Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

Threads Virtual, Programmation Structurée et Scoped Values : état du projet Loom en Java 21 et 22

Séminaire animé par José PAUMARD

Java propose un modèle de programmation concurrente intégré à son API depuis ses toutes premières versions. Deux évolutions majeures ont eu lieu depuis l'introduction de la classe Thread et du mot-clé synchronized : l'ajout de l'API java.util.concurrent en Java 5, puis le framework Fork / Join et l'API CompletableFuture en Java 8. Une troisième évolution a eu lieu en Java 21, avec l'ajout des threads virtuels, qui autorisent un modèle alternatif à la programmation réactive. A l'issue de cette journée, vous saurez précisément comment les threads virtuels fonctionnent, dans quel contexte vous pouvez les utiliser, et comment vous pouvez les utiliser pour améliorer vos applications concurrentes et réactives.

Détails

- Code : OX'days 2024 – 8
- Durée : 1 jour (7 heures)

Public
- Architectes
- Developpeurs

Pré-requis
- Avoir des notions de programmation concurrente (thread, race condition, synchronisation), notions de programmation réactive (CompletableFuture ou toute autre API).

Objectifs
- Savoir comment les threads virtuels fonctionnent
- Dans quel contexte les utiliser
- Comment vous les utiliser pour améliorer les applications concurrentes et réactives

Programme

Les threads virtuels

- comment optimiser le nombre de traitements par seconde des applications : programmation réactive et threads virtuels
- patterns de création de threads virtuels
- comment fonctionnent-ils, quel est leur coût en mémoire et quelles sont leurs performances

La programmation concurrente structurée, centrée sur la classe StructuredTaskScope :

- en quoi la programmation concurrente structurée règle-t-elle le problème des threads perdus
- comment fonctionne cet objet
- son utilisation pour lancer des tâches asynchrones
- ses modèles alternatifs : fermeture sur le premier résultat, fermeture sur la première erreur
- comment l'étendre pour exploiter les résultats de ces tâches
- gestion des timeout et de la fermeture de cet objet

Les variables ScopedValue

- pourquoi a-t-on besoin du modèle de variables ThreadLocal
- en quoi ce modèle est-il difficilement adaptable à la programmation réactive
- problèmes du modèle de variables ThreadLocal
- fonctionnement des variables ScopedValue
- utilisation dans des cas réels

A propos de José PAUMARD :

José est Java DevRel – Java Platform Group chez Oracle. Consultant en informatique depuis environ 25 ans. Maître de conférences en mathématiques et informatique à l'université Paris 13, docteur en mathématiques appliquées, il a enseigné les technologies Java au niveau Master et dernière année d'école d'ingénieur depuis environ 20 ans.

Il est fortement investi en qualité des applications logicielles et software craftsmanship depuis 2015 au travers de missions de coaching et de formation auprès de plusieurs sociétés (SGCIB, Renault, Crédit Agricole, Groupama, Accenture). La formation Software Craftsmanship qu'il a créée, commercialisée par la société Oxiane, est suivie par une centaine de développeurs chaque année.

Java Champion et Java Rockstar, il intervient régulièrement dans les conférences Java en France et à l'international : Devoxx (en France, en Belgique, en Angleterre), JavaOne, Oracle Code One, Goto, DevNexus, JFokus, etc.

Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

Modalités

- Type d'action :Acquisition des connaissances
- Moyens de la formation :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- Modalités pédagogiques :Exposés – Cas pratiques – Synthèse
- Validation :Exercices de validation – Attestation de stages