



*multiplie tes talents*

## **Cours en ligne**

# ***Développement Java pour le web***

# Programme général du Cours Développement Java pour le web

## Module 1 - Programmation J2ee

### A) Bases de programmation Java

#### Unité 1 : éléments de programmation de base

Caractéristiques principales de Java  
Syntaxe du langage  
Types de données et d'opérateurs  
Structures de contrôle  
Structures itératives  
Programmation orientée objets pour Java  
Classes et objets  
Propriétés, méthodes et événements  
Caractéristiques du OOP (eredit.incaps.polimorfismo)  
Visibilité des variables  
Aperçu des exceptions  
Systèmes I/O  
Environnement de développement (Eclipse)

#### Unité 2 : bases de données SQL

Ce qu'est le SQL  
Créer une base de données  
Remplir une base de données  
Interroger une base de données  
Mettre à jour une base de donne

#### Unité 3 : éléments de programmation avancée

Collections  
Exceptions/erreurs  
Asserts  
Threads  
Débogage et déploiement sur Eclipse

#### Unité 4 : (X)HTML et JavaScript

Ce qu'est l'(X)HTML  
Les tags HTML et leur structure  
Structure d'une page  
Importance et structure du tag form

Principaux objets HTML (input, bouton, textarea, bouton radio, checkbox)  
Tableaux  
Javascript comme langage "côté client"  
Principaux usages du Javascript  
Syntaxe de base du JavaScript

## **Unité 5 : éléments de base de programmation web**

MVC  
Servlet  
JSP  
Bean

## **B) Programmation O.O. avancée**

Cette partie est dédiée aux caractéristiques du langage qui sont peu connues habituellement, même des programmeurs experts, comme l'utilisation correcte des interfaces, le polymorphisme et l'hérité des constructeurs, les modificateurs, les exceptions, les threads... Toutefois nous considérons que la connaissance de ces caractéristiques est très importante. En effet, dans la programmation, on se trouve parfois devant des solutions compliquées qui peuvent devenir simples ou face à des bogues inexplicables qui pourraient pourtant être résolus. Ces caractéristiques avancées sont également fondamentales pour obtenir la SCJP.

### **Unité 1 : Comparaison entre objets**

Cette unité est dédiée à l'implémentation correcte du concept de comparaison entre les objets Java, en exploitant les caractéristiques mises à disposition par le langage.

### **Unité 2 : Collection Framework**

Cette unité est entièrement dédiée aux paquets probablement les plus utilisés dans l'absolu : java.lang e java.util. Rappelons que le premier est le seul paquet importé automatiquement dans tous nos programmes. Le deuxième contient des classes dont le programmeur ne peut se passer. Évidemment, ce module ne couvrira pas les 200 classes présentes dans ces paquets : nous chercherons plutôt à présenter les principaux concepts et la philosophie avec laquelle utiliser ces paquets.

### **Unité 3 : JAVA I/O**

Les applications ont souvent besoin d'utiliser des informations provenant de sources externes ou d'envoyer des informations à des destinataires externes. Par « informations », nous ne parlons pas seulement de chaîne de caractères, mais

également les objets, les images les sons, etc. Par « sources ou destinataires externes » à l'application, en revanche, nous entendons fichiers, disques, réseaux, mémoires ou tout autre programme. Dans ce module, nous verrons comment Java permet de gérer la lecture (input) des sources externes et l'écriture vers des destinations externes (output). Nous introduirons en particulier le paquet java.io, suplice et délice des programmeurs java. Le paquet en question est vaste et très complexe. Connaître chaque classe du paquet est une entreprise ardue et surtout inutile. Pour pouvoir gérer l'input -output dans java, il faut comprendre sa philosophie, qui est régulée par le patron de conception connu sous le nom de Decorator. Ne pas comprendre ce patron Decorator impliquerait des difficultés récurrentes à se débrouiller parmi les classes de java.io.

## **C) Web dynamique**

### **Unité 1 : Conteneur de servlets**

Dans cette unité nous introduisons les principaux outils de réalisation d'une application web sur Java. Nous verrons en particulier le fonctionnement et la configuration d'Apache Tomcat, conteneur OpenSource d'applications web.

### **Unité 2 : Servlet**

Dans cette unité nous introduisons le concept de servlet et de ses principales ramifications HttpServlet, la gestion des requêtes qui nous arrivent du « client » et la mise en place de la réponse vers le « client ».

### **Unité 3 : JavaServer Pages**

Ici, nous introduisons la principale technologie de visualisation disponible pour une application web basée sur J2EE : les Java Server Pages. Le but de cette technologie est de rendre les pages web dynamiques et d'en simplifier l'écriture et la manutention.

### **Unité 4 : Gestion de la session**

Dans cette unité nous introduisons l'élément principal de toutes les applications web, c'est-à-dire la technologie qui permet à l'application web de gérer une ou plusieurs sessions utilisateur en même temps et qui permet donc à l'application de visualiser des données et des informations différentes en fonction de l'utilisateur.

## **Unité 5 : Custom tag**

Dans cette unité, nous voyons comment la technologie à la base des JSP peut et doit être développée pour satisfaire les exigences particulières d'une application web.

## **Unité 6 : JSTL**

Ici nous introduirons les JSP Standard Tag Library, c'est-à-dire les outils qui permettent de simplifier la gestion d'une JSP.

## **D) EJB**

### **Unité 1 : conteneur EJB**

Ici nous introduisons les principaux éléments pour la gestion et la configuration d'un EJB, en particulier nous verrons la notion d'application server.

### **Unité 2 : classification des beans**

Dans cette unité nous classerons les EJB (EnterpriseJava Beans) en sessions, entités, messages.

### **Unité 3 : Session bean**

Ici, nous verrons en détail la structure des EJB 2 de type session. Nous introduirons les concepts de session bean stateless et de session bean statefull.

### **Unité 4 : Entité bean**

Nous verrons dans le détail les EJB2 de type entité, nous introduirons le concept d'ORM (Object Relational Mapping) et sa réalisation au travers des entités.

### **Unité 5 : EJB 3.0**

Ici, nous verrons les principales nouveautés introduites par EJB 3, en particulier le JPA (Java Persistence API)

## **Module 2 – UML et conception de logiciels**

### **A) Introduction à UML**

#### **Unité 1 : ce qu'est UML**

Introduction au langage UML (Unified Modeling Language), à ses caractéristiques et à ses finalités

## **Unité 2 : Introduction aux diagrammes**

Introduction aux principaux diagrammes UML pour la conception des logiciels et leur utilisation.

Diagrammes structurels

Diagrammes comportementaux

Diagrammes comportementaux d'interaction

Diagrammes des paquetages

Diagrammes des classes

## **Unité 3 : Architecture dirigée par les modèles (MDA : Model-driven architecture)**

Nous introduirons la méthodologie MDA pour le développement de logiciels systèmes.

Métamodèles et diagrammes

OMG

Roundtrip engineering

## **Unité 4 : Outils UML : critères de choix**

Introduction à Visual modeling

Facteurs à prendre en compte (coût, adéquation au standard, soutien au travail de groupe, etc.)

L'UML et vos exigences

## **B) Introduction aux patrons de conception (design patterns)**

### **Unité 1 : L'histoire des patrons de conception**

Introduction aux patrons de conception

### **Unité 2 : Caractéristiques des patrons de conception**

Étude des caractéristiques qui permettent de définir un patron de conception

Normes

Problèmes

Solutions

Conséquences

## **Unité 3 : Classification des patrons**

Patrons créationnels  
Patrons structurels  
Patrons comportementaux

## **Unité 4 : Exemples de patrons de conception**

Le J2EE  
Les Gof

## **C) Gestion des états**

### **Unité 1 : état de patron**

Simplification du code  
Explicitation des transitions entre états  
Les états, les transitions, la nidification

### **Unité 2 : processus d'entreprise**

Les cas d'usage : lieu et usage de la procédure. Scénarii, variantes, illustrations, relations et liens entre les cas.  
Le diagramme d'activité pour décrire l'activité et l'organisation des process  
Activités, actions élémentaires, transition, événements, couloirs, synchronisation  
Le diagramme des interactions: une variante du diagramme d'activité  
Le diagramme de séquence pour décrire l'organisation des process. L'échange entre les acteurs. L'axe temporel.

### **Unité 3 : Abstract Factory**

La création et le partage des états

## **D) Commandes et strategies**

### **Unité 1 : Diagramme des classes de gestion de commande des**

programmes  
Création des fonctionnalités annuler/répéter  
Le patron Commande  
Commandes de Sodoku  
Création et gestion des commandes

## **Unité 2 : Appel de commandes**

Composition des commandes

Créer des commandes macro à travers les commandes composites

## **Unité 3 : Création d'un nouveau jeu**

Définir et créer la stratégie

Mise en place de la stratégie de création

## **Module 3 – Frameworks Open Source**

### **A) Struts**

#### **Unité 1 : Struts Overview**

Introduction au Framework Struts e au patron Front Controller. Caractéristiques de l'ActionServlet et du fichier de configuration du Framework

#### **Unité 2 : éléments de base**

Introduction aux principaux éléments qui composent le Framework, les classes action et Actionform et leur configuration.

#### **Unité 3 : éléments avancés**

Introduction aux caractéristiques avancées du Framework

#### **Unité 4 : Tiles**

Introduction au Framework de visualisation Tiles.

#### **Unité 5 : Validator**

Introduction au Framework de validation Validator

#### **Unité 6 : Struts 2**

Principales nouveautés introduites par la nouvelle version du Framework Struts avec le patron Interceptor

### **B) JSF**

#### **Unité 1 : Faces**

Introduction aux caractéristiques du Framework de visualisation Faces



## **Unité 2 : Java server faces**

Implémentation Sun des caractéristiques Faces

## **Unité 3 : MyFaces**

Implémentation Apache des caractéristiques Faces

## **Unité 4 : Extensions**

Principales extensions du Framework

Facelets

Seam

Extensions

## **C)Hibernate**

### **Unité 1 : Hibernate**

Introduction au Framework de persistance Hibernate

### **Unité 2 : ORM**

Réalisation de la persistance objet des données avec Hibernate

### **Unité 3 : Requêtes**

Dans cette unité nous verrons comment mener et optimiser des recherches sur une base de données Hibernate.

## **D) Spring**

### **Unité 1 : Spring**

Introduction au Framework Spring e au patron IOC (Inversion of Control), gestion des Java bean avec des fichiers de configuration

### **Unité 2 : Dependency Injection**

Découverte du patron Dependency Injection, de sa réalisation et de son utilisation avec Spring

### **Unité 3 : Spring MVC**

Comment Spring réalise le patron Front Controller. Analyse des principales différences avec le Framework Struts

#### **Unité 4 : Spring Web Flow**

Étude du nouveau Framework des gestion des flux d'une application web.

#### **Unité 5 : Spring AOP**

Introduction à la programmation orientée aspects et à sa réalisation à travers Spring

#### **Unité 6 : Spring Integration**

Comment Spring permet la réitération entre les différentes technologies Java : Struts, Hibernate, Jpa, en respectant les règles de bonne programmation.

#### **Unité 7 : Apache Common Library**

Introduction aux bibliothèques Apache Common pour le support à la programmation Java

#### **Unité 8 : Jasper Report**

Introduction à la librairie pour la création dynamique de rapport Jasper Report

# Objectifs, modalités de suivi et évaluation

## Objectifs et modalités

L'objectif du cours est de fournir les connaissances techniques et méthodologiques nécessaires aux personnes qui souhaitent exercer la profession d'analyste programmeur dans le secteur J2ee.

Le cours sera disponible sur une plateforme de formation à distance (FAD) comprenant du matériel d'enseignement (vidéos et leçons) et des tests qui permettront de vérifier l'assimilation des notions.

Grâce à des outils de collaboration Internet comme le chat ou l'e-mail, les étudiants pourront bénéficier de l'accompagnement des enseignants dans leur parcours pédagogique.

## Structuration du cours

Le programme s'oriente vers les technologies de développement des logiciels sur plateforme Sun Microsystems, les plus demandées actuellement par le marché public et privé. Il part de zéro (les bases de la programmation) pour arriver au développement Java, web, Html, Javascript, SQL sur MySql, EJB et s'oriente aussi sur les Frameworks plus répandus comme Spring, Struts, JSF, Hibernate et sur les modalités d'analyse et de conception (UML).

Le cours se divise en 3 modules: J2ee, UML et conception de logiciels et Framework Open Source.

## Public cible et modalités d'évaluation

Le cours est destiné aux personnes ayant suivi des études supérieures qui souhaitent entreprendre une carrière d'analyste programmeur Java.

L'évaluation des élèves sera effectuée en ligne à travers des tests d'évaluation.