

Stage pratique de 4 jour(s)
Réf : JOT

Participants

» Testez vos connaissances :
www.orsys.fr

Pré-requis

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

Prix 2012 : 2150€ HT

Dates des sessions

Paris

27 mar. 2012, 29 mai 2012
25 sep. 2012, 27 nov. 2012

Java, optimisation développer des applications efficaces et fiables

OBJECTIFS

Le langage Java possède des caractéristiques qui simplifient le développement des applications. Cette facilité qui repose d'une part, sur les propriétés générales du modèle "objet" et d'autre part, sur des spécificités propres au langage, peut se traduire par des performances médiocres et une occupation mémoire importante. Cette formation a un double objectif : présenter un panorama synthétique des spécificités de Java qui jouent un rôle dans l'efficacité des applications, mais aussi présenter une méthodologie d'optimisation des applications ainsi que les concepts et les outils associés.

1) Introduction

2) Méthodologie et outils pour l'optimisation

3) L'optimisation et l'environnement d'exécution

4) L'optimisation et le langage

5) L'optimisation et les bibliothèques standard

6) L'optimisation et quelques Design Patterns

7) L'optimisation et les I/O

8) L'optimisation et le multithreading

9) L'optimisation et la gestion de la mémoire

10) L'optimisation et l'utilisation de bibliothèques C

11) et C++

Intervenant

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

» Testez vos connaissances : www.orsys.fr

1) Introduction

- Les éléments intervenant dans l'optimisation et les actions correspondantes.
- Le choix d'un environnement d'exécution performant.
- Les outils d'aide à l'optimisation.

2) Méthodologie et outils pour l'optimisation

- L'optimisation des performances est souvent présentée comme une activité principalement basée sur l'usage d'astuces. En fait une approche méthodologique rigoureuse et systématique existe qui par ailleurs conduit souvent à une amélioration de la conception.
- Les éléments de la méthodologie.
- Quelques outils de profiling et de visualisation de la mémoire (hprof, jmp, TPTP, profiler de NetBeans), quelques outils de tests et d'aide (findBugs, CheckStyle...).

Travaux pratiques

Utilisation des profilers hprof et jmp. Utilisation de junit. Profiling d'une application simple.

3) L'optimisation et l'environnement d'exécution

- Le choix du compilateur.
- Le choix de la JVM.

4) L'optimisation et le langage

- Une approche objective du langage Java montre que la plupart des problèmes de performances ne sont pas liés au langage lui-même mais à une mauvaise utilisation du langage.
- Identifier les vraies sources d'inefficacité.
- Connaître les bons usages qui les évitent.
- L'optimisation et la gestion des objets, les évaluations, les tests.
- L'optimisation et les nouveautés de Java5.

Travaux pratiques

Démonstrations de l'amélioration des performances de l'application témoin par une meilleure gestion des objets et des évaluations.

5) L'optimisation et les bibliothèques standard

- La manipulation des String.
- Les classes " Collection ".

Travaux pratiques

Démonstrations de l'amélioration des performances de l'application témoin par une meilleure manipulation des String et des Collection.

6) L'optimisation et quelques Design Patterns

- L'obtention de performances impose souvent de s'adapter aux caractéristiques d'un environnement donné. La flexibilité et la configurabilité des applications sont donc parfois un facteur important, dont le traitement reste encore aujourd'hui très ad hoc. Il existe pourtant des techniques connues. On présente d'une part, les solutions architecturales génériques et d'autre part, celles spécifiques à Java.
- L'utilisation des interfaces et des classes abstraites.
- L'utilisation de fabriques.

- L'utilisation d'adaptateurs, décorateurs, proxys.
- Les lanceurs d'application.

Travaux pratiques

Design Patterns pour l'optimisation des applications.

7) L'optimisation et les I/O

- Une programmation médiocre des IO peut avoir un effet désastreux sur les performances. Il est important de connaître les facteurs sur lesquels agir.
- La bufferisation, l'encodage des données.
- Le choix des bonnes classes de Stream.
- L'optimisation de la sérialisation.
- Les classes du package java.nio.
- L'optimisation spécifique aux IO réseaux.

Travaux pratiques

Démonstrations de l'amélioration des performances de l'application témoin par une meilleure gestion des IO.

8) L'optimisation et le multithreading

- Cette formation se focalise sur l'utilisation des threads par l'optimisation des IO (en particulier des IO réseau).
- Le modèle d'activité avant Java5.
- Rappels programmatiques (création, ordonnancement, synchronisation des activités).
- Le modèle d'activité de Java5. La nouvelle API.
- Les threads et l'optimisation.

Travaux pratiques

Démonstrations de l'usage des threads dans l'optimisation des performances de l'application témoin.

9) L'optimisation et la gestion de la mémoire

- Paramétrer le Garbage Collector pour améliorer les performances. Les techniques adaptées aux applications qui manipulent des objets de grande taille ou de très nombreux objets.
- Les principes généraux des GC (principaux algorithmes, collaboration avec l'allocateur).
- L'organisation de la mémoire des principales JVM.
- Le GC de la JVM de Sun. Le GC de la machine d'IBM. Fonctionnement et paramétrage.
- L'impact des finalize sur les performances.
- La saturation mémoire (cause, diagnostic, solutions).
- Les références faibles : SoftReference, WeakReference, PhantomReference.
- Les files de références.
- Les évolutions Java6 : le garbage collector

Travaux pratiques

Démonstration de l'effet des SoftReference. Utilisation d'outils de visualisation de la mémoire, de l'activité du GC;

10) L'optimisation et l'utilisation de bibliothèques C

11) et C++

- L'utilisation de code C ou C++ dans une application Java est une solution possible à des problèmes locaux de performances. On présente rapidement l'interface JNI prévue à cet effet.
- Les principes généraux de JNI : les étapes du développement. L'API JNIEnv. JNI et les performances.