Organisme de formation professionnelle continue
N° agrément : 91 30 03332 30
France : +33 (0)188 24 70 33 / 34
Site : www.oo2.fr
E-mail : contact@oo2.fr

Maitriser la programmation Java SE avancée

Date et durée

Code formation : JAVASE2-CF-FR
Durée : 6 jours
Nombre d'heures : 42 heures

Description

La programmation Java est l'épine dorsale pour le développement de logiciels basés sur de nombreuses technologies. Il est **utilisé dans un large éventail d'applications**, allant des systèmes d'exploitation aux applications web, jusqu'aux applications mobiles. En effet, le langage de développement principal d'Android est Java.
Destinée aux développeurs, architectes, intégrateurs et chefs de projet, cette formation Java SE s'adresse aux professionnels ayant une solide connaissance du langage Java. Elle vous permet de **maîtriser les concepts avancés de la programmation Java** et ainsi concevoir, déployer et maintenir des applications complètes et maintenables.

Objectifs

Au cours de cette **formation avancée Java SE**, vous atteindrez les objectifs pédagogiques suivants :

- maîtriser les concepts et les fonctionnalités avancées de Java SE ;
- acquérir des connaissances approfondies sur certains aspects de Java et sur ses évolutions ;
- développer des applications dans des environnements multithreads et concurrents ;
- implémenter différentes communications d'objets ;
- maîtriser les API pour la communication synchrone et asynchrone ;
- gérer la persistance des données avec la Java Persistence API (JPA) ;
- internationaliser vos applications ;
- réussir le projet fil rouge.

Points forts

Une formation de haut niveau sur Java SE, un formateur en génie logiciel et des travaux pratiques en groupes (projet Fil rouge).

Modalités d'évaluation

Travaux Pratiques

Pré-requis

Suivre la **formation avancée Java SE** nécessite le prérequis suivant :

- connaître les fondamentaux du langage Java ou avoir complété la formation suivante (recommandée) :

Les formations ci-dessous sont recommandées.

Programme

Gestion des exceptions

- La distinction entre les exceptions contrôlées et les exceptions non contrôlées.
- Le traitement des exceptions avec try-catch.
- L'utilisation du bloc finally.
- L'utilisation de l'instruction try-with-resources.
- La propagation des exceptions.
- Le principe de traduction des exceptions.
- La mise en place des exceptions personnalisées.

Allocation de la mémoire

- Les différents modes d'allocation (RAM).
- Les différentes zones de la mémoire.
- L'allocation et la référence.

Codification des classes générique

- Qu'est-ce qu'une classe générique ?
- Qu'est-ce qu'un paramètre de type ?
- Les limitations des paramètres de type.
- Les méthodes génériques .
- L'interface générique.

Utilisation des collections

- Qu'est-ce qu'une collection et la généricité ?
- L'interface collection.
- La liste, l'interface set et le tableau associatif.
- Le tri d'une collection.
- Les 2 classes d'utilitaires.

Utilisation des entrées et des sorties

- Qu'est-ce les flux en Java ?
- Les 2 classifications transversales.
- La différence entre les flux BINAIRE et les flux TEXTE.
- Les flux standards.
- Les principes de base du paquetage java.io.
- L'obtention de la référence d'un flux.
- Les classes de filtres.

Introspection et annotations

- Qu'est-ce qu'une introspection ?
- Le chargement dynamique d'une classe en mémoire.
- L'instanciation dynamique.
- L'accès aux informations d'une classe.
- Qu'est ce qu'une annotation ?
- La syntaxe des annotations.
- Les méta-annotations standard.

- L'exploitation des annotations par introspection.

Utilisation du socket réseau RMI

- Le principe de base du socket ?
- L'implémentation d'un réseau local.
- La réception et l'envoi de chaînes de caractères.
- La réception et l'envoi d'objets.
- Le serveur multithread.
- Les caractéristiques de RMI.
- Les serveurs RMI.
- L'utilisation de JNDI (Java Naming and Directory Interface).

Introduction à la programmation concurrente

- Qu'est-ce que la programmation concurrente ?
- Les processus et les threads.
- Les principaux aspects utiles des threads.
- Les threads en Java.
- Le cycle de vie d'un thread.
- L'accès concurrent à la mémoire.
- L'accès concurrent aux objets.
- La synchronisation temporelle.
- Les sémaphores.

Utilisation des patrons de conception

- Pourquoi utilisé un patron de conception ?
- Le pattern Proxy.
- Le pattern Singleton.
- Le pattern Usine.
- Le pattern Façade.
- Le pattern Décorateur.

Injection de dépendances

- Le couplage fort.
- Le couplage faible.
- L'injection par instanciation statique.
- L'injection par instanciation dynamique.
- L'injection de dépendance et l'inversion de contrôle.

Gestion de la persistance avec hibernate et JPA

- Présentation du mappage d'objets relationnels.
- (ORM hibernate).
- L'architecture de l'API de persistance Java.
- Les entités et les transactions.
- Les associations et l'héritage.