

Opleiding : Campus Atlas - JAVA, de basis van programmeren

Praktijkcursus - 5d - 35u00 - Ref. LJB

Prijs : 2610 € V.B.

NEW

Na afloop van de cursus zijn de deelnemers in staat om de Java-taal en bijbehorende technologieën te gebruiken om een applicatie te maken. Dit trainingsprogramma is gericht op medewerkers in de professionele branches die onder de OPCO Atlas vallen.

Pedagogische doelstellingen

Aan het einde van de training is de deelnemer in staat om:

- ✓ De basisconcepten van de Java-taal begrijpen en de syntax beheersen
- ✓ Bibliotheken en API's gebruiken
- ✓ De concepten van objectgeoriënteerd programmeren in Java begrijpen
- ✓ Een toepassing maken in Java

Doelgroep

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

Voorafgaande vereisten

Bekendheid met de principes van objectgeoriënteerd programmeren en ervaring met het gebruik van een programmeertaal om toepassingen te ontwikkelen.

Praktische modaliteiten

Quizzen, spellen in tweetallen of groepen, discussies, rollenspellen, intensief praktisch werk, geïntegreerde ontwikkelomgeving.

Leer methodes

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.

DEELNEMERS

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

VOORAFGAANDE VEREISTEN

Bekendheid met de principes van objectgeoriënteerd programmeren en ervaring met het gebruik van een programmeertaal om toepassingen te ontwikkelen.

VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakken als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ...
De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

Opleidingsprogramma

1 Algoritmen - Denken voordat je ontwerpt - Inhoud voor pre-training digitaal leren

- Inleiding tot algoritmen.
- Basisinstructies in pseudocode.

Digitale activiteiten

In deze online cursus leer je redeneren voordat je een programma ontwerpt door de basis van algoritmiek te ontdekken. In het bijzonder bestudeer je de fundamentele instructies in pseudocode.

2 Objecttechnieken

- De algemene principes van objectmodellering en programmeren.
- Abstractie en inkapseling: interfaces.
- De verschillende vormen van overerving, polymorfisme.
- Inleiding tot UML-modellering: het statische model, het dynamische model, het samenwerkingsmodel, scenario's.

Praktisch werk

UML-specificatie van een casestudy die een van de uitgangspunten zal zijn voor de volgende oefeningen.

3 Basis taalconstructies

- Variabelen: declaratie en typering.
- Velden en methoden definiëren.
- Bedieningsuitdrukkingen en instructies.
- Opgesomde tabellen en types, autoboxing.
- Compilatie-eenheden en pakketten.

Praktisch werk

Een reeks eenvoudige oefeningen om de ontwikkelomgeving onder de knie te krijgen.

4 Handhaving en verantwoordelijkheden

- Statische invoer.
- Toetsenbordinput/uitvoer.
- De java.time API en datumconversie.

Praktisch werk

Een eenvoudig programma maken en pakketten gebruiken.

5 Klassen definiëren en instantiëren

- Klassen en objecten.
- Velden, methoden en constructeurs.
- Zelfverwijzing en statische velden/methoden.
- Methoden met een variabel aantal argumenten.
- Methodologische aspecten: klasontwerp.

Praktisch werk

De casestudy plannen in subgroepen.

PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

TOEGANGSMODALITEITEN EN TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

TOEGANKELIJKHEID VOOR MINDERVALIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

6 De erfenis (deel 1)

- De verschillende vormen van overerving: uitbreiding en implementatie.
- Interfaces en interface-implementatie.
- Polymorfisme en de implementatie ervan.

Praktisch werk

Een eenvoudige klassenhiërarchie ontwerpen en bouwen.

7 De erfenis (deel 2)

- Uitbreiding, definitie van afgeleide klassen, constructeurs, verwijzingen.
- Klassenhiërarchieën bouwen, code factureren: abstracte klassen.
- Gelijktijdig gebruik van implementatie en uitbreiding.

Praktisch werk

Het implementeren van polymorfisme en genericiteit in de casestudy.
Klassenhiërarchieën en interfaces bouwen.

8 Uitzonderingen

- Blokken proberen, uitzonderingen genereren.
- Het selectiealgoritme `catch()`.
- Gecontroleerde en ongecontroleerde uitzonderingen.
- Gebruik van het laatste blok.

Praktisch werk

Introductie van uitzonderingen in de casestudy.

9 Collecties en algemeenheid

- Het concept van genericiteit en de voordelen van genericiteit.
- De verzamelinginterface en lijsttypen.
- Kaarten.

Praktisch werk

Gebruik van een generieke klasse en implementatie van lijsten en kaarten.

10 Functioneel programmeren

- Notie van functionele interface.
- `java.util.function` API, de vier categorieën van functionele interfaces.
- Verzamelingen, `forEach` en `removeIf` methoden.
- Syntaxis en gebruik van lambda-uitdrukkingen.

Praktisch werk

Lambda-expressies gebruiken met een functionele interface. Toepassing in lijsten en verzamelingen.

11 Stromen

- Relatie met functioneel programmeren.
- De essentiële operatoren: `filter`, `map`, `reduce`.
- Begrip van terminale en intermediaire operaties.
- Vereenvoudiging van het algoritme.

Praktisch werk

Toepassing van streams om een verzameling te verwerken.

12 Inleiding tot JDBC

- Principe en voordelen van Java Database Connectivity (JDBC).
- JDBC-architectuur en stuurprogramma's.
- Configuratie van de omgeving.

Praktisch werk

Een JDBC-stuurprogramma installeren en configureren. Voor het eerst verbinding maken met een database.

13 Basisverbindingen en query's

- Notie van Verbinding, Bestuurder.
- Verbindingen maken en sluiten.
- Beheer van hulpbronnen.

Praktisch werk

Veilige verbindingen tot stand brengen.

14 Statement en ResultSet

- Begrip van Statement en ResultSet.
- SELECT-, INSERT-, UPDATE- en DELETE-query's uitvoeren.
- Resultaten bijhouden en verwerken.

Praktisch werk

Basisverzoeken implementeren met JDBC.

15 VoorbereidStatement en best practice

- Statement en PreparedStatement, belangrijke verschillen.
- Preventie van SQL-injecties.
- Prestatieoptimalisatie.

Praktisch werk

Statements converteren naar PreparedStatement. Implementatie van verzoeken met parameters.

16 Beheer van transacties

- Begrip van transacties en best practices.
- Vastleggen en terugdraaien.
- Beheer van transactiefouten.

Praktisch werk

Implementatie van complexe transacties.

17 Nieuw in Java (recente versies)

- Ontwikkelingen in Java sinds Java 8.
- Nieuwe API's en functionaliteiten.
- Records, patroonherkenning, schakeluitdrukkingen.
- Tekstblokken en verbeterde API.
- Virtuele threads en andere verbeteringen.

Praktisch werk

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

18 Tools voor debuggen en bewaking

- Gebruik van debug-gereedschappen in IDE.
- Geavanceerde debugging-technieken.
- Registratie en bewaking.
- Goede ontwikkelingspraktijken.

Praktisch werk

Praktische debugsessie over de casestudy.

19 Optimalisatie en prestaties (1/2)

- Profileren en prestatieanalyse.
- Geheugenbeheer en Garbage Collector.

Praktisch werk

Prestatieanalyse van de casestudy.

20 Optimalisatie en prestaties (2/2)

- Huidige optimalisaties (JVM-argumenten).

Praktisch werk

Implementatie van optimalisaties.

21 Testen en codekwaliteit

- Moderne teststrategieën (TDD, BDD).
- JUnit 5 en de nieuwe functies.
- Geparameteriseerde tests, integratietests.
- Mocking voor gevorderden met Mockito.
- Codedekking en statische analyse.
- Prestatie- en belastingstesten.

Praktisch werk

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

22 Integratietraject (deel 1)

- Behoeftanalyse en ontwerp.
- Toepassingsarchitectuur.
- De projectstructuur opzetten.

Praktisch werk

Ontwerp van de uiteindelijke toepassing.

23 Integratief project (deel 2)

- Implementatie van kernfunctionaliteiten.
- Integratie van databases.
- Afhandeling van uitzonderingen.

Praktisch werk

Integratie van concepten in de uiteindelijke toepassing.

24 Integratief project (deel 3)

- Afronding van de ontwikkeling.
- Testen en debuggen.
- Documentatie.

Praktisch werk

Testen en valideren van de applicatie.

25 Junit, unit testing in Java onder de knie krijgen - Post-training digitale

leerinhoud

- Inleiding tot JUnit en configuratie.
- Junit annotaties begrijpen en gebruiken.
- Structureren van tests en best practices.
- Geavanceerd testen met Junit.
- Verdere beheersing van JUnit.

Digitale activiteiten

In deze online training ontdek je hoe je JUnit 5 onder de knie krijgt om effectieve unit tests in Java te schrijven, te organiseren en uit te voeren. Na een inleiding tot de fundamentele concepten leer je hoe je tests structureert, belangrijke annotaties gebruikt, uitzonderingen beheert, geparametriseerde tests en testsuites maakt en JUnit integreert met Maven of Gradle. Elk concept wordt geïllustreerd met praktische voorbeelden, zoals het testen van een rekenmachine of een inventaris.

Data en plaats

KLAS OP AFSTAND

2026 : 23 maa., 15 juni, 28 sep., 7 dec.

PARIS LA DÉFENSE

2026 : 8 juni, 21 sep., 30 nov.

LILLE

2026 : 15 juni, 28 sep., 7 dec.