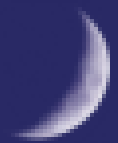


Catalogue de formations

UML, Java, .NET

janvier 2008

sewatech



UML 2, Analyse et conception	1
UML pour chefs de projets	3
UML pour développeurs	4
Bonnes Pratiques Java EE	5
Programmation orientée objet	6
Principes et conception Objet	7
Architecture Objet	8
Conception Objet et Design Patterns	9
Conception Java EE et Design Patterns	10
–	11
Initiation java avec Eclipse	12
Approfondissement java	14
Passer à java 5	16
Tests unitaires et refactoring	17
Tests unitaires pour JavaEE	18
JavaEE pour le Web - JSP et servlet	19
EJB 2, les bases	20
EJB 3, les bases	22
Mapping O/R avec EJB 3 et JPA	24
Framework Spring 2.0 / 2.5	25
Framework MVC - JSF	27
Framework MVC - Struts	28
Mapping O/R avec Hibernate	30
Web 2.0 avec AJAX et javascript	31
Développement XML en java	32
Initiation à .NET et C#	33
Tests unitaires avec .NET	34
–	35
Administration JBoss 4	36
Administration Apache Tomcat	38
–	40
Java : tour d'horizon	41
JavaEE : tour d'horizon	42
Sécurité des applications J2EE	43
Open Source / Java	44
.NET : tour d'horizon	45
Open Source / PHP	46
Architectures des applications Web	47
	48

UML, la notation standard de l'OMG, s'est imposée depuis plusieurs années pour la modélisation de systèmes informatiques. Elle permet de spécifier les éléments d'une application tout au long de la démarche, depuis les étapes les plus fonctionnelles jusqu'à la conception des éléments les plus techniques.

Ce cours vous permet d'acquérir les connaissances nécessaires à l'utilisation d'UML en expression des besoins et en analyse et d'appréhender la modélisation UML en conception.

Durée : 4 jours

Prix (intra) : 4600€

+ € par participant

Participants

Chefs de projets, architectes fonctionnels et techniques, analystes et concepteurs ayant quelques notions objet

Travaux pratiques

Une étude de cas est réalisée tout au long de la formation. Elle permet aux stagiaires de mettre en pratique les éléments de modélisation nécessaires à chaque étape du projet.

Les travaux pratiques sont réalisés avec l'outil Jude Community. Sur demande, d'autres outils de modélisation peuvent être utilisés (outils Rational, EclipseUML, Poseidon, Objecteering,...).

Programme de la formation

Approche Objet

- > Les origines de l'orienté objet
- > Les principes directeurs de l'objet
- > Le principe d'unicité et la définition d'un objet
- > Le principe d'abstraction et l'identification des classes
- > Le principe d'organisation avec l'héritage et la délégation
- > Le principe d'encapsulation

Introduction à UML

- > La démarche de projet
- > La modélisation objet
- > Les principes de mise en œuvre d'UML

Modélisation du comportement

- > Introduction
- > Le diagramme d'activités métier
- > Le diagramme de cas d'utilisation
- > Le détail des cas d'utilisation
- > La structuration du modèle de cas d'utilisation
- > L'expression des besoins non-fonctionnels

Modélisation de la structure

- > Introduction
- > Le diagramme de classes
- > Le diagramme d'objet
- > L'organisation du modèle de classes
- > Les classes de conception et leur projection en java
- > La notion d'interface
- > Le diagramme d'états-transitions
- > Le diagramme de composants
- > Le diagramme de déploiement

Modélisation des interactions

- > La réalisation de cas d'utilisation
- > Le diagramme de séquence
- > Le diagramme de communication

Finalisation du système et compléments de conception

- > La persistance des données
- > Introduction aux Design Patterns
- > Les outils de modélisation UML
- > Le démarche pour UML

- > Le Processus Unifié (UP ou Unified Process)
- > Rational Unified Process (RUP)
- > Les méthodes agile : eXtreme Programming (XP) et SCRUM
- > Conclusion

UML, la notation standard de l'OMG, s'est imposée depuis plusieurs années pour la modélisation de systèmes informatiques. Elle permet de spécifier les éléments d'une application tout au long de la démarche, depuis les étapes les plus fonctionnelles jusqu'à la conception des éléments les plus techniques.

Ce cours vous permet de comprendre les principaux diagrammes UML, en particulier dans le cadre de l'expression des besoins et de l'analyse.

Durée : 2 jours

Prix (intra) : 2200€

+ € par participant

Participants

Chefs de projets, architectes fonctionnels

Travaux pratiques

Une étude de cas est parcourue tout au long de la formation. Elle permet aux stagiaires de visualiser les éléments de modélisation pour chaque étape du projet.

Programme de la formation

Présentation d'UML

- > Intérêt de la modélisation Objet
- > Origines d'UML
- > Principales notions Objet

UML dans les méthodes objet

- > Différentes approches. Intégration d'UML
- > Introduction au Unified Process (UP), itérations, phases et activités
- > Déclinaison de UP : Rational Unified Process (RUP)
- > Autres possibilités (XP, Scrum,...).

Expression des besoins

- > Modéliser les besoins avec les diagrammes de cas d'utilisation
- > Affiner les diagrammes de cas d'utilisation avec les relations d'extension et d'inclusion
- > Structurer le modèle de besoins à l'aide de paquetages
- > Détailler les scénarii

Modélisation de la structure

- > Diagramme de classes
- > Identifier les classes d'analyse utiles
- > Détailler les classes : attributs, opérations, visibilité, associations, rôles, multiplicité, généralisation
- > Diagramme d'objets
- > Organisation du modèle
- > Diagrammes de composants et de déploiement

Modélisation des interactions

- > Réalisation de cas d'utilisation
- > Diagramme de séquence
- > Communication par message entre objets
- > Diagramme de communication / collaboration

Modélisation des comportements

- > Modèle d'activité métier
- > Contextes d'utilisation des diagrammes d'activité
- > Diagramme d'états-transitions

Conclusion

- > Quel niveau de détail donner au modèle ?
- > MDA et génération de code
- > Quels outils adaptés à chaque démarche ?

UML, la notation standard de l'OMG, s'est imposée depuis plusieurs années pour la modélisation de systèmes informatiques. Elle permet de spécifier les éléments d'une application tout au long de la démarche, depuis les étapes les plus fonctionnelles jusqu'à la conception des éléments les plus techniques.

Ce cours vous permet de comprendre les principaux diagrammes UML, en particulier dans leurs aspects techniques, liés à la conception et au développement.

Durée : 2 jours

Prix (intra) : 2200€

+ € par participant

Participants

Développeurs et concepteurs Objet

Travaux pratiques

Des exercices pratiques de conception, à partir d'une analyse pré-établie, ainsi que des exercices de rétro-conception seront réalisés. Nous utiliserons les outils StarUML, BoUML et Eclipse.

Programme

Introduction à UML

- > Démarche projet
- > Modélisation objet
- > Mise en œuvre d'UML

Expression de besoins et analyse

- > Introduction
- > Diagramme de cas d'utilisation
- > Diagramme de séquence
- > Diagramme de classes
- > Diagramme d'objets
- > Diagramme d'états-transitions
- > Organisation du modèle

Conception

- > Classes de conception
- > Interface
- > Diagramme de séquence
- > Diagramme de communication

Finalisation du système et compléments

- > Structuration du système
- > Persistance des données
- > Conception des écrans
- > Design Pattern
- > Outils de modélisation UML
- > Démarche
- > Processus Unifié
- > Rational Unified Process
- > eXtreme Programming

Conclusion

- > Conclusion
- > Références

Le développement java, et java EE (ou J2EE) recèle de nombreux pièges qui peuvent avoir des conséquences diverses : défaut de performance, productivité réduite, difficultés de maintenance,...

Cette formation permet de parcourir les principales bonnes pratiques permettant d'éviter ces écueils dans vos projets java EE.

Durée : 3 jours

Prix (intra) : 3450€

+ 50€ par participant

Participants

Développeurs, concepteurs et chefs de projets pratiquant le développement ou la conception non objet

Déroulement du stage

Cette formation alterne théorie et pratique. Des exercices pratiques permettent d'illustrer les sujets les plus concrets du cours. Ils sont réalisés en java avec le JDK 5.0 de Sun et les outils open source Eclipse 3.

Programme

Les bonnes pratiques de conception

- > Les enjeux de la conception
- > La conception avec ou sans UML
- > La réutilisation : techniques et limites
- > Le rôle des interfaces et classes abstraites dans la stratégies d'évolution d'un système
- > L'organisation du sous-systèmes ou modules
- > La gestion de l'évolutivité par les dépendances
- > Le rôle du paquetage dans la conception
- > La notion de responsabilité dans l'organisation du système
- > Les design patterns pour résoudre les problèmes de conception récurrents

Les bonnes architectures pour Java EE

- > L'importance de l'architecture dans la conception
- > L'architecture multi-couches pour orienter le graphe de dépendances
- > Les design patterns dans l'architecture
- > Les technologies Java EE dans l'architecture
- > Les frameworks Java EE

Les bonnes pratiques de développement

- > Les techniques pour économiser la mémoire (instanciation, pool et cache)
- > Les transactions
- > La sécurité

Les outils pour bien développer

- > Améliorer la productivité individuelle (eclipse, ant)
- > Améliorer la productivité de l'équipe (eclipse, subversion, maven)
- > Préparer l'exploitation avec de bonnes traces (Apache Log4J et Common Logging ou SLF4J)
- > Suivre la mémoire (jconsole, profiling)

Le suivi de la qualité

- > Les différents types de tests
- > La mise en oeuvre des tests unitaires automatisés (junit, jcover)
- > L'automatisation des tests d'intégration
- > Les outils de mesure de la qualité (CodeStyle, PMD, jDepend,...)

Les bonnes démarches de projet

- > L'organisation d'un projet autour d'UML : UP, le processus unifié
- > Centrer un projet sur les modèles : MDA (Model Driven Architecture)
- > Centrer un projet sur l'agilité des développeurs : eXtreme Programming

Cette formation s'adresse à des stagiaires qui n'ont jamais pratiqué l'objet et qui souhaitent apprendre la façon de développer avec un langage objet moderne comme java ou .NET (BV ou C#).

Durée : 2 jours

Prix (intra) : 2200€

+ € par participant

Elle commence par une approche généraliste du raisonnement objet, en introduisant les différents concepts élémentaires et avancés. Elle enchaîne ensuite sur les principes d'architecture qui présentent la façon traditionnelle d'organiser une application en couches. Enfin, à l'aide d'exemples simples manipulés par les stagiaires, la formation met en lumière la façon dont ces concepts sont mis en oeuvre dans un langage comme java.

Participants

Développeurs, concepteurs et chefs de projets souhaitant apprendre la programmation avec un langage objet

Déroulement du stage

Cette formation est alterne théorie et exemples. Des exercices pratiques permettent d'illustrer la dernière partie en java avec le JDK 5.0 de Sun et l'outil open source Eclipse 3.

Programme de la formation

Principes fondamentaux de l'Objet

- > De la programmation structurée à la programmation Objet
- > Qu'est-ce qu'un objet ?
- > Le mécanisme d'unicité et son application à la définition d'un objet
- > Le mécanisme d'abstraction et la définition des classes
- > Le mécanisme de classification pour organiser les classes dans une perspective de réutilisation
- > Les classes abstraites et les interfaces
- > Le mécanisme d'encapsulation pour améliorer la robustesse et l'évolutivité des systèmes

Principes de conception et de réutilisation

- > Les techniques Objet pour la réutilisation : héritage, délégation,...
- > Les limites de l'héritage
- > Les techniques complémentaires pour allier réutilisation et évolutivité : le polymorphisme, les interfaces

Principes d'architecture

- > L'importance de l'architecture dans une conception Objet
- > La place de la base de données dans l'architecture
- > L'enjeu de la gestion des dépendances entre classes et paquetages
- > L'architecture multi-couches : des écrans, des traitements et des données

Programmation objet

- > Les variables et la notation pointée
- > La création, la manipulation et la destruction d'objets
- > Les mécanismes objet avancés avec l'héritage et le polymorphisme
- > Les classes abstraites et les interfaces

Cette formation permet d'acquérir les mécanismes élémentaires du raisonnement objet. Après une présentation des principes fondamentaux, elle expose quelques bonnes pratiques en conception et en architecture.

A l'issue de ce cours, vous serez en mesure d'aborder dans les meilleures conditions l'apprentissage d'un langage de développement Objet (java, .net, php5,...). Vous pourrez aussi vous orienter vers l'analyse et la conception Objet avec UML.

Durée : 2 jours

Prix (intra) : 2200€

+ € par participant

Participants

Développeurs, concepteurs et chefs de projets pratiquant le développement ou la conception non objet

Déroulement du stage

Cette formation est essentiellement théorique. Le formateur fait participer les stagiaires par le biais de questions / réponses et d'exemples issus du monde réel ou de l'informatique.

Des exemples sont donnés tout au long du cours, avec UML, java et .NET.

Programme de la formation

Origines de l'Objet

- > De la programmation structurée à la programmation Objet
- > Évolution de la programmation vers la conception et l'analyse
- > Qualités attendues de l'Objet

Principes fondamentaux de l'Objet

- > Qu'est-ce qu'un objet ?
- > Le mécanisme d'unicité et son application à la définition d'un objet
- > Le mécanisme d'abstraction et la définition des classes
- > Le mécanisme de classification pour organiser les classes dans une perspective de réutilisation
- > Les classes abstraites et les interfaces
- > Le mécanisme d'encapsulation pour améliorer la robustesse et l'évolutivité des systèmes

Principes de conception et de réutilisation

- > Les techniques Objet pour la réutilisation : héritage, délégation,...
- > Les limites de l'héritage
- > Les techniques complémentaires pour allier réutilisation et évolutivité : le polymorphisme, les interfaces

Principes d'architecture

- > L'importance de l'architecture dans une conception Objet
- > L'enjeu de la gestion des dépendances entre classes et paquets
- > Mise oeuvre des bonnes pratiques pour rationaliser les dépendances : introduction aux design patterns
- > Le principe de façade pour organiser un système en modules
- > L'architecture multi-couches pour orienter le graphe de dépendances
- > Les frameworks pour faciliter la mise en application des principes de conception et d'architecture

Processus de développement Objet

- > Introduction à UML : modèle et diagrammes
- > L'organisation d'un projet autour d'UML : UP, le processus unifié
- > Centrer un projet sur les modèles : MDA (Model Driven Architecture)
- > Centrer un projet sur l'agilité des développeurs : eXtreme Programming

Synthèse

- > Risques et perspectives de l'Objet

Objectifs :

- > Comprendre les enjeux d'architecture en développement objet
- > Connaître les règles fondamentales en architecture objet

Durée : 1 jour

Prix (intra) : 1150€
+ 30€ par participant

Participants

Architectes techniques, développeurs et chefs de projets ayant des notions objets

Programme

Isolation des couches

- > Structuration en couches (présentation, métier,...)
- > Structure MVC

Architecture distribuée et serveur d'application

- > Historique des architectures
- > Types d'architectures distribuées
- > Architecture Web
- > Architectures java et .net
- > Serveur d'application

WebServices dans une architecture Objet

- > Définitions
- > Technique

La puissance du concept objet fait de la modélisation logicielle un métier de spécialistes dont seuls l'expérience et les retours terrain alimentent l'expertise.

Cette formation vous permettra d'acquérir une compétence opérationnelle sur le design des applications, et de gagner en productivité grâce à l'utilisation des patterns. Les nombreux cas pratiques vous apprendront à modéliser et réaliser des composants et des applications évolutives et réutilisables, et à comprendre les principaux patterns de conception.

Durée : 2 jours

Prix (intra) : 2300€

+ € par participant

Participants

Architecte, chef de projet, analyste, concepteur, développeur, responsable méthode, connaissant un langage objet, comme Java, C++ ou C#

Travaux pratiques

Les travaux pratiques de modélisation en UML et Java permettront aux participants de manipuler les principes du design et d'identifier les principaux patterns.

Programme de la formation

Présentation du design

- > Rappel des notions fondamentales de la programmation OO et d'UML
- > Les enjeux de la conception : accroître la réutilisation sans freiner les évolutions
- > La réutilisation par l'héritage : avantages et inconvénients

Principes fondamentaux en conception objet

- > La stratégie d'évolution avec le principe d'ouverture/fermeture (OCP)
- > Une réutilisation efficace par l'héritage et les interfaces : le principe de substitution de Liskov (LSP)

Principes d'organisation en packages

- > Le package comme unité de conception avec les principes d'équivalence livraison/réutilisation (REP) et de réutilisation commune (CRP)
- > Le découpage des packages grâce au principe de fermeture commune (CCP)
- > L'organisation entre package : principes des dépendances acycliques (ADP) et de relation dépendance/stabilité (SDP)

Principes de construction des classes

- > La gestion raisonnée des dépendances avec l'inversion de dépendance (DIP)
- > La réduction de la complexité apparente par la séparation des interfaces (ISP)
- > La répartition des responsabilités avec le principe de GRASP

Principes des Design Patterns

- > Origine et portée des patterns
- > Les design patterns comme réponse aux problèmes techniques

Les patterns fondateurs de Gamma et Gof

- > Le catalogue de patterns de la "bande des quatre"
- > Isoler la création des objets de leur utilisation avec les patterns créateurs : fabrique, singleton et prototype
- > Affiner l'affectation des responsabilités grâce aux patterns comportementaux : chaîne de responsabilité, patron de méthode et observateur
- > Améliorer l'organisation des classes avec les patterns de structure : adaptateur, façade et composite

La puissance du concept objet fait de la conception logicielle un métier de spécialistes dont seuls l'expérience et les retours terrain alimentent l'expertise.

Cette formation vous permettra d'acquérir une compétence opérationnelle sur le design des applications JavaEE (J2EE), et de gagner en productivité grâce à l'utilisation des patterns. Les nombreux cas pratiques vous apprendront à modéliser et réaliser des composants et des applications JavaEE évolutives et réutilisables, et à comprendre les principaux patterns de conception.

Durée : 3 jours

Prix (intra) : 3450€

+ € par participant

Participants

Architecte, chef de projet, analyste, concepteur, développeur, connaissant Java et JavaEE (J2EE)

Travaux pratiques

Les travaux pratiques de modélisation en UML et Java, sous Eclipse permettront aux participants de manipuler les principes du design et d'identifier les principaux patterns.

Programme

Présentation du design

- > Rappel des notions fondamentales de la programmation OO et d'UML
- > Les enjeux de la conception : accroître la réutilisation sans freiner les évolutions
- > La réutilisation par l'héritage : avantages et inconvénients

Principes fondamentaux en conception objet

- > La stratégie d'évolution avec le principe d'ouverture/fermeture (OCP)
- > Une réutilisation efficace par l'héritage et les interfaces : le principe de substitution de Liskov (LSP)

Principes d'organisation en packages

- > Le package comme unité de conception avec les principes d'équivalence livraison/réutilisation (REP) et de réutilisation commune (CRP)
- > Le découpage des packages grâce au principe de fermeture commune (CCP)
- > L'organisation entre package : principes des dépendances acycliques (ADP) et de relation dépendance/stabilité (SDP)

Principes de construction des classes

- > La gestion raisonnée des dépendances avec l'inversion de dépendance (DIP)
- > La réduction de la complexité apparente par la séparation des interfaces (ISP)
- > La répartition des responsabilités avec le principe de GRASP

Principes des Design Patterns

- > Origine et portée des patterns
- > Les design patterns comme réponse aux problèmes techniques

Les patterns fondateurs de Gamma et Gof

- > Le catalogue de patterns de la "bande des quatre"
- > Isoler la création des objets de leur utilisation avec les patterns créateurs : fabrique, singleton et prototype
- > Affiner l'affectation des responsabilités grâce aux patterns comportementaux : chaîne de responsabilité, patron de méthode et observateur
- > Améliorer l'organisation des classes avec les patterns de structure : adaptateur, façade et composite

Les patterns JavaEE (J2EE)

- > Rappel sur l'architecture JavaEE
- > Adapter les patterns à l'architecture multi-tiers
- > Optimiser une architecture distribuée avec le Transfer Object et le Session Façade
- > Améliorer l'évolutivité des couches avec le Data Access Object et le Business Delegate
- > Structurer la couche présentation grâce au MVC (Model-View-Controller) : Front Controller et Composite View

La puissance du concept objet fait de la conception logicielle un métier de spécialistes dont seuls l'expérience et les retours terrain alimentent l'expertise.

Cette formation vous permettra d'acquérir une compétence opérationnelle sur le design des applications JavaEE (J2EE), et de gagner en productivité grâce à l'utilisation des patterns. Les nombreux cas pratiques vous apprendront à modéliser et réaliser des composants et des applications JavaEE évolutives et réutilisables, et à comprendre les principaux patterns de conception.

Durée :

Prix (intra) : 0€

+ € par participant

Le développement d'applications Java nécessite avant tout de comprendre le raisonnement Objet ; il demande aussi de connaître les bases du langage, puis d'acquérir les techniques adaptées à l'architecture.

Il n'existe donc pas de cursus Java universel ; chaque équipe doit se constituer un ensemble de formations sur mesure, en fonction des choix des architectes.

Quelle que soit l'architecture dans laquelle vous serez amenés à développer, les bases de java sont les mêmes. Ce cours est le tronc commun nécessaire avant de se lancer dans tout développement java, que ce soit pour des écrans Swing, des pages JSP ou des applications J2EE.

Durée : 4 jours

Prix (intra) : 4200€
+ 50€ par participant

Ce cours vous permettra de connaître les bases du langage Java et de comprendre les concepts objet avec Java. Il vous apprendra à développer des classes Java et vous présentera les principales API.

Ce cours peut être complété par des modules sélectionnés dans le plan de Approfondissement java.

Participants

Développeurs et chefs de projets ayant une connaissance pratique d'un langage de programmation (C, C++, Visual Basic, Pascal,...)

Travaux pratiques

Des exercices pratiques permettent d'illustrer les différents sujets abordés dans le cours.

Ils sont réalisés avec le JDK 5.0 de Sun et les outils open source Eclipse 3 et Hsqldb. Cependant, si vous avez déjà fait des choix d'outils pour vos développements, nous pouvons nous y adapter.

Programme de la formation

Présentation de java

- > 15 ans d'histoire
- > L'environnement java : le JDK, le JRE et la machine virtuelle
- > Les outils de développement du marché : Eclipse, JBuilder,...
- > Les principales caractéristiques du langage

Premiers pas avec java

- > Les constructions de base d'un programme
- > Les types, identificateurs et variables
- > Les instructions conditionnelles et itératives
- > Les opérateurs

Notions Objet en java

- > Les notions élémentaires : développer une classe, détailler les champs et méthodes d'une classe
- > La création, la manipulation et la destruction d'objets : le mécanisme de ramasse-miettes (garbage collector)
- > Les tableaux de valeurs ou d'objets
- > L'organisation du code en packages
- > Les niveaux de visibilité pour les classes et leurs membres
- > Les mécanismes objet avancés avec l'héritage et le polymorphisme
- > Les classes abstraites et les interfaces

Mécanisme d'exceptions

- > Comment gérer les erreurs au sein d'une application ?
- > Le principe de propagation des exceptions
- > Les principales classes d'erreur et d'exception
- > Le traitement des exceptions avec les blocs try-catch-finally
- > La déclaration des exceptions (throws), cas des RuntimeException
- > Développer des classes d'exception personnalisées

Librairies standards du JDK

- > La classe Object
- > Manipulation de chaînes de caractères : classes String, StringBuilder et StringBuffer
- > Les types élémentaires et les enveloppeurs de types primitifs ; la technique du boxing et de l'autoboxing
- > Manipulation de dates et heures
- > Gérer des listes dynamiques avec les collections et les maps ; utilisation des generics

Accès aux bases de données avec JDBC

- > Principes de JDBC : une API commune et un driver spécifique
- > L'architecture de JDBC et les 4 types de drivers
- > Etablir une connexion avec une base

- > Envoyer des requêtes de sélection et lire le résultat dans un ResultSet
- > Executer des requêtes de mise à jour

L'initiation à Java permet de découvrir les bases du langage et de mettre le pied à l'étrier des développeurs amenés à collaborer à un projet. Cette initiation peut être insuffisantes par rapport à certains sujets.

C'est pourquoi nous proposons un ensemble de modules autonomes parmi lesquels vous trouverez des sujets importants et complémentaires à la formation [16]. Ces modules ont été conçus par 1/2 journée ou par journée pour être ajoutés à la formation d'initiation.

Durée : selon sélection
Prix (intra) : 1050€ / jour
+ € par module et par participant

Participants

Développeurs et chefs de projets ayant une première connaissance de java ou suivi le cours Initiation à java

Travaux pratiques

Des exercices pratiques permettent d'illustrer les différents sujets abordés dans le cours.

Ils sont réalisés avec le JDK 5.0 de Sun et les outils open source Eclipse 3 et Hsqldb. Cependant, si vous avez déjà fait des choix d'outils pour vos développements, nous pouvons nous y adapter.

Programme

Rappels et approfondissements

- > Mécanismes de redéfinition et surcharge
- > Développement de classes abstraites et d'interfaces
- > Développement de java beans
- > Gestion de la mémoire et mécanisme de ramasse-miettes

Collections et tableaux

- > Rappel : les principales classes et interfaces
- > La transformation tableaux - collections
- > Les algorithmes de tri
- > Les collection immuables
- > Autres manipulations de collections et de tableaux

Entrée / sorties

- > Les flux et filtres
- > Les classes d'entrées / sorties
- > La sérialisation d'objets
- > La lecture et l'écriture de fichiers
- > L'envoi et réception d'objets via le réseau
- > La compression des flux

Applications multi-threads

- > La classe Thread et l'interface Runnable
- > Les états et le cycle de vie des threads
- > Sémaphores, mutex et sections critiques
- > Gérer la priorité des threads
- > Groupe de threads

Expressions régulières

- > Principe des expressions régulières
- > Eléments de syntaxe : ., *, +, ?, \d, \s, \w, [], ()
- > Manipulation de chaînes de caractères avec le package java.util.regex
- > Formattage de chaînes et de flux avec les classes Formatter et Scanner
- > Utilisation des nouvelles méthodes format et printf de la classe PrintWriter

Internationalisation d'une application Java

- > La norme i18n
- > Les principes d'internationalisation des applications client/serveur et Web
- > La classe « Locale », représentant une culture
- > Adapter le formatage des nombres et dates à une culture
- > La gestion des libellés et messages via un « ResourceBundle »

Introduction à l'API de réflexion Java

- > Le type Class

- > Charger dynamiquement une classe
- > Lire les méta-données d'une classe
- > Invoquer dynamiquement une méthode

JDBC

- > Drivers et connexions aux bases de données
- > Exécution de requêtes et libération de ressources
- > Gestion de transactions
- > Appel de procédures stockées

Programmation graphique avec Swing

- > Présentation des JFC (AWT, swing, java 2D,...)
- > Développer une fenêtre simple (JFrame)
- > Développer des composants graphiques simples (JLabel, JButton,...)
- > La gestion des événements avec les listeners
- > Développer de composants plus complexes (JList avec un ListModel)

Gérer les traces d'une application

- > Principe de Apache Log4J
- > Installer et configurer Log4J
- > Utilisation du framework
- > Utilisation combinée avec Apache commons-logging

"La disponibilité de J2SE 5 est l'annonce la plus importante pour la communauté depuis Java 2", estime Graham Hamilton, Vice Président de Sun et architecte de J2SE 5. Il est vrai que la dernière version du JDK amène des progrès que toute la communauté de développeurs attendait depuis longtemps.

Ainsi, grâce à l'autoboxing, il en fini de la gymnastique entre types primitifs (int, boolean,...) et leurs équivalents objets (Integer, Boolean,...). De même, les itérations simples sur des collections ou des tableaux deviennent effectivement simples, grâce à la nouvelle version de l'instruction 'for'. Enfin, il sera maintenant possible de définir le type des objets contenus dans une collection, grâce aux types génériques ; fini le développement « à l'aveugle ».

Découvrez la facilité de développement obtenue par ces nouvelles fonctionnalités et profitez au plus vite des gains de productivité avec J2SE 5.

Durée : 2 jours
Prix (intra) : 2200€
+ 30€ par participant

Participants

Développeurs et chefs de projets, connaissant les versions antérieures de java
Pour des développeurs ne connaissant pas java, le cours "Initiation à java" est plus adapté.

Travaux pratiques

Des exercices pratiques permettent d'illustrer les différents sujets abordés dans le cours.

Ils sont réalisés avec le JDK 5.0 de Sun et Eclipse 3. Cependant, si vous avez déjà fait des choix d'outils pour vos développements, nous pouvons nous y adapter.

Programme

Introduction

- > Historique
- > Compilateur v5

Nouveaux mécanismes Objet

- > Redéfinition du type de retour
- > Autoboxing
- > Paramètres variables
- > Imports statiques

Nouvelles instructions

- > Instructions itératives
- > Programmation déclarative et méta-données

Nouveaux types

- > Énumérations
- > Generics

Autres nouveautés

- > Entrées / Sorties
- > Concurrence en application multi-threads
- > ...

Synthèse

- > Compatibilité et cohabitation avec l'existant
- > Quand passer à J2SE 5 ?

Les techniques de rectoring et de test unitaire sont particulièrement préconisées en java dans les démarches agiles de type eXtrem Programming.

La première partie de ce cours permet de comprendre la démarche d'amélioration du code préconisée dans l'eXtrem Programming, la démarche « Test Driven » ainsi que les techniques de tests unitaires proposée par le framework standard JUnit. La seconde partie permet de connaître les techniques classiques de refactoring et de savoir mettre en application ces techniques avec JUnit et Eclipse.

Durée : 2 jours

Prix (intra) : 2300€

+ € par participant

Participant

Développeurs et chefs de projets connaissant java et Eclipse

Travaux pratiques

Ce cours est constitué d'environ 50% de travaux pratiques. Il sont réalisés sous windows ou linux, avec le JDK 5.0 et eclipse 3.1

Programme

Principes et démarche

- > Principaux types de test
- > Principe du test unitaire
- > Automatisation des tests unitaires
- > Développement conduit par les Tests

Framework JUnit

- > Présentation et caractéristiques
- > Écriture d'un test simple
- > Assertions, échecs et erreurs
- > Mock Objects
- > Extension du framework

Introduction au refactoring

- > Définitions
- > Principes
- > Démarche

Refactoring dans une classe

- > Problèmes de dimension
- > Problèmes de nommage
- > Complexité inutile
- > Duplication
- > Logique conditionnelle

Refactoring entre classes

- > Héritage
- > Responsabilité
- > Modifications de code
- > Bibliothèques

Tests unitaires pour JavaEE

sewatech 

Cette formation permet de comprendre les enjeux et les techniques des tests unitaires, avec la mise en oeuvre de JUnit et d'outils complémentaires, comme les Mock Objects ou ANT, pour l'automatisation

Durée : 2 jours

Prix (intra) : 2100€

+ € par participant

Participants

Développeurs, architectes et chefs de projets connaissant le langage java.

Travaux Pratiques

Chaque technique est illustrée à l'aide d'exercices pratiques. Ceux-ci sont réalisés avec Eclipse et JUnit.

Programme

Principes et démarche

- > Enjeux
- > Types de tests
- > Démarche
- > Processus Unifié
- > Rational Unified Process
- > eXtrem Programming

Bases du framework JUnit

- > Présentation des tests unitaires
- > Framework JUnit
- > Cas de test
- > Initialisation et finalisation
- > Réutilisation
- > Suite de tests
- > Bonnes pratiques

Techniques avancé avec JUnit

- > Unité des tests
- > Objets de leurre
- > Automatisation des tests
- > Extensions à JUnit

Tests unitaires en architecture Java EE

- > Architecture Java EE
- > Tests d'EJB 3
- > Tests avec le framework Spring
- > Tests des classes persistantes d'Hibernate

Le développement d'applications Web en java se fait à partir des techniques de servlets et de JSP.

Ce cours vous permettra de réaliser des applications Web en java. Il vous apprendra à développer des servlets et des JSP et vous présentera les techniques de développement adaptées à l'architecture MVC.

Durée : 3 jours

Prix (intra) : 3300€

+ 50€ par participant

Participants

Développeurs et chefs de projets connaissant java (cf. Initiation à java)

Travaux pratiques

Des exercices pratiques permettent d'illustrer les différents principes abordés dans le cours.

Ils sont habituellement réalisés avec le JDK 5.0 de Sun et les outils open source Eclipse, Apache Tomcat et Hsqldb. Si vous avez déjà fait des choix d'outils pour vos développements, nous pouvons nous y adapter.

Programme

Architecture Web

- > Principes du Web : requêtes http, URL et pages HTML
- > Techniques de Web dynamique
- > Java Enterprise Edition (JavaEE) et les serveurs d'applications

Premiers pas

- > Servlets : classe HttpServlet, méthodes service, doGet et doPost
- > Cycle de vie d'une servlet : méthodes init et destroy
- > Requête et réponse http: HttpServletRequest, HttpServletResponse
- > Descripteur de déploiement : web.xml
- > JSP : intégrer du code au HTML avec des scriptlets, des déclarations et des expressions

Architecture MVC

- > Limites des servlets et des JSP
- > Principe de l'architecture MVC : Model-View-Controller

Développement des servlets

- > Délégation et redirection de requête : RequestDispatcher
- > Contexte, session

Principes des JSP

- > Les objets implicites : request, session, out,...
- > Les actions standards : jsp :useBean, jsp :getProperty,...

Gestion des erreurs

- > Les erreurs standards http (404, 403, 500,...)
- > Les gestion des exceptions au sein des JSP avec la directive page
- > La gestion déclarative des exceptions dans web.xml

Accès aux bases de données

- > JDBC et Datasource
- > Les pools de connexions et le contexte JNDI
- > La problématique des transactions

Librairies de balises

- > Utiliser des librairies de balises
- > Développer des balises personnalisées
- > Les librairies standards (JSTL) et le langage d'expression
- > Utilisation de la librairie core

Déploiement d'une application

- > Configuration avec le descripteur de déploiement web.xml
- > Structure d'une application

Les EJB représentent la solution standard pour développer des applications distribuées avec java et J2EE. Ils sont utiles pour le développement de composants métiers accessibles en RMI, la technique classique de java, en JMS, par messages asynchrones, ou en Web Service.

Durée : 3 jours

Prix (intra) : 3300€

+ € par participant

Ce cours vous permettra de comprendre le modèle de composants métiers de J2EE et vous apprendra à développer des composants EJB, session, entité ou MDB.

Participants

Développeurs, architectes et chefs de projets connaissant et pratiquant java et ayant des notions d'XML

Travaux pratiques

Les travaux pratiques représentent 50% de la formation. Ils sont réalisés sous Eclipse, avec le serveur d'application JBoss.

Sur demande, ils peuvent être réalisés avec d'autres outils de développement du marché, comme Borland JBuilder ou Rational Application Developer, et d'autres serveurs d'applications, comme BEA Weblogic ou IBM Websphere.

Programme

Introduction

- > Les architectures multi-tiers et les architectures distribuées
- > J2EE : les concepts, les composants et les services
- > Les principes des Enterprise Java Beans
- > Les différents types d'EJB et leur utilisation dans l'architecture

Premiers pas avec les Session Beans

- > La mise en oeuvre d'EJB stateless et stateful — Développement de la classe Bean et des interfaces Home et Remote — Configuration des EJB dans le descripteur de déploiement ejb-jar.xml — Le déploiement d'un EJB
- > La mise en oeuvre Client — Interrogation de l'annuaire JNDI — Invocation à distances des EJB
- > Techniques de développements : xdoclet

Développement de Session Beans

- > Les méthodes de callback : ejbCreate, ejbRemove,...
- > Le cycle de vie et le pool d'instances
- > Les interfaces locales et l'invocation locale des EJB
- > L'accès à une base de donnée via une datasource

Transactions

- > Principe des transactions distribuées : commit à deux phases
- > JTA : Java Transaction API
- > Gestion déclarative et attributs transactionnels
- > Gestion programmée des transactions

Sécurité

- > Principe de JAAS (Java Authentication and Authorisation Service)
- > Les rôles et les permissions sur les méthodes
- > Gestion déclarative de la sécurité d'accès à un EJB
- > Gestion programmée de la sécurité

Développement d'Entity Beans

- > Principe du mapping objet/relationnel
- > Gestion manuelle de la persistance : BMP
- > Gestion paramétrée de la persistance : CMP
- > Cycle de vie
- > Langage de requêtes EJB-QL
- > Mise en oeuvre des relations entre EJB entités

Développement de Message Driven Beans

- > Principe des MOM (Middlewares Orientés Message)
- > Mise en oeuvre avec JMS (Java Message Service)
- > Modèles de messages Publish/Suscribe et Point-To-Point

- > Développement d'un EJB message-driven
- > Développement d'un client de MDB

Synthèse

- > Intégration des EJB dans l'architecture J2EE
- > Les alternatives aux EJB
- > L'évolution des EJB : EJB 3

L'architecture EJB 3 nous apporte enfin la facilité de développement de composants métier. Les EJB représentent la solution standard pour développer des applications distribuées avec java et JavaEE, que ce soit avec les techniques traditionnelles telles que RMI, par messages asynchrones JMS, ou en Web Service.

Ce cours vous permettra de comprendre le modèle de composants métiers de JavaEE et vous apprendra à développer des composants EJB3, session, entité ou MDB.

Durée : 3 jours

Prix (intra) : 3450€

+ 50€ par participant

Participants

Développeurs, architectes et chefs de projets connaissant et pratiquant java 5 et ayant des notions d'XML

Travaux pratiques

Les travaux pratiques représentent 50% de la formation. Ils sont réalisés sous Eclipse, avec le serveur d'application JBoss.

Sur demande, ils peuvent être réalisés avec d'autres outils de développement du marché, comme Borland JBuilder ou Rational Application Developer, et d'autres serveurs d'applications, comme BEA Weblogic ou IBM Websphere.

Programme de la formation

Introduction

- > Les architectures multi-tiers et les architectures distribuées
- > La plateforme JavaEE 5 : les concepts, les composants et les services
- > Les principes des Enterprise Java Beans et leur utilisation dans l'architecture
- > Les nouveautés de JavaSE 5 et EJB 3

La mise en œuvre de composants orientés services

- > Les EJB stateless
- > Le développement des classes et interfaces d'EJB
- > Les annotations spécifiques EJB 3
- > Le déploiement d'un EJB
- > La vue client des composants : registre JNDI et invocation locale ou distante des EJB

Développement avancé de Session Beans

- > La mise en œuvre de composants avec état : EJB stateful
- > Le cycle de vie et le pool d'instances
- > Les méthodes de callback
- > L'activation et la passivation des composants
- > L'accès à une base de donnée via une datasource

Développement de composants orientés message

- > Principe des MOM (Middlewares Orientés Message)
- > Mise en oeuvre avec JMS (Java Message Service)
- > Modèles de messages Publish/Suscribe et Point-To-Point
- > Le développement d'un EJB Message Driven Bean
- > L'envoi de messages à un MDB
- > Le cycle de vie d'un MDB
- > La gestion des messages erronés

Transactions

- > Le principe des transactions distribuées : commit à deux phases
- > JTA : Java Transaction API
- > La gestion déclarative avec les annotation de transaction
- > La propagation du contexte transactionnel
- > La gestion des transactions via le contexte d'EJB
- > La manipulation distante des transactions via le "User transaction"

Gestion des exceptions

- > Les types d'exception : système et applicative
- > Les exceptions de base de données
- > La relation entre exceptions et transactions

Sécurité

- > Principe de JAAS (Java Authentication and Authorisation Service)
- > Les rôles et les permissions sur les méthodes
- > Gestion déclarative de la sécurité d'accès à un EJB
- > Gestion programmée de la sécurité

Développement d'Entity Beans

- > Les principes du mapping objet/relationnel
- > La nouvelle technique de persistance : JPA
- > Le développement d'un Entity Bean et la déclaration d'attributs persistants
- > Le gestionnaire d'entités, le contexte et l'unité de persistance
- > Le cycle de vie
- > Le langage de requêtes EJB-QL
- > Les relations élémentaires entre EJB entités

Synthèse

- > L'intégration des EJB dans l'architecture J2EE
- > Les frameworks alternatifs aux EJB

L'API de persistance de donnée JPA fait parti de la nouvelle spécification EJB 3. Cette API est l'aboutissement logique des travaux de Hibernate et des anciennes spécifications sur les EJB 2.0 et 2.1 .

Ce cours vous permettra de comprendre les principes fondamentaux et de développer une couche de persistance avec JPA

Durée : 3 jours

Prix (intra) : 3450€

+ 50€ par participant

Participants

Développeurs, Architectes et Chefs de projets connaissant et pratiquant Java et ayant des notions XML

Travaux Pratiques

Les travaux pratiques sont réalisés avec le JDK 5.0 de Sun et Jboss 4.0, sous windows.

Programme

Techniques de persistance Java

- > Définition de la persistance
- > Présentation des frameworks de persistance Java
- > Présentation de Java Persistence API

Développer une classe persistante simple

- > Coder la classe persistante
- > Effectuer le mapping de la classe persistante
- > Définir les propriétés de configuration
- > Effectuer une requête EJB QL
- > Sauvergarder un objet persistant

Mapping objet / relationnel avec JPA

- > Contexte et objectifs
- > Développement des classes persistantes
- > Effectuer le mapping des classes et propriétés
- > Effectuer le mapping des associations
- > Effectuer le mapping de l'héritage

Manipuler les objets persistants

- > Charger les objets persistants
- > Les opérations CRUD
- > Cycle de vie des objets
- > Synchronisation avec la base de données
- > La persistance en cascade

Utilisation avancée du mapping

- > Contrôler les INSERT et les UPDATE
- > Mapping des clés primaires composées
- > Mapping multi-tables
- > Mapping des associations many-to-many
- > Mapping des associations de type list et map

Le langage EJB QL

- > Exécuter des requêtes d'interrogation
- > Exécuter des sous-requêtes
- > Exécuter des requêtes avec jointures
- > Utiliser des projections avec EJB QL
- > Requêtes sur les ensembles

Transactions et accès concurrents

- > Présentation des propriétés d'une transaction
- > Gestion des transactions avec JPA
- > Techniques de verrouillage

Spring est un framework qui simplifie considérablement la programmation J2EE et encourage les bonnes pratiques de conception objet. Il fournit une couche d'abstraction qui permet d'intégrer facilement l'ensemble des technologies J2EE (EJB, JMS, Web Service...), ainsi que les principaux frameworks open source Java (Struts, Hibernate...). La richesse des fonctionnalités offertes et la simplicité de mise en oeuvre font de Spring Framework le conteneur léger le plus attractif du marché.

Ce cours vous permettra de comprendre les principes fondamentaux de Spring Framework (Inversion de Contrôle, AOP, couche d'abstraction). Il vous apportera tous les éléments nécessaires pour développer avec efficacité une application n-tiers en utilisant Spring Framework. Il tient compte des évolutions de Spring 2.0 et de Spring 2.5.

Durée : 5 jours
Prix (intra) : 5750€
+ € par participant

Participants

Développeurs et chefs de projet, ayant à réaliser le développement d'applications n-tiers et connaissant le langage Java ainsi que le développement Web

Travaux pratiques

Ce cours comporte de nombreux travaux pratiques permettant d'assurer une bonne compréhension des concepts abordés. Les travaux pratiques sont réalisés avec Spring 1.2, Eclipse 3.1, MySQL 4 et JBoss 4.0.

Programme

Les principes fondamentaux de Spring Framework

- > Les techniques de développement Java EE
- > Les frameworks spécialisés : MVC, mapping O/R, traces,...
- > Pourquoi un framework de plus ?
- > Présentation des conteneurs légers et de l'inversion de contrôle (IoC)
- > Le design pattern IoC dans le processus d'instanciation
- > Les fonctionnalités du framework Spring

Les premiers pas avec Spring

- > Le développement par interface
- > L'accès aux beans

La manipulation de beans Spring

- > Le conteneur de composants : BeanFactory et ApplicationContext
- > La définition des beans
- > Les techniques d'injection
- > La gestion des dépendances
- > Les méthodes de callback

Utiliser Spring pour développer une couche d'accès aux données persistantes

- > Présentation des techniques de persistances
- > Développer une DAO avec Spring JDBC
- > Développer une DAO avec Spring / iBatis
- > Développer une DAO avec Spring / Hibernate
- > Gestion des transactions

Créer des composants métier évolués avec Spring AOP, Spring Remoting et le module de sécurité Acegi

- > Présentation du module Spring AOP
- > Développement de composants distribués avec le remoting Spring
- > Sécuriser une application avec le module de sécurité Acegi

Développer une application web avec Spring Framework

- > Programmation Struts avec Spring
- > Présentation du framework Spring MVC

Simplifier la configuration de Spring avec les annotations

- > Déclarer un bean par annotation
- > Les annotations d'injection
- > La gestion des transactions
- > Les annotations AOP
- > Gérer les priorités entre annotations et configuration

Présentation de la couche d'abstraction J2EE

- > La plate-forme J2EE
- > Développement JMS avec Spring Framework
- > Développement de composants EJB avec Spring Framework
- > Développement de Web Services avec Spring Framework

Framework MVC - JSF

sewatech 

Cette formation permet comprendre le pattern MVC mis en oeuvre dans les JavaServer Faces et de savoir développer des applications avec les JavaServer Faces.

Durée : 3 jours
Prix (intra) : 3450€
+ € par participant

Participants

Développeurs, architectes et chefs de projets connaissant et pratiquant Java en architecture Web et ayant des notions d'XML.

Travaux Pratiques

Ils sont réalisés avec Eclipse 3.1, Tomcat 5.5 et MyFaces.

Programme

Introduction

- > Les technologies Java Web
- > JSF, les frameworks MVC et le développement RAD

Les premiers pas

- > La configuration web et JSF
- > Créer un backing-bean et un formulaire

Les concepts de JSF

- > Les composants graphiques, coté client et coté serveur
- > Les backing beans : contrôleurs secondaires du MVC
- > La conversion et la validation des données saisies
- > Les événements et les listeners
- > Le langage d'expression (EL) et les différences avec les EL JSP

Les composants standards

- > Les caractéristiques générales
- > Passer des paramètres aux composants
- > La notion de Facet
- > Les composants pour les formulaires
- > Les composants d'affichage

Configurer une application JSF

- > Configurer le fichier web.xml
- > La structure du fichier de configuration JSF (faces-config.xml)
- > Configurer les backing beans et la navigation

Les notions avancées

- > Le cycle de traitement d'une requête (phases, action *immediate*,...)
- > Les principales classes du framework (FacesContext, Application, ExternalContext)

Afficher des messages utilisateur

- > Le principe des messages JSF
- > Afficher un message dans une JSP
- > Créer un message applicatif

L'internationalisation

- > La classe Locale et la norme i18n
- > La gestion des libellés
- > Les messages de conversion / validation
- > Les messages applicatifs

Intégrer JSF avec les autres technologies

- > L'intégration dans les JSP et avec des balises non JSF (JSTL)
- > L'intégration avec Struts (struts faces, shale, tiles)

Le MVC (Modèle-Vue-Contrôleur) est le principe d'architecture le plus répandu dans le développement d'applications Web en java. Le framework Struts, développé en open source dans le cadre de la fondation Apache, est depuis plusieurs années l'implémentation de référence de ce principe.

Ce cours vous permettra de comprendre le pattern MVC tel qu'il est implémenté dans Struts et vous apprendra à développer des applications Web/JSP avec le framework Struts.

Durée : 3 jours

Prix (intra) : 3450€
+ 50€ par participant

Participants

Développeurs, architectes et chefs de projets qui connaissent et pratiquent java en architecture Web (JSP, Servlet) et ont des notions d'XML

Travaux pratiques

Des exercices pratiques permettent d'illustrer les différents sujets abordés dans le cours.

Ils portent sur la version 1.3 de Struts, ou sur une version antérieure (possibilité de mise en oeuvre avec les versions 1.2 ou 1.1). Ils sont réalisés avec le JDK 5.0 de Sun et les outils open source Eclipse 3 et Apache Tomcat. Cependant, si vous avez déjà fait des choix d'outils pour vos développements, nous pouvons nous y adapter.

Programme

Architecture J2EE

- > Rappels sur J2EE
- > Les technologies Java Web : JSP et servlet
- > Architecture MVC2

Premiers pas avec Struts

- > Implémentation du MVC2 dans Struts
- > Intégration dans l'architecture n-tiers
- > Installation dans un projet Eclipse
- > Modèle : les form beans (ActionForm)
- > Vue : les JSP avec les taglibs HTML
- > Contrôleur : les actions
- > Configuration (struts-config.xml)

Développer les actions

- > La structure du contrôleur (ActionServlet, RequestProcessor,...)
- > La configuration d'une action
- > La gestion des renvois : forward par action et forward global
- > Les actions intégrées au framework (ForwardAction, DispatchAction, EventDispatchAction,...)

Développer les form beans

- > La validation d'un form bean ; classes ActionErrors et ActionMessage
- > Le cycle de vie d'un form bean
- > Les form beans dynamiques (DynaActionForm)
- > L'utilisation de Map et List

Utiliser les bibliothèques de balise

- > taglib bean
- > taglib html
- > taglib logic
- > Concurrence avec JSTL

Gestion des erreurs

- > La validation de form beans
- > Les erreurs et exceptions dans les actions
- > Afficher les messages d'erreur
- > La gestion déclarative des exceptions
- > La classe ModuleException

Techniques d'internationalisation

- > Les principes i18n

- > L'externalisation des textes dans un catalogue
- > La gestion automatique de la langue
- > La gestion spécifique de la langue

Framework de validation

- > Les principe de validation déclarative
- > Le framework Apache Validator
- > Les règles de validation
- > La validation de form beans classiques ou dynamiques
- > La validation contextuelle
- > La validation client en javascript
- > L'internationalisation des règles de validation

Mise en page avec Tiles

- > Les principes du framework Tiles
- > L'installation avec Struts
- > La mise en oeuvre

Approfondissements

- > Mise en oeuvre de la chaîne de responsabilité (ou de commandes)
- > L'organisation d'une application en modules

Objectifs :

- > Comprendre les principes fondamentaux de Hibernate
- > Savoir développer une couche de persistance avec Hibernate

Durée : 3 jours

Prix (intra) : 3450€
+ 50€ par participant

Participants

Développeurs, architectes et chefs de projet connaissant et pratiquant Java et ayant des notions d'XML

Travaux pratiques

Ils sont réalisés avec Eclipse 3 et Hsqldb.

Programme

Qu'est-ce que la persistance ?

- > Définition
- > Les solutions de stockage des données
- > Accéder aux SGBDR avec Java
- > Framework de persistance
- > Hibernate

Premier pas

- > Définir une classe persistante
- > Définir le mapping
- > Les propriétés de configuration
- > Utiliser Hibernate
- > Exécuter une requête
- > Exemple complet

Le mapping

- > Le contexte
- > Objectifs
- > Coder les classes persistantes
- > Ecrire le schéma de la base de données
- > Effectuer le mapping

Manipuler les objets persistants

- > Cycle de vie des objets
- > Opérations CRUD de base
- > Synchronisation avec la base de données
- > Persistance en cascade
- > Charger les objets persistants

Mapping avancé

- > Collections de valeurs
- > Mapping des associations

HQL et Criteria

- > Requêtes de base
- > Jointure
- > Projections
- > Requêtes sur les ensembles

Transaction et accès concurrent

- > Propriétés d'une transaction
- > Gestion de l'atomicité
- > Gestion de l'isolation
- > Verrouillage pessimiste
- > Verrouillage optimiste
- > Modes de verrouillage

Gestion du cache

- > Objectifs
- > Cache de premier niveau
- > Cache de second niveau

Parmi les technologies du Web 2.0, AJAX permet d'étendre les possibilités ergonomiques des applications Web et de rendre les applications plus conviviales pour les utilisateurs.

Cette formation a pour objectif de fournir aux participants les notions fondamentales du Web 2.0 et de leur permettre de prendre en main les techniques nécessaires pour sa mise en oeuvre. Le dernier chapitre doit leur permettre de comprendre comment ces nouveaux éléments s'intègrent dans leur architecture.

Durée : 2 jours

Prix (intra) : 2200€
+ 50€ par participant

Participants

Développeurs, architectes et chefs de projets connaissant XML, HTML, le langage java et le développement Java EE pour le Web (cf. Java EE - Développement Web).

Travaux Pratiques

Chaque technique est illustrée à l'aide d'exercices pratiques. Ceux-ci sont généralement réalisés avec Eclipse WTP.

Programme de la formation

Les principes du Web 2.0

- > Bilan des architectures Web traditionnelles
- > Web 2.0 : les objectifs et les technologies
- > Le principe des RIA (Rich Internet Application)
- > AJAX et javascript

Le langage javascript

- > Présentation de javascript
- > Le modèle de programmation
- > Les principaux frameworks et outils Javascript (prototype, dojo, script.aculo.us,...)

La programmation Ajax

- > Les principes et techniques d'AJAX
- > La mise en oeuvre avec l'objet XMLHttpRequest
- > La mise en oeuvre avec les frameworks javascript
- > L'intégration d'AJAX avec JavaEE ; problématique de l'intégration dans le MVC (Struts, JSF,...)

La mise en oeuvre d'un framework AJAX (option DWR ou Ajax4Jsf)

- > Présentation et prise en main du framework
- > Mise en oeuvre dans environnement Java EE
- > Les fonctions avancées du framework

Java et XML représentent un binôme parfait pour le développement d'applications distribuées, ouvertes et évolutives.

Cette formation apportera le socle de connaissances nécessaires à la mise en oeuvre des techniques XML en environnement java

Durée : 2 jours

Prix (intra) : 2200€

+ € par participant

Participants

Développeurs, architectes et chefs de projet connaissant java

Travaux pratiques

Les exercices réalisés à chaque étape de la formation se font avec le JDK 5.0 et Eclipse. Ils peuvent être réalisés à la demande, avec des JDK plus anciens (1.3 ou 1.4) et avec d'autres outils de développement (IBM RAD, Borland JBuilder,...).

Programme

Langage XML

- > Introduction
- > Construction arborescente
- > Balises standards
- > Commentaires
- > En-tête
- > Espace de nommage

Validation de documents

- > Principes
- > XML Schema
- > DTD
- > Autres techniques

Manipulation de documents

- > Techniques de lecture et d'écriture
- > DOM, SAX, JDOM
- > Castor

C# est le principal langage de la plateforme .NET. Quel que soit le type d'application que vous serez amenés à développer, les bases de .NET et C# sont les mêmes. Ce cours vous permettra de connaître les bases du langage C# et de comprendre les concepts objet avec .NET. Il vous apprendra à développer des classes et vous présentera les principales API.

Durée : 4 jours
Prix (intra) : 4200€
+ € par participant

Participants

Développeurs, concepteur et chefs de projets qui pratiquent un langage de programmation (C, C++, Visual Basic, Pascal,...)

Travaux pratique

Des exercices pratiques permettent d'illustrer les différents sujets abordés dans le cours. Ils sont réalisés avec le framework .NET 1.1 et VisualStudio 2003 ou avec SharpDevelop.

Programme

Présentation de .NET

- > Historique
- > Framework .net
- > Outils de développement

Premiers pas

- > Constructions de base
- > Types, identificateurs et variables
- > Instructions et opérateurs

Notions Objet en C#

- > Notions élémentaires
- > Organisation du code
- > Mécanisme d'exceptions

Librairies standards

- > Types élémentaires
- > Collections
- > Entrées / sorties

Accès aux bases de données avec ADO.NET

Objectifs :

- > Comprendre la démarche « Test Driven » préconisée dans l'eXtrem Programming
- > Connaître les techniques de tests unitaires et le framework NUnit pour .NET

Durée : 1 jour

Prix (intra) : 1100€
+ € par participant

Participants

Développeurs et chefs de projets connaissance .NET et C#

Travaux pratiques

- > 1/3 de théorie et de 2/3 de travaux pratiques
- > Windows, .NET framework 1.1, Visual Studio 2003 ou SharpDevelop, csUnit ou NUnit, EasyMock.NET, nMock

Programme

Principes et démarche des tests unitaires

- > Les principaux types de test
- > Les principes du test unitaire
- > Automatiser les tests unitaires
- > Le développement conduit par les Tests

Framework NUnit

- > Présentation et caractéristiques
- > Écriture d'un test simple
- > Assertions, échecs et erreurs

Techniques avancées

- > Leurres : Mock Objects
- > Leurres dynamiques
- > Test d'une application connectée à une base de données

Objectifs :

- > **Comprendre la démarche « Test Driven » préconisée dans l'eXtrem Programming**
- > **Connaître les techniques de tests unitaires et le framework NUnit pour .NET**

Durée :**Prix (intra) : 0€
+ € par participant**

L'administration d'un serveur d'applications Java EE est une tâche particulièrement délicate car elle requière avant tout des compétences système, mais demande de se plonger dans ce monde des composants Java.

Les formations de ce domaines essaient de faire la synthèse entre les connaissances Java nécessaires au métier et les procédures classiques d'exploitation de ces serveurs.

JBoss est le leader des serveurs d'application JavaEE libres. Son implantation sur le marché est en constante progression.

Ce cours vous présentera les caractéristiques de JBoss.

Il vous apprendra à installer et configurer JBoss, à déployer et optimiser des applications JavaEE dans JBoss.

Il vous présentera aussi les principes d'administration avancée de JBoss (surveillance, optimisation,...).

Durée : 3 jours

Prix (intra) : 3450€

+ 50€ par participant

Participants

Administrateurs JavaEE ayant une connaissance générale de JavaEE (Servlet, JSP, EJB,...)

Remarque : le séminaire « JavaEE : tour d'horizon » est un excellent moyen d'atteindre ces prérequis

Travaux pratiques

Pour chaque partie, mise en oeuvre pratique avec déploiement d'applications.

Ils sont réalisés avec le JDK 5.0 de Sun et JBoss 4.0 ou 4.2, sous windows, linux ou unix (solaris,...).

Programme

Introduction à JavaEE (rappels)

- > Les principes fondamentaux de java et de JavaEE
- > Les principaux composants de JavaEE (JSP, servlet, EJB, JMS,...)
- > L'essentiel d'XML

Introduction à JBoss

- > JBoss, le serveur d'application certifié JavaEE de RedHat
- > Les principes de l'Open Source et les types de licences associées
- > L'architecture de JBoss (kernel, JMX et AOP)

Les bases de l'administration JBoss

- > L'installation de JBoss et les prérequis
- > Le principe des configurations
- > Le déploiement d'applications JavaEE (ear), d'applications Web (war) et de composants EJB (jar)
- > Le chargement de classes et son impact sur le déploiement des librairies

La gestion des configurations

- > Les configurations standards (minimal, default, all) et configurations personnalisées
- > L'architecture de déploiement
- > Le déploiement de datasources et de connecteurs JCA (rar)
- > Le déploiement de services (sar) et de MBeans (JMX)
- > L'inspection du serveur avec JMX (console, twiddle et accès RMI)
- > L'intégration avec Tomcat ; le remplacement de Tomcat par Jetty

Le suivi du serveur

- > La gestion des traces avec Log4J et Chainsaw
- > L'inspection du serveur avec la console web, ses fonctions d'alertes et ses graphiques
- > L'analyse des queues et sujets JMS avec Hermes
- > Le suivi des requêtes SQL avec le proxy JDBC P6Spy et IronGrid
- > Les outils de monitoring du JDK : jconsole, jps, jstat,...

Les fonctions avancées d'administration

- > Les techniques d'invocation d'EJB (http, pooled,...)
- > Affiner la chaîne d'intercepteurs
- > La technique de script BeanShell
- > Les notifications en MBeans
- > La planification de tâches (scheduler et time service)

L'optimisation des performances

- > Les réglages de la machine virtuelle (hotspot, Xmx, Xms,...)
- > Affiner le dimensionnement des pools (EJB, DataSource, threads)
- > Les autres optimisations classiques
- > La mise en cluster pour la tolérance de panne (failover) et la répartition de charge (load balancing)

La sécurité du serveur et des applications

- > Les principes de sécurisation du serveur (réseau, security manager, applications critiques)
- > Le modèle de sécurité JBoss (JBossSx)
- > La gestion des autorisations et des authentifications en JavaEE (JAAS)
- > La sécurisation des échanges avec SSL

Quelques compléments

- > L'élimination des composants inutiles d'une configuration
- > Les notifications en MBeans (suite)
- > Le gestion centralisée de la configuration (netboot)
- > Le binding pour faire cohabiter plusieurs instances de JBoss sur un serveur

Apache Tomcat est le serveur de référence pour les JSP et servlets. Il peut être utilisé seul ou intégré dans d'autres serveurs d'application comme JBoss.

Ce cours vous apprendra à installer et configurer Tomcat, à y déployer, sécuriser et optimiser des applications Web.

Durée : 3 jours

Prix (intra) : 3450€

+ € par participant

Participants

Développeur désireux d'acquérir une expertise sur Tomcat Administrateurs J2EE ayant une connaissance générale de J2EE (servlet, JSP,...)

Travaux pratiques

Pour chaque partie, mise en oeuvre pratique avec déploiement d'applications. Ils sont réalisés avec le JDK 5.0 de Sun et Tomcat 5.5, sous windows ou linux.

Programme

Introduction à J2EE et à Tomcat

- > Les principes fondamentaux de java et de J2EE
- > Les principes de mise en oeuvre des servlets et des JSP
- > L'essentiel d'XML
- > La fondation Apache
- > Apache Tomcat, serveur d'application Web

Installation de Tomcat

- > Le choix et l'installation de la machine virtuelle
- > L'installation et le lancement de Tomcat
- > L'installation avec Apache Ant
- > L'architecture de Tomcat

Configuration

- > Les principes de configuration
- > La console d'administration
- > Le déploiement d'application Web (war)

Fonctions avancées

- > La gestion des sessions
- > Le principe des valves
- > Les valves de traces
- > Les valves de sécurité
- > L'installation de librairies
- > Le gestion de ressources (datasources)

Connecteurs

- > Les connecteurs Coyote
- > Optimiser les performances des connecteurs
- > L'intégration avec un serveur Web avec le protocole AJP
- > L'intégration avec Apache Web Server
- > L'intégration avec Microsoft IIS

Sécurité

- > La protection du serveur
- > Les principes d'authentification et autorisation
- > La technique d'authentification JAAS 133
- > Le protocole SSL

Gestion des traces

- > Apache Commons Logging
- > Java Standard Logging
- > Apache Log4J
- > L'exploitation des traces (Chainsaw)
- > L'intégration des systèmes de trace dans Tomcat

Performances et tests

- > Les outils standard de monitoring (jconsole, jstat,...)

- > L'optimisation de la JVM
- > Quelques réglages divers
- > Optimiser les JSP avec Jasper
- > Mettre Tomcat en cluster
- > Les tests de charge avec Apache JMeter

Apache Tomcat est le serveur de référence pour les JSP et servlets. Il peut être utilisé seul ou intégré dans d'autres serveurs d'application comme JBoss.

Ce cours vous apprendra à installer et configurer Tomcat, à y déployer, sécuriser et optimiser des applications Web.

Durée :

Prix (intra) : 0€

+ € par participant

Ces formations s'adressent à toutes les personnes des métiers de l'informatique qui souhaitent connaître l'essentiel sur un sujet dans un minimum de temps.

Le champs d'application de Java est particulièrement étendu, tant au niveau des architectures que des domaines fonctionnels. De plus les outils disponibles sur le marché, dans le domaine commercial ou open source, sont nombreux et le choix peut s'avérer difficile.

Cette journée permet de faire un tour d'horizon des possibilités de java et des techniques à mettre en oeuvre pour développer et déployer des applications.

Durée : 1 jour

Prix (intra) : 1500€

+ € par participant

Participants

Cette journée s'adresse à tout public informatique désirant mieux connaître le monde de java.

Exemples et démonstrations

Les exemples sont présentés avec Eclipse et JBoss.

Programme

Introduction à java

- > Historique
- > Java Standard / Java Entreprise
- > Fonctionnalités et architectures

Développements avec Java

- > Kits de développement (Eclipse, JBuilder,...)
- > Développement client / serveur avec Eclipse Visual Editor
- > Développement Web avec Eclipse Web Tool

Déploiements avec Java

- > Machines virtuelles et déploiement d'applications (poste client)
- > Rôle du serveur d'application (exemple avec JBoss)
- > Déploiement de composants Web, Métier et Web Service
- > Exemple basé sur une architecture mixte (client / serveur et Web)

Synthèse

- > Champs d'application de java
- > Solutions "métier" java
- > Java et l'Open Source

JavaEE : tour d'horizon

sewatech 

Cette journée permet de faire un tour d'horizon des techniques proposées par l'édition entreprise de java et de comprendre la façon de développer et déployer une application J2EE.

Durée : 1 jour
Prix (intra) : 1500€
+ € par participant

Participants

Cette journée s'adresse à tout public informatique désirant mieux connaître le monde de java.

Exemples et démonstrations

Les exemples sont présentés avec Eclipse et JBoss.

Programme

Introduction

- > Historique
- > Architecture 3 tiers

Applications Web et client / serveur

- > Client riche : swing
- > Client léger : servlet et JSP
- > Structure et frameworks MVC

Composants métiers

- > EJB, POJO
- > JMS
- > Frameworks J2EE

Accès aux données

- > JDBC, transactions
- > EJB, JDO
- > Frameworks de mapping O/R

Intégration de l'existant

- > JCA
- > Web Services

Serveur d'application

- > Services d'un serveur d'application
- > Formats de déploiement (war, jar, ear,...)
- > Sécurité

Cette formation présente les principes de sécurisation des applications Web développées en java. Elle expose les préoccupations principales en terme de sécurité, puis détaille les techniques proposées par java et J2EE.

Durée : 1 jour
Prix (intra) : 1500€
+ € par participant

Participants

Cette formation est destinée aux développeurs et chefs de projets, connaissant les architectures java et J2EE, qui veulent étudier les possibilité de sécurisation des applications Java.

Exemples et démonstrations

De nombreux exemples sont présentés avec JBoss et Tomcat.

Programme

Présentation des concepts liés à la sécurité

- > Principes d'identification et d'authentification
- > Gestion des autorisations et permissions
- > Problématiques de confidentialité et de non-répudiation
- > Techniques de cryptage symétrique et asymétrique
- > Attribution de clés publiques et de clés privées, autorités de confiance
- > Protection par pare-feu et zone démilitarisée (DMZ)

Authentification et autorisations J2EE avec JAAS

- > Principe de JAAS (Java Authentication and Authorization Service)
- > Notion de Sujet et de Principal
- > Architecture évolutive avec les notions de Contexte et de Module
- > Modules JAAS classiques : base de données, LDAP, NTLM, Kerberos,...
- > Mise en oeuvre du single sign on

Authentification et autorisations programmatiques

- > Prises de sécurité en architecture MVC
- > Principe des filtres et application à la sécurité
- > Framework Acegi Security
- > Techniques propriétaires : exemple des valves de Tomcat

SSL avec Java

- > Fonctionnalités de Java Secure Socket Extension (JSSE).
- > Certificats X.509
- > Protocoles TLS et SSL
- > Cryptage à base de clés de session

Sécurisation des services Web

- > Problématique de sécurité dans l'échange de données applicatives
- > Application des techniques de sécurité Web aux services Web
- > Signatures digitales XML, cryptage XML des informations
- > Authentification personnalisée : utilisation des en-têtes SOAP.
- > Extension de sécurité SOAP (Digital Credentials & Digital Signature Extensions).
- > Web Services Security Specifications (WS-Security).

Open Source / Java

sewatech 

Cette journée permet de comprendre le modèle Open Source, ses avantages et ses inconvénients et d'élaborer une architecture d'entreprise en java, à base de solutions Open Source.

Durée : 1 jour

Prix (intra) : 1500€

+ € par participant

Participants

Tout public informatique ; aucun pré-requis particulier

Programme

Open Source

- > Historique
- > Définitions et licences

Tour d'horizon des solutions Open Source

- > Infrastructure (systèmes d'exploitation, serveurs Web, bases de données,...)
- > Développement (langages, outils)
- > Organisation des communautés Open Source
- > Types d'organisations
- > Exemple de la fondation Apache

Environnement de développement

- > Eclipse, NetBeans,...
- > CVS, Ant
- > Frameworks OpenSource (Apache Struts, Spring,...)

Environnement de déploiement

- > Linux
- > JBoss, JonAS, Apache Tomcat ou Geronimo
- > MySQL, PostgreSQL,...

Solutions « métier »

- > Portail
- > Gestion de contenu et groupware
- > ERP

.NET : tour d'horizon

sewatech 

Cette journée permet d'appréhender le champ d'utilisation de .NET et de comprendre la façon de développer et déployer une application .NET.

Durée : 1 jour

Prix (intra) : 1500€

+ € par participant

Participants

Cette journée s'adresse à tout public informatique désirant mieux connaître le monde de java.

Exemples et démonstrations

Les exemples sont présentés avec Visual Studio .NET.

Programme

Introduction

- > Historique
- > Microsoft et la standardisation (ISO)
- > Concurrence avec Java

Framework .NET

- > Fonctionnalités et architectures
- > Kits et outils de développement
- > Runtime (CLR) et déploiement d'applications

Applications Web et client / serveur

- > WinForms
- > ASP.NET, WebForms

Applications distribuées

- > NET remoting
- > Web Services
- > COM+

Accès aux données

- > ADO.NET
- > Mapping O/R

Stratégies de migrations

- > De VB6 vers .NET
- > De C++ vers .NET
- > De COM vers .NET

Open Source / PHP

sewatech 

Cette journée permet de comprendre le modèle Open Source, ses avantages et ses inconvénients et d'élaborer une architecture Web en PHP, à base de solutions Open Source.

Durée : 1 jour

Prix (intra) : 1500€

+ € par participant

Participants

Tout public informatique

Programme

Open Source

- > Historique
- > Définitions et licences

Tour d'horizon des solutions Open Source

- > Infrastructure (systèmes d'exploitation, serveurs Web, bases de données,...)
- > Développement (langages, outils)

Architectures LAMP

- > Linux (et alternatives)
- > Apache (et alternatives)
- > MySQL (et alternatives)
- > PHP (et alternatives)

Architectures WAMP

- > Développement PHP
- > Historique de PHP
- > Fonctionnement d'une application PHP
- > Outils de développement
- > Évolution PHP4 / PHP5
- > Comparaison avec Java et .NET

La mise en place d'une architecture Web demande de prendre en compte tous les critères permettant de maîtriser les coûts de développement, de maintenance et d'exploitation. Pour y parvenir, des choix fondamentaux interviennent très tôt dans le projet, nécessitant d'avoir une vue globale sur les possibilités offertes par les technologies récentes.

Cette formation vous permettra de faire un tour d'horizon des principes essentiels servant de socle à l'élaboration d'une architecture Web et de connaître les principales possibilités techniques à évaluer, en java ou .NET, avec ou sans XML.

Durée : 1 jour

Prix (intra) : 1500€

+ € par participant

Participants

Développeurs, concepteurs, architectes, chefs de projet souhaitant se lancer dans le développement ou l'intégration d'applications Web

Programme du cours

Architectures et systèmes d'informations

- > Les principes d'architecture
- > L'organisation en couches
- > Un historique des architectures
- > Les architectures distribuées
- > Architecture et urbanisation

Architecture et technologies Web

- > Les architectures Web
- > Les techniques de serveur Web
- > Les techniques de client Web
- > Les apports du Web 2.0

Élaboration des couches applicatives

- > Rappel : architectures n-tiers
- > La notion de services métier
- > Java EE pour les services métier
- > Les services métier en .NET
- > La couche d'accès aux données, avec Java EE ou .NET
- > L'intégration d'un serveur d'applications

XML dans l'architecture

- > Le langage XML
- > La validation de document
- > La manipulation de documents
- > La transformations avec XSL et XSLT
- > Les Web Services
- > Les architectures orientées services (SOA)

Synthèse

- > Quelle technologie pour le Web ?
- > Quelle architecture pour le Web ?