

# DÉVELOPPEMENT JAVA AVANCÉ ET ACCÈS AUX DONNÉES

**CODE STAGE : AS550**

## OBJECTIFS

Maîtriser le développement Java avec l'IDE Eclipse

Comprendre comment utiliser les principaux Frameworks Java

Disposer des connaissances nécessaires au développement d'une application Java

Savoir tester, debugger et optimiser ses applications

## DURÉE

4 jours

## PUBLIC

Développeurs

Chargés de développement d'applications informatiques

## PRÉ-REQUIS

Avoir suivi la formation Les fondamentaux de la programmation Java (J-IP2) ou maîtriser la programmation orientée objet Java

Connaissance des concepts de bases de données relationnelles et du langage SQL

## PROGRAMME

PRÉSENTATION DE L'ÉCOSYSTÈME JEE

Le standard JEE et ses composantes (EJB, JSP/Servlet et JSF)

Les frameworks JEE Spring et Hibernate

Comparaison détaillée entre les deux solutions (EJB Vs. Spring Hibernate)

MISE EN PLACE DE LA COUCHE PRÉSENTATION DE L'APPLICATION WEB

Rappel sur le patron de conception MVC

Introduction aux JSP/Servlet

Mise en oeuvre de la partie présentation et contrôle (réutilisée dans la partie deuxième solution voir ci-dessous)

1ÈRE APPROCHE : ARCHITECTURE D'UNE APPLICATION JEE AVEC LES EJB 3.2

## PRÉSENTATION DES EJB 3.2

La spécification EJB 3.2 : architecture et principes du conteneur lourd

La triade des EJB : session, entité et message-driven beans

Packaging et déploiement des EJB 3

Localisation des EJB 3 avec JNDI

Injection de dépendances

Comment créer un client standalone et un client Web

## LES EJB ENTITY ET MAPPING OBJET-RELATIONNEL AVEC JPA

Présentation des EJBs Entity

Introduction à JPA (Java Persistence API)

Mapping objet-relationnel, rôle de la clé primaire, mapping des associations (one-to-many, many-to-one etc.)

héritage

EQL, le langage de requête orienté-objet

## 2ÈME APPROCHE : ARCHITECTURE D'UNE APPLICATION JEE AVEC SPRING ET HIBERNATE

### SPRING

Notion de conteneur léger

Instanciation des objets par le conteneur Spring (constructeur et par setters)

injection de dépendance et inversion de contrôle

Programmation Orientée Aspect avec Spring AOP

Access aux données : présentation du JDBC Template

### HIBERNATE

Hibernate : présentation et architecture

Installation et configuration

Persistence XML des classes et des attributs, associations et héritage

### HIBERNATE, OPTIONS AVANCÉES

Optimisation avec utilisation du cache de second niveau

Traitements en mode BatchVersioning, transactions et accès concurrents

Les requêtes avec HQL, Criteria, By The Example, Detached Criteria

Couplage avec Spring (Hibernate Template) pour une solution ORM plus transparente

## CONCLUSION

Revue des points clés des deux approches

Les plus et moins de chaque approche

