

Agendavorschlag zum Seminar

Java Vertiefung

Dauer: 4 Tage

Autor:
Stephan Karrer

Stand:
1.10.2018

Zu dieser Agenda

Diese Agenda ist ein Vorschlag bzgl. der Inhalte und Unterrichtszeiten für das Seminar

Java Vertiefung

Bitte überprüfen Sie, ob diese Konzeption Ihren Erfordernissen entspricht. Änderungen oder Ergänzungen hinsichtlich der Seminarinhalte oder des Zeitplans nehmen wir in Absprache mit Ihnen, soweit möglich, gerne vor.

Zeitliche Struktur des Seminars

Wir gehen von 8:30 - 16:30 täglicher Seminarzeit aus. Individuelle Regelungen zu den Unterrichtszeiten und den Pausenregelungen stimmen wir mit Ihnen ab.

Hard/Software - Medien

Die Teilnehmer benötigen PCs unter Windows oder Linux mit folgender zusätzlicher Software:

- Java Development Kit Version 8 (JDK 8)
- Eclipse für Enterprise Developers als Entwicklungsumgebung
- Datenbanksystem Oracle Express Edition
- EclipseLink als JPA-Provider
- Internet-Zugang

Testen mit JUnit

- Zentrale JUnit-Klassen und Annotationen für JUnit 4
- Assertions
- JUnit in Eclipse
- Lebenszyklus einer Testsuite
- Organisation der Testklassen, Sammeln von Testfällen in Suites
- Testen von Exceptions

Bemerkung:

JUnit kann dann bei den folgenden Praxisbeispielen durchgängig eingesetzt werden

Vertiefung: Vererbung, Abstrakte Klassen, Interfaces

- Vererbung ist Design-Thema
- Komposition vor Vererbung: Warum Vererbung einschränkt
- Details zur statischen und Instanz-Initialisierung
- Konstruktoren defensiv programmieren
- Private Konstruktoren und Factory-Methoden
- Abstrakte Klasse versus Interface
- Lose Kopplung bei der Codierung, enge Kopplung zur Laufzeit
- Ausblick: Dependency Injection
- Schnittstellenerweiterungen mit Java 8:
Statische und Default Methoden in Interfaces

Generische Datenstrukturen

- Worum geht es
- Typparameter, generische Klassen und generische Methoden
- Beispiel: Eigene generische Klassen und Interfaces
- Wildcards: Syntax und Sinn
- Unterschied zu den klassischen typisierten Arrays

Design Patterns

- Was sind Muster in der Software-Entwicklung?
- Kategorien von Mustern

Erzeugungsmuster

- Factory
- Builder
- Prototyp
- Singleton

Strukturmuster

- Adapter
- Bridge
- Decorator
- Facade
- Composite

Verhaltensmuster (Behavioral Pattern)

- Template Method
- Command
- Observer
- Visitor
- Strategy

Ausnahmebehandlung im Überblick

- Ausnahmen behandeln mit try-catch-finally
- Ausnahmen weiterleiten
- Checked Exceptions und Unchecked Exceptions, RuntimeException
- Erweiterungen mit Java 7
- Eigene Ausnahmen

Datenstrukturen in Java: Collection-Klassen

- Index-sequentielle, verkettete und gehashte Datenstrukturen: Vor- und Nachteile
- Die Bedeutung von equals() und hashCode()
- Struktur der Collection-API: Interfaces, Klassen und Algorithmen
- Das Iterator-Konzept
- Verwendung gehashter Datenstrukturen
- Verwendung von Optional

Erweiterungen durch Google Guava und Apache Commons

- Unterstützung bei einfachen Datentypen
- Vorbedingungen
- Ausnahmebehandlung
- Ordnen
- Immutable Collections
- Zusätzliche Collection-Typen: Multisets, Multimaps, Tables, ...
- Vereinfachung der I/O-Operationen

Einführung in Threads

- Thread-Konzept
- Runnable und Parameterübergabe bzw. Ergebnisabfrage
- Synchronisationsproblem, Synchronized-Blöcke und Lock-Objekte
- Deadlock-Vermeidung mit Wait und Notify

Verwendung von Thread-Pools

- Runnable und Callable
- Executor-Schnittstellen
- Future und Completable Future
- Fork-Join-Pool

Einführung in Lambda-Ausdrücke und Streams

- Motivation
- Schreibweisen
- Methodenreferenzen
- Streams von Arrays und Collections
- Terminale und intermediäre Operationen
- Typische Map-Reduce-Operationen

Optional (sofern Zeit bleibt): Einführung in JPA

- Objekt-Relationales Mapping
- JPA-Architektur
- Entities und Entity Manager
- PersistenceContext und PersistenceUnit
- CRUD-Operationen
- Synchronisation mit der Datenbank