

## L4 - Java industriel

### *Le développement d'applications industrielles en Java(TM)*

Java est une marque déposée de Oracle

#### Objectifs

- Maîtriser les concepts du langage Java
- Sécuriser vos applications Java en gérant les exceptions du langage
- Maîtriser le mécanisme des threads
- Apprendre à mettre en oeuvre des Applets
- Maîtriser l'appel de fonctions C/C++ dans des programmes Java via les interfaces JNI et SWIG
- Utiliser les collections d'objets en Java
- Maîtriser les principales classes utilitaires de Java
- Optimiser le code Java

Les exercices pourront se faire soit en local sur le PC soit en utilisant le SDK MicroEJ pour viser un microcontrôleur embarqué

#### Matériel

- Un PC par binôme avec
  - le JDK
  - Eclipse, environnement de développement dédié aux applications Java ou MicroEJ SDK
- Un support de cours imprimé (formation présentielle seulement) et PDF
- Source des exercices

#### Pré-requis

- Connaissance d'un langage de programmation type C ou C++ (niveau stage L2, L3)

#### Environnement du cours

- Cours théorique
  - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
  - Cours dispensé via le système de visioconférence Teams (si à distance)
  - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
  - Les activités pratiques représentent de 40% à 50% de la durée du cours
  - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
  - Exemples de code, exercices et solutions
  - Pour les formations à distance:
    - ▶ Un PC Linux en ligne par stagiaire pour les activités pratiques, avec tous les logiciels nécessaires préinstallés.
    - ▶ Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
    - ▶ Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
  - Pour les formations en présentiel:
    - ▶ Un PC (Linux ou Windows) pour les activités pratiques avec, si approprié, une carte cible embarquée.
    - ▶ Un PC par binôme de stagiaires s'il y a plus de 6 stagiaires.
  - Pour les formations sur site:
    - ▶ Un manuel d'installation est fourni pour permettre de préinstaller les logiciels nécessaires.
    - ▶ Le formateur vient avec les cartes cible nécessaires (et les ramène à la fin de la formation).
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session (demi-journée en présentiel) une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

## Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

# Course Outline

## Premier Jour

### Introduction

- Historique de Java
- Caractéristiques du langage Java
  - Portabilité
  - sécurité
  - Robustesse
  - Simplicité
  - Multithreading
- Le JDK ( Java Development Kit )
- La machine virtuelle
- Les notions de base de JAVA
  - Les types de données
  - Les opérateurs
  - Les contrôles de flux

### Programmation objet en Java

- Classes
  - Les classes
  - L'encapsulation
  - L'héritage
  - Le polymorphisme
  - Les interfaces

**Exercice :** Ecrire un programme producteur-consommateur (usine) en java

- Classes et interfaces imbriquées
  - Classes internes
  - Classes anonymes
- Changement de type et opérateur instanceof
- Packages
  - définition
  - import
  - ordre de recherche

**Exercice :** Réécriture de l'usine en utilisant des classes anonymes

### Aspects avancés

- Généricité en Java
  - Classes génériques (paramétrées)
  - Méthodes génériques

**Exercice :** Paramétrage de l'usine grâce à des types génériques

- Les exceptions Java
  - Présentation des exceptions et de leur mécanisme
  - Capture et propagation des exceptions
  - Les classes d'exception
  - Les exceptions métier

**Exercice :** Contrôle de l'usine par des exceptions

## Deuxième Jour

### Programmation Multitâches en Java

- Qu'est-ce qu'un thread
- Les Java threading API
- Les techniques de synchronisation entre threads
- L'ordonnancement des threads
- La communication asynchrone entre threads

**Exercice :** Créer deux usines qui travaillent en parallèle, l'une consommant les produits de l'autre

### Les classes utilitaires de Java

- Manipulation des chaînes de caractères
  - La classe String
  - La classe StringBuffer
- Les Entrées/Sorties
  - Le package java.io
  - Lire et écrire sur les entrées/sorties standard
  - Lire et écrire des fichiers texte

**Exercice :** Écrire un programme qui lit un fichier texte et l'imprime un mot par ligne

- Les calculs mathématiques
  - La classe java.lang.Math
- Gestion des dates
  - La classe Calendar
- Internationalisation
  - La classe Locale
- Accès à l'environnement
  - La classe System
  - La classe Runtime

**Exercice :** Modifier ce programme pour trier le résultat (par la commande "sort")

## Troisième Jour

### Gestion des données en Java

- Les Collections d'Objets
  - Les types de collections
  - Les classes collection abstraites
  - Les classes d'implémentation
    - ▶ Vector
    - ▶ HashTable
- L'interface Iterator
- Comparaison d'objets et tri des collections
- Utilisation rationnelle des collections

**Exercice :** Réécriture du programme précédent pour compter le nombre d'occurrences de chaque mot et afficher les 10 plus fréquents

### Les applets

- Caractéristiques d'une applet
- Les API associées aux applets
- Déclaration d'une applet dans un fichier HTML
- Applets et sécurité
  - Signature et certificat
  - Comment générer une applet signée
- Communication entre applets

- Communication entre applets et navigateur

## L'interface JNI

- Interfaçage du code Java avec des méthodes natives C/C++
- Présentation de l'interface JNI (Java Native Interface)
- Appel d'une méthode native
- Convention de nommage de la méthode appelée
- Passage et utilisation des objets Java dans du code C/C++
- SWIG (Simplified Wrapper and Interface Generator) pour interfaçer du code C/C++ avec du code Java.

## Quatrième Jour

### Paquetages, Interfaces et fichiers ".jar"

- Créer un paquetage
- Créer une interface
- Créer un fichier Jar

### La sécurité en Java

- La sécurité de la plate-forme Java2
- Le Chargeur de classes
- Le Domaine de sécurité
- Le Contrôleur d'accès
- Le Gestionnaire de sécurité
- Les Fichiers de règles de sécurité
- Les permissions
- La classe FilePermission
- Le cryptage
  - Les signatures numériques
  - Les certificats

### Optimisation

- Compilation éclair ou JIT
- Compilation statique ou Aot
- Quelques règles pour écrire du code efficace
- Outils de monitoring