Liste des formations

Principes et conception objet

Comprendre les mécanismes élémentaires du raisonnement objets et d'être capable d'aborder un langage de modélisation ou de programmation orienté objet.

Origines de l'Objet (Jour 1)
- 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 (Jour 1)
- 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 (Jour 1)
- 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 (Jour 2)
- L'importance de l'architecture dans une conception Objet
- L'enjeu de la gestion des dépendances entre classes et paquetages
- 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 (Jour 2)
- 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

Public:

Développeurs, concepteurs et chefs de projets

Pré-requis:

Connaissance et pratique du développement ou de la conception non objet

Durée:

1 ou 2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Programmation orientée objet

Introduction les différents concepts élémentaires et avancés du raisonnement objet.
Présentation des principes d'architecture d'une application en couches.

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

Public:

Développeurs, concepteurs et chefs de projets souhaitant apprendre la programmation avec un langage objet moderne comme java ou .NET (VB ou C#).

Pré-requis:

Aucun

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Mise en oeuvre des Design Patterns

-Apprendre à modéliser et réaliser des composants et des applications évolutives et réutilisables.
-Comprendre les principaux patterns de conception.

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 substition 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 compléxité 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 de création d'objets : fabrique, singleton et prototype
- Affiner l'affectation des responsabilités grâce aux patterns comportementaux : chaîne de responsabilité, patron de méthode et observateur
- Améliorer la structuration des classes avec les patterns de structure : adaptateur, façade et composite

Les patterns dans l'architecture
- Adapter les pattern 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

Public:

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

Pré-requis:

Pratique du langage de développement

Durée:

2 ou 3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Conception Java EE et Design Patterns

Apprendre à modéliser et réaliser des composants et des applications JavaEE évolutives et réutilisables.
Comprendre les principaux patterns de conception.

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

Public:

Architecte, chef de projet, analyste, concepteur, développeur.

Pré-requis:

Connaissances de Java et JavaEE (J2EE)

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Bonnes pratiques Java EE

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

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

Public:

Développeurs, concepteurs et chefs de projets

Pré-requis:

Connaissances préalables du développement ou la conception Java EE

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Développement d'applications JEE 8

Apprendre à développer simplement une application basée sur JSF, EJB, JPA, Bean Validation et CDI, ainsi que JAX-WS et RS.

Introduction
- Un historique de Java et JEE
- La compétition entre standards et frameworks
- La nouvelle philosophie de JavaEE : retour à la simplicité
- Les architectures des applications JEE
- Les profils : Web et complet
- Les outils : IDE et serveurs applications

Composants et dépendances
- Les composants métier EJB 3.1
- Les EJB avec ou sans état
- Les interfaces locales, distantes ou pas d'interface ?
- L'EJB Singleton
- Le Timer Service
- Les méthodes asynchrones
- L'injection de composants : Managed Beans et DI 1.0
- Le modèle de composants CDI
- Les portées (scopes) prédéfinis
- Les producteurs de beans
- La sécurité des composants

Gestion de la persistance
- Le mapping objet / relationnel avec JPA 2
- Le PersistenceContext
- Les annotations de mapping
- Les associations
- L'API Criteria
- La gestion des transactions avec JPA et EJB

Gestion de l'affichage
- Le principe des JSP et servlets 3.0
- La prise en compte des requêtes asynchrones
- Le développement de page JSF 2
- Un framework orienté composants
- Les templates Facelets
- Les Managed Beans
- Le langage d'expression
- La gestion d'évènements
- Les convertisseurs et validateurs
- La définition de la navigation
- Les composants AJAX : PrimeFaces, RichFaces,...

Services transverses
- Le framework de validation
- Bean Validation
- Web Services avec JAX-WS
- Services RESTful avec JAX-RS

Synthèse
- Retour sur l'architecture JEE 8
- Avantages par rapport aux versions précédentes
- Comparaison avec Spring Framework

Public:

développeurs et architectes connaissant Java et souhaitant apprendre à développer des applications JEE 8.

Pré-requis:

Connaissances de Java

Durée:

5

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Tests unitaires JUnit pour JavaEE

- Apprendre les bonnes pratiques nécessaires à la réalisation de tests unitaires efficaces et à l'élaboration d'une architecture pleinement compatible avec les tests unitaires.
- Etre en mesure d'exécuter vos tests dans un environnement d'intégration continue.

Principes et démarche
- Les enjeux de la qualité logicielle
- Les types de tests dans un projet
- L'intégration des tests dans la démarche
- Les tests dans le Processus Unifié et dans RUP (Rational Unified -Process)
- Les tests en démarche agile : eXtrem Programming et SCRUM
- La pratique du TDD (Test Driven Development)

Bases du framework JUnit
- Présentation des tests unitaires
- Le framework jUnit
- Développer un cas de test
- L'initialisation et finalisation d'un cas de test
- La réutilisation des portions de test
- Les suites de tests
- La restitution des résultats de tests

Mock Objects
- Nos tests sont-ils réellement unitaires ?
- Différencier les tests unitaires des tests d'intégration
- Le principe des objets de leurre (Mock)
- Les frameworks de Mock
- La mise en œuvre avec Mockito

Bonnes pratiques pour le développement de tests unitaires
- L'organisation des tests en packages
- L'indépendance et l'isolation des tests
- Trouver la bonne granularité
- Réaliser des tests aux limites

Bonnes pratiques pour l'écriture de code testable
- Le développement par composants
- La délégation plutôt que l'héritage
- Une gestion souple des dépendances avec l'inversion de contrôle et l'injection

Couverture des tests
- Les métriques de couverture de tests
- Les objectifs de couverture
- L'évaluation de la couverture des tests avec Cobertura et Sonar

Outils complémentaires à JUnit
- La concurrence avec TestNG
- Tester les applications Web avec HttpUnit
- DBUnit pour tester les applications avec base de données

Intégration continue des tests
- L'automatisation avec Ant ou Maven
- Le principe de l'intégration continue
- La place des tests en intégration continue
- La mise en œuvre avec Hudson

Tests unitaires en architecture JavaEE
- Rappel sur les architectures JavaEE
- Les tests de composants EJB 3
- Les tests avec le framework Spring
- Les tests des classes d'affichage (Struts, JSF)
- Les tests des classes persistantes d'Hibernate

Synthèse et Conclusion
- Intégrer les tests unitaires dans la démarche
- Intégrer JUnit dans l'environnement
- Automatiser les autres types de tests (avec Fitnesse, Selenium,...)

Public:

Développeurs, architectes et chefs de projets

Pré-requis:

Connaissance du langage java

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Développement de Services Web en Java

Ce cours explique aux participants comment développer un nouveau type de composants logiciels baptisé Services Web (Web Services).
Les services Web reposent sur l'utilisation de XML pour échanger des données (SOAP), pour établir le format d'échange (XML, XML-Schéma), pour enregistrer et décrire les services (UDDI, WSDL).
Les services Web peuvent être écrits dans différents langages, ils font partie intégrante des environnements .NET de Microsoft et J2EE de SUN

Les Web Services : un modèle de composants logiciels

Architecture des Web Services

Rappel sur XML
- XML, namespaces et DTD
- Présentation des schémas XML

Le protocole SOAP

Description d'un service avec WSDL

Mise en œuvre de services web avec Axis :
- Installation
- Configuration, déploiement

Publication d'un service avec UDDI

Aspects sécurité

Services Web J2EE

Public:

Toute personne souhaitant utiliser et développer des services Web

Pré-requis:

Pour suivre ce cours, les participants doivent connaître et pratiquer le langage Java.

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Développement XML en Java

- Comprendre la structure d'un document XML
- Etre capable de manipuler des documents XML en java

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

Public:

Développeurs et Chefs de projets

Pré-requis:

Connaissance et pratique de Java

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Tests unitaires et Refactoring

Les techniques de refactoring 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

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
- Librairies

Public:

Chefs de projets et développeurs

Pré-requis:

Connaitre et pratiquer Java avec Eclipse

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Performance des applications Java/J2EE

Exposer les problématiques autour de la performance des applications J2EE. Montrer les techniques les plus optimum pour réaliser des applications performantes.

Introduction
- Architecture distribuée
- Architecture applicative orientée service
- Architecture applicative orientée objet
- Coût et bénéfices des frameworks
- Qualité et performance

Machine virtuelle
- Compilation JIT

Java : le langage
- Bonnes et mauvaises pratiques
- Membres statiques et méthodes virtuelles
- Upcast et downcast
- Appels de méthode et empilement des paramètres
- Évaluation des expressions
- Déclaration de variables
- Coût des structures de contrôle

Gestion de la mémoire
- Garbage Collector
- Les objets PhantomReference, ReferenceQueue, SoftReference, WeakReference
- Les fuites de mémoire
- Le coût du GC

Threads
- Rappel sur le fonctionnement des threads
- Les objets synchronisés
- Les objets pour la synchronisation
- Notification

Exceptions
- Coût des exceptions
- Modèle de mise en œuvre efficace

API
- Collection
- JDBC
- JSP
- XML

Java 5
- Améliorations de la performance avec Java 5
- Generics
- StringBuilder
- Mesure du temps
- Annotations
- Paramétrage du GC
- Paramétrage orienté serveur

Frameworks
- Journalisation (log4j)
- MVC (Struts, JSF)
- Persistance (ORM Hibernate, EJB)
- Gestion transactionnel objet (Spring)

Profilage
- RAM
- CPU
- Les outils du JDK
- Les outils open source

Public:

Développeurs J2EE

Pré-requis:

Connaître Java.

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Java - Compléments sur mesure

Les compléments possibles pour un cursus

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'ecriture 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'internationnalisation 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 « RessourceBundle »

Introduction à l'API de réflexion Java
- Le type Class
- Charger dynamiquement une classe
- Lire les méta-données d'une classe
- Invoquer dynamquement une méthode

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 SLF4J ou Apache commons-logging

Public:

Développeurs et Chefs de projets

Pré-requis:

Avoir une première connaissance de java ou suivi le cours Initiation à java

Durée:

Variable

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Initiation à Java et UML

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.
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.

Présentation de java
- 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
- 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, 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 exception (throws), cas des RuntimeException
- Développer des classes d'exception personnalisées

Librairies standards
- Le classe Object
- Manipulation de chaînes de caractères : classes String et StringBuffer
- Les types élémentaires et les enveloppeurs de types primitifs
- Manipulation de dates et heures
- Gérer des listes dynamiques avec les collections et les maps

Accès aux bases de données avec JDBC
- Principes de JDBC : une API commune et un driver spécifique
- Envoyer des requêtes de sélection et lire le résultat dans un ResultSet
- Envoyer des requêtes de mise à jour

Programmation graphique avec Swing
- Présentation des JFC (AWT, swing, java 2D,...)
- Développer des composants graphiques simples (JLabel, JButton,...)
- La gestion des événements avec les listeners
- Développer de composants plus complexes (JList avec un ListModel)

Principales nouveautés de Java5
- Les collections typées avec les generics
- Les nouvelle instruction de boucle (for each)
- Les types énumérés et l'autoboxing

UML pour développeur (1 jour)
Généralités
Diagramme de classes
- Définition des classes, types de classes
- Attributs, associations, opérations
- Héritage et agrégation
- Classes abstraites et Interfaces
Diagrammes d'interactions (collaboration / Séquence)
- Utilité du diagramme de collaboration
- Règles d'affectation des opérations aux classes
- Projection du modèle de classe en Java

Public:

Développeurs et Chefs de projets

Pré-requis:

Connaissance et pratique d'un langage de programmation (C, C++, Visual Basic, Pascal,...)

Durée:

5 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Initiation Java

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.

Il vous permettra de:
- Connaître les bases du langage Java et de comprendre les concepts objet avec Java.

Présentation de java
- Quelques rappels historiques
- L'environnement java : le JDK, le JRE et la machine virtuelle
- Les outils de développement du marché : Eclipse, Netbeans,...
- 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 paquetages (ou 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
- Les types énumérés

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 exception (throws), cas des RuntimeException
- Développer des classes d'exception personnalisées

Librairies standards du JDK
- Le 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
- La manipulation de dates et heures
- Les listes dynamiques avec les collections et les maps ; utilisation des generics

Accès aux bases de données avec JDBC
- Les principes de JDBC : une API commune et un driver spécifique
- L'architecture de JDBC et les 4 types de drivers
- Établir une connexion avec une base
- Les requêtes de sélection et la lecture du résultat dans un ResultSet
- Exécuter des requêtes de mise à jour
- La gestion des transactions, en mode automatique ou manuel
- L'appel de procédures stockées

Le programme de ce cours peut être adapté, et complété par des modules sélectionnés dans le plan de Approfondissement java.

Public:

Développeurs et Chefs de projets souhaitant commencer un apprentissage de Java.

Pré-requis:

Connaissance et pratique d'un langage de programmation (C, C++, Visual Basic, Pascal...)

Durée:

4 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Profilage et tests de montée en charge

- Exposer l'état de l'art en matière de techniques de mesure de charges et de profilage.
- Découvrir les facteurs de charges.
- Mener des tests de profilage des applications.
- Apprendre à construire un plan de charge.
- Apprendre à diagnostiquer des mesures de charge.
- Comprendre les facteurs d'amélioration des performances.

Qu'est-ce qu'une mesure de performance ?
- Les critères de performance des applications : architecture, qualité du code, réglage des serveurs, disponibilités du réseau
- Performance et charge des applications
- Les facteurs aggravants

À quelle phase du projet étudier les performances ?
- En phase de développement : pour vérifier le comportement de l'application en accès simultanés et déboguer le code
- En fin de projet : pour régler les serveurs et augmenter les performances

Quoi mesurer ?
- L'activité du système
- L'activité du réseau
- L'occupation mémoire
- L'occupation du processeur
- Les moniteurs des serveurs Web, de composants et de bases de données

Comment faire les mesures ?
- Mettre en place de la plateforme de test
- Installer un outil de simulation de charge
- Définir un protocole de mesures
- Conformer la plateforme de test à la plateforme de production
- Simuler, mesurer, collecter

Comment interpréter les mesures ?
- Mesurer la charge
- Mesurer la performance

Quels sont les outils de mesure de performance ?
- Les outils de profilage du code
- Les simulateurs de montée en charge

Comment améliorer les performances des applications ?
- Ordres SQL, MPD et réglages du SGBD
- Configuration et trafic sur le réseau
- Bon usage et réglage du serveur web
- Bon usage et réglage du serveur de composants
- Le code, bonnes et mauvaises pratiques
- Qualité versus rapidité

Conclusion
- Anticiper les tests de montée en charge
- Veiller à la qualité du code
- Opérer un « refactoring » régulier

Public:

Chef de projet, développeurs et exploitants d'applications J2EE.

Pré-requis:

Aucun

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Administration Apache Tomcat

Apprendre à installer et configurer le serveur d'applications Tomcat
Apprendre à déployer et optimiser des applications Web dans Tomcat.

Introduction à JavaEE et à Tomcat
- Les principes fondamentaux de java et de JavaEE
- Les principes de mise en œuvre des servlets et des JSP
- L'essentiel d'XML et son usage dans Tomcat
- Présentation de la fondation Apache
- Apache Tomcat, serveur d'application Web
- Architecture de déploiement : autonome ou avec un serveur Web frontal

Installation et Configuration de Tomcat
- La préparation du serveur : système, machine virtuelle Java
- L'installation et le lancement de Tomcat, en mode standard ou service
- Le lancement de Tomcat
- L'architecture de Tomcat
- Les principes de configuration
- Le déploiement d'application Web (war)
- L'installation des librairies
- L'installation d'une DataSource

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

Sécurité
- La protection du serveur
- Les principes d'authentification et autorisation
- Les valves de sécurité
- Le protocole SSL

Monitoring et gestion des traces
- Introduction à la gestion des traces
- Apache Commons Logging, Java Logging API et Apache Log4J
- L'intégration de Log4J dans Tomcat
- Les valves de traces de requêtes
- Les outils standard de monitoring (jconsole, jstat,...)
- Les fonctionnalités du Manager de Tomcat
- Les outils de monitoring JMX

Optimisation des performances
- Le réglage de la JVM : mémoire et garbage collector
- Le réglages des pools : connexions aux bases de données, threads
- Optimiser les JSP avec Jasper
- Déployer Tomcat en cluster

Public:

Développeurs désireux d'acquérir une expertise sur Tomcat et administrateurs JavaEE

Pré-requis:

Connaissance générale de J2EE (servlet, JSP...)

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Administration JBoss AS 7

Comprendre d'un point de vue administration les changements par rapport aux versions précédentes sont profonds :
- modification complète des fichiers de configuration
- nouvelle console
- nouvelle interface en ligne de commande
- nouveau système de modules
- ...

Le plan présenté ci-dessous est susceptible d'évoluer.

Introduction à JavaEE et JBoss
- Présentation de Java et de JavaEE
- Typologie des applications JavaEE
- Profils de JavaEE 6
- Présentation de JBoss
- Principale tâches d'administration

Bases de l'administration de JBoss AS
- Installation, démarrage et arrêt
- Principes de configuration
- Mode autonome ou domaine
- Déploiement d'applications (ear, war, jar,...)
- Déploiement automatique ou manuel

Outils d'administration
- Console d'administration
- Interface en ligne de commande
- Interface HTTP / JSON
- API Java

Gestion des ressources et des accès
- Accès Web : HTTP, HTTPS, AJP
- Gestion des librairies et des dépendances avec JBoss Module
- Accès aux base de données : datasource
- Ports utilisés et gestion des conflits

Suivi et surveillance du serveur
- Les traces d'accès Web
- Les traces du serveur avec LogManager

Sécurité du serveur et des applications
- Les objectifs de sécurisation du serveur
- Le modèle de sécurité JBoss AS
- La gestion des autorisations et des authentifications en JavaEE (JAAS)
- La sécurisation des accès et des consoles
- La sécurisation des échanges avec SSL

Amélioration des performances
- Le tuning de la machine virtuelle
- La dimensionnement des pools (EJB, DataSource, threads)
- Le retrait de composants inutiles
- Le clustering pour la tolérance de panne (failover) et la répartition de charge (load balancing)

Administration JMS
- Un rappel des principes de JMS
- La configuration des destination dans HornetQ
- La répartition de la charge avec HornetQ
- La tolérance de panne

Public:

Cette formation s'adresse aux administrateurs connaissant JavaEE (Servlet, JSP, EJB,...), aux développeurs et architectes souhaitant connaître le fonctionnement de JBoss AS 7, ainsi qu'aux intégrateurs JavaEE.

Pré-requis:

Connaissance générale de Java EE

Durée:

4 jours

Prix:

Nous contacter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Développer une application JEE avec Spring

-Assimiler les principes fondamentaux de Spring Framework (IoC, AOP, gestion transactionnelle, Spring Security, SWF, SWS).
-Manipuler les technologies importantes du développement JEE dans un contexte Spring (Struts, JSF, Hibernate, JMS, Web Services).
-Decouvrir les évolutions apportées par Spring 2.0 / 2.5 et les nouveautés de Spring 3.

Les principes fondamentaux de Spring Framework
- Présentation de la notion de conteneur léger dans le cadre d'un développement JavaEE
- Intérêt du design pattern Inversion de Contrôle (IoC) et des techniques AOP
- Mise en oeuvre du conteneur Spring avec l'ApplicationContext
- La configuration des composants par XML et par annotations
- Les différentes techniques d'injection des dépendances avec Spring
- Cycle de vie des beans Spring : portée des beans et méthodes de callback
- Notion de post-processeur : exemple d'utilisation pour externaliser des propriétés de configuration
- Les apports de Spring 3 en matière de configuration

Couche de persistance et gestion transactionnelle
- Le modèle de programmation d'accès aux données : traduction des exceptions, classes de support et classes template
- Ecrire des classes DAO avec Spring JDBC. Cas des clés auto-générées, fonctionnalités Java 5
- Utiliser Spring pour simplifier le développement Hibernate : les différentes stratégies
- La notion de propagation des transactions entre services métier
- Gestion déclarative des transactions par configuration XML et par annotations

Construction d'applications Web avec Spring Framework
- Configuration et instanciation du contexte Spring dans un environnement web
- Utiliser des composants métier Spring depuis une application Struts ou JSF
- Principe et mise en oeuvre de Spring MVC
- La notion de view resolver
- La taglib "form" pour développer des formulaires
- Utiliser les annotations pour configurer les controllers
- Les nouvelles annotations de Spring 3
- Implémenter des flux d'écrans avec Spring Web Flow 2

Gérer la sécurité des applications avec Spring Security
- Présentation et principe de Spring Security 2 (Acegi)
- Configurer la chaîne de filtre pour sécuriser une application web
- Formulaire de login personnalisé, gestion du logout, authentification anonyme, fonction "remember me"
- Améliorer la protection de l'application par cryptage des mots de passe
- Personnaliser la persistance des données d'authentification et d'habilitation
- Utiliser la taglib de Spring Security dans les JSP

Ecrire des composants AOP
- Définitions et concepts de la programmation AOP
- Définir des Poincuts, Advices et Aspects pour créer ses propres modules AOP
- Créer des proxys pour appliquer un aspect à un composant Spring
- Configuration par XML et par annotations AspectJ

Techniques de remoting JMS et Web Services
- Développer des composants JMS avec Spring
- Ecrire des services asynchrones du type Message Driven POJO
- Développer un web service de type "contract-first" avec Spring Web Services

Tester les composants d'une application Spring
- Les bonnes pratiques de conception pour les tests
- Utiliser des ressources autonomes et des objets de mock
- L'intégration de JUnit et TestNG

Public:

Développeurs, chefs de projet, architectes

Pré-requis:

Les participants doivent connaître les langages Java, SQL et HTML.
La connaissance de XML et des spécificités Java 5 est un plus.

Durée:

5 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Etat de l'art des frameworks et architectures JEE

1. Echange avec les participants : affiner les attentes des participants en fonction des expériences, souhaits et besoins des participants
2. Principes d'architectures : solutions / inconvénients
3. Technologie Java EE et frameworks associés

1. Introduction

- Les objectifs de l'architecture
  - Rationaliser le développement
  - Améliorer la réutilisation
  - Uniformiser le code
  - ...
- L'écosystème Java
  - Les standards Java SE et Java EE
  - Les projets Open Source
  - Les solutions propriétaires

2. Principes d'architectures

- Les principaux types d'architectures
  - Les architectures à 1, 2, 3,... couches
  - Les architectures distribuées
  - Les architectures Web et client / serveur
- Les principaux patterns d'architecture
  - L'accès aux données avec le pattern DAO
  - Le traitement et règles de gestion dans la couche services
  - Le transfert de données par Data Transfer Objects (DTO) ou DataValue
  - L'organisation de la couche présentation avec MVC
  - La gestion des transactions en architectures n-tiers et Web
- Les architectures Web et orientées services
  - Les protocoles de communication
  - Les formats d'échange : XML, JSON,...
  - Les services Web dans une architecture Objet
  - Les services REST
  - L'intégration de services

3. Frameworks Java EE

- Les architectures Web et n-tiers avec Java EE
  - Les blueprints officiels
  - Les serveurs d'applications Java EE
  - Les technologies standards Java EE
- La couche Persistance
  - Le Mapping O/R : JPA, Hibernate
  - Les outils Data Mapper : MyBatis, Spring JDBC
- Les technologies de la couche service
  - Les standards : EJB et CDI
  - L'injection de dépendances avec Spring
  - La gestion déclarative des transactions
- Les technologies et frameworks Web
  - Les techniques de base : HTML, CSS, JavaScript
  - Les standards Java : JSP, servlet, JSF
  - Les concurrents de JSF : Spring MVC, Play!, Grails,...
  - Les principes d'AJAX
  - Les frameworks spécialisés pour AJAX : GWT, Vaadin, Wicket
  - Les outils et fwk JavaScript : jQuery, AngularJS, BackboneJS,...
- Les techniques d'intégration
  - Intégration verticale ou horizontale
  - L'exposition de ressources avec JAX-RS et Spring MVC
  - Les Web Services : JAX-WS, Axis, CXF et Spring WS
  - Le messaging avec JMS
- Synthèse

Public:

Cette formation s'adresse aux architectes, concepteurs et développeurs souhaitant comprendre les enjeux d'une bonne architecture et l'intérêt des frameworks dans l'écosystème Java.
Elle s'intègre dans une

Pré-requis:

Aucun

Durée:

2 jours

Prix:

Nous contacter.

Date

Nous consulter.

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

WebService : développement avec JAX-WS

- Connaîtrez les techniques de développement de services Web avec Java.
- Etre capable de mettre en œuvre JAX-WS au sein d'applications JavaEE et avec le framework Spring.

Présentation des WebServices
- Un historique des techniques d'interopérabilité
- Le socle technique des WebServices : http, XML
- Un rappel sur XML
- Les protocoles des Services Web
- La terminologie associée
- Les services dans une architecture
- Quelques cas d'utilisation et contre-exemples

Présentation des protocoles
- Les messages XML avec SOAP (Simple Object Access Protocol)
- La description des service avec WSDL (Web Service Description Langage)
- Les annuaires de services UDDI (Universal Description, Discovery and Integration)
- La pile des protocoles WS-*

Solutions et standards Java
- Le traitement des données XML avec JAX-P, JAX-B, SAAJ
- L'appel de procédures distantes avec JAX-RPC et JAX-WS
- L'intégration dans les serveurs d'applications
- Le kit WSDP (Java Web Service Developer)
- Les implémentations de référence : Metro, Jersey,...
- Les outils Apache : Axis et CXF
- Les possibilités offertes par Spring Framework
- Les démarches : description-first ou code-first

Mise en œuvre de JAX-WS
- Un rappel sur la pile des techniques JAX-*
- Le développement d'un service par annotation
- La personnalisation du service
- Le binding des arguments avec JAX-B
- Les utilitaires wsconsume et wsgen

Mise en œuvre avec Spring Framework
- Le choix entre Apache CXF et Spring-WS
- CXF : principe et mise en œuvre
- L'intégration de CXF dans Spring
- Les principes de Spring-WS
- Le développement de services avec Spring-WS
- Le développement JAX-WS avec Spring

Public:

Architectes, concepteurs et développeurs pratiquant Java et souhaitant intégrer des Web Services dans leurs applications Java Standard ou Java EE.

Pré-requis:

Aucun

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Développer avec les frameworks Open Source

- Présenter les frameworks classiques, JSF (ou struts), Spring et Hibernate.
- Apprendre à démarrer le développement avec ces frameworks.

Pour la première, il faut choisir entre Struts et JSF.

1° PARTIE : option JSF
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 managed beans : contrôleurs secondaires du MVC
- La conversion et la validation des données saisies

Configurer une application JSF
- Configurer les fichiers web.xml et faces-config.xml
- Configurer les managed beans et la navigation

1° PARTIE : option Struts
Introduction au MVC
- Principe du MVC avec les JSP et les servlets
- Implémentation du MVC dans Struts

Premiers pas avec Struts
- Modèle : les form beans
- Vue : les JSP avec les taglibs HTML
- Contrôleur : les actions

Techniques avancées
- Librairies de balises bean et logic
- Gestion des erreurs

2° PARTIE : Spring
Les principes fondamentaux de Spring Framework
- Présentation des conteneurs légers et de l'inversion de contrôle (IoC)

La manipulation de beans Spring
- La définition des beans
- Les techniques d'injection

Intégration de Spring avec les autres frameworks
- Développer une DAO avec Spring / Hibernate
- Gestion des transactions
- Programmation Struts ou JSF avec Spring

3° PARTIE : Hibernate
Présentation d'Hibernate

Coder une première classe persistante avec Hibernate
- Développement d'une classe persistante
- Définir les propriétés de configuration
- Lire des objets persistants à l'aide d'une requête HQL

Présentation du mapping objet / relationnel avec Hibernate
- Développement des classes persistantes
- Effectuer le mapping des cas les plus courants

Manipuler les objets persistants
- Cycle de vie des objets
- Les opérations CRUD

Public:

Architectes et développeurs devant démarrer rapidement un projet en architecture n-tiers.

Pré-requis:

Connaissance du langage Java ainsi que du développement Web.

Durée:

5 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Spring : les bases

Etre en mesure de développer des applications JavaEE basées sur Spring.

Présentation des principes fondamentaux de Spring Framework
- Les techniques de développement Java EE
- Les frameworks Java spécialisés : MVC, mapping O/R,...
- Les conteneurs légers
- L'IoC : inversion de contrôle
- L'AOP : programmation orientée aspects
- Les fonctionnalités du framework Spring

Première classe métier
- L'implémentation dans une classe
- La configuration du conteneur
- L'accès au bean
- La configuration XML ou par annotations

Manipulation les beans Spring
- Le BeanFactory et d'ApplicationContext
- La définition des beans et méthodes de fabrique
- L'injection des dépendances
- L'héritage de beans et les beans abstraits

Accès aux données avec Spring JDBC et Hibernate
- La couche d'abstraction pour JDBC
- L'intégration avec Hibernate
- La gestion déclarative des transactions

Application web avec Spring / Struts
- Rappels sur la programmation avec Struts
- L'intégration des actions Struts

Sécurité avec Spring Security
- Présentation du module Spring Security / Acegi
- La gestion des autorisations
- La gestion de l'authentification

Web Services avec Spring Framework
- Les principe des Web Services
- L'intégration d'Apache CXF avec Spring

Public:

Architectes, concepteurs et développeurs.

Pré-requis:

Aucun

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

JavaServerFaces (JSF)

Maîtriser les concepts fondamentaux de JSF (composants graphiques, backing bean, cycle de traitement des requêtes...).
Apprendre à utiliser des composants Ajax, ainsi que la programmation avec Facelets.

Développement Web avec Java et JSF
- Présentation des technologies utilisées dans le développement d'applications Web avec JEE
- Rappel du principe MVC 2 pour les applications à base de Servlet/JSP
- Fonctionnalités de JSF et positionnement dans l'écosystème Java Web

Démarrer avec JavaServer Faces : créer une première page JSF
- Ecrire un formulaire de login avec des tags graphiques JSF
- Créer le backing-bean du formulaire
- Configurer l'application avec les fichiers faces-config.xml et web.xml
- Structure de l'archive de déploiement

Concepts de base
- Les composants graphiques JSF : des Java Beans accessibles depuis les JSP et le code Java
- Notion de page Stateful : la représentation "serveur" et la représentation "client"
- Le modèle de programmation évènementiel de JSF
- Les backing beans utilisés comme contrôleurs secondaires
- Les étapes principales du cycle de traitement d'une requête HTTP
- Le langage d'expression JSF : différences avec les EL JSP, Unified EL

La bibliothèques des composants standards
- Les propriétés générales des composants JSF : visibilité, style CSS, notion de binding...
- Passer des paramètres aux composants et notion de facet
- Composants d'affichage de texte et d'image
- Les composants de saisie et les "value change event"
- Les composants de commande et les "action event"
- Sélection de données dans les formulaires : combo-box, listes, cases à cocher et boutons radio
- Regroupement de composants et layout avec le PanelGroup et le PanelGrid
- Afficher une source de données dynamique avec un DataGrid

Gestion des messages utilisateur
- Structure des messages JSF : la classe FacesMessage
- Afficher un message avec HtmlMessage et HtmlMessages
- Créer un message applicatif avec la classe FacesContext

Validation des saisies
- Principe de la validation JSF
- Méthode de validation dans un backing-bean
- Les validators standards de l'API JSF
- Modèle de programmation pour écrire un validator personnalisé

Configuration avancée d'une application JSF
- Paramétrage de la servlet JSF et des implémentations JSF-RI et MyFaces
- Structure de la configuration JSF : comment organiser la configuration en plusieurs fichiers
- Configurer les backing beans : injecter des propriétés et effectuer des références entre beans
- Définir les règles de navigation : navigation par page, navigation globale, navigation par action

Internationaliser une application JSF
- Principe de l'internationalisation en Java
- La gestion des locales avec JSF : gestion par défaut, gestion personnalisée
- Les catalogues de traduction : libellés, messages applicatifs, messages d'erreur de validation

Notions avancées
- Revue détaillée du cycle de traitement des requêtes HTTP
- Principe de l'attribut "immediate" des composants de commande et de saisie
- Mise en oeuvre d'un PhaseListener
- Interagir avec le framework : utiliser les classes FacesContext, VariableResolver, ELResolver, NavigationHandler, ViewHandler...

Intégrer JSF avec les autres technologies
- Intégration avec les JSP et les tags JSTL
- Programmation Ajax avec la bibliothèque Rich Faces
- Présentation de la programmation avec Facelets

Public:

Développeurs, architectes et chefs de projets

Pré-requis:

Connaître le développement Java en environnement Web (API Servlets et JSP).
La connaissance de XML est un plus.

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Mapping O/R avec JPA

- Comprendre les logiques de mapping entre une base de données relationnelle et les classes d'un langage orienté objet.
- Découverte des techniques d'annotations, du langage de requête JPQL (Java Persistence Query Language) et des transactions, en environnement JavaSE ou JavaEE, avec ou sans les EJB.

Techniques de persistances Java
- La problématique de la persistance
- Les frameworks de persistance pour Java
- Présentation de JPA : Java Persistance API
- Développer une classe persistante simple

La classe persistante
- Le mapping de la classe persistante, avec les annotations JPA
- Les propriétés de configuration standard
- Les propriétés de configuration spécifiques à Hibernate ou Toplink
- Une requête JPQL / EJB QL
- Sauvegarder un objet persistant
- Mapping objet / relationnel avec JPA

Contexte et objectifs
- Le développement des classes persistantes
- Le mapping des classes et propriétés
- Le mapping des associations
- Le mapping de l'héritage
- Manipuler les objets persistants

Le chargement des objets persistants
- Les opérations CRUD
- Le cycle de vie des objets
- La synchronisation avec la base de données
- La persistance en cascade
- Utilisation avancée du mapping

Contrôler les INSERT et les UPDATE
- Le mapping des clés primaires composées
- Le mapping multi-tables
- Le mapping des associations many-to-many
- Le mapping des associations de type list et map
- Le langage JPQL / EJB QL

Les requêtes d'interrogation
- Les sous-requêtes
- Les requêtes avec jointures
- Les projections avec JPQL / EJB QL
- Les requêtes sur les ensembles
- Transactions et accès concurrents

Présentation des propriétés d'une transaction
- La gestion des transactions en environnement JavaSE
- Les transactions en environnement Java Web, sans les EJB
- Les transactions JTA, en environnement JavaEE, avec les EJB
- Les techniques de verrouillage : optimiste ou pessimiste

Public:

Développeurs, Architectes et Chefs de projets

Pré-requis:

- Connaissance et pratique de Java
- Avoir des notions XML

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Application Web (JSP & Servlets)

Apprendre l'ensemble des techniques nécessaires pour démarrer un projet Java Web, ainsi que le principe du pattern MVC.

Présentation des architectures Web
- Les principes du Web : requêtes HTTP, URL et pages HTML
- Techniques de Web dynamique
- La plateforme JEE 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

Public:

Développeurs et Chefs de projets

Pré-requis:

Les participants doivent connaître le langage Java, ainsi que le langage HTML.

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Initiation à Scrum

Appréhender Scrum et les principes des méthodes agiles. Acquérir le vocabulaire spécifique et de comprendre les valeurs qui guident cette méthode. Etre en mesure d'évaluer dans quelle mesure Scrum peut être adapté à votre organisation

Introduction aux méthodes agiles
- Les principes de gestion de projet
- Du processus en V vers les processus agiles
- La gestion des risques au cœur de l'organisation
- Les origines des méthodes agiles
- Les principales méthodes agiles (XP et scrum) et leurs grands principes
- Le fonctionnement des cycles de Scrum

Les rôles dans Scrum
- Une organisation par auto-gestion : la responsabilité collective de la livraison
- Le Directeur de produit et le 'Product Backlog'
- Le ScrumMaster pour coacher et protèger l'équipe

L'organisation du projet agile
- La construction et la gestion du Release Plan
- Les sprints
- L'organisation de l'espace de travail
- L'animation de l'équipe et de sa communication
- La collaboration dans l'équipe

La gestion des besoins et des exigences
- Établir la vision
- Extraire le Product Backlog
- Le démarrage : itération zéro

Déroulement d'une itération
- La planification de l'itération
- La construction et le suivie de l'itération backlog
- L'organisation en Features Teams
- La rétrospective d'itération et le calcul de la vélocité

L'organisation au quotidien
- Le Scrum Meeting
- L'affectation des tâches
- Le Test Driven Development (TDD)

Synthèse
- La planification dans scrum
- Vendre scrum aux décideurs

Public:

Chefs de projets ou responsables qualité dans le développement logiciel

Pré-requis:

Aucun

Durée:

1 jour

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

UML pour Chefs de Projet

Comprendre les principaux diagrammes UML dans le cadre de l'expression des besoins et de l'analyse.

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 scenarii

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 ?

Public:

Chefs de projets, architectes fonctionnels

Pré-requis:

Aucun

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

UML pour développeurs

Comprendre les principes de diagrammes d'UML dans leurs aspects techniques, liés à la conception et au développement.
Savoir évaluer l'utilité d'UML pour votre organisation.

Introduction à UML

- La démarche projet
- La modélisation objet
- La mise en œuvre d'UML

Introduction à l'expression de besoins et à l'analyse

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

Utilisation d'UML en conception

- Le diagramme de classes en conception
- La différence entre classes d'analyse et classes de conception
- La notion d'interface
- Les diagrammes de séquence et de communication

Finalisation du système et compléments

- La structuration du système : architecture n-tiers
- La persistance des données
- Les Design Patterns
- Les outils de modélisation UML
- UML dans Le processus unifié et dans RUP (Rational Unified Process)
- La modélisation dans les méthodes agiles (XP, Scrum)

Conclusion

Public:

Chefs de projets et concepteurs

Pré-requis:

Pratique d'un langage orienté objet, comme Java, C++, C# ou PHP.

Durée:

2 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

UML 2 : Analyse et Conception

Acquérir les connaissances nécessaires à l'utilisation d'UML en expression des besoins et en analyse.
Introduction à la modélisation UML en conception.

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'objets
- 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
- Les démarches pour UML
- Le Processus Unifié (UP ou Unified Process)
- Rational Unified Process (RUP)
- Les méthodes agiles : eXtreme Programming (XP) et SCRUM
- Conclusion

Public:

Chefs de projets, architectes fonctionnels, analystes et concepteurs souhaitant utiliser UML dans leurs projets.

Pré-requis:

Une pratique de projets informatiques est nécessaire et des notions d'objets sont un plus pour pouvoir suivre efficacement cette formation. Aucune pratique de programmation requises.

Durée:

4 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Introduction à la programmation iPhone

- Initiation au développement d'applications iPhone.
- Découverte des bonnes pratiques de conception et de design d'interface.

Découverte de l'environnement de développement

L'essentiel d'Objective-C

Premier pas avec le SDK iPhone

Méthode de développement d'un projet iPhone
- Principes ergonomiques et design patterns d'interface

Conception de l'interface graphique
- Le développement de l'interface
- Contrôler les écrans de l'application
- Assembler les écrans de l'application
- Développer et animer les vues
- Liste d'éléments

La manipulation des données
- Lire et enregistrer les données

Communiquer avec l'extérieur

Persistance d'objets avec CoreData

Manipuler des données multimédias

Utiliser les API de notifications

La publication des applications
- Publier sur l'AppStore

Public:

Toute personne souhaitant apprendre à développer des applications pour IPhone.

Pré-requis:

Ce cours s'adresse aux développeurs ayant déjà une expérience significative de la programmation avec un langage objet : Java, C++ ou mieux Objective-C.

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Initiation aux techniques du web sémantique

Le fonctionnement du Web sémantique repose sur l'utilisation de descriptions exprimées en RDF (Resource Description Framework). L'utilisation et la conception de descriptions RDF supposent une bonne compréhension des utilisateurs de la logique formelle de description des ressources sur le Web sémantique.

Une ontologie représente une compréhension commune et partagée d'un

Introduction au Web sémantique
- Principes du Web sémantique
- Technologies et architectures sémantiques
- XML, RDF et ontologie

Formats RDF
- Définition
- RDF / XML
- N3 et N-Triple
- Mise en oeuvre dans un wiki sémantique
- Mise en œuvre dans réseaux et sites de partage

Conception d'ontologie
- Introduction
- RDF Schema
- Langage OWL
- Langage OWL-Lite
- Langage OWL-DL
- Langage OWL-Full
- Inférence

Réutilisation d'ontologies
- Dublin Core
- FOAF

Intégration RDF / Web
- Architectures d'intégration
- Microformats
- RDFa
- GRDDL

Stockage et Recherche
- Triple Store
- SPARQL

Architectures et solutions sémantiques
- SOA sémantique et mashups
- Architecture Jena

Public:

Cette formation est conçue pour les développeurs, concepteurs et architectes connaissant les technologies Web et souhaitant franchir un palier sémantique.

Pré-requis:

Bonnes connaissances des Technologies Web.
Outils utilisés : Semantic Mediawiki, Protégé et Jena
Nous étudierons aussi la façon dont ces techniques sont utilisées au sein de sites comme Twine ou DBpedia.

Durée:

3 jours

Prix:

4990 CHF HT / pers.

Date

Nous consulter

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Ajax et le Web 2.0

- Apprendre à développer des applications Web présentant une fluidité impossible à atteindre avec une approche traditionnelle.
- Découvrir les différentes composantes d'AJAX et leur mise en œuvre au travers de la réalisation d'applications Web dynamiques.

Introduction
- Différences entre le modèle Web traditionnel et le modèle AJAX
- Quand et comment utiliser AJAX

Rappel des caractéristiques du langage JavaScript
- Syntaxe de base
- Fonctions et objets prédéfinis
- Gestion des événements
- Closure et Prototypes

Chargement asynchrone des données grâce à XML
- Description du mécanisme
- Frame cachées
- Iframes cachées
- L'objet XMLHttpRequest

Principaux inconvénients des sites Ajax
- Gestion des historiques
- Gestion des favoris
- Nouvelles conventions graphiques

Manipulation de documents XML avec Dom
- L'arbre DOM
- Passage du format Fichier plat au format DOM
- Parcours de documents XML
- Modification de documents XML

Le format de données JSON
- Syntaxe de base
- Cas d'utilisation

Gestion des différents browser
- Principales différences
- Développer un code portable

Présentation de XSL-T
- Introduction à XSL-T et XPath
- Utilisation combinée XSL-T/AJAX
- Recherche d'informations avec XPath et XSL-T

Exemples de framework AJAX
- Ajax côté serveur (Java et J2EE): DWR, JSON-RPC-Java
- Ajax côté client (JavaScript): Sarissa, Dojo
- Ajax côté serveur et client (Java, JavaScript): XWire, Swato

Présentation détaillée du toolkit Dojo
- Principe de fonctionnement
- Widgets proposées
- Gestion des événements
- Positionnement des widgets
- Gestion du bouton back et des favoris

Public:

Programmeurs désireux de développer des pages Web en utilisant les dernières technologies dites du "Web 2.0".

Pré-requis:

Pour suivre ce cours efficacement il est recommandé de connaître un langage de programmation: JavaScript, Java... et de connaître le langage HTML.

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Sécurité des applications J2EE

- Mieux connaître les techniques proposées par l'édition entreprise de java.
- Comprendre la façon de développer et déployer une application J2EE.

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

Public:

Tout public informatique

Pré-requis:

Aucun pré-requis particulier

Durée:

1 jour

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch

Informatica - Powercenter Développeur

- Formation pour les développeurs et mainteneurs sur les projets mettant en oeuvre INFORMATICA POWERCENTER pour atteindre une bonne maîtrise, voire une expertise sur le produit.

- Cette formation permet également une remise à niveau et une mise à jour pour les développeurs et chefs de projet ayant suivi une formation sur des versions antérieures.

1 Présentation générale & architecture

Points forts et Difficultés majeures

Architecture serveur
- implications sur le déploiement
- option : le « versionning »
- option : gestion « unicode »

Organiser la sécurité

2 PowerCenter 8 : approfondissements sur les transformations et optimisation de la conception

Processus de développement Informatica
- impact sur la conception et l'optimisation
- vers un « code » dynamique
- option : industrialiser développements et tests

Familles de transformation :
- Informatica et SQL des SGBD-R
- Les capacités spécifiques des transformations Informatica (agrégat, router, joiner, sorter, ...)

Lookups :
- les différents modes d'utilisation
- option : cache dynamique et persistant

Normalizer

3 Sources et Cibles

Paramétrages des Sources :
- paramétrage avancé
- paramétrage dynamique

Sources hiérarchiques :
- option : cobol ou vsam
- option : XML

Gestion de plusieurs fichiers en lecture

4 Les sessions et workflows

Gérer enchaînements et évènements

Notion de partitionning

Options de mise à jour des sessions : modes de chargement des données

-option : gestion des transactions

Usage du mode « recovery »

Interfaçage système : variables d'environnement et ordonanceurs (option)

5 Gestion des erreurs

Traitement des erreurs :
- gestion des tolérances
- utiliser les « logs » et corriger
- pilotage des traitements

6 Paramétrage

Public:

Les développeurs et mainteneurs de projet utilisant Informatica PowerCenter.

Pré-requis:

Cette formation est basée sur une mise en oeuvre de développements.

La pratique pendant au moins trois mois et sur au moins un projet est recommandée

Durée:

3 jours

Prix:

Nous consulter

Date

Tel: +41 (0) 22 301 72 20
www.oosphere.ch