

Programmation avancée en Java

LJ002

Durée: 5 jours

Public:

Développeurs Java, ingénieurs logiciels et architectes d'applications.

Objectifs:

Approfondir la connaissance de Java notamment dans les domaines de la programmation multi-tâches, des tests et du logging.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base du langage Java.

Programme:

- Le multi-threading : Fonctionnement. Ordonnancement et priorité. Exclusion mutuelle. Synchronisation. Thread démon. Communication par flux "pipe".
- L'API de concurrence. : Les exécuteurs de tâches. Les queues. Les maps atomiques. La représentation du temps et de ses unités. Les synchroniseurs. Les traitements asynchrones anticipés. Les variables atomiques. Les verrous "haute performance".
- Les annotations. : Objectif. L'API Reflection. Annotations standards. Les méta-annotations. Fabriquer ses annotations. Annotation Processing Tool (APT)
- Les nouvelles I/O : La gestion des flux standards : l'API Scanner.
- La gestion et la supervision de la JVM : L'API de management JMX
- Les tests : Objectif. Le framework JUnit.
- Traçabilité des applications : Objectif. L'API Java Logging.
- Java Management eXtension : Supervision avec JMX
Principe des MBeans, et exemples de MBeans standards fournis à partir du jdk 1.5

Programmation avancée en Java

- Réseau : Les classes principales d'accès au réseau
Programmation par socket, sérialisation. Transferts d'objets au travers du réseau. Création d'une application client/serveur réseau.
Accès aux ressources partagées, synchronisation, verrous. Utilisation des threads.
- Sécurité : Introduction à la sécurité Java.
Protection du système vis à vis des applications tierces: SecurityManager, ClassLoader.
JCE. Chiffrement, chiffrement asymétrique, hachage
- Intégration : exécution de commandes système depuis Java.
appel de programme en C depuis Java: classes natives.
appel de Java depuis le C.
Instanciation d'une JVM.