

Formation Java avancé

Approfondir ses connaissances en Java

Référence : JAVA-AV-04

Durée : 4 jour(s)

Objectifs

- Pousser plus avant la maîtrise du langage Java
- Comprendre et maîtriser les architectures des applications concurrentes
- Maîtriser les API de communication entre applications Java

Répartition: 50% Théorie, 50% Pratique

Public: Architecte, Développeur, Chef de projet

Pré-requis: Connaissance de Java

Programme

Contrôler la qualité du code avec les Assertions

- Présentation et cas d'utilisation
- Syntaxe des assertions
- Activer sélectivement les assertions à la compilation, au runtime

Types paramétrés (« Generics »)

- Présentation et avantages
- Notion d'effacement de type
- Types bornés et indéfinis
- Comprendre les API et collections paramétrées
- Développer des classes et méthodes paramétrées
- Limitations

Réflexion

- Analyser une classe
- Accéder aux champs et méthodes
- Créer une nouvelle instance
- Cas particuliers : génériques, tableaux et enums
- Proxies dynamiques

Maîtriser les Threads

- Notion de Thread et cycle de vie
- Lancer un thread : Thread et Runnable
- Arrêter proprement un thread
- Interruption prématurée du traitement
- Threads démons, priorité, groupes de threads
- Variables Threadlocal

Thread Safety

- Notion de classe « thread-safe », immutabilité
- Synchronisation, volatilité et visibilité mémoire
- Variables atomiques et collections synchronisées

Architectures concurrentes

- Synchronisation des threads avec wait, notify et join
- Le framework Executor
- Queues, Latches, Barrières et Sémaphores
- Design patterns concurrents

Sérialisation

- Rappels sur la sérialisation
- Serializable et Externalizable
- Sérialisation et désérialisation personnalisées

NIO

- Comparaison avec java.io
- Comprendre les Buffers
- Manipuler des données volumineuses avec le mapping en mémoire
- Opérations asynchrones avec les Channels et les Selectors




Classloaders

- Définition et hiérarchie des classloaders
- Quand une classe est-elle chargée ?
- Fonctionnement interne : findClass(), defineClass() et loadClass()
- Implémenter un classloader personnalisé

SPI et découverte dynamique de services

- Notions de SPI et de Service
- Découverte dynamique
- Use-case : système de plugins

OFFERT EN INTER-ENTREPRISE

-  Le petit déjeuner
croissants, jus d'orange, café)
-  Le déjeuner
-  Une qualification téléphonique
si nécessaire avec l'un de nos
consultants

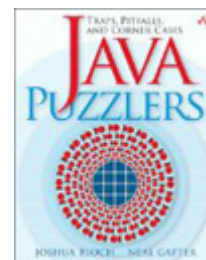
Tel: +33(0)1 45 26 19 15
Fax : +33(0)1 75 43 49 92
Email : training@zenika.com

Auteur du cours



Arnaud Cogoluènes est architecte Java EE et expert Spring. Il dispense pour SpringSource les formations Core Spring et EnterpriseIntegration. Arnaud est co-auteur de la seconde édition de *Spring par la Pratique Spring*, de *Spring Dynamic Modules in Action* et travaille actuellement sur le livre *Spring Batch in Action*.

Livre offert ! (*)



Java Puzzlers

(*) Les livres sont offerts uniquement pour les formations inter-entreprise. Zenika se réserve le droit de changer le livre proposé à tout moment.