



☎ +44 7989 401397

✉ info@olsensoft.com

Modern Java

(3 days)

Course overview

This course looks at the new language features and APIs from Java 9 to Java 17. The most important change is undoubtedly the introduction of modules, as this will affect all Java applications going forward. There are also several new language features and APIs, including Reactive Streams, a standard HTTP2 client API, process API improvements, and more. We'll take a full look at all these new features, and discuss strategies for adopting and integrating them into your existing codebase.

What you'll learn

- Java Platform Module System (JPMS)
- New language features in modern Java
- New APIs in modern Java
- Reactive streams
- Adoption strategies

Prerequisites

- Good experience in Java SE 8

Course details

- **Introduction:** Java in the 'Post-Java-8' era; Tool support; Oracle JDK vs. OpenJDK; Using JShell
- **Java Modules:** Getting started with Java modules; Using the modular JDK; Creating a modular application; Creating a multi-module app; Modularized JARs
- **Going Further with Java Modules:** Standalone applications; Services; The unnamed module; Automatic modules; Migration strategies
- **New Language Features:** Private methods in interfaces; Type interference; Miscellaneous improvements; Multi-version JARs
- **Collection Enhancements:** Immutable collections; Enhancements to the Stream interface; Enhancements to the Optional class
- **Pattern Matching:** Getting started with pattern matching; Pattern matching techniques
- **Records:** Getting started with records; Defining members in a record; Additional techniques for records
- **Switch Usage:** switch statement enhancements; switch expressions
- **Sealed Classes:** Getting started with sealed classes; Additional techniques with sealed classes

- [Reactive Streams](#): Overview of reactive programming; Understanding the Java Flow API; Implementing publishers and subscribers; Implementing processors; Practical applications
- [Miscellaneous New APIs](#): Process API improvements; Walking the stack trace; HTTP/2; Garbage collection; Enhanced @Deprecated annotation; Minor tweaks