

Leuville Objects

EXPERTS EN TECHNOLOGIES JAVA, SOA ET UML,
DU SERVEUR AU MOBILE

# CATALOGUE DES FORMATIONS

Septembre 2009 – Décembre 2009

Accompagnement pédagogique, Ateliers sur mesure, LEUVILLE OBJECTS se positionne aujourd'hui comme la société de prestation de services et de formations spécialiste des technologies Objet et experte dans la conduite de projets et de formations C++, C#, Java et JavaEE, Services Web, SOA et UML.

Nous bâtissons et concrétisons nos formations sur des exercices, études de cas et travaux pratiques réels, déjà vécus, développés et appliqués par nos formateurs. Cette mise en condition vous permet une montée en compétence rapide et durable. Nos formations sont dispensées en inter et toutes compatibles en intra, issues de notre expertise alliant expérience du terrain et veille technologique constante.

Nous accueillons un maximum de 5 stagiaires par session afin de privilégier les relations stagiaires/formateur, assurer un suivi attentif et créer un véritable climat d'échange. Nos Ateliers constituent une nouvelle approche de la formation inter-entreprises, entièrement tournée vers la qualité de la relation entre le formateur et son stagiaire.

Laurent Nel, gérant et fondateur de LEUVILLE OBJECTS

Découvrez tous les avantages des formations LEUVILLE OBJECTS et les modalités d'inscription en nous contactant au 01 39 50 20 00 ou sur www.leuville.com

Filière Java / JavaEE ..... 4
Les formations proposées ..... 4
Les bases du langage Java ..... 5
Applications graphiques en Java avec Swing ..... 6
Traitements XML en Java ..... 7
Servlets et JSP ..... 8
Java Server Faces ..... 9
Développer avec le framework Struts 1.x ..... 10
Développer avec le framework Struts 2.x ..... 11
Composants métiers JavaEE : les EJB 3.0 ..... 12
Filière C++ ..... 13
Les formations proposées ..... 13
Développer en C++ - Niveau 1 ..... 14
Développer en C++ - Niveau 2 ..... 15
Filière C# ..... 16
Les formations proposées ..... 16
Les bases du langage C# ..... 17
Filière Web Services ..... 18
Les formations proposées ..... 18
Développer des Web Services SOAP en Java ..... 19
Développer des services RESTful en Java ..... 20
Filière UML et l'Objet ..... 21
Les formations proposées ..... 21
Les concepts Objet ..... 22
Savoir modéliser avec UML ..... 23
Design Patterns et éléments de conception ..... 24
Calendrier des sessions ..... 25
Cours du samedi ..... 25
Calendrier des sessions inter-entreprises ..... 25

# Filière Java / JavaEE

La filière Java / JavaEE propose des formations couvrant l'ensemble des technologies basées sur le langage Java. Les débutants souhaitant s'initier au langage y trouveront les formations de base et les personnes expérimentées seront surement interessées par des formations plus poussées sur les technologies web (Servlet, JSP, JSF, ..) ou EJB, par exemple.

## Les formations proposées

- **CP-JAVA1** Les bases du langage Java
- **CP-SWING** Applications graphiques en Java avec Swing
- **CP-JXML** Traitements XML en Java
- **CP-WJSP** Servlets et JSP
- **CP-WJSF** Java Server Faces
- **CP-STR1** Développer avec le framework Struts 1.x
- **CP-STR2** Développer avec le framework Struts 2.x
- **CP-EJB3** Composants métiers JavaEE : les EJB 3.0

# Les bases du langage Java

| | |
|---|---|
| **Code** | CP-JAVA1 |
| **Durée** | 3 jours |
| **Tarif** | 990 € HT |

## Description

Ce cours est une première approche du langage Java. Après présentation des concepts Objet et de la syntaxe du langage, le stagiaire apprend à développer des applications interactives en Java. Des exercices pratiques permettent d'illustrer et de mettre en oeuvre les concepts abordés.

## Participants

Ce cours s'adresse à toute personne amenée à travailler avec le langage Java et / ou souhaitant s'initier à ce langage.

## Pré-requis

- La connaissance d'un langage informatique est un plus, mais n'est pas obligatoire.
- La connaissance préalable des concepts Objet est nécessaire.

## Programme

### Syntaxe de base du langage Java

*Présentation des éléments syntaxiques de base du langage : la structure en blocs, la déclaration des variables, la déclaration d'une méthode, ...*

- Déclaration et initialisation d'une variable
- Manipulation des variables, opérations arithmétiques
- Signature et déclaration d'une méthode
- La structure en blocs

### Contrôle du flot d'exécution

*Apprentissage des boucles (for, foreach, while, do while) et des conditions (if, switch, opérateur ternaire, assertions).*

- Itérer avec les boucles for / while
- Tests à base de if / else if / else
- Les expressions booléennes en Java

### Classes et interfaces

*Déclaration de classes et d'interfaces en Java, contrôle de la visibilité des membres, classes abstraites, ...*

- Déclaration d'une classe / interface en Java
- Contrôle de la visibilité d'une classe et de ses membres
- Héritage, polymorphisme

### Les exceptions

*Présentation du système de gestion des erreurs par exception. Présentation des exceptions courantes. Différence en erreurs, exceptions et RuntimeException.*

- Exceptions et erreurs, exceptions gérées / non gérées
- Gérer ou propager une exception
- Les blocs try / catch / finally

### Entrées / Sorties en Java

*Apprentissage des classes de base du package java.io pour lire, écrire (du texte, des objets, ...) dans des flux (fichiers, flux réseaux, ...).*

- Le package java.io
- Accès au système de fichier
- Les classes Reader / Writer
- Lecture / Ecriture bufferisée
- Sérialisation

### Les énumérations

*Apprentissage des énumérations, types apparus avec la version 5 de Java.*

- Déclaration d'une énumération
- Manipulation des membres d'une énumération
- Déclaration de membres dans un type énumération

### Collections et généricité

*Présentation des structures de données dites "Collection" en Java (listes, piles, queues, ...). Présentation de la généricité en Java.*

- Les méthodes equals et hashcode
- Collection, List, Set et Map
- Les implémentations, leurs différents usages
- La généricité en Java

### Les annotations

*Présentation des annotations, nouveau type de données apparus dans Java 5. Utilisation des annotations standard, création et utilisation d'une annotation.*

- Présentation et utilité des annotations
- Présentation des annotations communes du langage Java
- Créer et utiliser ses propres annotations

### Le packaging d'une application Java

*Description du format d'archive jar et des métadonnées associées.*

- Le format jar
- Métadonnées d'une archive jar

**Code** CP-SWING

**Durée** 3 jours

**Tarif** 990 € HT

# Applications graphiques en Java avec Swing

## Description

La bibliothèque Swing est une API, incluse dans le langage Java, permettant de développer des applications graphiques (clients lourds ou applets) en Java. Elle est basée sur l'implémentation d'un ensemble de patterns et propose, de base, un ensemble complet de composants graphiques réutilisables. Ce cours présente cette bibliothéque, les composants disponibles, les techniques de mise en page de ces composants et la gestion des évènements. Tout au long du stage, des exercices pratiques permettent d'illustrer et de mettre en oeuvre les concepts abordés.

## Participants

Ce cours s'adresse aux développeurs Java souhaitant découvrir ou approfondir leurs connaissances de la bibliothèque graphique Swing.

## Pré-requis

- La connaissance préalable du langage Java est nécessaire.

## Programme

### Introduction à swing

*Présentation générale de la bibliothèque, différences avec la bibliothèque awt, principes des composants / conteneurs, ...*

- Quelques exemples d'applications Swing
- Le design pattern composite et son application en Swing
- Comparaison avec d'autres langages d'applications graphiques

### Conteneurs et composants

*Description des conteneurs et des composants de base de la bibliothèque Swing. principe d'ajout des composants dans les conteneurs.*

- Les composants de base : boutons, labels, cases à cocher, ...
- Les conteneurs de base : panels, fenêtres, ...
- La classe JFrame et ses dérivées
- Structure arborescente du modèle swing

### Les gestionnaires de présentation

*Présentation des différentes gestionnaires de présentation (layout manager) disponibles avec swing. Comment gérer l'aspect de l'application en cas de redimensionnement de la fenêtre principale ?*

- Principe des LayoutManager
- FlowLayout, GridLayout, CardLayout, et les autres
- Positionnement absolu des éléments avec le layout "null"

### Les composants avancés

*Présentation des composants avancés basés sur le modèle MVC.*

- Les listes
- Les tableaux
- Les arborescences
- Le composant JScrollPane

### Gestion des évènements

*Ajouter de l'interactivité entre les composants et l'utilisateur en gérant les évènements. Principes des "écouteurs" d'évènement.*

- Les Events et les Listeners
- Evènements courants et classes associées
- Créer et propager ses propres évènements

### Quelques considérations Objet

*Comment organiser ses classes et ses méthodes dans du code swing ? Où et comment initialiser ses composants ? Où placer la gestion des évènements ?*

- Problématiques liées à la programmation d'interfaces graphiques
- L'initialisation d'un conteneur
- L'astuce de l'initialisation par l'accesseur
- Création de nouveaux composants par héritage / délégation

### Développer ses propres composants

*Développer des nouveaux composants par redéfinition des méthodes paint et paintComponent.*

- Les méthodes paint et paintComponent
- Utilisation d'une image pour créer un rendu de composants

### Java2D

*Utiliser la classe Graphics2D pour dessiner à la volée ses composants graphiques.*

- La classe Graphics2D
- Les méthodes de dessin
- Ajout de texte dans un graphique

### Animation en Swing

*Créer des animations dans une application Swing. Techniques d'animation pour développer un jeu en java.*

- Techniques à base de Timer
- Applications swing avec boucle principale
- La méthode repaint et le rafraichissement partiel de l'écran

# Traitements XML en Java

**Code** CP-JXML

**Durée** 3 jours

**Tarif** 990 € HT

## Description

XML est devenu un incontournable des systèmes informatiques de l'entreprise. Il est utilisé pour stocker des données, échanger des informations, configurer des systèmes... Depuis la version 1.4 du langage Java, l'API XML fait partie intégrante du JDK et celle-ci n'a pas arrêté de se compléter jusqu'à aujourd'hui.

## Participants

Ce cours s'adresse aux développeurs Java désireux de connaître les techniques de traitement de documents XML en Java.

## Pré-requis

- La connaissance du langage Java est nécessaire.
- Une bonne connaissance de XML est recommandée, même si un rappel est prévu au programme.

## Programme

### Rappels : les bases de XML

*Rappels sur les principes de base des langages à balises, les règles syntaxiques, la validation.*

- Introduction à XML
- Les règles syntaxiques de XML
- La validation par schéma
- Les espaces de nommage XML

### Le parsing DOM

*Principes et mise en oeuvre du parsing DOM en Java.*

- Les principes du DOM
- Parser un document XML avec DOM en Java
- Créer un document XML avec DOM en Java
- Validation

### Le parsing SAX

*Principes et mise en oeuvre du parsing SAX en Java.*

- Les principes du SAX
- Parser un document XML avec SAX en Java
- Créer un document XML avec SAX en Java
- Validation

### Le parsing Stax

*Principes et mise en oeuvre du parsing Stax en Java.*

- Les principes du Stax
- Parser un document XML avec Stax en Java
- Créer un document XML avec Stax en Java
- Validation

### Transformations XSLT

*Principes et mise en oeuvre de la transformation à base de feuille de style XSL en Java.*

- Feuilles de style XSL
- Application d'une feuille de style à un document en Java
- La résolution des entités et des URL

### La transformation XSL-FO avec Apache FOP

*Utilisation de l'API Apache FOP pour effectuer de la transformation XSL-FO, utile pour, par exemple, générer des documents PDF à partir d'un document XML et d'une feuille de tyle XSL.*

- Formating Object
- Apache FOP : présentation de l'API

### Le mapping XML / Java avec JAXB

*Manipulation de l'API et des outils associés à JAXB pour créer des mappings XML / Java.*

- Le principe du mapping
- Les outils xjc / schemagen
- Les annotations JAXB
- Marshalling et Unmarshalling

### XLink et XPointer

*XLink et XPointer sont deux technologies XML permettant de créer des liens entre documents.*

- Utilisation de XLink et XPointer dans des documents XML
- Résolution des liens lors d'un parsing en Java

### XQuery avec XQJ

*XQJ (JSR 225) est l'API permettant d'exécuter des requêtes XQuery en Java.*

- Présentation de la JSR 225
- Utilisation de l'implémentation de référence

# Servlets et JSP

**Code** CP-WJSP

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Le développement d'applications Web que ce soit pour Internet ou pour des Intranet a pris une place importante dans le monde de l'informatique. Elle impose de prendre en compte les contraintes et besoins liés aux évolutions des technologies en apportant un dynamisme à celles-ci. Cette formation permettra de découvrir comment mettre en place des applications de haut niveau en utilisant des sessions, des balises utilisateurs ...

## Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et souhaitant apprendre à mettre en place des applications Web Java à l'aide de JSP et de servlets.

## Pré-requis

- Maîtrise du langage Java nécessaire
- La connaîssance du langage HTML est un plus

## Programme

### Les servlets et JSP

*Rappels sur les notions de base du Web et la structuration normalisée des applications Web Java.*

- Rappel sur HTTP
- La structure normalisée d'une URL
- La structure d'une application Web

### Les servlets

*Présentation des principes de base associés aux servlets.*

- Structure d'une servlet
- Le cycle de vie d'une servlet
- Utilisation de formulaires

### Les pages JSP

*Présentation des principes de base associés aux pages JSP.*

- Structure d'une page JSP
- Syntaxe des pages JSP
- Le cycle de vie d'une page JSP
- scripts, directives, expressions et actions
- Les objets pré définis
- La gestion des erreurs

### L'utilisation de sessions

*Présentation de la gestion de sessions utilisateurs dans les applications.*

- Pourquoi utiliser des sessions ?
- Principe de fonctionnement des sessions
- Les méthodes de suivi des sessions
- Le cycle de vie des sessions
- Le modèle MVC
- Utilisation des JavaBeans

### Utilisation de bases de données dans les applications

*Les applications Web s'appuient souvent sur des informations qui se trouvent en bases de données. Il est donc important de voir comment utiliser celles-ci dans une application JSP/servlets.*

- Présentation de JDBC
- Utilisation de JNDI pour accèder aux données
- Utilisation de pools de connexion
- Accès aux ressources via JNDI

### La sécurité des applications Web Java

*En dehors de l'utilisation de protocoles sécurisés, il est important de regarder comment réduire l'accès aux différentes parties d'une application Web.*

- La sécurité déclarative
- Utilisateur, rôle et realm
- Sécurité par programmation

### Création de balises utilisateurs (taglibs 1.2)

*Présentation des balises utilisateurs.*

- Les différents types de balises utilisateurs
- Les balises sans corps
- Les balises avec corps

### JavaServer Pages Standard Tag Library

*Présentation de la librairie de balises ansi que du langage unified EL.*

- Le langage Unified Expression Language
- Les différentes catégories de balises

### Les taglibs 2.0

*L'utilisation des anciennes versions de taglibs pouvant être compliquée, de nouvelles possibilités ont été offertes avec Java EE5*

- Les balises SimpleTag
- Les fichiers tag

# Java Server Faces

**Code** CP-WJSF

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Le développement d'applications Web en Java peut s'avérer complexe lorsque celles-ci deviennent importantes ? Il est donc nécessaire d'utiliser des outils permettant d'en alléger le développement par l'intermédiaire d'un Framework. L'utilisation de JSF peut être la solution car il a l'avantage de faire partie de Java EE5.

## Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et souhaitant apprendre à mettre en place des applications Web Java à l'aide du Framework JSF.

## Pré-requis

- Maîtrise du langage Java nécessaire
- La connaissance du langage HTML
- La connaissance des JSP et des servlets

## Programme

### Les concepts de JSF

*Avant de commencer à utiliser JSF, il est important de comprendre les concepts qui y sont associés.*

- Le modèle MVC2
- Le modèle événementiel de JSF
- La servlet FacesServlet
- Le fichier facesconfig.xml
- Les ressources bundle

### Les composants standard

*Présentation des différents composants fournis avec JSF.*

- Les différentes familles de balises
- Utilisation du langage Unified EL avec JSF

### Cycle de traitement d'une requête

*Analyse du fonctionnement de JSF lors de l'envoi d'une requête.*

- Les différents éléments appelés

### Utilisation de JavaBean dans JSF

*Présentation des différents types de Beans qui peuvent être utilisés avec JSF.*

- Rappel sur les JavaBeans
- Déclaration d'un Bean pour une application (Manageg bean)
- Les Backing Beans

### La navigation entre pages

*Présentation du modèle qui permet de définir les enchainements d'appels entre les différentes pages d'une application.*

- Définition des règles de navigation
- Navigation statique et dynamique

### Conversion et validation de données

*Présentation des possibilités offertes pour améliorer la gestion des données saisies par l'utilisateur.*

- Le processus de conversion et de validation
- Utilisation de la conversion standard
- Utilisation de la validation standard
- Créer ses propres valideurs et convertisseurs

### Les événements JSF

*Présentation des différents événements rencontrés avec JSF et présentation de leur utilisation.*

- Cycle de vie des événements
- Les types d'événements
- Les balises d'écoute d'événements

### Création de composants personnalisés

*Présentation de la mise au point de composants personnalisés.*

- Les classes permettant de créer des composants
- Encodage et décodage

### Les librairies complémentaires

*Présentation d'extensions existantes pour JSF.*

- Présentation de certaines librairies
- La librairie Apache Tomahawk

# Développer avec le framework Struts 1.x

**Code** CP-STR1

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Struts est un framework de la fondation Apache permettant de développer des applications Web en Java conformes au modèle MVC2. Il propose une API ainsi qu'une configuration à base de fichiers XML afin de fournir un modèle de développement. Ce cours présente les différents constituants de ce framework et la façon de les assembler afin de créer une application complète.

## Participants

Développeurs et chefs de projet amenés à travailler avec le framework Struts 1.

## Pré-requis

- La connaissance du langage Java est obligatoire
- Des connaissances sur Servlet et JSP sont très fortement recommandées.

## Programme

### Servlets, JSP et le modèles MVC2

*Historique du développement des applications web en Java*

- Application web JavaEE
- Servlet et JSP
- Le modèle MVC2

### Introduction au framework Stuts

*Présentation des principes de base de Struts*

- Le modèle MVC2 appliqué
- ActionServlet, ActionForm et Action
- Taglib Struts

### Configuration de Struts

*Description des divers fichiers de configuration utilisés par Struts.*

- Descripteur de déploiement web
- struts-config.xml

### Les ActionForms

*Gestion de la saisie utilisateur avec les ActionForms.*

- Portée d'un ActionForm
- Cycle de vie d'un ActionForm
- Validation de la saisie

### Les DynaActionForms

*Intérêt et utilisation des DynaActionForms*

- Présentation des DynaActionForms
- Configuration
- Validation de la saisie

### Actions prédéfinies

*Présentation des classes Actions prédéfinies dans le framework Struts.*

- DispatchAction
- ForwardAction
- IncludeAction
- LookupDispatchAction
- SwitchAction

### Les taglibs Struts

*Revue des balises Struts utilisables dans les JSP.*

- Balises HTML
- Balises bean
- Balises logic
- Balises tiles

# Développer avec le framework Struts 2.x

**Code** CP-STR2

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Apache Struts 2 est un framework facilitant le développement d'applications Web Java d'entreprise. Le framework est conçu de façon à faciliter l'ensemble du cycle de vie du logiciel, de son développement, son déploiement jusqu'à sa maintenance et son évolution. Apache Struts 2 était connu, à l'origine, en tant que WebWork 2. Après avoir travailler indépendamment pendant plusieurs années, les communautés WebWork et Struts se sont associées pour créer Struts 2. Cette nouvelle version de Struts est plus simple d'utilisation et plus proche de ce que Struts aurait dû être.

## Participants

Développeurs ou chefs de projet amenés à travailler avec le framework Struts 2.

## Pré-requis

- La connaissance du langage Java est obligatoire
- Des connaissances sur Servlet et JSP sont très fortement recommandées.

## Programme

### Servlets, JSP et le modèles MVC2

*Historique du développement des applications web en Java*

- Application web JavaEE
- Servlet et JSP
- Le modèle MVC2

### Architecture Struts 2

*Présentation des différents composants du framework et de leurs interactions.*

- Le Servlet Filter Dispatcher
- Principe des interceptors
- Les actions

### Configuration d'une application Struts 2

*Présentation des fichiers de configuration utilisés par Struts 2 ainsi que des éléments de configuration JavaEE nécessaires.*

- Configuration de base
- Configuration des règles de navigation en XML
- Configuration des Interceptors

### Annotations

*Présentation des annotations utilisables (optionnelles) avec Struts 2 afin d'obtenir une application avec zéro configuration.*

- Configuration à base d'annotations
- Configuration des Actions et des Results

### Développement d'une Action

*Présentation des classes Action et de la façon de les développer.*

- Développer une classe Action
- Injection des valeurs saisies
- Rediriger vers un Result

### Les types de Result

*Présentation des types de Result pré-définis dans le framework.*

- Chain, Dispatcher, HttpHeader
- Redirect, Stream, XSL
- PlainText, Tiles
- Intégration avec FreeMarker et Velocity

### Balises Struts 2

*Présentation des balises Struts 2 utilisables dans les JSP.*

- Control tags
- Data tags
- Struts UI tags
- Non-Form UI tags

### OGNL

*Object Graph Navigation Language est le langage de navigation à travers des objets utilisé dans les balises Struts 2.*

- Syntaxe
- Référencer une propriété
- La gestion des tableaux, listes, et autres structures
- Les variables
- Invocation de méthodes

### Validation et localisation

*Présentation du processus de validation et de localisation du framework Struts 2.*

- Les validateurs de base
- Configuration des Validators
- Localisation des textes et des formats

# Composants métiers JavaEE : les EJB 3.0

**Code** CP-EJB3

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Les EJB ont pendant longtemps souffert d'une mauvaise image : trop complexes à développer, fastidieux à configurer, pas toujours réellement portables, ... Cependant, avec la venue de JavaEE 5 et de la spécification EJB 3.0, les choses changent. Les EJB deviennent beaucoup plus simples à développer et à utiliser et méritent donc que l'on revoit notre jugement à leur sujet.

## Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et la plateforme JavaEE et souhaitant apprendre à développer, utiliser et intégrer des EJB.

## Pré-requis

- Maîtrise du langage Java nécessaire
- Connaissances de base sur les serveurs d'application JavaEE

## Programme

### Les Objet Métier

*Présentation de la vision de l'Objet Métier par l'OMG et ses implémentations dans diverses technologies.*

- Objet métier : présentation
- Architectures à base d'objets répartis : CORBA, DCOM, RMI
- Limite des architectures à base d'objets répartis

### Caractéristiques de base de JavaEE

*Rappels sur les principes de base de la plateforme JavaEE et du rôle des conteneurs.*

- Les serveurs d'applications JavaEE
- Rôle des conteneurs
- Architecture canonique JavaEE
- Les API de JavaEE
- Panorama des principaux serveurs JavaEE

### Introduction aux EJB

*Présentation des différents types d'EJB, leurs différences et leur place dans une architecture.*

- EJB Session Stateless
- EJB Session Stateful
- EJB Message Driven
- La disparition des EJB Entity
- Les services offerts par le conteneur

### Développer des EJB Session Stateless

*Techniques et bonnes pratiques de développement des EJB Session Stateless.*

- Cycle de vie d'un EJB Session Stateless
- Développer les interfaces et l'implémentation
- Présentation des annotations utiles
- Lien entre EJB Session Stateless et Web Service

### Développer des EJB Session Stateful

*Techniques et bonnes pratiques de développement des EJB Session Stateful.*

- Cycle de vie d'un EJB Session Stateful
- Développer les interfaces et l'implémentation
- Présentation des annotations utiles

### Java Persistence API

*Présentation de l'API qui vient remplacer les EJB Entity.*

- Le mapping Objet / Relationnel
- Notion d'entité
- Entity Manager et Persistence Unit
- Annotations JPA

### Développer des EJB Message Driven

*Développer des EJB évènementiels basés sur JMS.*

- Java Messaging Service
- Cycle de vie d'un EJB MDB
- Les annotations utiles

### Services techniques EJB

*Présentation de la façon de configurer les services techniques pris en charge par le conteneur.*

- Les EJB et la gestion de la sécurité
- Les EJB et la gestion des transactions

### Les Intercepteurs

*Présentation des Intercepteurs EJB, utilité, développement.*

- Le concept d'intercepteur EJB
- Intercepter une invocation

# Filière C++

La filière C++ propose des formations permettant de s'initier au langage C++ des bases du langage à l'utilisation de librairies permettant d'améliorer la qualité du code. Les débutants souhaitant s'initier au langage y trouveront une formation de base et les personnes expérimentées seront surement interessées par une formation plus poussée sur l'utilisation de STL ou Boost, par exemple.

## Les formations proposées

- **CP-CPP1** Développer en C++ - Niveau 1
- **CP-CPP2** Développer en C++ - Niveau 2

# Développer en C++ - Niveau 1

**Code** CP-CPP1

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Le langage C++ bien qu'ayant une réputation de complexité est encore beaucoup utilisé aujourd'hui. Il est important pour l'utiliser de comprendre les concepts associés ainsi que les problématiques mémoire qu'il est possible de rencontrer.

## Participants

Toutes personnes souhaitant découvrir le langage C++.

## Pré-requis

- La connaissance d'un langage informatique est préférable

## Programme

### Les notions de base en C++

*Avant de pouvoir utiliser le langage, il est nécessaire de découvrir les notions de base qui y sont associées.*

- Les types de base
- Les boucles et les conditions
- Les pointeurs
- Le modificateur const
- Les références
- Les espaces de nommage

### Les concepts Objet

*C++ étant un langage orienté Objet, il est important de regarder quels en sont les principes et comment modéliser cela.*

- Les concepts Objet
- Le langage UML
- Les diagrammes de classe

### Les fonctions

*Présentation des fonctions et de leurs utilisations possibles.*

- Structure et déclaration d'une fonction
- Les paramètres par défaut
- La surcharge de fonctions
- La redéfinition de fonction
- Les fonctions constantes

### Les classes en C++

*Présentation de l'utilisation des classes, structures et unions en C++.*

- La structure d'une classe
- Les structures et les unions
- Les interfaces
- Les constructeurs et le destructeur

### L'héritage

*Présentation de la mise en place des principes de base de l'héritage.*

- Mise en place de l'héritage
- La redéfinition des fonctions
- Les fonctions virtuelles
- Les classes abstraites

### La surcharge des opérateurs

*Présentation du mécanisme de surcharge des opérateurs.*

- Pourquoi surcharger les opérateurs ?
- Les différents types de surcharges
- Surcharges de quelques opérateurs +, - ...

### Les templates

*Mise en place de la généricité pour les classes et les fonctions.*

- Le rôle des templates
- Les fonctions templates
- Les classes templates

### La gestion des exceptions

*Présentation de la gestion des erreurs en C++.*

- Les blocs try/catch
- Définir des exceptions
- Lancer une exception

### Les entrées/sorties

*Etude des classes permettant de gérer les entrées/sorties standard ainsi que les accès aux fichiers.*

- Les différentes classes
- Lecture des entrées clavier
- Affichage dans la console
- Accès aux fichiers

# Développer en C++ - Niveau 2

**Code** CP-CPP2

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Le langage C++ est complexe, et offre de nombreuses possibilités. Que ce soit la gestion de la mémoire ou une meilleure utilisation du langage, il est important de s'assurer que le code mis en place offrira le plus de sécurité.

## Participants

Ce cours est destiné à des personnes utilisant déjà le langage C++ et qui souhaitent approfondir leurs connaissances.

## Pré-requis

- Une première expérience du langage C++

## Programme

### Introduction à la STL

*Présentation des éléments de base de la STL.*

- La classe string
- Les itérateurs
- Les conteneurs associatifs
- Les conteneurs séquentiels

### Le typage dynamique

*Présentation du typage dynamique avec RTTI.*

- Avantages et inconvénients du typage dynamique
- La conversion de type dynamique
- L'opérateur d'identification de type

### Les smart pointers

*Présentation des smart pointers et de leur utilisation.*

- Qu'est-ce qu'un smart pointer ?
- Avantages et inconvénients
- Conversion implicite

### Compatibilité entre C et C++

*Présentation des éléments de code qui ne sont pas compatibles entre C et C++.*

- Le code C incompatible avec le C++

### La partie utilities de la STL

*Dans STL, utilities offre de nombreux outils qui permettent de simplifier la mise au point d'applications C++.*

- La classe paire
- Les pointeurs automatiques
- Les fonctions auxiliaires

### Les fonctions Objet de la STL

*Présentation des fonctions Objet (souvents appelées functors) de la STL.*

- Présentation des fonctions Objet
- Utilisation des fonctions Objet

### Les composants numériques de la STL

*Présentation de l'utilisation des composants numériques (nombres complexes ...).*

- Les nombres complexes
- La classe valarray
- Les fonctions numériques

### Les allocators

*Présentation des allocators et de leur utilité.*

- Rôle des allocators
- L'allocator par défaut
- Les allocators utilisateurs

### La librairie Boost

*Présentation de la librairie Boost.*

- Présentation de la meta-programmation
- Calcul et optimisation de code
- Génération automatique de code

# Filière C#

## Les formations proposées

- **CP-CS01** Les bases du langage C#

# Les bases du langage C#

**Code** CP-CS01

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Le langage C# est le langage de programmation orienté Objet le plus complet de la plateforme .NET. Simple et élégant, il est à la fois efficace et facile d'accès et peut donc être assimilé par tous. L'usage d'IDE dédié (Visual Studio 2008 pour ne citer que le principal) vient simplifier encore plus les développements avec ce langage.

## Participants

Ce cours s'adresse aux développeurs désireux de s'initier au langage C#.

## Pré-requis

- La connaissance des concepts Objet est nécessaire.
- La connaissance d'un langage Objet (C++, Java, ...) est un plus, mais n'est pas obligatoire.

## Programme

### Syntaxe du langage C#

- Expressions, types et variables
- Boucles et conditions
- Espaces de nommage

### Principes Objet appliqués en C#

- Définition d'une classe
- Héritage et polymorphisme
- Interfaces

### Notions avancées du langage C#

- Properties, Indexeurs, Structures
- Evènements, Delegates
- Enumérations, Attributs

### Autres principes Objet

- Le contrôle du polymorphisme
- Héritage multiple

### Gestion des exceptions en C#

- Principe des exceptions
- Le bloc try / catch
- Le bloc finally

### Entrées / Sorties en C#

- Opérations sur les fichiers
- Lire et écrire dans des fichiers

### C# ADO.NET

*Apprendre à utiliser l'APi ADO.NET pour accéder à une base de données en C#.*

- Accéder à une base de données
- Mode connecté / déconnecté
- Utiliser les commandes paramétrées
- Invoquer des procédures stockées

### Introduction à LINQ

*Language Integrated Query (LINQ) est un composant du framework .NET ajoutant un support natif du système de requêtage à tous les langages .NET.*

- Présentation de LINQ
- Fournisseurs LINQ

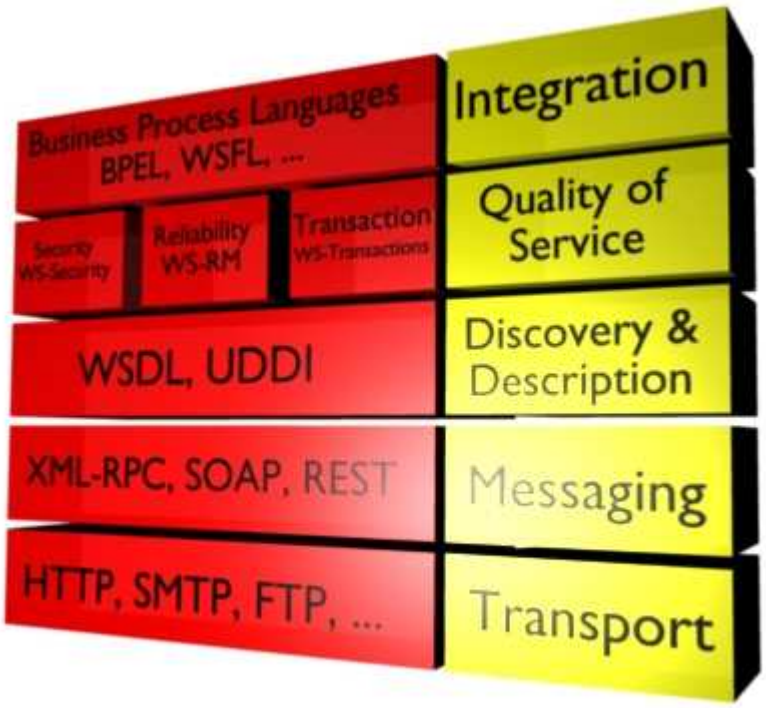
### Introduction aux WinForms

*Présentation de la librairie de développement de clients lours en .NET*

- Présentation de WinForm
- Quelques éléments de programmation
- Intégration dans Visual Studio

# Filière Web Services

Les Web Services sont la solution aux problématiques d'intégration entre systèmes hétérogènes La filière Web Service propose des cours de présentation et de mise en oeuvre de Web Services dans les technologies Java. On y parlera aussi bien des Web Services W3C (SOAP + WSDL + WS-X) que des services dits RESTful.




*La pile des technologies Web Service*

## Les formations proposées

- **CP-SOAP** Développer des Web Services SOAP en Java
- **CP-REST** Développer des services RESTful en Java

# Développer des Web Services SOAP en Java

**Code** CP-SOAP

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Les organismes de standardisation W3C, OASIS et WS-I ont défini un ensemble de spécifications permettant de standardiser les échanges de messages XML entre systèmes informatiques. Il s'agit des spécifications définissant les Web Services à base de messages SOAP. De tels Web Services permettent de faire communiquer entre eux des systèmes jusque là incompatibles. La mise en oeuvre de tels services en Java a été standardisée dans l'API JAX-WS. Ce cours est un tour d'horizon des spécifications SOAP, WSDL et JAX-WS.

## Participants

Ce cours s'adresse aux développeurs ou architectes souhaitant participer au développement de Web Services SOAP en Java.

## Pré-requis

- La connaissance préalable du langage Java est nécessaire.
- Une expérience préalable de XML est un plus, un rappel des concepts est cependant au programme.
- Une expérience préalable de l'API JAXB est recommandée.

## Programme

### Introduction aux Web Services

*Définition des Web Services W3C, intérêt supposé et présentation des standards sous-jacents.*

- Les problématiques d'interopérabilité
- La solution apportée par les Web Services
- SOAP et WSDL

### Rappels : les bases de XML

*Rappels sur les principes de base des langages à balises, les règles syntaxiques, la validation.*

- Introduction à XML
- Les règles syntaxiques de XML
- La validation par schéma
- Les espaces de nommage XML

### SOAP

*Présentation du format des messages SOAP, utilisé dans les Web Services.*

- Structure de base d'un message SOAP
- Les balises Enveloppe, Header, Body et Fault
- RPC vs Document
- Soap encoded vs Literal
- Les recommandations du WS-I

### WSDL

*Le format WSDL est le format standard de description d'un Service Web.*

- Structure d'une description WSDL
- Description abstraite d'un service
- Description concrète d'un service
- Les recommandations du WS-I

### JAX-WS

*Présentation de l'API JAX-WS, API standard de développement de Web Services en Java.*

- Présentation de JAX-WS
- Présentation de JAXB
- Possibilités offertes par l'API
- Les annotations
- Implémentations de la spécification

### Web Service en Java

*Mise en oeuvre de la spécification JAX-WS pour développer des services et des clients.*

- Développer un service depuis son implémentation
- Personnalisation du bindign XML / Java
- Générer un client depuis une description WSDL

### Les Handlers

*Un Handler est un composant coté client ou coté service chargé d'intercepter et de traiter des requêtes et/ou des réponses.*

- Les différents types de Handlers avec JAX-WS
- Développer un Handler
- Configurer et déclarer une chaîne de Handlers avec JAX-WS

### l'API Provider

*L'API Provider, incluse dans JAX-WS permet de développer des services Web traitant directement les messages en XML.*

- Présentation de l'API
- Les différents types de Provider
- Configuration et lancement d'un provider

### l'API Dispatch

*L'API Dispatch, incluse dans JAX-WS, permet de développer des clients de Service Web traitant directement les messages en XML.*

- Présentation de l'API
- Développer un client Dispatch

# Développer des services RESTful en Java

**Code** CP-REST

**Durée** 3 jours

**Tarif** 990 € HT

## Description

REST est aujourd'hui de plus en plus utilisé pour la réalisation d'architectures orientées services dans laquelle des systèmes hétérogènes nécessitent de communiquer entre eux par échange de messages sur HTTP. Plus qu'une alternative à SOAP, il s'agit d'une autre façon, complémentaire, de voir la communication entre applications. La spécification JAX-RS (JSR 311) décrit un standard de développement de services REST en Java.

## Participants

Ce cours s'adresse aux responsables de projet, développeurs et architectes désirant comprendre ce qu'est REST et la spécification relative à sa mise en œuvre en Java.

## Pré-requis

- Connaissances de base des applications Internet à base de Servlets
- Maîtrise du langage Java5

## Programme

### Présentation des concepts

*Présentation des concepts REST, tel qu'ils ont été décrits par Roy Fielding.*

- Representational State Transfer
- Intégration de REST dans une architecture informatique

### Le protocole HTTP

*Le fonctionnement de REST est, en partie, calqué sur les possibilités offertes par le protocole HTTP.*

- Historique de HTTP
- Structure des messages HTTP
- Les méthodes HTTP
- Sécurisation d'une communication HTTP

### JAX-RS

*JAX-RS est la spécification Java qui standardise le développement de services dits RESTful.*

- Java API for RESTful Web Services
- Classe ressource
- Les annotations JAX-RS
- L'URI pour identifier les ressources
- Le passage de paramètres

### Glassfish Jersey

*Jersey est un sous projet de la communauté Glassfish et la principale implémentation de JAX-RS.*

- Présentation de Jersey
- Intégration dans un environnement de développement
- Développement d'une classe ressource
- Déploiement
- La sécurité des services avec Jersey

### WADL

*Web Application Description Language est le standard de description d'une application web, et est parfaitement adapté à la description de services RESTful.*

- Présentation du formalisme XML
- Aperçu de la spécification

### Invocation d'un service RESTful en Java

*Comment développer un client ? Quelle API utiliser ?*

- Récupération d'une ressource en utilisant uniquement JavaSE
- L'API client de Jersey
- Jakarta Commons HTTP Client

### JSON

*JavaScript Object Notation est un standard de description d'objets, facilement exploitable en JavaScript.*

- Usage des flux JSON dans une architecture RESTful
- Formalisme
- JSON et Java

### Traitements des documents XML en Java

*Aperçu des techniques de manipulation des documents XML en Java, en vue de traiter des messages XML en REST.*

- Rappels sur XML : balises, validations, ...
- Parsing DOM, SAX et Stax
- JAXB
- XML dans REST

### Eléments d'architecture

*Place des services RESTful dans une architecture globale.*

- Place des services RESTful dans une architecture multi-niveaux
- REST comme couche d'accès aux données
- REST comme couche d'objets métiers
- REST en tant que services frontaux

# Filière UML et l'Objet

Depuis maintenant 10 ans, UML et les concepts Objet ont pris une place de plus en plus importante au sein du Système d'Information de l'entreprise. Hier cantoné aux développeurs et architectes, l'usage de UML commence aujourd'hui a apparaître également chez les Maîtrise d'Ouvrage. Cette filière permet à tous de s'initier au langage UML et aux concepts Objets sous-jacents.

## Les formations proposées

- **CP-OBJT** Les concepts Objet
- **CP-UML1** Savoir modéliser avec UML
- **CP-PATT** Design Patterns et éléments de conception

# Les concepts Objet

**Code** CP-OBJT

**Durée** 3 jours

**Tarif** 990 € HT

## Description

L'usage des langages Objet et de la modélisation avec UML est de plus en plus répandu dans les entreprises. Si la syntaxe d'un langage ou le formalisme UML s'apprennent relativement facilement, les concepts sous-jacents (les concepts de l'orienté Objet) sont peut être plus difficiles à appréhender. Cette formation se donne pour objectif de clarifier les concepts tels que l'Objet, la classe, l'encapsulation, l'héritage ou le polymorphisme. Les langages Objet ainsi que UML y sont également présentés.

## Participants

Ce cours s'adresse aux développeurs, architectes, chefs de projet ou tout autre personne ayant à participer à un projet manipulant des concepts Objet (en phase d'analyse, de conception ou de développement).

## Pré-requis

- Connaissances de base en conception d'applications et en développement logiciels

## Programme

### Paradigme de programmation orientée Objet

*Rappel des différentes approches de programmation*

- Une taxonomie des principaux paradigmes de programmation
- L'historique des langages orientés Objet

### Les concepts de base du modèle Objet

*Les notions de base et la présentation des principes qui, si ils sont suivis, permettent d'aboutir à une "conception efficace". Présentation des critères d'efficacité d'une conception.*

- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme
- Principes fondamentaux en conception Objet
- Erreurs à éviter

### Les concepts avancés du modèle Objet

*Notions plus avancées en programmation orientée Objet.*

- Classes et méthodes abstraites
- Interfaces
- Généricité

### Méthodologie orientée Objet

*Différences entre le formalisme et la méthode.*

- La modélisation, l'apport de modèles, différences entre la modélisation fonctionnelle et Objet
- Processus de développement et cycles de vie de logiciels (processus unifié: RUP, 2TUP, ...)
- Langages de modélisation

### Modélisation Objet

*Historique et philosophie d'UML. UML et le processus de développement Objet.*

- L'historique d'UML, modes d'utilisation, outillage
- Présentation générale d'UML 2: les différents types de diagrammes, différences entre diagrammes statiques et dynamiques
- La place d'UML dans le processus de développement
- Stéréotypes et extensions d'UML

### UML (Unified Modeling Language)

*Présentation du langage*

- Diagrammes principaux
- De la modèlisation à l'implémentation: comment passer de modèle UML à code

### Langages de développement ObjetInitiation aux langages Java et C#

- Caractéristiques des langages de programmation Objet
- Java et C#: présentation générale, les concepts Objet avec ces langages et quelques exemples

### Design Patterns

*Qu'appelle-t-on Design Patterns ? A quoi servent ils ? Comment se présentent ils ?*

- Objectifs des Design Patterns
- Formalisme
- Catégories des patterns GOF
- Les autres patterns

### Les plates-formes Objet actuelles et les composants

*Présentation des plates-formes Objet et les technologies .NET et J2EE*

- Définitions: plate-forme, composant, librairies de classes
- Technologies .NET de Microsoft et J2EE de SUN.
- Comparaison de leurs points forts et leurs points faibles

# Savoir modéliser avec UML

**Code** CP-UML1

**Durée** 3 jours

**Tarif** 990 € HT

## Description

Ce cours permet de préciser la place de la modélisation UML, et plus particulièrement ses divers diagrammes, dans le processus de développement d'une application. Ce cours s'adresse aussi bien aux concepteurs désireux de formaliser leur modèle Objet en UML, qu'aux responsables MOA souhaitant communiquer avec leur Maîtrise d'Oeuvre en utilisant un langage commun.

## Participants

Développeurs, concepteurs, analystes, architectes, chefs de projet, responsables MOA désireux de découvrir ou d'approfondir leurs connaissances de UML.

## Pré-requis

- Aucun pré-requis pour ce cours. La connaissance préalable des concepts Objet est cependant un plus.

## Programme

### Rappels : les concepts Objet

*Rappel des concepts Objet, base de toute conception efficace.*

- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme

### Introduction à UML

*Historique et philosophie de UML. Différence entre le formalisme et la méthode.*

- La formalisation, l'apport du modèle, différence entre la modélisation fonctionnelle et Objet, les langages de modélisation
- L'historique d'UML, modes d'utilisation, outillage
- Présentation générale de UML 2 : les diagrammes, modélisation statique et dynamique
- Stéréotypes et l'extension d'UML

### Processus de développement

*La place d'UML dans le processus de développement.*

- Processus de développement et cycle de vie logiciel
- Le Processus Unifié : RUP, 2TUP, ...
- Relations entre 2TUP et UML

### Diagramme de cas d'utilisation

*Formalisme et utilité des diagrammes de cas d'utilisation.*

- Acteurs, Cas d'utilisation, frontières du système
- Relations entre entités dans le Use Case Diagram

### Diagramme de classes, d'objets et de packages

*Formalisme du diagramme de classes. Représentation des relations entre classes.*

- Représentation des classes, attributs, opérations
- Relations entre classes : association, agrégation et héritage
- Visibilité des membres d'une classe
- Classes abstraites et interfaces
- Diagrammes d'objets
- Diagramme de packages

### Diagramme de séquences et diagramme de communication

*Formalisme de présentation d'une séquence d'interaction entre objets.*

- Scénario, ligne de vie d'un objet
- Echange de messages synchrones / asynchrones
- Les cadres UML 2
- Diagramme de communication

### Diagramme d'activités

*Formalisme de présentation d'un workflow ou processus en UML.*

- Notion d'action, d'activité
- Les couloirs d'activité

### Diagramme d'états transitions

*Formalisme de présentation du changement de l'état d'un objet.*

- Etats et transitions
- Gardes, actions

### Diagramme de structure composite

*Formalisme de présentation de l'intérieur d'un système en UML.*

- Notions fondamentales du formalisme

### Diagramme de composants / déploiement

*Représentation d'une architecture en UML.*

- Diagramme de composants
- Diagramme de déploiement

# Design Patterns et éléments de conception

**Code** CP-PATT

**Durée** 3 jours

**Tarif** 990 € HT

## Description

La pratique de la conception Objet n'est pas chose aisée. Une fois compris les principes fondateurs de l'Objet, on réalise que leur connaissance n'est pas suffisante et que quelques règles de bonnes pratiques seraient les bienvenues lors d'une conception. Le but de ce cours est de présenter ces bonnes pratiques. Il s'agit aussi bien de Design Patterns (plus particulièrement les patterns GOF) que des principes généraux de conception.

## Participants

Ce cours s'adresse aux développeurs, concepteurs ou architectes désireux d'approfondir leurs connaissances de la conception Objet.

## Pré-requis

- Une bonne maîtrise des concepts Objet est nécessaire
- Une expérience de la conception Objet est fortement recommandée

## Programme

### Rappels : les concepts Objet

*Rappel des concepts Objet, base de toute conception efficace.*

- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme

### Principes fondamentaux de conception

*Présentation des principes qui, si ils sont suivis, permettent d'aboutir à une "conception efficace". Présentation des critères d'efficacité d'une conception.*

- Objectifs de la conception efficace
- Principe d'ouverture / fermeture
- Principe de substitution de Liskov
- Principe de connaissance minimale

### Principe d'organisation en package

*L'élément de base de la réutilisabilité est le package. De cela découle un ensemble de principes.*

- Principe d'équivalence / réutilisation
- Principe de réutilisation commune
- Principe de fermeture commune
- Principe des dépendances acycliques
- Principe de relation dépendance / stabilité
- Principe de stabilité des abstractions

### Principes de construction des classes

*Ensemble de principes permettant de déterminer quelles classes créer et les relations à créer entre elles pour obtenir une conception efficace.*

- Principe d'inversion de dépendance
- Principe de séparation des interfaces
- Principes de GRASP

### Design Patterns

*Qu'appelle-t-on Design Patterns ? A quoi servent ils ? Comment se présentent ils ?*

- Objectifs des Design Patterns
- Formalisme
- Catégories des patterns GOF
- Les autres patterns

### Patterns de création

*Revue des Design Pattern GOF de type "Création".*

- Factory, Singleton
- Prototype, Builder, AbstractFactory

### Patterns structurels

*Revue des Design Pattern GOF de type "Structure".*

- Bridge, Composite, Facade, Decorator
- Proxy, Adapter, Flyweight

### Patterns comportementaux

*Revue des Design Pattern GOF de type "Comportement".*

- Iterator, Observer, State
- Template Method, Strategy, Mediator, Chain of responsability
- Commande, Visitor, Interpreter

# Calendrier des sessions

## Cours du samedi

Il est parfois difficile de prendre le temps de consommer les heures de formation acquises au titre du DIF.

Leuville Objects lance une formule innovante consistant à répartir les trois journées d'un cours sur trois samedi successifs. Cette formule permet de concilier la nécessité de se former avec les contraintes de planification de vos projets, le tout à un tarif très intéressant.

- **CP-UML1** : 10 octobre - 17 octobre - 24 octobre 2009
- **CP-JAVA1** : 21 novembre - 28 novembre - 5 décembre 2009

## Calendrier des sessions inter-entreprises

| | Sep 2009 | Oct 2009 | Nov 2009 | Dec 2009 |
|---|---|---|---|---|
| CP-JAVA1 | 21/09 - 23/09 | | | |
| CP-SWING | 28/09 - 30/09 | | | |
| CP-JXML | | 14/10 - 16/10 | | |
| CP-WJSP | | 14/10 - 16/10 | | |
| CP-WJSF | | 19/10 - 21/10 | | |
| CP-STR1 | | 26/10 - 28/10 | | |
| CP-STR2 | | | 30/11 - 02/12 | |
| CP-EJB3 | | 26/10 - 28/10 | | |
| CP-CPP1 | 21/09 - 23/09 | | | |
| CP-CPP2 | | 19/10 - 21/10 | | |
| CP-CS01 | | | 02/11 - 04/11 | |
| CP-SOAP | 16/09 - 18/09 | | | |
| CP-REST | | | 02/11 - 04/11 | |
| CP-OBJT | 16/09 - 18/09 | | | |
| CP-UML1 | 28/09 - 30/09 | | | |
| CP-PATT | | | 30/11 - 02/12 | |