Java, programmation avancée

★★★★★

SII-298 **5 Jours (35 Heures)**

Description

Cette formation a un double objectif. D'une part, approfondir certains aspects avancés du langage (les types génériques, les annotations, la programmation réflexive, le chargement des classes). D'autre part, présenter un panorama synthétique des principales librairies concernant les thèmes suivants : la gestion des Threads, les communications via le réseau, l'administration et la supervision des applications. Ce cours insiste par ailleurs sur les techniques architecturales associées.

À qui s'adresse cette formation ?

Pour qui

Développeurs, ingénieurs, chefs de projets proches du développement.

Prérequis

Bonnes connaissances du langage Java. Expérience requise en programmation Java.

Les objectifs de la formation

- Maîtriser les aspects avancés du langage Java
- Mettre en oeuvre la programmation concurrente à base de Threads
- Mettre en oeuvre différentes techniques de communication (Socket, RMI, JMS)
- Comprendre le modèle de sécurité et le chargement des classes Java
- Administrer une application Java via JMX, surveiller la JVM
- Mettre en oeuvre la programmation réflexive

Programme de la formation

Quelques aspects avancés du langage

- Les Inner Classes.
- Les classes anonymes.
- La redéfinition covariante des méthodes (jdk1.
- 5).
- Les nouvelles boucles for (jdk1.
- 5).
- Les Import Static (jdk1.
- 5).
- L'auto-boxing, auto-unboxing (jdk1.
- 5).
- Les varargs (jdk1.
- 5).
- Les types énumérés (jdk1.
- 5).
- Utilisation et définition.
- Les types génériques (jdk1.
- 5).
- L'utilisation et la définition de types génériques simples.
- La généricité et la relation de sous-typage.
- Les types génériques à l'exécution, les types génériques et l'instanciation, les types génériques et les tableaux.
- Les méthodes génériques.
- L'utilisation simultanée des types génériques et non génériques.
- Les annotations (jdk1.
- 5 et jdk1.
- 6).
- Principes.
- Les annotations prédéfinies (
- override,
- deprecated,
- generated.

La programmation concurrente

- Définitions relatives à la programmation concurrente : les Threads.
- La création/destruction des Threads.
- Ordonnancement des Threads.
- La synchronisation des Threads.
- Le verrouillage des méthodes et des instructions (synchronized).
- Les moniteurs.
- Principes méthodologiques pour la conception d'applications multithreadées.
- Les problèmes posés par la programmation multithread.
- Le problème de l'interblocage (caractérisation, évitement, prévention, détection).
- Le problème de la famine.
- La librairie concurrente de JDK1.
- 5.
- Les nouvelles collections.
- Les nouveaux outils de synchronisation : les verrous partagés/exclusifs, les sémaphores, les barrières cycliques.
- Travaux pratiques Construction d'une application concurrente avec synchronisation et résolution d'interblocage.

La communication par socket

- Rappels sur les principaux concepts réseaux.
- La programmation en mode non connecté (par datagram).
- Le modèle Peer to Peer.
- La communication en mode connecté (par stream).
- Le modèle client/serveur.
- Serveur séquentiel vs serveur concurrent.
- Utilisation de la sérialisation.
- La librairie nio.
- Les buffers, channels.
- Les sélecteurs et leur utilisation.
- Travaux pratiques Programmation de l'étude de cas en mode client-serveur (séquentiel et concurrent).

La communication par invocation de méthode distante : RMI

- Les principes généraux.
- Les classes de base.
- Le service de nommage.
- Le processus de développement du client et du serveur.
- Les contraintes de sécurité et de chargement de classes.
- Travaux pratiques Programmation de l'étude de cas à l'aide de RMI.

La communication par messages : JMS

- Les principes généraux de la communication par messages.
- Les concepts de JMS, les interfaces et les classes correspondantes.
- Les différentes formes de messages.
- La communication en mode point à point.
- La communication en mode publish/subscribe.
- Travaux pratiques Programmation de l'étude de cas à l'aide de JMS.

La sécurité et le chargement de classes

- Les aspects avancés et moins connus de la JVM.
- Les mécanismes liés à la JVM pour assurer la sécurité.
- Le modèle de sécurité de Jdk1.
- 2 : les permissions, les politiques, les domaines de sécurité.
- Le gestionnaire de sécurité et le contrôleur d'accès.
- Les mécanismes de chargement de classes.
- Le développement de chargeurs de classes spécifiques.

Présentation de JMX (Java Management Extension)

- La distinction entre l'application à administrer et l'application d'administration.
- Les concepts de JMX : MBean, MBeanServer, Agent, Connector.
- Les API JMX.
- La mise en oeuvre des différentes catégories de Mbeans (standard, dynamiques, model).
- L'utilisation des adaptateurs et des connecteurs.
- Les consoles d'administration.
- Travaux pratiques Mise en place d'une couche d'administration-supervision pour l'étude de cas.

La programmation réflexive

- Objectifs et principes.
- Le chargement de classes.
- L'objet Class.
- Découverte dynamique des informations (propriétés et méthodes) relatives à une classe ou à un objet.
- Instanciation dynamique.
- Invoquer une méthode.
- La réflexivité des annotations en Java 5.