

# ) Seminar )

## Clean Code und Software Craftsmanship

Softwaresysteme verfallen im Laufe der Entwicklungszeit, die Lesbarkeit und Wartbarkeit können rapide abnehmen. Auffällig wird dieser Umstand aber meist erst, wenn die Entwicklungsproduktivität einbricht, man also neue Features nur noch sehr langsam veröffentlicht und Änderungen zu ungeahnten Seiteneffekten führen. Um dem Qualitätsverlust vorzubeugen, empfiehlt sich der konsequente Einsatz von Praktiken der Clean Code Initiative. Trainieren Sie in diesem Workshop an praxisnahen Beispielen testgetriebene Entwicklung und die sinnvolle Verwendung von Entwurfsmustern, sowohl in purem Java als auch mit Hilfe moderner Frameworks und Tools. Identifizieren Sie zudem problematische Abschnitte (Code Smells), verbessern Sie die Software durch gezielte Refactorings und nutzen Sie Review-Techniken, um Wissen zu transferieren und die Qualität hoch zu halten.

Anhand vieler praktischer Übungen lernen Sie die wichtigsten Richtlinien und Vorgehensweisen kennen. Vergleichen und diskutieren Sie die verschiedenen Lösungen mit den anderen Teilnehmern und dem Trainer. Üben Sie sich in den handwerklichen Fähigkeiten moderner Softwareentwicklung, um nicht nur funktionierenden, sondern auch gut verständlichen, wartbaren, erweiterbaren und robusten Code entwickeln zu können. Machen Sie Ihre Softwareprojekte so fit für die Zukunft.

### Seminar-Inhalt

#### Software Craftsmanship

- Softwareentropie
- Professionalität und Berufsethik
- Clean Code Initiative und Software Craftsmanship Manifest
- Code Katas und Coding Dojos

#### Grundlegende Konzepte

- Sinnvolle und verständliche Namensgebung
- Kapselung, Kohäsion und Kopplung
- Modularisierung und Management von Abhängigkeiten
- Objektorientierte Prinzipien (SRP, DRY, Open-Closed-Principle)
- Separation of Concerns - Trennen von Querschnittsbelangen (AOP)

#### Design Patterns und Refactoring

- Klassische Entwurfsmuster (Strategie, Observer, Factory, ...)
- Erkennen von Code Smells
- Toolgestütztes Refactoring
- Wartung von Legacy Code

#### Effektive Entwicklung mit Java

- Immutability und unveränderbare Datenstrukturen
- Besonderheiten von java.lang.Object
- Fehler- und Null-Referenz-Behandlung
- Funktionsliterale (Lambdas) und Higher Order Functions
- Wiederverwendung durch Komposition und Currying

#### Testautomatisierung und Qualitätssicherung

- Testgetriebene Entwicklung (TDD)
- Continuous Integration und Delivery
- Code Metriken mit Lint-Tools
- Wissenstransfer durch Code Reviews

#### Ausblick

- Durchführung von Coding Dojos und Retreats
- Integration in agile Vorgehensweisen
- Zusammenhang zu modernen Softwarearchitekturen

### Zielgruppe

Java Softwareentwickler, technische Projektleiter, Architekten

### Voraussetzungen für Seminar-Teilnahme

Gute Grundkenntnisse in Java z.B. durch Besuch unserer [Java Schulung](#).

### Seminar-Dauer

4 Tage

### Vorteile einer Seminar-Teilnahme

- Sie werden im Anschluss lesbaren, verständlichen, wartbaren, anpassbaren und zukunftsfähigen Code schreiben.
- Sie lernen oder wiederholen grundsätzliche Design Prinzipien und Muster.
- Besuchen Sie ein kompaktes Training und werden Sie ein professioneller Software-Handwerker.
- Unsere Kurs-Unterlagen sind immer auf dem aktuellsten Stand

### Seminar-Preis

2.130 EUR pro Person  
(inkl. Unterlagen u. Tagesverpflegung zzgl. MwSt.)

### Seminar-Termine

9.4. - 12.4.2018, 3.9. - 6.9.2018

### Inhouse-Kurse

Alle unsere Seminare können wir Ihnen auch für eine Durchführung in Ihrem Hause mit einer speziellen Kalkulation für Exklusiv-Seminare anbieten. Hierbei können die Inhalte aller unserer Seminare beliebig für Ihr individuelles Training zu einem Wunschseminar zusammengestellt werden.