

Maîtrise des techniques de programmation Java et optimisation des performances

(JAV201FR, 4 jours)

Description

Ce cours certifié SETC aborde l'environnement de développement Java des points de vue architecture et implémentation. Ce cours de niveau intermédiaire couvre tout à partir de l'architecture interne de la machine virtuelle Java (JVM) jusqu'aux aspects performance associés aux différentes techniques d'attribution de la mémoire. L'héritage, le polymorphisme, les structures de données, les algorithmes et les fils d'exécution sont également couverts. Le cours se termine avec un survol de l'implémentation des techniques de remaniement (aussi appelé refactorisation, ou refactoring) en Java ainsi que l'utilisation des modèles de conception (design patterns) avec Java.

Tarifs

- Tarification: \$3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Architecture Java

Symptômes et causes profondes des problèmes de développement logiciel

Choix d'une méthodologie

Le processus unifié de Rational

Symptômes et causes profondes des problèmes de développement logiciel

Les niveaux d'application : applications à 1, 2 et 3 niveaux

Applications distribuées : Technologie et infrastructure

Applications Web : Technologie et infrastructure.

Paquets et fichiers

Journalisation avec Java

Gestion des exceptions

Implications sur les performances

Considérations relatives à l'optimisation des performances

Architecture JVM

Le moteur d'exécution Java : Architecture interne

Architecture interne de Java

Architecture de la mémoire

Allocation d'objets

Collecte d'ordures

Types de référence

Héritage et polymorphisme

Héritage

Polymorphisme

Fonctions et classes abstraites

Interfaces

Implications en termes de performances

Structures de données et algorithmes

À propos des collections Java

Tableaux de hachage et cartes de hachage

Accès en cache

Travailler avec des chaînes de caractères

Génériques

Threading Java

À propos des threads Java

Conditions de course

Blocages et synchronisation

Pools de threads

Équilibrage de charge

Outils et utilitaires

Aperçu des outils JDK

Outils de surveillance, de gestion et de dépannage

Résumé des outils de diagnostic post-mortem

Résumé des outils pour les processus suspendus

Débogage Eclipse

Détails sur l'optimisation des performances

Optimisation de la compilation dans Rational Developer

Optimisation du serveur d'application

Configuration de la JVM

Outils de gestion utiles

Utiliser JStat

Utiliser JConsole

Utilisation de HProf

L'outil d'analyse du tas de Java (JHAT)

Langage de requête d'objets (OQL)

Utilisation de jdb

Refonte des applications

Le processus de refactoring

Aperçu des odeurs de code

Code dupliqué

Méthode longue

Grande classe

Longue liste de paramètres

Changement divergent

Chirurgie de l'arme à feu

Envie de fonctionnalités

Amas de données

Obsession primitive

Déclarations de commutation

Classe paresseuse

Généralité spéculative

Champ temporaire

Chaînes de messages

Homme du milieu

Classes alternatives avec des interfaces différentes

Classes de données

Requête refusée

Commentaires

Modèles de conception

Classes d'utilité

Méthode des modèles

Usine de classe

Modèle Singleton

Composite

Inversion de contrôle

Sécurité J2EE

Autorisation et authentification

Ajout de groupes

Ajout d'utilisateurs

Création de rôles

Création de politiques

Mise en œuvre de la sécurité déclarative

Sécurité des applications clientes