

Java, programmation multithread

Cours Pratique de 4 jours - 28h

Réf : JQT - Prix 2024 : nous consulter

La programmation multithread en Java prend de plus en plus d'importance grâce à la généralisation des architectures multiprocesseurs : elle simplifie la conception et le développement des applications comportant un parallélisme intrinsèque et offre des solutions efficaces aux problèmes de performance.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Maîtriser les modèles de programmation multithread et les bibliothèques standards correspondantes

Connaître les principales structures de données adaptées à la programmation multithread

Connaître les principaux bugs et avoir un aperçu des solutions

Connaître les outils de test et débogage

Comprendre les liens entre la programmation multithread et les performances

LE PROGRAMME

dernière mise à jour : 05/2019

1) Les modèles, les interfaces et classes de programmation multithread

- Les concepts de base : tâche, ressource d'exécution, activité, service d'exécution, futur.
- Les différentes mises en œuvre des concepts (Runnable, Callable<T>, ExecutorService, Future<T>...).
- Les exceptions non catchées, les groupes de threads.
- Le futur complétable.

Travaux pratiques : Programmation d'une application combinant les différents modèles.

2) Les contraintes sur le comportement correct des activités

- Quelques méthodes formelles de spécifications.
- Une méthode semi-formelle de spécification.
- La mise en œuvre des spécifications.

Travaux pratiques : Utilisation de méthodes formelles de spécifications.

3) La synchronisation et communication des threads

- Les status "synchronized", "wait", "notify" et la programmation de moniteurs.
- Les interfaces et classes de synchronisation : verrous, sémaphores, les barrières cycliques.
- Les queue.

Travaux pratiques : Utilisation des interfaces et classes de synchronisation.

4) L'exécution de tâches en parallèle

- Les ExecutorService.
- Le modèle fork/join (RecursiveTask<T>, RecursiveAction, ForkJoinPool).

Travaux pratiques : Utilisation des services d'exécution et du modèle fork/join.

5) Les structures de données dédiées à la programmation multithread

- Les collections spécialisées.

PARTICIPANTS

Développeurs, architectes logiciels et chefs de projets.

PRÉREQUIS

Connaissances de base en Java et de l'utilisation d'un IDE (Eclipse, IntelliJ...).

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...
Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Le stockage local des données de thread : ThreadLocal<T>.

- Les classes Atomic.

Travaux pratiques : Utilisation des structures de données.

6) Les threads et les performances

- L'impact de la création de threads.

- L'impact de la synchronisation.

- L'impact des caches mémoire.

- Les threads et les IO, les BD et le graphique.

- Les threads et l'ordonnancement.

Travaux pratiques : Optimisation des programmes.

7) Les modèles alternatifs

- Les modèles asynchrones : JReact.

- Les modèles d'acteurs (Akka Actor4J...).

- Les modèles réactifs synchrones.

8) Outils dédiés au développement de programmes concurrents

- JConsole, jstack.

- La librairie JArmus.

- La logique temporelle de Lamport : TLA+.

- Les erreurs courantes : famine (contention), endormissement (dormancy), interblocage (deadlock), terminaison prématurée.

LES DATES

Nous contacter