Course : Campus Atlas - JAVA, the fundamentals of programming

Practical course - 5d - 35h00 - Ref. LJB
Price : 2610 € E.T.

NEW

On completion of the course, participants will be able to use the Java language and associated technologies to create an application. This training program is intended for employees of professional branches covered by the OPCO Atlas.

Teaching objectives

At the end of the training, the participant will be able to:

- Understand the basic concepts of the Java language and master its syntax
- Using libraries and APIs
- Understand the concepts of object-oriented programming in Java
- Creating a Java application

Intended audience

Pour les adhérents à l'OPCO Atlas : développeurs, chargés de développement d'applications informatiques, chefs de projet proches du développement…

Prerequisites

Familiarity with the principles of object-oriented programming and experience of a programming language in application development.

Practical details

Quizzes, games in pairs or in groups, discussions, role-playing, intensive hands-on work, integrated development environment.

Teaching methods

60% pratique – 40% théorie. Pour optimiser le parcours d'apprentissage, des modules e-learning peuvent être fournis avant et après la session présentielle ou la classe virtuelle, sur simple demande du participant.

Course schedule

PARTICIPANTS

Pour les adhérents à l'OPCO Atlas : développeurs, chargés de développement d'applications informatiques, chefs de projet proches du développement…

PREREQUISITES

Familiarity with the principles of object-oriented programming and experience of a programming language in application development.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they've developed.

1 Algorithmics - Think before you design - Pre-training digital learning content

- Introduction to algorithms.
- Basic instructions in pseudo-code.

Digital activities

In this online training course, you'll learn how to reason before designing a program by discovering the basics of algorithmics. In particular, you'll study the fundamental instructions in pseudo-code.

2 Object techniques

- General principles of object modeling and programming.
- Abstraction and encapsulation: interfaces.
- The different forms of inheritance, polymorphism.
- Introduction to UML modeling: static model, dynamic model, cooperation model, scenarios.

Hands-on work

UML specification of a case study that will be one of the guiding threads of the following exercises.

3 Basic language constructs

- Variables: declaration and typing.
- Defining fields and methods.
- Control expressions and instructions.
- Enumerated tables and types, autoboxing.
- Compilation units and packages.

Hands-on work

A series of simple exercises to get to grips with the development environment.

4 Enforcement and responsibilities

- Static imports.
- Keyboard input/output.
- The java.time API and date conversion.

Hands-on work

Create a simple program and use packages.

5 Class definition and instantiation

- Classes and objects.
- Fields, methods, constructors.
- Self-referencing and static fields/methods.
- Methods with a variable number of arguments.
- Methodological aspects: class design.

Hands-on work

Programming of the case study in sub-groups.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

6 Inheritance (part 1)

- The different forms of inheritance: extension and implementation.
- Interfaces and interface implementation.
- Polymorphism and its implementation.

Hands-on work
Design and build a simple class hierarchy.

7 Inheritance (part 2)

- Extension, definition of derived classes, constructors, references.
- Building class hierarchies, factoring code: abstract classes.
- Simultaneous use of implementation and extension.

Hands-on work
Implementing polymorphism and genericity in the case study. Building class hierarchies and interfaces.

8 The exceptions

- Try blocks, exception generation.
- The catch() selection algorithm.
- Controlled and uncontrolled exceptions.
- Use of the finally block.

Hands-on work
Introduction of exceptions in the case study.

9 Collections and genericity

- Notion of genericity and the benefits of genericity.
- The collection interface and list types.
- Maps.

Hands-on work
Use a generic class and implement lists and maps.

10 Functional programming

- Notion of functional interface.
- API java.util.fonction, the four categories of functional interfaces.
- Collections, forEach and removeIf methods.
- Syntax and use of lambda expressions.

Hands-on work
Using lambda expressions with a functional interface. Application in lists and collections.

11 Streams

- Relationship with functional programming.
- Essential operators: filter, map, reduce.
- Notion of terminal and intermediate operations.
- Algorithm simplification.

Hands-on work
Application of streams to process a collection.

12 Introduction to JDBC

- Principle and benefits of Java Database Connectivity (JDBC).
- JDBC architecture and drivers.
- Environment configuration.

Hands-on work
JDBC driver installation and configuration. First database connection.

13 Basic connections and queries

- Notion of Connection, Driver.
- Create and close connections.
- Resource management.

Hands-on work
Establish secure connections.

14 Statement and ResultSet

- Notion of Statement and ResultSet.
- Execute SELECT, INSERT, UPDATE and DELETE queries.
- Tracking and processing results.

Hands-on work
Implement basic requests with JDBC.

15 PreparedStatement and best practices

- Statement and PreparedStatement, important differences.
- Prevent SQL injections.
- Performance optimization.

Hands-on work
Statement to PreparedStatement conversion. Implement parameterized queries.

16 Transaction management

- Notion of transactions and best practices.
- Commit and rollback.
- Transaction error management.

Hands-on work
Implementation of complex transactions.

17 What's new in Java (recent versions)

- Java evolution since Java 8.
- New APIs and features.
- Records, pattern matching, switch expressions.
- Text Blocks and enhanced APIs.
- Virtual Threads and other improvements.

Hands-on work
Refactoring d'ancien code avec les nouvelles fonctionnalités. Exercices pratiques sur les records et pattern matching. Expérimentation avec les text blocks.

18 Debugging and monitoring tools

- Use IDE debugging tools.
- Advanced debugging techniques.
- Logging and monitoring.
- Good development practices.

Hands-on work
Practical debugging session on the case study.

19 Optimization and performance (1/2)

- Profiling and performance analysis.
- Memory management and Garbage Collector.

Hands-on work
Case study performance analysis.

20 Optimization and performance (2/2)

- Current optimizations (JVM arguments).

Hands-on work
Implementation of optimizations.

21 Testing and code quality

- Modern test strategies (TDD, BDD).
- JUnit 5 and its new features.
- Parameter tests, integration tests.
- Advanced mocking with Mockito.
- Code coverage and static analysis.
- Performance and load testing.

Hands-on work
Mise en place d'une suite de tests complète pour l'étude de cas. Configuration d'outils d'analyse de code.

22 Integrative project (part 1)

- Needs analysis and design.
- Application architecture.
- Setting up the project structure.

Hands-on work
Design of the final application.

23 Integrative project (part 2)

- Implementation of core functionalities.
- Database integration.
- Exception handling.

Hands-on work
Integration of concepts into the final application.

24 Integrative project (part 3)

- Finalizing development.
- Testing and debugging.
- Documentation.

Hands-on work
Application testing and validation.

25 Junit, mastering unit testing in Java - Post-training digital learning

content

- Introduction to JUnit and configuration.
- Understand and use Junit annotations.
- Test structuring and best practices.
- Advanced testing with Junit.
- Learn more about JUnit.

Digital activities
In this online training course, you'll discover how to master JUnit 5 to write, organize and execute efficient unit tests in Java. After an introduction to fundamental concepts, you'll learn how to structure tests, use key annotations, handle exceptions, create parameterized tests and test suites, and integrate JUnit with Maven or Gradle. Each concept will be illustrated by practical examples, such as testing a calculator or an inventory.

Dates and locations

REMOTE CLASS
2026 : 23 Mar., 15 June, 28 Sep., 7 Dec.

LILLE
2026 : 23 Mar., 15 June, 28 Sep., 7 Dec.

PARIS LA DÉFENSE
2026 : 16 Mar., 8 June, 21 Sep., 30 Nov.