

Price: R7,800.00 excl. VAT
Duration: 3 days
Code: JARCH

JEE Architecture

Description

Java Enterprise Edition (JEE) is a collection of technologies for the Java platform that is designed to support large, complex software systems in a corporate environment. This course will give you an overview of all the technologies involved. You will learn how JEE systems are organised, the relationships between the technologies, and principles to guide your design of JEE systems.

Objectives

After you have completed the JEE Architecture course, you will be able to:

- Understand the JEE architecture.
- Choose the appropriate JEE technology for your needs.
- Choose the appropriate framework for an application.
- Identify and use the core JEE design patterns.

Intended Audience

You should attend the JEE Architecture course if:

- You are a Java programmer and you need to develop systems using the JEE architecture.
- You are an architect, project manager or systems analyst and you need to understand the JEE architecture.
- You need a detailed overview of JEE to understand the role of the various technologies.

If you need an in-depth understanding of specific technologies within JEE, then you should think about attending one of our more specialised courses:

- Java Servlets and JSP course
- JavaServer Faces course
- Enterprise JavaBeans course
- Java Web Services course

Prerequisites

Before you attend the JEE Architecture course:

- You must have attended our Java Programming course or already be comfortable with the fundamentals of the Java programming language.

Course Contents

Distributed Computing Overview

- Monolithic software development.
- Client-server model.
- Multi-tier development.
- Origins of and need for distributed computing.
- Component-based architectures and JEE containers.
- CORBA and Java IDL.
- RMI and RMI-IIOP.
- Web Services: SOAP vs RESTful.

JEE Application Servers and Containers

- JEE Application Servers (JAS).
- Servlet and EJB containers.
- IoC/DI containers.
- Container Services.

JEE Technology, Container Services and APIs

- Servlets and Java Server Pages (JSP).
- Enterprise JavaBeans (EJB).
- Java Naming and Directory Interface (JNDI).
- Remote Method Invocation (RMI)
- RMI-IIOP (RMI over Internet Inter-ORB Protocol).
- Java Database Connectivity (JDBC).
- Java Persistence API (JPA).
- Java Transaction API (JTA) and Transaction Service (JTS).
- Java Message Service (JMS).
- JavaMail and Java Activation Framework (JAF).
- JEE Connector Architecture (JCA).
- Java Authentication and Authorization Services (JAAS).

IoC, DI and MVC Frameworks

- MVC architecture.
- Inversion of Control (IoC) and Dependency Injection (DI).
- The use and benefits of IoC and DI containers.
- JavaServer Faces.
- Struts and WebWork.
- Spring MVC and Spring Web Flow.
- GWT.
- Grails.
- Play.
- Vaadin.
- Wicket.

Enterprise JavaBeans

- Comparison between EJB2 and EJB3.
- EJB interfaces, components and lifecycle.
- Stateless, stateful and singleton session beans.
- Message driven beans and message queue (MQ) servers.
- Point-to-point versus publish and subscribe messaging.
- EJB3 Entity beans and the JPA.
- Java annotations vs XML deployment descriptors.

Design Patterns and Principles

- Gang of Four (GoF) Design Patterns.
- Core JEE Design Patterns.
- SOLID design principles
- GRASP guidelines.

- JEE Best Practices.

*** The lecturer reserves the right to modify the contents of the course to suit the needs of the delegates.*