

Java Modules and New Features (to Java 11)

Course Summary

Description

Java's new release cycle introduces new features at a more rapid pace, requiring increased attention from users. This course provides in-depth coverage of the important new features through Java 11 LTS. It also provides an overview of useful new features in Java 12+ that are likely to appear in Java 17 (the next LTS release).

Technical coverage starts with the new Java Platform Module System (JPMS), which presents a fundamental shift in how applications are organized, and interconnected to the libraries they use. We introduce modules from the ground up, show how to use them, describe your choices in usage, as well as caveats and pitfalls. We also provide guidelines for migration from non-modular applications.

The course covers many other new and powerful features that will be useful to developers (see the detailed outline for specifics). The course also includes several extra optional labs and guided self-study exercises in these additional areas.

The course is very hands-on, including numerous code examples and programming labs that reinforce the concepts presented, so that attendees can immediately employ what they've learned in their current projects.

Objectives

At the end of this course, students will be able to:

- Understand the new Java release cycle and Long Term Support (LTS) releases
- Be familiar with significant deprecated and removed features, and how to work around them
- 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, how it supports modular applications as well as legacy classpath-based code, and the implications of strong encapsulation on each
- Migrate classpath-based applications to Java 11, understanding the stages of migration and options available
- Recognize the issues with 3rd party libraries in a migration effort, and how to work with them on the modulepath and classpath
- Use local-variable type inference with `var`, including lambda parameters
- Gain a practical working knowledge of the JShell REPL tool, including working with code snippets and variables, configuration, and using external libraries
- Use the HTTP Client to access HTTP resources from Java, as well as understand the other options available
- Work with various HTTP request and response types, using both synchronous and asynchronous techniques
- Outline the new factory methods in Java Collections and understand native immutable collections
- Describe the motivation for multi-release JAR files (MR-JARs), understand their structure, and how to create them
- Understand the runtime behavior of MR-JARs in both legacy and modern JVMs, and principles and strategies for working with them effectively
- Outline the principles of Reactive Programming and how it differs from traditional synchronous invocation models
- Describe Reactive Streams and the role of the Flow API that defines the Java platform's support for them
- Understand the characteristics and benefits of custom runtime images
- Use `jdeps` to analyze application dependencies, and create custom runtimes with `jlink`, for both modular and classpath-based applications
- Explore some of the more important additional features and APIs, including new features for interfaces, the Process API, new JDK tools and command line options

Java Modules and New Features (to Java 11)

Course Summary (cont'd)

Topics

- Preface: Java State of the Union
- Introduction to Modules
- Working with Java 9 Modules
- Type Inference
- JShell
- Http Client
- Other New Features
- Custom Runtime Images

Audience

Suitable for Java developers that want to ramp up on the new features in Java 11 and beyond, with special focus on Java 9+ modules

Prerequisites

Working knowledge of Java programming, including use of inheritance, interfaces, and exceptions.

Duration

Two days

Java Modules and New Features (to Java 11)

Course Outline

- I. *Java State of the Union*
 - A. New Release Cycle
 - B. What's In
 - C. What's Out
 - D. The Move to Java Modules
- II. *Introduction to Modules*
 - A. Motivation and Overview
 - B. Types of Modules
 - C. Modular JDK
 - D. Our Approach
- III. *Working with Java 9 Modules*
 - A. Defining and Using Modules
 - B. Services
 - C. Compatibility and Migration
 - D. Conclusion
- IV. *Type Inference*
 - A. Local-Variable Type Inference
 - B. Brief Overview of Lambdas
 - C. Local-Variable Syntax for Lambdas
- V. *JShell*
 - A. Introduction to JShell
 - B. Working with Code
 - C. Using Libraries (Modules, Jars, etc.)
- VI. *Http Client*
 - A. Overview
 - B. API
 - C. Usage and Features
- VII. *Other New Features*
 - A. Collection Factory Methods
 - B. Multi-Release JARs
 - C. Reactive Programming
 - D. Miscellaneous
 - E. Upcoming Features in Java 12+
- VIII. *Custom Runtime Images*
 - A. Application-Specific Runtimes
 - B. Benefits
 - C. Creating Runtime Images with jlink