

Updated on 11/10/2024

Sign up

Java Training: Advanced Programming

4 days (28 hours)

Presentation

If you've mastered the basics of Java, this course is for you. You'll also learn about advanced language concepts such as reflexive programming and concurrent programming.

You'll discover how to test, [debug](#) and [optimize](#) your Java applications efficiently. We'll also introduce you to the most popular frameworks: Spring, Hibernate and JSF.

Our advanced Java training course will give you the techniques you need to create complex Java programs. You'll learn how to optimize and secure your applications in the best possible way.

As with all our training courses, this one will introduce you to the latest version of Java ([JDK 23](#)).

Objectives

- Master the new advanced features of the Java language
- Know how to use the main Java frameworks and libraries
- Apply OOP techniques to build classes and create objects
- Develop Java applications using relational databases
- Securing your Java application and understanding class loading
- Test, debug and optimize your application
- Understanding and applying reflective and concurrent programming

Target audience

- Developers
- Technical architects
- Applications Development Manager

- IT Project Manager

Prerequisites

Have taken our [Java](#) training course.

Our Java training program: Advanced programming

What's new in Java 19

- Overview of recording models
- Support for Linux/RISC-V instruction set architecture
- Overview of function and foreign memory APIs
- Overview of virtual threads
- Fourth vector API incubation
- Third overview of pattern matching for switch

Reflective programming

- What is reflective programming?
- Reflexivity about Java
- Annotations
- Get the object of a Class class
- Determine the object of the class
- Constructors Methods and fields
- API reflection
- Instantiation and dynamic invocation

Concurrent programming and multi-threading

- Threads and multithreaded programming
- The life cycle of a thread
- The problems and benefits of multithreading
- Multithreading with GUI
- Semaphore and synchronization between threads
- Interrupting threads
- Overview of competing collections
- Creating and managing threads
- Managing the competition

Java Management Extension (JMX)

- JMX architecture
- Introducing the instrumentation layer (MBeans)
- Related concepts (Dynamic MBeans, Open MBeans, Model MBeans and MXBeans)

- The agent layer
- The distributed services layer
- The administration console (JConsole)
- Controlling a JVM

Java Message Service (JMS)

- Why JMS?
- JMS 1 and 2 API
- Message management (creation, prioritization, response requests, delivery and expiry times)
- Message types and message objects
- JMS interfaces and classes
- The different types of communication (point-to-point, publish, publish by queue...)
- Filter or group messages
- Securing and administering JMS

Java Server Faces (JSF)

- Introducing JSF concepts
- The MVC model
- Configuring a JSF application
- The request life cycle
- Facelets and Ajax

SQL and database access with JDBC

- A reminder of relational databases
- JDBC API interfaces
- Basic SELECT query
- WHERE clause
- ORDER BY clause
- Merging data from several tables: INNER JOIN
- INSERT statement
- UPDATE statement
- DELETE statement
- Installing MySQL and MySQL Connector/J
- Creating a database in MySQL

Lambda expressions

- Interfaces, lambda expressions and inner classes
- Interfaces
- Lambda expressions
- Internal classes
- Service chargers
- Proxies

- Anonymous class, lambda expression concept
- Handling collections with lambda expressions
- Code SQL : Listeners

Introduction to JSTL

- What is JSTL (Java Server page Standard Tag Library)?
- Tag classification
- Main JSTL functions
- Using custom tags
- How to reduce your code?
- How can you make your code more powerful with JSTL?

Servlets

- What is a servlet?
- HTTP methods; GET, POST, PUT, DELETE, TRACE, OPTIONS
- Servlet life cycle
- Servlet configuration context
- Developing an application using servlets

Networking

- URL manipulation
- Reading a file on a Web server
- Setting up a simple server using Stream sockets
- Setting up a simple client using continuous sockets
- Client/server interaction with Stream Socket connections
- Datagrams: connectionless client/server interaction
- Client/Server/Multithreaded Server

Bookshops

- Introducing the Java Standard Library
- Using the Java Virtual Machine (JVM)
- Easier programming with Java Standard Library
- Reusing Java components with Apache Commons
- Handling JSON, CSV and XML files with the Jackson library

Getting started with Spring

- Introducing Spring
- Framework capabilities
- Aspect-oriented programming
- REST API

Discover Hibernate

- Introducing Hibernate
- Framework configuration
- Managing an occurrence
- Class inheritance
- Generate your code

Testing your applications

- Good testing practices
- The different types of test
- Frameworks for unit testing
- Test Driven Development
- Troubleshooting

Securing your web applications

- Declarative and programmatic security
- The roles
- Basic authentication
- Digest authentication
- Authentication by form
- Standard login form
- Safety constraints
- Web resources collection

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.