. Tag: Java Message Service (JMS)** ..... 2

Die Java Message Service API ist eine standardisierte Programmierschnittstelle für den Austausch von Nachrichten aus dem Java Enterprise Edition Umfeld. Der JMS Workshop behandelt die beiden Kommunikations-Modelle Point-to-Point (PTP) und Publish/Subscribe, und zeigt anhand der Implementierung von Apache ActiveMQ wie eine robuste und portable Nachrichtenverarbeitung in Java realisiert werden kann. ..... 3

**Inhalte:** ..... 3

**3. Tag: Spring Framework** ..... 3

Bei diesem Workshop handelt es sich nicht um einen Springlehrgang für Anfänger und Profis auf eigenem Pferd, sondern es werden einige, interessante Teile des Java Frameworks „Spring“ behandelt. Das Spring Framework bietet eine so genannte "leichtgewichtige" Alternative zur Java Enterprise Edition, denn in Gegensatz zu Java EE sind nur wenige Abhängigkeiten zum Framework vorhanden. Es werden ausschließlich normale Java Objekte verwendet und die verwendeten Dienste der Java EE Plattform wie z. B. der Java Message Service werden nur deskriptiv konfiguriert. Der Vorteil liegt in der einfachen Entwicklung sowie der einfachen Testbarkeit der Anwendung. Der praktische Umgang mit Spring ist Schwerpunkt dieses Workshops. ..... 4

**Inhalte:** ..... 4

**4. Tag: OSGi** ..... 4

Die *Open Services Gateway initiative* (OSGi) ist eine Spezifikation die es Softwareentwicklern erleichtert, sehr komplexe Anwendungen zu modularisieren und zu verwalten. Zentrales Merkmal der Java-basierten OSGi-Service-Plattform ist eine strikte Entkopplung einzelner Module durch explizites Importieren und Exportieren von Klassen. OSGi ist insbesondere als Baukasten für die Plugin-Entwicklung bekannt, so basieren z.B. alle Plugins der Entwicklungsumgebung Eclipse auf OSGi Technologie. Anhand der OSGi-Implementierung Equinox wird gezeigt, wie voneinander abhängige Module als Bundles implementiert und konfiguriert werden. ..... 5

# 1. Tag: Vorbereitungstag

Der erste Tag richtet sich an alle, die bislang wenig Praxiserfahrung mit der Programmiersprache Java haben. Die Teilnehmer erhalten einen Einblick in die universelle Java-entwicklungsumgebung „Eclipse“ und lernen die Grundlagen der Java Architektur und Programmierung. Ein sehr wichtiger Bestandteil des Workshops ist das beispielhafte Erarbeiten einiger Klassen der umfangreichen Java API. Das „Schmankerl“ zum Schluss ist die Programmierung einer kleinen Java Benutzeroberfläche.

**Voraussetzungen:** Kenntnisse einer Programmiersprache (z.B. C++)

## Inhalte:

- **Einführendes Beispiel: Hello World**
  - Installation des Java Development Kits (JDK)
  - Kurze Einführung in das JDK
  - Erstellen eines Beispiels „HelloWorld“
- **Umgang mit der Entwicklungsumgebung Eclipse**
  - Erstellen von Projekten, Assistenten
  - Debugging
  - Refactoring
- **Grundlegende Sprachelemente der Programmiersprache Java**
  - einfache Anweisungen (Variablen, Typen, Zuweisungen)
  - Bedingte Anweisungen, Schleifen
  - Listen und Felder (Arrays)
  - Klassen, Methoden, Vererbung und Interfaces
  - Fehlerbehandlung
- **Die Java Bibliothek - Java API**
  - Beispielhafte Verwendung der Java API anhand von „java.io“
  - Lesen und Schreiben einer Datei
  - nützliche Filesystem Methoden
  - Client/Server Kommunikation mit dem Remote Method Invokation (RMI) API
- **GUI Programmierung mit Swing**
  - Design einer Swing Benutzeroberfläche in Eclipse
  - Eventbehandlung
  - Programmierung einer Beispielbenutzeroberfläche

# 2. Tag: Java Message Service (JMS)

Die Java Message Service API ist eine standardisierte Programmierschnittstelle für den Austausch von Nachrichten aus dem Java Enterprise Edition Umfeld. Der JMS Workshop behandelt die beiden Kommunikations-Modelle Point-to-Point (PTP) und Publish/Subscribe, und zeigt anhand der Implementierung von Apache ActiveMQ wie eine robuste und portable Nachrichtenverarbeitung in Java realisiert werden kann.

## **Inhalte:**

- **Einführendes Beispiel in die JMS-API**
  - JMS Client, JMS Producer, JMS Consumer, JMS-Message
  - Connections, Sessions
  - Nachrichten über Listener empfangen oder pollen
- **JMS-Message im Detail**
  - Message-Typen, Message-Body
  - Header und Properties
  - Delivery-Mode, Time-to-Live, Prioritäten
  - Message-Selector
- **Anmelde-/Versendesystem (Publish-Subscribe Messaging)**
  - An- und Abmelden an einem Topic, Topic
  - TopicConnectionFactory, TopicConnection, TopicSession
  - TopicPublisher, TopicSubscriber
  - Durable und Non-Durable Subscriptions
- **Nachrichtenwarteschlangen (Point-to-Point Messaging)**
  - Message-Queue, Queue
  - QueueConnectionFactory, QueueConnection, QueueSession
  - QueueSender, QueueReceiver
  - Queue anschauen
- **Spezielle Nachrichtenübertragungsmethoden**
  - Garantierte Übertragung
  - JMS Acknowledgement Modes
  - Nachrichtengruppen
  - JMS-Nachrichten in Transaktion
  - Nachrichten-Priorität

## **3. Tag: Spring Framework**

Bei diesem Workshop handelt es sich nicht um einen Springlehrgang für Anfänger und Profis auf eigenem Pferd, sondern es werden einige, interessante Teile des Java Frameworks „Spring“ behandelt. Das Spring Framework bietet eine so genannte "leichtgewichtige" Alternative zur Java Enterprise Edition, denn in Gegensatz zu Java EE sind nur wenige Abhängigkeiten zum Framework vorhanden. Es werden ausschließlich normale Java Objekte verwendet und die verwendeten Dienste der Java EE Plattform wie z. B. der Java Message Service werden nur deskriptiv konfiguriert. Der Vorteil liegt in der einfachen Entwicklung sowie der einfachen Testbarkeit der Anwendung. Der praktische Umgang mit Spring ist Schwerpunkt dieses Workshops.

## **Inhalte:**

- **Grundlagen des Spring Application Frameworks**
  - Das Prinzip „Inversion of Control“ (IoC) und „Dependency Injection“
  - Überblick über das Framework
  - Einbindung in Eclipse – das Spring IDE Plugin
- **Bean Konfiguration in Spring**
  - einfache Werte injizieren
  - Bean-Referenzen konfigurieren
  - Setter-Injection, Konstruktor-Injection
  - verschiedene Fabrikmethoden
- **Testen mit Spring**
  - Anwendungen mit JUnit und Spring testen
  - Isolierte Integrationstests
- **Remote-Zugriffe mit Spring**
  - Integration von RMI in Spring
  - RmiServiceExporter und RmiProxyFactoryBean
- **Spring und Java Message Service (JMS)**
  - Einbindung des JMS in das Spring Framework
  - Asynchrone Nachrichtenverarbeitung mit JMS
  - Verwendung von JmsTemplates
  - Beispiel: ActiveMQ aus Spring nutzen

## **4. Tag: OSGi**

Die *Open Services Gateway initiative* (OSGi) ist eine Spezifikation die es Softwareentwicklern erleichtert, sehr komplexe Anwendungen zu modularisieren und zu verwalten. Zentrales Merkmal der Java-basierten OSGi-Service-Plattform ist eine strikte Entkopplung einzelner Module durch explizites Importieren und Exportieren von Klassen. OSGi ist insbesondere als Baukasten für die Plugin-Entwicklung bekannt, so basieren z.B. alle Plugins der Entwicklungsumgebung Eclipse auf OSGi Technologie. Anhand der OSGi-Implementierung Equinox wird gezeigt, wie voneinander abhängige Module als Bundles implementiert und konfiguriert werden.

## **Inhalte:**

- **Einführung in das OSGi-Framework**
  - Überblick über das OSGi Framework
  - Einsatzmöglichkeiten des Frameworks
- **Erster Service und Bundle programmieren**
  - Java-Archive eines OSGi-Frameworks
  - Bundle Activator und Lebenszyklus
  - `manifest.mf`-Datei und Installation
  - Auflösen von Abhängigkeiten
  - Bundles und Klassenpfade
- **OSGi System-Services**
  - Standard-Services und Framework-Services
  - Service-Schnittstellen und Implementierung
  - Bereitstellung von Service Listenern
- **Eclipse als Entwicklungsplattform für OSGi**
  - Installation von Equinox
  - Einsatz der Managementconsole
  - Erstellen von Bundles mit Equinox