+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated 06/06/2025

Sign up

Java Training: Advanced Programming

4 days (28 hours)

Presentation

Our JAVA advanced programming 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. If you've mastered the basic functionalities of Java, this course is for you. You'll also learn about the language's advanced concepts, such as reflective programming and concurrent programming.

In a modern, professional way, you'll learn how to test, debug and optimize your Java applications efficiently, through all the latest Java developments. We'll also introduce you to the most popular frameworks: Spring and Hibernate.

Thanks to a hands-on teaching approach based on a captivating red thread project, you'll explore practical techniques: effective concurrency management with multithreading and virtual threads, performance optimization by mastering the advanced mechanisms of Garbage Collector, and dynamic implementation through reflection and metaprogramming.
Each workshop will enable you to immediately integrate these complex concepts into your project.

This course also covers in depth the essential frameworks of the Java world, such as Spring Boot, Hibernate and JPA, and will guide you towards the creation of robust, secure and maintainable applications. You'll learn how to design high-performance REST microservices, efficiently manage relational data access, optimize SQL queries and exploit ORMs to facilitate data persistence.

Finally, you'll keep abreast of the latest technological developments by discovering all the new features brought by recent versions of the JDK (from Java 8 to Java 24). Modules, Records, Pattern Matching, Scoped Values, Structured Concurrency: this course will enable you to understand and apply these innovations, making you a true expert in advanced Java programming, ready to take on all current and future technical challenges.

As with all our training courses, this one will introduce you to the latest version of Java (JDK 24).

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
- Secure your Java application and understand class loading
- Test, debug and optimize your application
- Understand and apply reflexive and concurrent programming

Target audience

- Developers
- Technical architects
- Application development managers
- IT project managers

Prerequisites

- Completion of our Java training course or equivalent level

Java training program: Advanced programming

Internal classes, genericity and SOLID design in Java

- Internal classes/anonyms
- Advanced genericity
- Functional interfaces
- Enums & immutability
- Patterns & SOLID
- Anonymous class, lambda expression concept
- Practical workshop: Setting up the initial structure of game classes (hero, monster, generic inventory).

Functional programming (Lambdas & Streams)

- Lambdas
- Standard functional interfaces
- API Stream
- Optional
- Parallel Streams
- Practical workshop: Managing a monster list via Streams.

Concurrency & Multi-threading

- Threads and multithreaded programming
- The thread life cycle
- Threads/Runnables
- Synchronization
- ExecutorService
- Advanced locks
- Semaphore and synchronization between threads
- Interrupting threads
- Overview of concurrent collections
- Thread best practices
- Practical workshop: Real-time game (spawn competing monsters).

Memory & Garbage Collector

- JVM memory model
- GC operation
- GC algorithms
- Memory monitoring
- Practical workshop: Memory stress testing, simple optimization.

Reflective & Metaprogramming

- What is reflective programming?
- Reflective API
- Getting the object of a class
- Determining the class object
- Constructors Methods and fields
- Dynamic invocation
- Dynamic instantiation
- Annotations
- Dynamic proxies
- Practical workshop: Dynamic loading of new monsters.

Security & Class Loaders

- JVM security model
- Dynamic class loading
- Plugins/modularity
- Bytecode verification
- Authentication
- Advanced custom ClassLoader management
- Avoiding classic vulnerabilities
- Tips & best practices for securing your application
- Practical workshop: Secure external plugins for new monsters.

Reminder of relational databases & SQL

- Tables (relations): sets organized into rows (tuples) and columns (attributes).

- Primary keys: unique identifier of a record.
- Foreign keys: reference to a primary key in another table (relationship).
- Relationships: types (1:1, 1:N, N:M).
- Referential integrity: consistency of references between tables (via foreign keys).
- Reminder of normal forms (normalization)
  - 1st normal form (1FN)
  - 2nd normal form (2FN)
  - 3rd normal form (3FN)
  - Boyce-Codd normal form (FNBC)
- Basic SELECT query
- WHERE clause, ORDER BY clause
- INSERT, UPDATE, DELETE statements
- Practical workshop: Optimizing a query using the execution plan.

Relational databases & SQL with Java

- Different interfaces in the JDBC API
- SQL connection, PreparedStatement and Transactions
- DAO pattern
- Pooling connections
- The different joins
- Query optimization and execution plan
- Practical workshop: Saving scores JDBC players.

ORM with Hibernate/JPA

- ORM principles
- Hibernate configuration
- Entity mapping
- Entity lifecycle
- JPA/HQL requests
- Lazy loading/cache
- ORM inheritance
- Practical workshop: Game persistence via Hibernate.

Spring Framework

- Aspect-oriented programming
- Dependency injection
- Spring configuration/annotations
- Spring Boot
- REST MVC API
- Spring Data JPA
- AOP
- Spring Security (intro)
- Hands-on workshop: creating a secure REST microservice API for a game action using Spring Boot.

Software quality

- JUnit unit testing
- Mockito/mocks
- TDD
- Efficient debugging
- Troubleshooting strategies
- Log analysis
- Practical workshop: Testing the game's combat rules.

Performance optimization

- Introduction to Java Management Extension (JMX)
- CPU/memory profiling (JFR, VisualVM, Java Flight Recorder, YourKit)
- Algorithm & structure optimization
- Advanced Garbage Collector settings (G1, ZGC, Shenandoah)
- Efficient concurrency management & virtual threads
- Application caching (EHCache, Caffeine)
- Benchmarking (JMH) & load testing (JMeter, Gatling)
- Practical workshop: Profiling and optimizing critical points in the game (e.g. combat, monster generation). Measuring performance improvements.

Main new features of the Java language

- Presentation of Java 21 (LTS) and overview of Java 22/23/24
- Pattern matching
- Virtual threads
- Improvements to modern APIs
- Vector API
- Advanced APIs
- The future LTS 25 (September 2025)

Detailed evolution from JDK 8 to JDK 24

- Modules (JPMS)
- Type inference (var)
- Text Blocks
- Switch Expressions
- Records
- Sealed classes
- Pattern Matching
- Virtual Threads
- Scoped Values
- Stream Gatherers
- Structured Concurrency
- String Templates
- Class-File API
- Importing modules
- Primitive types in patterns
- Vector API
- Early class loading
- Security Manager deactivation
- Compact Object Headers

- Key Derivation Function API

Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced IT 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 forthcoming course, 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 training: 60% hands-on, 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.

Certification

A certificate will be awarded to each trainee who has completed the entire course.