>> JAVA

Object-Oriented Analysis and Design Using UML

EGOS! the education company

Java ist eine plattformunabhängige Programmiersprache, die ursprünglich von der Firma Sun Microsystems, entwickelt wurde. Java wird in allen IT-Bereichen eingesetzt und ist für unzählige Betriebssysteme und Plattformen, vom mobilen Telefon bis hin zur Echtzeit-Großrechneranlage geeignet.

Ihr Nutzen

Die Teilnehmer erwerben Kenntnisse über Softwareentwicklungsprozesse, objektorientierte Verfahren und die Unified Modeling Language (UML). Der Kurs konzentriert sich auf Aufgabenstellungen, mit denen sich Softwareentwickler konfrontiert sehen, und vermittelt einen praktischen, vollständigen Ablaufplan für objektorientierte Analyse und Design (OOAD).

Preis pro Teilnehmer

EUR 2650,- exklusive der gesetzlichen MwSt.

Seminardauer

5 Tag(e)/Day(s)

Voraussetzungen

Fundamentals of the Java Programming Language~4272
oder entsprechende Kenntnisse.

Empfohlen wird auch:
Java Programming Language~4273

Hinweise

Version: 14

Seminarinhalte

Tag 1:
* Der objektorientierte Softwareentwicklungsprozess (OOSD)
- Beschreibung, Darstellung, Methoden
- Zweck und Aktivitäten der wichtigsten Arbeitsabläufe

* Prinzipien der Objekt-Orientierung
- Einfluss auf den Software-Entwicklungsprozess

* Auswählen einer objektorientierten Methodik
- Überblick, “best practices“
- Vergleich der gebräuchlichen Methoden
- Auswahl der geeignetsten Methodik für ein Projekt

* Projekt-Vision festlegen und dokumentieren
- Funktionelle Erfordernisse des Systems mit Interviews bestimmen
- Bestimmung von Risiken und Einschränkungen

Tag 2:
* Erfassen der Systemanforderungen
- Validieren und Verfeinern auf Basis der Projekt-Vision
- System Requirements Specification (SRS)

* Analyse der Systemanforderungen - Anwendungsfälle (Use cases)
- Elemente eines UML Anwendungsfall-Diagramms
- Anwendungsfall-Diagramm auf Basis der SRS erstellen
- Anwendungsfall-Szenarien
- Anwendungsfälle vererben, Abhängigkeiten erkennen

* Elemente eines Aktivitätsdiagramms
- Validieren von Anwendungsfällen mit einem Aktivitätsdiagramm

Tag 3:
* Abstraktionen ermitteln
- CRC-Karten (Class Responsibility Collaboration)

* Problem-Modell definieren
- Elemente eines UML Klassendiagramms
- Domänenmodell mittels Klassendiagrammen definieren
- Elemente eines UML Objektdiagramms

* Robustheitsanalyse
- Elemente eines UML Kollaborationsdiagramms
- Designmodell für Anwendungsfälle
- Elemente eines UML Sequenzdiagramms

Tag 4:
* Das Architekturmodell, fundamentale Architektur-Konzepte
- Entwerfen einer Systemarchitektur
- Berücksichtigung von nichtfunktionalen Anforderungen
- Berücksichtigung von Entwicklungsbeschränkungen
- Workflow und Diagramme

* Architektur-Modelle für:
- Client- und Präsentationsschicht (Benutzerschnittstellen)
- Business Schicht, verteilte Systeme
- Ressource- und Integrationsschicht

Tag 5:
* Das Lösungsmodell
- Erstellen eines Systemdesigns
- Verfeinern des Domänenmodells

* Design unter Verwendung von Design-Pattern
- Elemente von Software-Patterns
- Composite, Strategy, Observer, Factory

* Objektstatus
- Zustandsdiagramm, Zustandsübergangsdiagramm (statechart)

EGOS! PDF MVC Content Application Framework v9.2611.10.1. ©2026 EGOS! The Education Company, Alle Rechte vorbehalten. Created on 17.11.2026 09:11:11. ID4197. JAVA Object-Oriented Analysis and Design Using UML

©2026 EGOS! The Education Company, Alle Rechte vorbehalten.