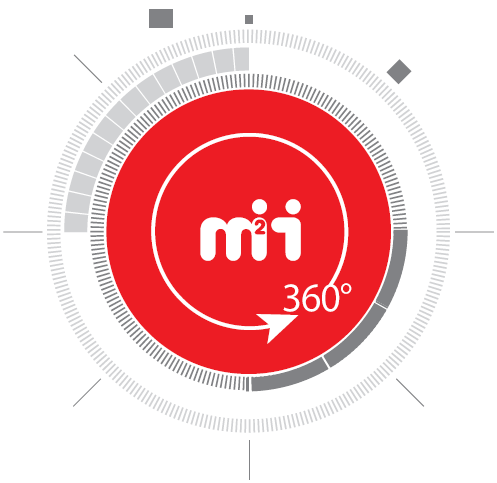
Logo m2i gris rouge2011.eps

**SOPRASTERIA ACADEMY**

**CATALOGUE**

**PARCOURS JAVA – proposition n°4**



**DESCRIPTIFS**

TABLE DES MATIERES

[Algorithmique et base de la programmation à l’aide de Java 3](#_Toc456958767)

[UML – Analyse et Conception 4](#_Toc456958768)

[Langage SQL 5](#_Toc456958769)

[HTML 5 et CSS 3 - Création de pages Web 6](#_Toc456958770)

[JavaScript et Jquery 8](#_Toc456958771)

[Java - Les fondamentaux et le développement JavaSE 10](#_Toc456958772)

[Hibernate(ORM, HQL, Persistance, JPA…) 11](#_Toc456958773)

[Spring 4 Framework 13](#_Toc456958774)

[JavaEE - Développement Web 15](#_Toc456958775)

[JSF 2 - Java Server Faces 16](#_Toc456958776)

[AngularJS 18](#_Toc456958777)

[Usine logicielle Java (SVN, MAVEN…) 20](#_Toc456958778)

[TP-JW TP de Synthèse 21](#_Toc456958779)

Algorithmique et base de la programmation à l’aide de Java

**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.

|  |  |
| --- | --- |
| **Référence :** PRG-IN  **Durée :** 4 jour(s) (28h)  **Certification :** Aucune  **Appréciation** : Evaluation qualitative de fin de stage | **Modalités et moyens pédagogiques** :   * Exposés * cas pratiques * synthèse |

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 réécrire (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 - Connaître les différents diagrammes - Analyser un problème et le représenter avec UML - Formaliser les exigences sous forme de use cases - Détailler les interactions entre objets avec les diagrammes UML - Utiliser les dossiers de conception rédigés en UML.

|  |  |
| --- | --- |
| **Référence :** UML-CO  **Durée :** 2 jour(s) (14h)  **Certification :** Aucune  **Appréciation** : Evaluation qualitative de fin de stage | **Modalités et moyens pédagogiques** :   * Exposés * 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 privilèges 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.

|  |  |
| --- | --- |
| **Référence :** SQL-FND  **Durée :** 3 jour(s) (21h)  **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.

|  |  |
| --- | --- |
| **Référence :** HTM-FND  **Durée :** 2 jour(s) (14h)  **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

iFrames

* **WEBFORMS2**

Champs de saisie

Listes déroulantes

Boutons radio

Champs date

PlaceHolder

Slider

Validation automatique

* **HTML5**

Balises sémantiques

Audio et vidéo

Canvas et SVG

WebGL

* **STYLES CSS**

Syntaxe des sélécteurs 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).

|  |  |
| --- | --- |
| **Référence :** JVS-JQ  **Durée :** 3 jour(s) (21h)  **Certification :** Aucune  **Appréciation** : Evaluation qualitative de fin de stage | **Modalités et moyens pédagogiques** :   * Exposés * cas pratiques * synthèse |

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

Evénements 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 jQuery

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.

|  |  |
| --- | --- |
| Référence : JAV-SE  Durée : 5 jour(s) (35h)  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

* **JAVA ORIENTE OBJET**

Classes et héritage

Packages et visibilité

Encapsulation, getters et setters

Interfaces et polymorphisme

Enumérations

* **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 généricité

* **ENTREES-SORTIES**

Accès au clavier et à la console

Utilisation des Streams

Gestion des flux de caractères

Lecture et écriture de fichiers

* **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

Hibernate(ORM, HQL, Persistance, 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.

|  |  |
| --- | --- |
| **Référence :** JAV-HIB  **Durée :** 3 jour(s) (21h)  **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 / bidirectionelle 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

Fichier de configuration XML

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

Les types utilisés par Hibernate

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

Les bonnes pratiques

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 d'une requête

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

Le verrouillage Optimiste

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.

|  |  |
| --- | --- |
| **Référence :** SPRG-FRW  **Durée :** 5 jour(s) (35h)  **Certification :** Aucune  **Appréciation** : Evaluation qualitative de fin de stage | **Modalités et moyens pédagogiques** :   * Exposés * cas pratiques * synthèse |

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

Application Context

* **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 Suite

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.

|  |  |
| --- | --- |
| **Référence :** JAV-DW  **Durée :** 3 jour(s) (21h)  **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 - Détailler 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.

|  |  |
| --- | --- |
| **Référence :** JAV-JSF  **Durée :** 3 jour(s) (21h)  **Certification :** Aucune  **Appréciation** : Evaluation qualitative de fin de stage | **Modalités et moyens pédagogiques** :   * Exposés * cas pratiques * synthèse |

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

* **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

Evénements JSF

Listener et PhaseListener

* **COMPOSANTS ET AFFICHAGE**

Facelets

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

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.

|  |  |
| --- | --- |
| **Référence :** JVS-ANG  **Durée :** 4 jour(s) (28h)  **Certification :** Aucune  **Appréciation** : Exercices de validation - Attestation de stages | **Modalités et moyens pédagogiques** :   * Exposés * Cas pratiques * Synthèse |

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.
* **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

* **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

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

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.

|  |  |
| --- | --- |
| **Référence :** JAV-CI  **Durée :** 2 jour(s) (14h)  **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

* **ANT**

Automatisation des tâches

Tâches de base

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

TP-JW 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 lors d’action globale de formation Sopra. Ce projet propose une simulation simple du système solaire, partant de la conception jusqu’à la mise en œuvre selon diverses technologies client et serveur.

|  |  |
| --- | --- |
| **Référence :** TP-JW  **Durée :** 3 jour(s) (21h)  **Certification :** Aucune  **Appréciation** : Exercices de validation - Attestation de stages | Modalités et moyens pédagogiques :   * Cas pratiques * Synthèse |

Prérequis : Connaissances de UML2 , de l’objet, du triptyque HTML5/CSS3/JavaScript 2015, 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

* **Algorithmie**

- Installation de Eclipse Neon.

- Utilisation de types de base pour représenter des notions : distance, nom, taille, diamètre d’uns sphère.

- Création d’un algorithme répétitif de rotation d’une sphère

- Ajout d’alternative dans la répétitive, notion de tests et bouléens, permettre à la sphère de s’agrandir selon sa distance.

- Utilisation de variables locales et globales.

- Création de fonctions statiques, notions d’arguments.  
- Création d’un algorithme de tri des sphères selon leur taille.

- Utilisation de librairies complémentaires afin de colorer les sphères.

* **UML2**

- Installation d’un produit de modélisation (Enterprise Architect par exemple)  
- Etude de cas d’utilisations simples dans le domaine de l’astronomie : « Un visiteur démarre une simulation de planètes»

- Etude de divers scénarii possibles issus des cas d’utilisation de la simulation planétaire

- Etude du diagramme de squence

- Conception du diagramme de classes : Une planète ou un Satellite est une sorte de Corps Spatial qui contient un Aspect et des éléments internes

- Etude du diagramme d’Etat-Transition : Naissance, vie, états, mort des Planètes et Satellites.

- Etude du diagramme de package et de déploiement

- Conception des interfaces accessibles à l’utilisateur et au sein des modules.

* **HTML-CSS3**

- A partir des éléments UML conçus préalablement, Structuration de plusieurs pages HTML permettant de proposer la présentation du Système solaire d’une façon statique au départ.

- Utilisation pour ce faire des principales balises sémantiques de HTML5.

- Pour une planète, donner ses caractéristiques, son nombre de satellites.

- Après avoir structuré, apporter les éléments de présentation avec CSS3, positionner différemment les planètes, les colorer, etc..  
- Utilisation des principaux sélecteurs CSS3 pour ce faire

* **Javascript et jquery**

- Transformation de l’aspect statique des planètes et satellites en aspect dynamique.

- Gestion objet des événements écrans en respectant un modèle MVC.

- Permettre aux planètes de tourner visuellement autour du soleil et des satellites autour de leur planète selon le déclenchement utilisateur (UML, HTML5, CSS3). Exploitation du DOM pour ce faire et de l’algorithmie étudiée en début de formation.

- Création de modules et de dépendances, chaque Corps Spatial devenant un module, avec un module consacré à l’affichage.

- Utilisation de Jquery afin de modifier le TP réalisé avec HTML5/CSS3 et utilier Jquery afin de le fluidifier. Les caractéristiques des planètes apparaissent selon des évènements souris et autres.

- Utilisation de Jquery/Ajax afin d’afficher les caractéristiques des planètes et satellites issues de données présentes sur le serveur. Utilisation d’une WebApp toute faite sous Tomcat pour la partie serveur.

* **Java fondamentaux**

- Conception d’une application graphique simple afin de permettre de réaliser une simulation en Java de la rotation des planètes et satellites. Les ajouts de Java7 et 8 sont utilisés.

- Toutes les notions de l’objet sont vues dans cet exercice : Les planètes sont des classes qui héritent de classes virtuelles Corps Spatial, les satellites héritent également de Corps Spatial. Les Corps Spatiaux incluent des classes Eléments d’Orbite et d’Aspect. Plusieurs collections sont utilisées ; on retrouve tout ce qui a été fait dans JavaScript, mais appliqué à Java.

- Les coordonnées des planètes sont enregistrées dans un fichier utilisant NIO2, JDBC est utilisé  pour enregistrer les mêmes coordonnées en base MySQL.

* **JPA/HIBERBATE**

- Intégration, dans le programme réalisé préalablement, de sérialisation des objets planète et satellite via JPA s’appuyant sur JPA/Hibernate comme provider. Représentation 1->N planète Satellite. Création induite des tables, alimentation des tables via JPA/Hibernate.

- Exploitation des données de coordonnées planétaires en lecture avec JPA. Utilisation des API Criteria et JPQL afin de remonter les données à l’utilisateur.

* **JEE/Développement web**

- développement d’une application Web utilisant les Servlet et JSP afin de « dynamiser » la première application développée avec HTML5/CSS3 de départ. L’objectif est ici d’afficher la liste des planètes et de leurs satellites, et, à la sélection d’une planète, d’afficher les propriétés de cette planète. Les données étant sur le serveur dans un composant métier, il y é découpage de cette application en trois modules MVC. Le modèle simule le système solaire, la vue est une JSP, le Contrôleur est la Servlet.

- Ajout de quelques actions à la Servlet.

* **JSF 2**

- Reprise du précédent TP en remplaçant le modèle par des Managed Beans JSF2, les vues sont remplacées par des fichiers XHTML utilisant des balises JSF2. Utilisation de composants de formatage et de contrôle JSF2

- Le but est le même que précédemment : afficher les caractéristiques des planètes lorsqu’elles sont sélectionnées.

* **spring4 framework**

- Création d’un nouveau petit programme permettant à ce que Spring4 s’occupe de la création des planètes et satellites, associe l’un et l’autres (Une planète contient des satellites). On délègue à Spring Core le soin d’instancier et des gérer les objets. Utilisation des patterns principaux de Spring.

- Utilisation de Spring MVC afin de représenter ls caractéristiques des planètes.

-**usine logicielle maven et svn**

- Exportation des divers projets de planète réalisés sous Eclipse sous SVN. Utilisation des diverses commandes SVN depuis Eclipse.

- Création de plusieurs projets MAVEN et récupération des projets sous les répertoires normés MAVEN. Utilisation de MAVEN sous Eclipse, ajout de dépendances, utilisation des cycles de construction MAVEN.