

Durée : 5 jours soit 35 heures

Référence : IF-JAVAC



Public visé :

- Développeurs d'applications JAVA



Pré-requis :

Pour suivre ce stage, il est nécessaire de maîtriser la programmation orientée objet en Java ou d'avoir suivi la formation référencée "JAVA" ou équivalente.



Objectifs pédagogiques :

- Approfondir certains aspects avancés du langage
- Comprendre et utiliser les expressions lambda
- Comprendre les principes de base de la nouvelle API
- Maîtriser la communication par socket
- Comprendre la communication distribuée : RMI
- Maîtriser l'accès aux bases de données avec JDBC
- Comprendre les principes de SWING et utiliser les principaux conteneurs



Compétences acquises à l'issue de la formation :

- Approfondir certains aspects avancés du langage
- Comprendre et utiliser les expressions lambda
- Comprendre les principes de base de la nouvelle API
- Maîtriser la communication par socket
- Comprendre la communication distribuée : RMI
- Maîtriser l'accès aux bases de données avec JDBC
- Comprendre les principes de SWING et utiliser les principaux conteneurs



Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure. La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz. L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration. En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.



Moyens et supports pédagogiques :

Cadre présentiel

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique



Cadre téléprésentiel

Session dispensée via notre solution iClassRoom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique



Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert...).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquisée ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin

Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.



Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :

Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr

Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.

Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- [AGEFIPH](#)
- [FIPHFP](#)
- MDPH de votre département

Description / Contenu

Module 1 : Quelques aspects avancés du langage

- Les Inner Classes
- Les classes anonymes
- Les nouvelles boucles for

IFORM : Agence de Balma : 16 avenue Charles de Gaulle, bâtiment 13, 31130 Balma

Siège social : Mbe 308 46 rue André Vasseur 31200 TOULOUSE

Tél : 05 61 34 39 80 • SIRET : 431 421 742 00096 • APE : 6202A

- L'auto-boxing, auto-unboxing
- Les types énumérés
- La généricité
- Les annotations

Module 2 : Les expressions lambda

- Notion de classe anonyme imbriquée, améliorations apportées par le concept d'expression lambda.
- Les aspects syntaxiques (déclaration, implémentation, passage de paramètres, portée des variables...).
- Le concept de "foncteur" à travers les interfaces "fonctionnelles"
- Utiliser les lambda-expressions pour manipuler les collections.

Module 3 : Nouvelle API Date et Time

- Les difficultés engendrées par les API de Date
- Les principes de base de la nouvelle API.
- La gestion du temps machine avec java.time.Instant et java.time.Duration
- Le temps humain avec java.time.LocalDate, java.time.LocalTime, java.time.LocalDateTime

Module 4 : La communication par socket

- Rappels sur les principaux concepts réseaux
- La communication en mode connecté
- Le modèle client/serveur
- Utilisation de la sérialisation

Module 5 : La communication distribuée : RMI

- Les principes généraux
- Les classes de base
- Le service de nommage
- Le processus de développement du client et du serveur

Module 6 : Accès aux bases de données avec JDBC

- Utilisation de l'API JDBC, des pilotes JDBC
- PreparedStatement et CallableStatement
- Requêtes
- Les transactions
- Utilisation des RowSet
- Le pattern DAO

Module 7 : Introduction à SWING

- Présentation et origine de Swing
- Evolutions des toolkits graphiques Java
- Les principes de base

Module 8 : Principaux conteneurs SWING

- JFrame et JDialog : fenêtres et boîtes de dialogue
- JPanel : conteneur générique
- JScrollPane : conteneur scrollable

Module 9 : Principaux layouts SWING

- Notions de Layout
- Layouts standard

Module 10 : Principaux composants SWING

- JLabel : intégration texte simple ou HTML
- Composants texte : JTextField, JTextArea
- Boutons : JButton, JComboBox

Module 11 : Internationalisation d'une application

- Définir la locale
- Gérer des messages dans des ResourceBundle
- Formater des dates et des nombres

Module 12 : Le multi-tâche

- Lancer un thread en Java
- Problématiques de la programmation multitâche (synchronisation...)
- Créer des variables atomiques
- Utiliser des Read-Write Locks
- Collections Thread-safe
- Concurrent Synchronizers (Semaphore, Phaser...)
- Executors et ThreadPools