Kurs: Intermediate Advanced Java 11+

Dauer: 5 Days

Beschreibung

This is a modern, fast-paced course suitable for developers with some previous experience in working with Java and Object-Oriented (OO) programming. The course can also be delivered to developers experienced in other OO languages (Python, C++, C#) but with limited Java exposure, as well as used as an advanced Java course for more experienced developers.

It includes an accelerated, yet thorough, hands-on review of Java foundational concepts, with attention given to OO design and implementation principles. This review also includes an overview of newer language features such as the Date/Time API (Java 8+), type inference with var, and switch expressions.

It then moves on to comprehensive coverage of more advanced topics in Java and OO development to provide participants with a strong grounding to use Java in a sophisticated and productive manner. This includes in-depth coverage of functional programming with lambdas and streams, as well as the Java Platform Module System (JPMS). Java modules presents a fundamental shift in how applications are organized, and interconnected to the libraries they use. Migration strategies are also covered, including a step-by-step case study.

This course covers far more than an introductory course, including important topics such as UML and Design Patterns, and using composition vs. inheritance, which are all key to creating well-structured OO systems. After these important areas, it moves on to the advanced Java topics described above. It teaches a number of useful techniques that enhance productivity and good system design - which may otherwise take Java developers years to absorb on their own.

Unit testing is stressed throughout the course, with most labs implemented as JUnit tests.

The course is very hands-on, including numerous code examples and programming labs that reinforce the concepts presented, so that you can immediately employ what you've learnt in your current project.

Be prepared to work hard and learn a great deal!

Ziele

- Solidify Java foundational knowledge, including the important contracts of class Object
- Understand the uses and consequences of inheritance and composition, and reinforce the role of interfaces
- Reinforce fundamental OO principles such as cohesion, coupling, and polymorphism
- Use the JUnit testing framework and become fluent in writing assertions to verify correct program behavior
- Familiarity with UML modeling in class diagrams and sequence diagrams
- Use advanced techniques for object creation, including factories and singletons
- Use established design patterns for object composition, including Strategy, Decorator, and Façade
- Write and use generic classes and methods
- Learn the use cases for inner classes and refactor existing code to use them when appropriate
- Create and use custom annotations
- Be familiar with reflection and how to use it
- Understand the role of functional interfaces
- Understand lambda expressions and method references, and use them to pass behavior (methods)
- Use the Stream API to perform complex processing of collections and other input sources
- Create and use Java modules, understanding module descriptors, modular JARs, exports and dependencies, and the modulepath
- Understand the structure and behavior of the modular JDK, and how it supports modular applications as well as legacy classpath-based code
- Migrate classpath-based applications to Java 11, understanding the stages of migration and options available

Kurs Inhalt

Session 1: Review – Basics

- Java Environment
- Classes and Objects
- Packages, Enums, Arrays
- Exceptions
- Date and Time API
- New Language Features

Session 2: Review (Inheritance and Interfaces)

- UML Overview
- Inheritance
- Interfaces
- New Interface Features (Java 8+)
- Guidelines

Session 3: Junit

- Overview
- Tests, Assertions, and Fixtures
- Best Practices and Test-Driven Development Overview (TDD)

Session 4: Collections and Generics

- Collections Overview
- Lists, Sets, and Maps
- Writing Generic Classes

Session 5: Techniques of Object Creation

- Design Patterns Overview
- Controlling Object Creation
- Singleton Pattern
- Simple Factory
- Factory Method Pattern
- Other Techniques

Session 6: Using Composition and Inheritance Effectively

- Inheritance and Composition - Pros and Cons
- Strategy Pattern
- Decorator Pattern
- Façade and Other Patterns

Session 7: Inner Classes

- Overview and Motivation
- Defining and Using Inner Classes
- Static Nested Classes

Session 8: Annotations

- Overview
- Using Annotations
- Writing Custom Annotations

Session 9: Reflection

- Overview and API
- Working with Objects Reflectively

Session 10: Lambda Expressions

- Using Lambda Expressions
- Method References

Session 11: Streams

- Overview
- Understanding the Stream API
- Stream Processing
- Collectors

Session 12: Introduction to Modules

- Motivation and Overview
- Types of Modules
- Modular JDK
- Our Approach

Session 13: Working with Modules

- Defining and Using Modules
- Services
- Compatibility and Migration

Conclusion