

# Cursus de formation Développeur JAVA



## DESCRIPTIFS

## TABLE DES MATIERES

<b>Algorithmique .....</b>	<b>3</b>
<b>UML – Analyse et Conception.....</b>	<b>4</b>
<b>Langage SQL .....</b>	<b>5</b>
<b>HTML 5 et CSS 3 - Création de pages Web .....</b>	<b>6</b>
<b>JavaScript et Jquery.....</b>	<b>7</b>
<b>Introduction à la programmation objet .....</b>	<b>8</b>
<b>Java - Les fondamentaux et le développement JavaSE .....</b>	<b>9</b>
<b>Usine logicielle Java (SVN, MAVEN...) .....</b>	<b>10</b>
<b>Hibernate(ORM, HQL, Persistance, JPA...).</b>	<b>11</b>
<b>Spring 4 Framework .....</b>	<b>12</b>
<b>JavaEE – introduction au architectures / Développement Web .....</b>	<b>14</b>
<b>Développer avec XML et Java .....</b>	<b>15</b>
<b>JavaEE - Développement Web .....</b>	<b>16</b>
<b>JSF 2 - Java Server Faces .....</b>	<b>17</b>
<b>AngularJS.....</b>	<b>18</b>
<b>TP de Synthèse .....</b>	<b>19</b>

# Algorithmique

**A l'issue de ce stage les participants seront en mesure de :** Disposer des connaissances nécessaires à l'apprentissage d'un langage de développement - Connaître les structures de base de la programmation (boucles, conditions) - Savoir quelles sont les grands paradigmes de programmation (procédural, objet) - Comprendre la notion d'objet et les concepts associés - Identifier les apports de la modélisation UML - Disposer d'un premier point de vue sur les approches Java - Découvrir les variables et le typage des données.

<b>Durée :</b> 10 jour(s) (70h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	<ul style="list-style-type: none"> <li>• Exposés</li> <li>• cas pratiques</li> <li>• synthèse</li> </ul>
<b>Appréciation :</b> Evaluation qualitative de fin de stage	

**Prérequis :** Rigueur logique et bonne connaissance de l'outil informatique.

**Public concerné :** Apprentis développeurs, analystes évoluant vers la programmation.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ INTRODUCTION

Les différentes phases : sources, compilation, binaire  
Interpréteur et compilateur  
Ne pas tout récrire (bibliothèques, éditions de liens et exécutables)

### ■ ALGORITHME

Les "atomes" pour s'exprimer  
Apprendre à formuler un traitement  
Utilisation d'un pseudo langage  
Exemples d'algorithme bien conçu, "mal" conçu, et ...faux !  
Représentation avec organigramme

### ■ UN PREMIER PROGRAMME

Présentation de l'environnement de développement  
Un premier programme simple en Java  
Compilation et exécution

### ■ VARIABLES ET TYPES DE DONNEES

Pourquoi typer les variables ?  
Exemples de types (entier, réel, caractères...)  
Exemples de problèmes liés aux types  
Les opérateurs disponibles (+, /, \*/ % ...)  
Le confort des tableaux et des structures  
Typage statique et dynamique

### ■ LA "GRAMMAIRE" D'UN LANGAGE

Les blocs de programme (début ... fin)  
Le test d'une expression (si ... alors ... sinon ...)  
La boucle (tant que ...)

### ■ STRUCTURER SON PROGRAMME

La nécessité d'utiliser des procédures ou des fonctions  
Différences conceptuelles et syntaxiques entre les deux  
Passer des arguments à une fonction (prototype, code retour)  
Les bibliothèques  
Ne pas réécrire ce qui existe déjà (librairies)  
Écrire ses propres librairies  
Comment les utiliser dans d'autres programmes

### ■ L'IMPORTANCE DE LA DOCUMENTATION

Les bonnes habitudes pour anticiper les erreurs (convention de nommage)  
Les commentaires utiles

### ■ L'APPROCHE OBJET

Les concepts objets  
Les objectifs du monde objet  
Les classes et les objets  
Les attributs  
Les méthodes  
L'encapsulation  
L'instanciation  
L'héritage  
Traduction des concepts objets en langage  
Les packages et les espaces de noms  
Les classes, les méthodes et leur visibilité  
Les attributs et leur visibilité  
L'instanciation, l'appel de méthodes et la référence aux variables

### ■ CONSTRUIRE UNE SOLUTION

Analyse et conception  
Comprendre le besoin du client  
Savoir imaginer une architecture logicielle adaptée  
Imaginer les acteurs de l'application

### ■ APPROCHE D'UML

Spécification  
Dans quel cas utiliser UML  
Diagrammes de cas d'utilisation  
Diagrammes de classe  
Diagrammes de séquence  
Utilisation d'UML dans le monde réel

## UML – Analyse et Conception

**A l'issue de ce stage les participants seront en mesure de :** Concevoir des applications objets avec UML - Comprendre ce qu'est un design pattern - Connaitre les différents diagrammes - Analyser un problème et le représenter avec UML - Formaliser les exigences sous forme de use cases - Détailier les interactions entre objets avec les diagrammes UML - Utiliser les dossiers de conception rédigés en UML.

<b>Durée :</b> 5 jour(s) (35h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	• Exposés
<b>Appréciation :</b> Evaluation qualitative de fin de stage	• cas pratiques • synthèse

**Prérequis :** Avoir une bonne connaissance de la programmation.

**Public concerné :** Développeurs, architectes logiciels et chefs de projets.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ INTRODUCTION

Apports d'une méthode de modélisation

Historique

La normalisation d'UML

Apports d'UML2

### ■ DIFFICULTES DU DEVELOPPEMENT LOGICIEL

Etat des lieux : les difficultés

Approches et solutions

### ■ CONCEPTS OBJETS

Approche procédurale et décomposition fonctionnelle

La transition vers l'approche objet

Les objets

Les classes et leurs relations

Généralisation et hiérarchies de classes

Le polymorphisme

Interfaces

Patrons et classes génériques

Exceptions

### ■ UML ET LE DEVELOPPEMENT DU LOGICIEL

La nécessité de structurer le développement applicatif

Cycles de développement logiciel

UML et le cycle en V

UML dans les développements itératifs

### ■ DIAGRAMMES UML

Types de diagrammes et éléments communs

Notes

Stéréotypes, contraintes et valeurs marquées

Paquetages

Relations

### ■ CAS D'UTILISATION (USE CASES)

Qu'est-ce qu'un cas d'utilisation ?

Acteurs et use cases

Représenter les use cases

Organisation des use cases

### ■ LE MODELE OBJET STATIQUE

Diagrammes de classes

Diagrammes d'objets

Diagrammes de composants

Diagrammes de déploiement

Diagramme de structures composites (UML2)

### ■ LE MODELE DYNAMIQUE

Diagrammes d'interactions

Diagrammes d'activités

Diagrammes d'états transitions

Les diagrammes de vue d'ensemble d'interactions (UML2 )

Les diagrammes de Timing (UML2 )

### ■ SENSIBILISATION AUX DESIGN PATTERNS

Formalisation des Design Patterns

Catégories de patterns (GoF, GRASP)

Exemples : singleton, fabrique ...

## Langage SQL

A l'issue de ce stage les participants seront en mesure de : Rappeler les principaux concepts des SGDBR (Système de Gestion des Bases de Données Relationnelles) et d'algèbre relationnelle utilisés dans le langage SQL - Interroger une base de données avec la clause SQL SELECT - Utiliser les commandes SQL de mise à jour des données - Connaître les commandes SQL de début et fin de transaction BEGIN, COMMIT et ROLLBACK - Appréhender les concepts de gestion des priviléges systèmes et objets avec les commandes SQL GRANT et REVOKE - Créer, modifier et supprimer certaines catégories d'objets (table, index, vues...) avec CREATE, ALTER et DROP.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Evaluation qualitative de fin de stage

**Modalités et moyens pédagogiques :**

- Exposés
- cas pratiques
- synthèse

**Prérequis :** Maîtriser les concepts de base du modèle relationnel.

**Public concerné :** Informaticiens et gestionnaires de données.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ LE RELATIONNEL

Les rappels (règles de CODD, vocabulaire, les formes normales)

### ■ L'ALGEBRE RELATIONNELLE ET LA LOGIQUE DU SQL

Projection, sélection, union, différence, produit cartésien, jointure, intersection et division

### ■ L'INTERROGATION DES DONNEES

Projection d'une ou plusieurs colonnes (avec ou sans éliminer les doublons : ALL ou DISTINCT)

Les conditions de recherche (WHERE) et les opérateurs booléens (NOT, AND, OR)

Suite de valeurs (IN), intervalle (BETWEEN)

Recherche d'une chaîne de caractères (LIKE avec % et \_)

Les opérateurs arithmétiques dans la projection et la sélection

Recherche de valeurs nulles (IS NULL et IS NOT NULL)

Les tables dérivées et le tri des résultats (ORDER BY)

Les regroupements de valeurs (GROUP BY) et leurs conditions (HAVING)

Les traitements de groupes (SUM, AVG, MIN, MAX, COUNT)

La structure case (CASE WHEN THEN ELSE)

### ■ LES JOINTURES

Les jointures (JOIN, ON, USING)

Les auto-jointures, les jointures externes (RIGHT, LEFT, FULL)

Les jointures naturelles (NATURAL JOIN)

Opérateurs algébriques sur plusieurs tables (UNION, MINUS, INTERSECT)

### ■ LES SOUS-INTERROGATIONS

Les sous-interrogations ramenant une ou plusieurs valeurs (<, >, =, ANY, ALL)

Sous interrogation de groupe et la corrélation

Sous-interrogations scalaires

### ■ MISE A JOUR DES DONNEES

Mise à jour des données de la base (INSERT, UPDATE, DELETE)

## HTML 5 et CSS 3 - Création de pages Web

**A l'issue de ce stage les participants seront en mesure de :** Comprendre la structuration d'une page HTML5 - Ajouter des styles CSS aux éléments d'une page - Utiliser les blocs et les tableaux - Créer des formulaires avec WebForms 2 - Tester les nouveautés HTML5 et CSS3.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Exercices de validation - Attestation de stages

### Modalités et moyens pédagogiques :

- Exposés
- Cas pratiques
- Synthèse

**Prérequis :** Aucun.

**Public concerné :** Webmasters, concepteurs Web, développeurs, chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ STRUCTURE D'UNE PAGE

Tags principaux

Simplification avec HTML5

Les éléments et leur placement

### ■ BALISES STRUCTURANTES

Blocs Div et Span

Listes

Tableaux

iFrame==

### ■ HTML5

Balises sémantiques

Audio et vidéo

Canvas et SVG

WebGL

### ■ STYLES CSS

Syntaxe des sélecteurs CSS

Pseudo sélecteurs

Principales propriétés CSS

Règles CSS

Frameworks CSS

### ■ STRUCTURE DES ELEMENTS

Le modèle de boîte

Eléments block et inline

La propriété display

Padding, marges et bordures

### ■ POSITIONNEMENT

Positionnement dans le flux

Positionnement absolu et relatif

Positionnement fixé

Positionnement flottant

### ■ CSS3

Nouveaux sélecteurs

Fonts, couleurs et bordures

Positionnement en colonnes

Animations

Media Queries====

## JavaScript et Jquery

**A l'issue de ce stage les participants seront en mesure de :** Connaître les bases de JavaScript et du DOM - Juger de l'intérêt de jQuery pour la programmation cross-browser - Gérer les évènements et les manipulations dynamiques - Savoir les règles d'or de la programmation avec JavaScript - Utiliser le Framework jQuery pour créer simplement des interfaces graphiques - Assurer la compatibilité des applications Internet riches (RIA) sur tous navigateurs. - Réaliser des appels synchrones (Ajax).

<b>Durée :</b> 5 jour(s) (35h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	<ul style="list-style-type: none"><li>• Exposés</li><li>• cas pratiques</li><li>• synthèse</li></ul>
<b>Appréciation :</b> Evaluation qualitative de fin de stage	

**Prérequis :** Connaissance de XHTML et CSS.

**Public concerné :** Développeurs, architectes, chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ UN LANGAGE POUR LE WEB

Historique de JavaScript  
JavaScript et Ajax  
Librairies JavaScript  
Avenir de JavaScript

### ■ BASES DU LANGAGE

Syntaxe de base  
JSON  
Pièges du typage dynamique  
Programmation objet  
Constructeurs et "this"  
Fonctions et programmation fonctionnelle  
Objet Window ou le contexte global  
Contextes d'exécution

### ■ DOCUMENT OBJECT MODEL

Les objets du navigateur  
Récupérer des éléments  
Ajouter des balises  
Modifier les contenus  
Manipulation des CSS  
Gestion des événements  
Cross browsing  
Amélioration avec jQuery

### ■ AJAX

Architecture des sites Web avec et sans Ajax  
Asynchronisme dans le navigateur  
XMLHttpRequest et HTML5  
Gestion d'erreurs et timeout  
Ajax avec jQuery

### ■ LE FRAMEWORK JQUERY

Pourquoi utiliser un framework JavaScript ?  
Présentation de jQuery

### ■ SELECTEURS CSS ET EVENEMENTS

La fonction \$()  
 Sélecteurs de styles, d'attributs et sélecteurs spécifiques  
Événements simples et composés  
L'objet Event et son utilisation  
Se désabonner à un événement

### ■ LE DOM (DOCUMENT OBJET MODEL)

Définition et représentation DOM  
Manipulation du DOM côté client et côté serveur  
Insertion, déplacement et suppression d'éléments  
Manipulation des tableaux  
Travail avec les formulaires  
Rôle dans Ajax

### ■ AJAX AVEC JQUERY

Le concept  
Quelques exemples d'utilisation  
Chargement de données à la demande  
Transmission de données au serveur  
Ajax et modèle événementiel  
Formats de données disponibles et contraintes de sécurité

### ■ REALISATION D'INTERFACES AVEC JQUERY UI

jQuery et les feuilles de styles CSS  
Construction d'interfaces interactives

- Onglets
- Sliders
- Accordéons
- Drag and drop...

### ■ BILAN

Comparaison avec d'autres frameworks  
JavaScript : forces et faiblesses de jQuer

## Introduction à la programmation objet

**A l'issue de ce stage les participants seront en mesure de :** Maîtriser à la fois la conception d'application orientée objet, ainsi que sa traduction en programmation objet - Identifier dans un projet les entités éligibles à devenir des classes - Traduire les entités et relations entre entités en classes - Passer à la programmation objet en présentant les principaux concepts - Comprendre l'utilité des Frameworks dans une approche objet - Choisir et utiliser les Frameworks.

<b>Durée :</b> 3 jour(s) (21h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	<ul style="list-style-type: none"> <li>• Exposés</li> <li>• cas pratiques</li> <li>• synthèse</li> </ul>
<b>Appréciation :</b> Evaluation qualitative de fin de stage	

**Prérequis :** Avoir les connaissances de base de la programmation.

**Public concerné :** Concepteurs et développeurs amenés à conduire un projet de développement en objet.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ DE L'ANALYSE A L'OBJET

Le langage UML comme charnière entre MOA et MOE  
Les principaux diagrammes dans UML qui conduisent à l'objet  
Identification des Design Patterns dans les diagrammes UML

### ■ POURQUOI L'OBJET ?

L'importance de la réutilisabilité  
La notion de composant  
Différences entre conception et développement fonctionnel et objet  
Différences entre principe procédural et objet  
Y a-t-il un seul paradigme objet ?

### ■ LES BASES ET CONCEPTS DE L'OBJET

Les objets sont partout  
Non dissociation des données et des procédures  
Le moule à objets : la classe  
Les objets comme instances de classe  
Les fonctions de classe : les méthodes  
Les attributs de classe : les propriétés  
Notion de visibilité des attributs d'objets  
Présentation de l'inclusion d'objets  
Présentation de l'héritage d'objets  
Une notion essentielle : l'interface

### ■ CONCEPTION D'UNE ARCHITECTURE INFORMATIQUE A BASE D'OBJETS

Les Design Pattern principaux  
Découpage d'une solution en tiers Data, métier et présentation  
Conception des objets d'accès aux données  
Conception des objets métier

Conception des objets graphiques

Conception d'une architecture MVC

Le liant : les interfaces

### ■ LES OUTILS DE DEVELOPPEMENT OBJET

Passage de la conception au code : les générateurs de code  
Présentation des outils et plugins principaux  
Présentation d'Eclipse  
Présentation de NetBeans  
Un exemple dans le monde de la mobilité : Android Studio  
Manipulation des objets dans ces outils d'intégration de technologie

### ■ UTILISATION DES OBJETS DANS LES LANGAGES ARCHITECTURES PRINCIPAUX

Les objets dans JEE et Java  
Les objets dans .NET, C# et VB.NET  
Les objets dans PHP  
Interopérabilité des objets entre architecture et langages : les Web Services

### ■ APPORT DES FRAMEWORKS OBJETS

Définition des Frameworks  
Les Frameworks, exemple type de réutilisabilité  
Conception d'application en utilisant des Frameworks  
Exploiter les objets d'une application dans un Framework choisi  
Importance de la conception applicative, l'approche Framework  
Les principaux Frameworks  
Les Frameworks objets de persistance en Java, .NET, PHP  
Les Frameworks objets de gestion graphique en Java, .NET, PHP, JavaScript  
Les Frameworks de gestion objet : Spring et CDI (une norme)

## Java - Les fondamentaux et le développement JavaSE

A l'issue de ce stage les participants seront en mesure de : Utiliser correctement le langage Java - Utiliser les classes essentielles de Java SE et les collections - Accéder aux données avec JDBC - Accéder aux flux d'entrée-sortie.

Durée : 15 jour(s) (105h)

Certification : Aucune

Appréciation : Exercices de validation - Attestation de stages

Modalités et moyens pédagogiques :

- Exposés
- Cas pratiques
- Synthèse

**Prérequis :** Avoir une connaissance pratique d'un langage orienté objet ou avoir suivi le cours OBJ-INT "Introduction à la programmation objet".

**Public concerné :** Développeurs, architectes et chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ DECOUVERTE DU JAVA DEVELOPMENT KIT

Fonctionnement interne (JRE, JIT)

Garbage collector

Les APIs du JDK

Différences JavaSE et JavaEE

Tour d'horizon des possibilités

Déploiement des applications Java

#### ■ CONCEPTS DU LANGAGE

Exceptions

Annotations

Classes internes et anonymes

Expressions lambda

Types génériques

AutoBoxing

Introduction à l'API de sérialisation

#### ■ CLASSES DE BASE

Les types incontournables : object, string...

Gestion des nombres et dates

Encodage et internationalisation

Collections et générativité

#### ■ ENTREES-SORTIES

Accès au clavier et à la console

Utilisation des Streams

Gestion des flux de caractères

Lecture et écriture de fichiers

#### ■ OPERATEURS

Opérateurs d'égalité, relationnels et conditionnels

Opérateurs d'assignation, arithmétiques et unaires

Opérateurs binaires et de décalages

#### ■ EXPRESSIONS, INSTRUCTIONS ET BLOCS

L'instruction if-else

L'instruction switch

L'instruction while et do-while

L'instruction for

Instructions de branchement

#### ■ CLASSES ET OBJETS

Déclaration de classes

Déclaration de variables membres

Définitions de méthodes

Utilisation de constructeurs

Passage d'arguments à des méthodes ou des constructeurs

Création d'objets

Utilisation d'objets

Retour de valeurs à partir de méthodes

Utilisation du mot-clé this

Contrôle d'accès aux membres d'une classe

Instances et membres de classes

Classes imbriquées

Types énumération

Annotations

#### ■ INTERFACES ET HERITAGE

Déclaration d'une interface

Implémentation d'une interface

Utilisation d'une interface comme un type

Concepts de l'héritage

Surcharge de méthodes

Polymorphisme

Utilisation du mot-clé super

La super-classe Object

Classes et méthodes finales

## Usine logicielle Java (SVN, MAVEN...)

**A l'issue de ce stage les participants seront en mesure de :** Partager le code avec SVN - Scripter avec Ant - Builder avec Maven - Tester avec JUnit - Intégrer en continu avec Jenkins - Inspecter avec Sonar.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Exercices de validation - Attestation de stages

**Modalités et moyens pédagogiques :**

- Exposés
- Cas pratiques
- Synthèse

**Prérequis :** Avoir des compétences Java est un plus.

**Public concerné :** Développeurs, chefs de projets, responsables qualité.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ CONCEPTS

Usine logicielle et intégration continue

Règles de build

L'usine au quotidien

Lien avec l'agilité et l'eXtrem Programming

#### ■ GESTION DE VERSIONS

Fonctionnement de SVN et GIT

Organisation du référentiel : trunk, tags et branches

Check-in et commit, accéder au référentiel

Merge et gestion des conflits

Règles de travail en commun

Créer des tâches

#### ■ MAVEN

Cycle de construction

Gestion des dépendances

Commandes Maven

Définition du POM

POM parent et modularité

Proxy Maven : Nexus

Les plug-ins Maven

#### ■ TESTS UNITAIRES

TU, TDD et test first

JUnit pour les tests unitaires

Mocks et librairies complémentaires

#### ■ INTEGRATION CONTINUE

Principaux produits

Utilisation de Jenkins

Création d'un job

Intégration des build Maven et tests JUnit

#### ■ INSPECTION DU CODE

Principales métriques

Production et utilisation de rapports

Sonar dashboard

## Hibernate(ORM, HQL, Persistence, JPA...)

**A l'issue de ce stage les participants seront en mesure de :** Développer une application Java EE (Web, EJB) en réalisant la persistance des objets avec le Framework Hibernate - Connaître et assimiler un Framework ORM (Object Relational Mapping) dans une architecture n-tiers - Maîtriser les concepts d'Hibernate - Utiliser les bonnes pratiques de développement Hibernate - Utiliser les annotations et les outils Hibernate.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Evaluation qualitative de fin de stage

**Modalités et moyens pédagogiques :**

- Démonstrations – Cas pratiques – Synthèse et évaluation des acquis

**Prérequis :** Avoir une bonne pratique de Java.

**Public concerné :** Architectes, chefs de projets, concepteurs, développeurs et ingénieurs.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ OBJECTIFS

Définition de la persistance objet

Classes, fichier de mapping, configuration, cycle de vie des objets

Mappage des associations unidirectionnelle / bidirectionnelle et des collections

#### ■ L'ARCHITECTURE D'HIBERNATE

Vue en couche

Cycle de vie des instances

Intégration JNDI, JDBC, JTA

Gestion des sessions

#### ■ CONFIGURATION

APIs permettant la configuration par programmation

Ouverture des sessions

Gestion des connexions JDBC

Dialectes SQL, cache de second niveau et cache de requêtes

Intégration à un serveur d'application Java EE

#### ■ REALISATION DE CLASSES PERSISTANTES

Les Plain Ordinary Java Object (POJO)

Mise en oeuvre de l'héritage

Redéfinition des méthodes equals et hashCode

#### ■ LES BASES DE L'ORM (OBJECT RELATIONAL MAPPING)

Les déclarations de mapping

Plusieurs mappings pour une même classe

Les annotations Java 5

#### ■ ETABLIR LA PERSISTANCE DES COLLECTIONS

Les types de collections persistantes

Les clés étrangères, les éléments et les index d'une collection

Associations \*..\* et 1..\*

#### ■ ETABLIR LA PERSISTANCE DES ASSOCIATIONS

Associations unidirectionnelles

Associations bidirectionnelles

Mappings complexes

#### ■ ETABLIR LA PERSISTANCE DES RELATIONS D'HERITAGE

Différentes stratégies

Utilisation d'un discriminant

Gestion du polymorphisme

Comparaison des méthodes

- Avantages et inconvénients
- Dans quels cas les utiliser ?

#### ■ HIBERNATE QUERY LANGUAGE (HQL) ET GESTION DES OBJETS PERSISTANTS

Stocker et charger un objet

Syntaxe du language HQL, les clauses from, select, where, order by, group by

Jointures, agrégation, requêtes polymorphiques

Les requêtes imbriquées

Externalisation de requêtes nommées

Chargement tardif des propriétés

Gestion du cache de second niveau

Le cache de requêtes

#### ■ MANIPULATION DES ENTITES

Le concept d'Entity Manager

Gestion des Transactions, JTA (java Transaction API)

Les requêtes : requêtes nommées et requêtes dynamiques, JPQL (Java Persistence Query Language)

Définition et exécution d'une requête

Requêtes paramétrées

Syntaxe JPQL

#### ■ TRAVAUX PRATIQUES

Utilisation des fonctionnalités de requêtage de JPA

#### ■ PERFORMANCES

Utilisation de Cascade

Optimisation du chargement et Lazy Loading

## Spring 4 Framework

**A l'issue de ce stage les participants seront en mesure de :** Comprendre le positionnement de Spring par rapport à EJB et JEE - Créer une architecture en couches - Mettre en oeuvre la programmation par Aspect - Paramétrer le conteneur léger de Spring et utiliser les annotations - Accéder à vos données avec Spring JDBC et Spring ORM - Créer des pages web avec Spring MVC - Sécuriser vos applications avec Spring Security.

<b>Durée :</b> 5 jour(s) (35h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	<ul style="list-style-type: none"> <li>• Exposés</li> <li>• cas pratiques</li> <li>• synthèse</li> </ul>
<b>Appréciation :</b> Evaluation qualitative de fin de stage	

**Prérequis :** Bonne connaissance du langage Java et bonne vue d'ensemble de la plate-forme JEE ou avoir suivi le cours JAV-SE "Java - Les fondamentaux et le développement JavaSE".

**Public concerné :** Développeurs, architectes et chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ VUE D'ENSEMBLE

Origines et vision d'architecture de Spring  
L'inversion de contrôle ou injection de dépendances  
La programmation orientée aspects  
SpEL  
Bilan des solutions apportées par Spring  
Evolution de Spring

#### ■ SPRING CORE

Construction des instances  
Injections des instances  
Annotation ou configuration  
Spring profiles  
Nommage des beans et bean factory  
Bean Scopes  
Auto Wiring  
ApplicationContext

#### ■ PROGRAMMATION AOP DANS SPRING

Concepts de la programmation orientée Aspects  
Utilisation d'AOP dans Spring  
Types d'advice (Around, after, throws...)  
Advisors et pointcuts  
MethodMatcherPointcut  
AspectJ Pointcut expressions  
Proxies et ProxyFactoryBean

#### ■ SPRING ET L'ACCES AUX DONNEES

Le pattern DAO  
Les JDBC template  
NamedParameters  
RowMappers  
BatchSqlUpdate  
Spring et les ORM : Hibernate, iBatis)  
Bean validation  
Gestion des transactions  
PlatformTransactionManager

#### ■ SPRING ET LE WEB

Spring MVC pour le Web  
Bean Validation côté Web  
Conversions et affichage  
Intégration d'autres frameworks serveurs  
Spring et Ajax / jQuery  
Support des services RestFul  
RestTemplate et AsyncRestTemplate  
Support des WebSockets avec Spring 4

#### ■ LES AUTRES SERVICES

Spring remoting  
Spring WebFlow  
Scheduling et Quartz  
Spring Batch  
Outils : Spring IDE et Tool Suit

## Classes abstraites

### ■ NOMBRES ET CHAINES DE CARACTERES

Les classes Numbers  
Formatage des nombres  
Caractères  
Conversion entre nombres et chaînes  
Manipulation des caractères dans une chaîne  
La classe StringBuilder

### ■ UTILISATION DES GENERIQUES

Introduction aux types génériques  
Constructeurs et méthodes génériques  
Sous-typage  
Le type Erasure

### ■ PACKAGES

Création d'un package  
Nommage d'un package  
Utilisation des membres de package  
Utilisation des fichiers sources et des classes

### ■ COLLECTIONS

Introduction aux collections  
L'interface Collection  
Les interfaces Set, Queue, List et Map  
Les interfaces SortedSet et SortedMap  
Les implémentations Set, List, Map et Queue  
Implémentation de collections personnalisées

### ■ IO

Les entrées sorties en java  
Le pattern décorateur  
Les Readers, les Writers et les Streams : utiliser les bonnes classes.  
Les fichiers Properties  
La sérialisation et la désérialisation d'objets.

### ■ PROGRAMMATION RESEAU

Rappel sur les concepts de programmations réseaux  
Les modes de communication par stream et par datagram

### ■ LES THREADS

Définition de la programmation multi-thread.  
Création de thread, contrôle d'exécution.  
Communication entre les threads.  
Synchroniser l'accès aux données.

### ■ JDBC

Tour d'horizon des solutions de persistance  
Architecture de JDBC  
- Le DriverManager  
- Les URL JDBC  
- Les types de drivers  
Les grandes interfaces utilisées  
- Connection  
- Statement  
- PreparedStatement  
- CallableStatement  
- ResultSet  
Liens entre JDBC et JPA

## JavaEE – introduction au architectures / Développement Web

**A l'issue de ce stage les participants seront en mesure de :** Comprendre la plateforme JavaEE - Développer des Servlets et JSP - Utiliser JSP EL - Créer des Custom Tags - Mettre en oeuvre le Pattern MVC - Gérer le contexte applicatif avec la session.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Exercices de validation - Attestation de stages

**Modalités et moyens pédagogiques :**

- Exposés
- Cas pratiques
- Synthèse

**Prérequis :** Connaissance pratique en Java ou avoir suivi le cours JAV-SE "Java - Les fondamentaux et le développement JavaSE".  
Avoir des notions de base HTML.

**Public concerné :** Développeurs, architectes, chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ PLATEFORME JAVAEE

Les défis des applications d'entreprise

Les technologies JavaEE

Scénarios d'applications JavaEE

Typologie des architectures

Les serveurs JavaEE

#### ■ SERVLETS

Architecture d'une application Web : descripteur de déploiement, fichier .WAR et packaging

Contexte de l'application et ServletConfig

Utilisation de HttpSession

Requêtes et réponses

Annotations

Modèle de threading

Utilisation du RequestDispatcher

Filtres et Listeners

Sécurisation de l'application Web avec SecurityRealms

#### ■ ACCES AUX DONNEES

Rappel JDBC

Utilisation de JNDI et des DataSources

Pooling de connexion

#### ■ JSP

Syntaxe : scriptlet et expressions

Variables implicites

Directives

Synthaxe XML

Utilisation de Bean

Expression Language

TagLibs et custom Tags

JSTL

#### ■ CONCEPTION D'APPLICATIONS

Session HTTP

Cookies

Gestion du contexte

Pattern Modèle Vue Contrôleur (MVC)

Implémentation MVC simple

Exemple d'un framework MVC

## Développer avec XML et Java

**A l'issue de ce stage les participants seront en mesure de :** Optimiser la lecture des documents XML - Connaître les API Java pour XML - Sérialiser avec JAXB.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Exercices de validation - Attestation de stages

**Modalités et moyens pédagogiques :**

- Exposés
- Cas pratiques
- Synthèse

**Prérequis :** Connaître Java et avoir des notions sur les fondamentaux de XML (syntax...).

**Public concerné :** Concepteurs et développeurs devant intégrer les technologies XML dans leurs applications.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ API JAVA POUR XML

API Standard JAXP

Parsers compatibles

Encodage des documents

Modèles de parsing

DOM4J et JDOM

Fabrication noeuds

Génération d'un fichier XML

Navigation et parcours rapides

Gestion de la validation

Gestion des Namespaces

#### ■ ACTIVATION DE TRANSFORMATIONS XSLT

Processeurs XSL

Manipulation

Cas d'utilisation dans un contexte Web

#### ■ SERIALISATION AVEC JAXB

Principes de la sérialisation XML

Annotations JAXB

Outils de mapping

#### ■ STAX (STREAMING API)

API bidirectionnelle et événementielle

Cursor API

Iterator API

#### ■ SAX (SIMPLE API FOR XML)

Modèle événementiel

Handlers SAX

Mémorisation du contexte

Gestion des namespaces

Gestion de la validation

Gestion des erreurs

#### ■ DOM (DOCUMENT OBJECT MODEL)

Modèle objet

Nodes et Elements

Analyse de l'arbre

## JavaEE - Développement Web

**A l'issue de ce stage les participants seront en mesure de :** Comprendre la plateforme JavaEE - Développer des Servlets et JSP - Utiliser JSP EL - Créer des Custom Tags - Mettre en oeuvre le Pattern MVC - Gérer le contexte applicatif avec la session.

**Durée :** 5 jour(s) (35h)

**Certification :** Aucune

**Appréciation :** Exercices de validation - Attestation de stages

**Modalités et moyens pédagogiques :**

- Exposés
- Cas pratiques
- Synthèse

**Prérequis :** Connaissance pratique en Java ou avoir suivi le cours JAV-SE "Java - Les fondamentaux et le développement JavaSE".  
Avoir des notions de base HTML.

**Public concerné :** Développeurs, architectes, chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ PLATEFORME JAVAEE

Les défis des applications d'entreprise

Les technologies JavaEE

Scénarios d'applications JavaEE

Typologie des architectures

Les serveurs JavaEE

#### ■ SERVLETS

Architecture d'une application Web : descripteur de déploiement, fichier .WAR et packaging

Contexte de l'application et ServletConfig

Utilisation de HttpSession

Requêtes et réponses

Annotations

Modèle de threading

Utilisation du RequestDispatcher

Filtres et Listeners

Sécurisation de l'application Web avec SecurityRealms

#### ■ ACCES AUX DONNEES

Rappel JDBC

Utilisation de JNDI et des DataSources

Pooling de connexion

#### ■ JSP

Syntaxe : scriptlet et expressions

Variables implicites

Directives

Synthaxe XML

Utilisation de Bean

Expression Language

TagLibs et custom Tags

JSTL

#### ■ CONCEPTION D'APPLICATIONS

Session HTTP

Cookies

Gestion du contexte

Pattern Modèle Vue Contrôleur (MVC)

Implémentation MVC simple

Exemple d'un framework MVC

## JSF 2 - Java Server Faces

**A l'issue de ce stage les participants seront en mesure de :** Savoir créer et utiliser des pages JSF avec Facelets - Maîtriser l'architecture d'une application JSF et son modèle MVC - Comprendre le rôle des managed-beans - Détailier le cycle de vie d'une requête - Connaitre les principaux composants standards JSF - Intégrer des composants JSF Ajax - Lier les composants JSF et les autres composants JEE.

<b>Durée :</b> 5 jour(s) (35h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	<ul style="list-style-type: none"><li>• Exposés</li><li>• cas pratiques</li><li>• synthèse</li></ul>
<b>Appréciation :</b> Evaluation qualitative de fin de stage	

**Prérequis :** Connaissance du langage Java et de la programmation Web en Java ou avoir suivi la formation JAV-DW "JavaEE - Développement Web".

**Public concerné :** Développeurs, architectes et chefs de projet technique.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ PRESENTATION ET ARCHITECTURE

Architecture MVC et implémentation JSF

Etat actuel de la norme JSF

Environnement de développement

Tour d'horizon des constituants d'une application JSF

Utilisation des annotations JSF 2.0

Configuration et déploiement dans un conteneur de servlet

Evaluations avec EL

Templating avec facelets

Composition de composants

Les composants JSF de base ("JSF Core Tags")

Internationalisation

### ■ COMPOSANTS ET AFFICHAGE AVANCES

Les principaux Frameworks de composants JSF

La librairie PrimeFaces

L'intégration native d'Ajax avec JSF 2.0

Affichage et traitement partiel

Personnalisation de composants

Création de composants

### ■ CONCEPTION AVEC JSF

Scope et managed beans

Gestion d'état avec JSF

Optimisation de la gestion d'état serveur

Optimisation client avec HTML5

JSF et Web Profile

Intégration avec CDI

### ■ CYCLE DE VIE

La servlet FacesServlet

Cycle de traitement des pages JSF

Les managed-beans

Les Backing beans

Règles de navigation

FacesContext

Validateurs et convertisseurs de données

Validateurs standards et spécifiques

Événements JSF

Listener et PhaseListener

### ■ COMPOSANTS ET AFFICHAGE

Facelets



# AngularJS

**A l'issue de ce stage les participants seront en mesure de :** Connaître les problématiques des Single Page Application (SPA) - Savoir y répondre avec AngularJS - Utiliser et définir des directives - Accéder au serveur depuis Angular - Gérer la navigation entre les vues et l'historique - Lier les composants et les modèles.

<b>Durée :</b> 5 jour(s) (35h)	<b>Modalités et moyens pédagogiques :</b>
<b>Certification :</b> Aucune	<ul style="list-style-type: none"><li>• Exposés</li><li>• Cas pratiques</li><li>• Synthèse</li></ul>
<b>Appréciation :</b> Exercices de validation - Attestation de stages	

**Prérequis :** Avoir une connaissance pratique de JavaScript et jQuery ou avoir suivi le cours JVS-IN "JavaScript".

**Public concerné :** Développeurs, architectes, chefs de projets techniques.

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ■ ARCHITECTURE D'UNE SPA

Rôles du client et du serveur  
Accès aux données par un service REST  
Angular côté client  
JSON au milieu  
Gestion de l'état applicatif  
Synchronisation des données  
Navigation dans une application mono-page  
Gestion des URL

Directives  
Scopes  
Modules  
Providers  
Services

### ■ LES DIRECTIVES EN DETAIL

Directives pré-définies  
Directives Custom  
Scope et cycle de vie  
Accès au DOM, événements et templates  
Transclusion

### ■ ANGULAR UN FRAMEWORK TESTABLE

Tests unitaires avec Angular  
ngMock  
End to End testing  
Karma Test Runner  
Protractor

### ■ ANGULAR PARTOUT ?

Angular comparé aux autres frameworks MVC  
Modularité et applications multi-vues  
Angular UI  
Intégration d'Angular avec d'autres librairies  
Modularité d'Angular  
Les limites d'AngularJS

### ■ PRINCIPES DE BASE D'ANGULAR

Model View Whatever : les types de MVC  
Le MVC à la mode Angular  
Bénéfices du Two Way Data Binding  
Injection de dépendances

### ■ FIGURES IMPOSEES

Gestion des formulaires  
Angular Templates et expressions  
Navigation, hashbang et deeplinking  
Accès serveur simplifié  
Contrôleurs et modèles  
Décoration par les CSS  
Internationalisation  
Routage et gestion de l'historique

### ■ PARTICULARITES D'ANGULAR

Processeur HTML  
Filtres

## TP de Synthèse

**A l'issue de cette mise en situation, les participants seront en mesure de :** Appliquer autour d'un projet pratique et d'une façon exhaustive l'ensemble des points acquis.

**Durée :** 5 jour(s) (35h)

**Modalités et moyens pédagogiques :**

**Certification :** Aucune

- Cas pratiques
- Synthèse

**Appréciation :** Exercices de validation - Attestation de stages

**Prérequis :** Connaissances de UML2 , de l'objet, du triptyque HTML5/CSS3/JavaScript, de Java, de JPA2.1/Hibernate5, de Java/Web 3.1, de JSF2, de Spring4, de Maven3

**Public concerné :** Développeur full stack java

Cette formation :

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

### PROGRAMME

#### ■ TP DE SYNTHESE DEVELOPPEUR JAVA

Travaux de synthèse des TP réalisés lors du cursus de formation en fil rouge

Test et recette de l'application

Préparation soutenance de présentation du projet

...