Catalogue de formations

http://www.sewatech.fr

Liste des formations

Initiation au langage java

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.

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.

4 jours

Tarif

- intra : 4680 euros HT (maxi 6 participants)
- inter : 1960 euros HT

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 valeurs littérales et leur mise en forme
- 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
- Le multi-catch
- 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
- La manipulation de dates et d'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
- La clôture manuelle ou automatique des ressources

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

Approfondissement java

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 Initiation à java. Ces modules ont été conçus par 1/2 journée ou par journée pour être ajoutés à la formation d'initiation.

selon sélection

Tarif

- intra : 1170 euros HT / j (maxi 6 participants)

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 collections immuables
- Autres manipulations de collections et de tableaux

Entrées / 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

Nouvelles entrées / sorties (JavaSE 7)

- L'accès au système de fichiers (java.nio.file)
- La gestion des chemins (Path, Paths)
- La manipulation de fichiers (Files)

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
- L'API fork / join (JavaSE 7)

Expressions régulières

- Principe des expressions régulières
- Éléments de syntaxe : ., *, +, ?, \d, \s, \w, [], ()
- Manipulation de chaînes de caractères avec le package java.util.regex
- Formatage 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 « RessourceBundle »

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 et de l'isolation
- Appel de procédures stockées

Gérer les traces d'une application

- Principe de Apache Log4J
- Installer et configurer Log4J
- Utilisation du framework
- Utilisation combinée avec Apache commons-logging
- Utilisation combinée avec SLF4J
- (le même sujet peut être proposé en remplaçant Log4J par Logback)

Tests unitaires et refactoring

Les techniques de rectoring et de test unitaire sont particulièrement préconisées en java dans les démarches agiles et font partie des pratiques fondamentales de l'eXtrem Programming.

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

2 jours

Tarif

- intra : 2580 euros HT (maxi 6 participants)
- inter : 1180 euros HT

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

Java 8 : developper avec les lambdas et streams

La version 8 de Java a apporté pas mal de modifications dont une nouvelle syntaxe : les expressions lambda. Ces dernières ont un impact énorme sur les APIs et sur notre façon d'écrire du code.

Dans cette formation, vous apprendrez à utiliser cette nouvelle syntaxe. Vous verrez comment l'utiliser pour exploiter les nouvelles APIs, en particulier celle des collections.

2 jours

Tarif

- intra : 2440 euros HT (maxi 6 participants)
- inter : 1080 euros HT

Introduction

- Classe anonyme, fonction et lambda : question de lisibilité
- Impact sur les collections : du pattern d'itération à map/reduce

Expression lambda

- Présentation de la nouvelle notation '-?'
- Présentation des différentes formes de lambda
- Compatibilité avec les interfaces (fonctionnelles) existantes
- L'inférence de type dans les lambda
- La notation par *method reference*
- Lambda et variable *final*

Interfaces fonctionnelles

- L'objectif de rétro-compatibilité
- Définir une interface fonctionnelle
- L'annotation @FunctionalInterface
- Les nouvelles interfaces fonctionnelles : Function, Predicate,...
- Les méthodes *default*

Collections et Streams

- Les changements dans l'API de collection
- Les nouveaux patterns pour Collection et Map
- Passage de Collection à Stream
- Création de streams (types primitifs, String,...)
- Le pattern *filter*, *map*, *collect*
- Collectors standards et personnalisés
- Optional, à la place de null

Programmation parallèle et concurrente en Java

En programmation Web, avec Java EE ou Spring Framework, l'essentiel de l'aspect multi-tâches est caché au développeur. Mais lorsqu'il s'agit de développer des batchs ou des traitements massifs, il faut souvent revenir aux racines des Threads et des fonctionnalités du JDK.

Dans cette formation, on commence pas manipuler les Threads de façon brute, comme en 1999, puis on voit les différentes classes et possibilités offertes par le JDK pour faciliter le développement.

2 jours

Tarif

- intra : 2580 euros HT (maxi 6 participants)
- inter : 1180 euros HT

Introduction

- Les bénéfices des traitements parallèles
- Les risques liés aux accès concurrents

Premiers pas

- La classe Thread
- Les tâches Runnable
- Les techniques de blocage de thread
- Comment interrompre un thread ?
- La gestion des erreurs

Gestion moderne des threads

- Le framework Executor
- Les pools de threads
- Les tâches Callable, le Future
- L'interruption de Threads gérés
- Le pool de Fork / Join

Accès concurrents

- Problèmes d'accès concurrent aux données et ressources
- Le dead lock, leur détection
- Les objets atomiques
- Les techniques avancées de blocage : barrières et loquets
- Les collections concurrentes

Nouveautés Java 8

- L'utilisation de Lambda avec les classes et interfaces existantes
- Les traitements parallèles avec les tableaux
- Les collections et les streams parallèles
- La programmation de style *reactive*, avec CompletableFuture
- Quelques autres nouveautés : classes et méthodes

Java 9 : développer des modules

En 2009, la mort du classpath a été annoncée. Il devait être remplacé par un système modulaire et tous nos problèmes de dépendance et de sécurité devaient se résoudre d'eux-mêmes. Le projet Jigsaw est finalement intégré au JDK 9, en septembre 2017.

La modularité se retrouve à la fois dans le JDK et nos applications. Dans cette formation, nous expliquons comment la modularité se retrouve dans le JDK et nos applications. Nous voyons quel problèmes ça résout et quels nouveaux problème ça pose.

Il faut concevoir différemment les dépendances et l'encapsulation, préparons-nous.

2 jours

Tarif

- intra : 2440 euros HT (maxi 6 participants)
- inter : 1080 euros HT

Introduction

- Classpath et classloader
- Risque de classes hybrides
- Défauts de sécurité
- Problème d'obésité

Modularité du JDK

- De rt.jar aux modules
- Modules standards et module de base
- Modules dépréciés
- Modules non-standards
- Classes et packages supprimés
- Compilation, packaging et exécution

Modularité des applications

- Dépendances entre modules
- Export de packages
- Application multi-modules

Encapsulation des modules

- Évolution de la visibilité public
- Règles de répartition de packages
- Export de packages : globaux et limités
- Deep reflection
- Ouverture de package et de module

Dépendances entre modules

- Dépendances transitives
- Dépendances statiques
- Modules automatiques
- Mode mixte : module path / class path

Migration d'applications

- Option d'accès illégal

- Ajout de modules racine
- Ajout de dépendances
- Évaluation préalable des dépendances

Native

- Construction d'image personnalisée
- Compilation native

Docker pour les développeurs

La propagation de **Docker** est impressionnante depuis sa création en 2013. Pour quelle raisons ? Quels avantages peut-on en tirer en terme d'architecture et d'outillage ?

Ce cours vous permettra de comprendre le fonctionnement de Docker. Il vous permettra d acquérir les pratiques nécessaires à son installation et son utilisation en environnement de développement.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Principes de base

- Le fonctionnement de Docker
- Quelques usages, en production ou en développement
- L'écosystème de Docker

Premiers pas

- Installer Docker sous Linux, MacOS ou Windows
- Récupérer une image depuis le Hub
- Lancer un conteneur (hello-world)

Démarrer des conteneurs

- La commande *run*
- L'isolation des conteneurs
- Arrêter et redémarrer un conteneur
- Supprimer un conteneur

Construire une image

- Les commandes *commit* et *build*
- La structure d'un Dockerfile
- Les principales instructions Dockerfile
- Contrôler et limiter la taille d'une image ?

Mise en réseau

- Les types de réseau Docker
- L'exposition de ports
- Les liens entre containers
- Les réseaux virtuels entre containers

Utilisation de volumes

- Monter un volume pour partager un répertoire
- Les volumes de données
- Les conteneurs de données

Registre d'images

- Organisation du Hub
- Les catégories d'images : officielles, publiques, privées
- Publier sur le Hub
- Installer et utiliser un miroir ou un registre local

Docker Compose

- Un environnement multi-conteneurs
- Installer Compose
- La structure du fichier docker-compose.yml
- Les principales commandes

Docker dans l'environnement de développement

- Les bases de données (MySQL, Postgres, Oracle, MongoDB)
- Les serveurs Web (Apache, nginx) ou d'applications (NodeJS, Tomcat)
- L'IDE (Eclipse)

Intégration continue

- Utiliser des conteneurs de build
- Intégrer la construction des images dans la chaine de build
- Utiliser Docker pour les tests d'intégration

Conclusion

- Utiliser Docker en architecture MicroServices

Administration JBoss / WildFly

WildFly est le successeur de JBoss AS 7 et sert de base aux dernières versions de **JBoss EAP**. Cette formation couvre WildFly et est compatible à 90% avec JBoss EAP 6 et à 95% avec JBoss EAP 7. La politique de distribution de JBoss / WildFly, avec des versions communautaires et *productisées*, sera expliquée pendant la session.

Ce cours vous permettra de comprendre le fonctionnement et les principes de configuration de WildFly. Il vous permettra d acquérir les pratiques nécessaires à son administration (déploiement, logging, monitoring, sécurité...) et la connaissance des outils comme JBoss CLI qui vous permettront de mieux automatiser vos tâches. Vous étudierez aussi les leviers qui permettent d obtenir la qualité de service attendue pour vos applications.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Introduction à Java EE et WildFly

- Présentation de Java et de Java EE
- Typologie des applications Java EE
- Présentation de WildFly, JBoss AS et JBoss EAP

Installation de WildFly

- Installation, démarrage et arrêt
- Installation en service
- Modes alternatifs d'installation

Configuration standalone

- Mode autonome ou domaine
- Principaux éléments de configuration
- Outils d'administration : console web, jboss-cli, APIs
- Configuration réseau

Déploiement d applications et de modules

- Déploiement d applications (ear, war, jar,...)
- Déploiement automatique ou manuel
- Déploiement par script
- Gestion des dépendances avec les modules
- Déploiement de DataSource
- Déploiement de destinations JMS

Administration d un domaine WildFly

- Introduction au mode domaine
- Configuration d un Domain Controller
- Configuration d un Host Controller
- Pilotage d'un domaine

Gestion des traces

- Traces de la JVM
-

Traces d accès Web
- Traces du serveur
- Traces des applications (Log4J, SLF4J)
- Traces des outils d'administration

Inspection du serveur
- Console d administration
- Commandes par script
- Monitoring par commandes HTTP
- Outils du JDK

Optimisation des performances
- Tuning de la machine virtuelle
- Gestion de la mémoire et du Garbage Collector
- Dimensionnement des pools (EJB, DataSource, threads)

Sécurité du serveur et des applications
- Objectifs de sécurisation du serveur
- Sécurisation des interfaces d administration
- Gestion des autorisations et des authentifications pour les applications
- Sécurisation des échanges avec SSL

(en option, 1 journée) Clustering WildFly
- Objectifs du clustering : tolérance de panne (failover) et à la répartition de charge (load balancing)
- Répartition des invocations EJB
- Répartition des requêtes HTTP
- Synchronisation des états
- Répartition de la charge JMS avec ActiveMQ Artemis (HornetQ)
- Tolérance de panne JMS avec ActiveMQ Artemis

JBoss / Wildfly en cluster

Wildfly est le leader des serveurs applicatifs Java EE libres et de plus en plus d'entreprises le choisissent au détriment de ses concurrents propriétaires.

Ce cours avancé s adresse à tous ceux qui souhaitent mettre en place un environnement Wildfly ou JBoss EAP en cluster. Les différents types de réplication et de répartition de charge sont abordés, sur les aspects HTTP, EJB et JMS. A chaque étape, un atelier vous permet de mettre la théorie en pratique. A la fin du cours, vous aurez les armes pour déployer des applications Java EE avec un haut niveau de disponibilité et une bonne scalabilité.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Introduction

Concepts du clustering

- Quelques définitions : scalabilté, membership, farming, failover, ...
- Typologie des clusters
- Répartition de charge contre haute-disponibilité

Fonctionnalités de clustering de Wildfly

- Les sous-systèmes impliqués dans le clustering
- Les mécanismes de répartition
- La réplication des objets à état
- Clustering vertical contre clustering horizontal
- Les configurations prédéfinies dans Wildfly
- L'administration en domaine

Communication entre nSuds avec **JGroups**

- Rappels réseau : IP, TCP / UDP, multicast / unicast
- Les canaux et piles de protocoles
- Les protocoles présents dans les configurations par défaut de JBoss
- Les principaux paramètres de configuration

Répartition HTTP

Apache en distributeur de charge

- Comment éviter le Single Point of Failure ?
- Choix entre une répartition HTTP et un distributeur matériel
- Les techniques de répartition Apache : mod_jk, mod_proxy, mod_cluster

La solution JBoss : **mod_cluster**

- Les différents modules Apache
- La configuration dans Wildfly
- La configuration coté Apache
- Le calcul de charge des nSuds
- La gestion via JBoss CLI

Réplication de sessions HTTP

Cache distribué avec **Infinispan**

- Les types de distribution et de réplication supportés par Infinispan
- La notion de cache-container
- Les options de synchronisation : concurrence et isolation
- Adaptation de la pile JGroups en fonction du type de réplication
- Monitoring d'Infinispan

Réplication de session HTTP

- Réplication ou distribution ?
- Les Configurations par défaut
- Les descripteurs de déploiement Web

Clustering d EJBs

EJBs session

- Rappels sur la configuration des pools
- Les algorithme de répartition de charge
- EJBs distants / EJBs locaux
- Répartition de charge des EJB stateless
- Répartition de charge et synchronisation des EJB stateful
- Configuration par défaut d'Infinispan

Entités JPA

- Infinispan et l'invalidation
- Le cache de second-niveau d'Hibernate
- Configuration par défaut de JBoss
- Les annotations spécifiques Hibernate

Clustering JMS

Fonctionnalités **ActiveMQ Artemis**

- Groupe de serveur
- Découverte automatique
- Répartition de charge
- Répartition des connexions clientes
- Redistribution de messages

Clustering ActiveMQ Artemis

- Configuration prédéfinies Jboss
- Mise en place de la redistribution
- Usine à connexions HA
- Mise en place de serveurs de backup
- Clients JMS d'un cluster

Administration Tomcat

Apache Tomcat est le serveur le plus populaire pour le déploiement d'applications Java Web. Open Source, facile à mettre en place et capable de très bonnes montées en charge : Tomcat cumule beaucoup de qualités. En revanche, il demande une montée en compétences pour exploiter correctement ces qualités.

Ce cours vous apprendra à installer et configurer Tomcat, à y déployer des applications Web. Vous saurez aussi assurer son suivi et son tuning, ainsi que celui de la machine virtuelle. La sécurité du serveur et des applications sera aussi approfondie. Enfin, l'installation derrière un serveur frontal Apache, ainsi que les architectures de haute disponibilité seront abordées.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Introduction à Java EE et à Tomcat
- Les principes fondamentaux de Java et de Java Enterprise Edition
- Les principaux composants de Java EE
- Une introduction à XML
- La fondation Apache
- Apache Tomcat

Installation et configuration
- L'installation de Tomcat : ligne de commande ou service
- Le démarrage de Tomcat : scripts et variables
- L'architecture Tomcat : Host, Engine, Service et Connector
- Les principes de configuration

Déploiement dans Tomcat
- Le déploiement d'applications
- Le répertoire de déploiement
- Le déploiement par contexte
- Le gestionnaire d applications
- L'installation des librairies
- L'installation d'une DataSource

Sécurité
- La sécurisation du serveur
- L'authentification et les autorisations
- La configuration des Realms
- L'authentification JAAS
- Le protocole SSL

Monitoring et Gestion des traces
- Une introduction aux traces
- Java Logging API et JULI
- Log4J, configuration et intégration à Tomcat

Administration Tomcat en cluster

Ce cours avancé vous permettra de mettre en place des architectures en cluster avec Tomcat.

Vous verrez le déploiement en répartition de charge avec un serveur frontal Apache et vous verrez les modes de réplication de session.

2 jours

Tarif

- intra : 2580 euros HT (maxi 6 participants)
- inter : 1180 euros HT

Architecture de déploiement et services de Clustering
- Installation de Tomcat autonome
- Installation avec un frontal Web
- Présentation des solutions : IIS ou Apache, HTTP ou AJP, mod_proxy ou mod_jk
- Définition d'un Cluster
- Services de répartition de charge et de
- Services de haute disponibilité

Installation avec Apache en reverse proxy
- Apache avec mod_proxy
- Impact de cette architecture sur les logs
- Gestion des en-têtes HTTP : RemoteIpValve, préservation du host
- Gestion des contextes Web
- Gestion des hôtes virtuels
- Partage et déport de ressources statiques
- Chiffrement SSL entre Apache et Tomcat
- Support du protocole AJP
- Installation alternative avec le mod_jk

Gestion du Load Balancing
- Répartition de charge entre plusieurs Tomcat avec mod_proxy_balancer
- Suivi de la répartition avec balancer_manager
- Algorithmes et critères de répartition
- Affinité de session et gestion des routes, impact sur les sessions
- Limites de cette architecture : la sensibilité aux pannes

Service de haute disponibilité
- Sessions et tolérance de panne
- Composant Cluster dans Tomcat
- Synchronisation des sessions par réplication
- Gestion dynamique des membres du cluster par multicast
- Gestion statique des membres
- Installation de grappes de Tomcat
- Déploiement d'applications par farming

Compléments

Java Optimisation

L'objectif de cette formation est de comprendre comment optimiser les performances d'une application Java.

Elle fait le tour des outils permettant l'inspection, le monitoring et le profiling de la machine virtuelle et des applications, en se concentrant sur les outils fournis avec le JDK. Puis elle aborde les aspects théoriques du fonctionnement de la machine virtuelle Java, en particulier les threads et la mémoire.

Enfin, elle termine avec un atelier pour replacer tous ces sujets dans la démarche d'optimisation.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Introduction

- La démarche d'optimisation
- Les objectifs de performance
- L'importance des tests et de leur environnement

Outils d'inspection et de monitoring

- Profiling vs monitoring
- Utiliser JMX pour le monitoring
- Les protocoles et connecteurs JMX
- Comment développer un MBean ?
- Les outils du JDK
- Un focus sur Visual VM
- Quelques outils tiers

Optimisation mémoire

- La structure de la mémoire Java
- Le paramétrage de la mémoire
- Le(s) Garbage Collector(s)
- Comprendre les erreurs OutOfMemoryError
- Générer et analyser un Heap Dump
- Le profiling mémoire

Optimisation des threads

- Lire et comprendre une Stacktrace
- Générer et analyser un Thread Dump
- Suivre la consommation CPU par thread
- Détecter un deadlock

Spécificités Tomcat

- Le réglage des pools
- Le pool de connexions (Datasources)
- Les connecteurs Coyote
- Développer et configurer des valves et listeners

Atelier de synthèse

Bonnes Pratiques Java EE

Le développement java, et Java EE 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.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

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, IntelliJ)
- Améliorer la productivité de l'équipe (Git, Maven, Jenkins)
- Préparer l'exploitation avec de bonnes traces (Apache Log4J et SLF4J)
- Suivre la mémoire (JConsole, VisualVM)

Le suivi de la qualité

- Les différents types de tests
- La mise en Suvre des tests unitaires automatisés
- L'automatisation des tests d'intégration
- Les outils de mesure de la qualité (SonarQube, PMD,...)

Java EE : architectures et frameworks

La première partie fait l'objet de questions / réponses afin que chacun puisse relater ses propres expériences. Cette partie a pour but de fixer les objectifs par rapport à l'architecture et aux frameworks.

La deuxième partie présente les principes d'architectures, les solutions apportées par celles-ci, ainsi que leurs inconvénients. Cette partie peut intéresser tous les développeurs et concepteurs, sans pré-requis techniques.

La troisième partie se concentre sur les technologies Java EE et les frameworks associés.

2 jours

Tarif

- intra : 3420 euros HT (maxi 6 participants)
- inter : 1480 euros HT

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 alternatives Open Source
  - Les solutions propriétaires

Principes d'architecture

- 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

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, jOOQ

- 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, JAX-RS
  - Les concurrents de JSF : Spring MVC
  - Les principes d'AJAX et des single page applications
  - Les outils et frameworks JavaScript : jQuery, Angular,...

- Les techniques d'intégration
  - Intégration verticale ou horizontale
  - Les Web Services : JAX-WS, Axis, CXF et Spring WS
  - Le messaging avec JMS

- Synthèse
  - Les bons assemblages et incompatibilités
  - Les accélérateurs : Spring Boot, JHipster et JBoss Forge
  - Une stratégie d'évolution vers un framework

Sécurité des applications Java EE

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 JavaEE.

1 jour

Tarif

- intra : 1710 euros HT (maxi 6 participants)
- inter : 740 euros HT

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 chiffrement 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 Java EE 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 Spring Security
- Techniques propriétaires : exemple des valves de Tomcat

SSL avec Java

- Fonctionnalités de Java SSE (Secure Socket Extension)
- Certificats X.509
- Protocoles TLS et SSL
- Chiffrement 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).

Développement d'applications Java EE

Cette formation vous apprendra à élaborer une architecture avec les techniques de JavaEE.

Grâce à ses ateliers pratiques, vous saurez comment développer simplement une application basée sur JSF, EJB, JPA, Bean Validation et CDI, ainsi que JAX-WS et RS.

5 jours

Tarif

- intra : 6450 euros HT (maxi 6 participants)
- inter : 2950 euros HT

Introduction

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

Composants et dépendances

- Les composants métier EJB 3.x
- 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
- 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.x
- 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 Java EE
- Comparaison avec Spring Framework

CDI, le coeur de Java EE

CDI, Context and Dependency Injection, est une nouveauté majeure de Java EE 6. Java EE 5 ne permettait l'injection que d'EJB et de ressources gérées par le serveur d'application. En comparaison de Spring Framework, cette version de la spécification manquait cruellement de souplesse. CDI permet d'injecter des composants plus variés et propose des techniques d'événements, d'interception ou de décoration. CDI devient donc un élément essentiel de Java EE.

Cette formation permet de prendre en main les techniques proposées par CDI : injection, interception, événements. Elle aborde aussi les techniques d'extension à CDI de Weld.

3 jours

Tarif

- intra : 3660 euros HT (maxi 6 participants)
- inter : 1620 euros HT

Principes d'architecture

- La plate-forme Java EE
- Les frameworks d'injection
- Les techniques d'injection

Premiers pas avec CDI

- Implémenter le composant
- Tester le composant
- Configurer CDI : fichier beans.xml

Manipuler des beans

- La définition d'un bean CDI
- L'injection de bean
- Le cycle de vie et les portées
- Les fabriques de beans

Enrichissement des beans

- L'interception de méthodes
- La décoration
- Le mécanisme d'événements

Intégration dans Java EE

- L'intégration avec JSF
- L'intégration avec JPA
- L'intégration avec EJB

Extensions à CDI

- Weld : implémentation de référence et extensions
- La gestion des exceptions
- la persistance et les transactions
- Les techniques d'extension

EJB 3, les bases

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.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Introduction

- Les architectures multi-tiers et les architectures distribuées
- La plateforme Java EE : les concepts, les composants et les services
- Les principes des Enterprise Java Beans et leur utilisation dans l'architecture

La mise en Suvre 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 Suvre 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'Entities

- Les principes du mapping objet/relationnel
- La nouvelle technique de persistance : JPA
- Le développement d'une entité 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 entités

Synthèse

- L'intégration des EJB dans l'architecture Java EE
- Les frameworks alternatifs aux EJB

Mapping O/R avec JPA

L'API de persistance de donnée JPA est l'aboutissement logique des travaux autour de Hibernate ou Toplink. Elle vise à standardiser l'usage de ces frameworks de mapping objet / relationnel.

Ce cours vous permettra de comprendre les logiques de mapping entre une base de données relationnelle et les classes d'un langage orienté objet. Il abordera les techniques d'annotations permettant de mettre en oeuvre ce mapping, ainsi que le langage de requête JPQL (Java Persistence Query Language). Enfin, il traitera la question des transactions, en environnement Java SE ou Java EE, avec ou sans les EJB.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

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

JUnit pour tests unitaires et d'intégration

Cette formation permet de comprendre les enjeux et les techniques des tests unitaires et d'intégration, avec la mise en Suvre de jUnit et d outils complémentaires, comme les Mock Objects ou Maven pour l automatisation.

Elle aborde aussi toutes les bonnes pratiques nécessaires à la réalisation de tests efficaces et à l'élaboration d'une architecture pleinement compatible avec les tests unitaires. A l'issue de cette formation, vous serez aussi en mesure d'exécuter vos tests dans un environnement d'intégration continue.

3 jours

Tarif

- intra : 3660 euros HT (maxi 6 participants)
- inter : 1620 euros HT

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 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 et les runners
- Le traitement des exceptions

Assertions

- Les assertions de JUnit
- De meilleurs assertions avec HamCrest
- Améliorer la fluidité des assertions avec AssertJ

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 Suvre avec Mockito
- Les fonctionnalités supplémentaires de PowerMock

Bonnes pratiques de tests

- L'organisation en packages
- Les conventions de nommage
- L'indépendance et l'isolation des tests

- Trouver la bonne granularité
- Gérer la durée et la fréquence des tests
- Réaliser des tests aux limites

Automatisation des tests
- Le principe de l'intégration continue
- La place des tests en intégration continue
- L'automatisation avec Maven
- La configuration des plug-ins Surefire et Failsafe
- La mise en Suvre avec Jenkins CI

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
- Le problème des méthodes *static*
- La gestion des dates

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

Tests d'intégration
- La différence avec les tests unitaires
- L'intégration avec la base de données
- Les outils DbUnit et DbSetup
- Tester les applications Web avec HttpUnit et Selenium

Tests en architecture JavaEE
- Rappel sur les architectures JavaEE
- Les tests de composants EJB 3
- Les tests des classes d'affichage JSF
- Les tests des classes persistantes JPA
- Arquillian pour faciliter les tests d'intégration

Tests avec Spring Framework
- Tests unitaires avec Spring
- Bonnes pratiques Spring
- Spring / JUnit
- Gestion des scopes
- Ressources autonomes et mocks

Fonctionnalités avancés de JUnit
- Les Rules
- Les tests paramétrés et les théories
- L'organisation des tests en catégories
- Les outils complémentaires Unitils
- Les plugins pour Eclipse : MoreUnit et Infinitest

Synthèse et Conclusion

- Intégrer les tests unitaires dans la démarche
- Que faut-il tester en priorité ?
- Quels types de tests sont les plus rentables ?

Spring Framework pour le développement d'application Java

Spring est un framework qui simplifie considérablement la programmation Java EE et encourage les bonnes pratiques de conception objet. Il fournit une couche d'abstraction qui permet d'intégrer facilement l'ensemble des technologies Java EE (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 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.

5 jours

Tarif

- intra : 6450 euros HT (maxi 6 participants)
- inter : 2950 euros HT

Les principes fondamentaux de Spring Framework

- Les techniques de développement Java
- 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 configuration XML ou par annotations

La manipulation de beans Spring

- Le conteneur de composants : BeanFactory et ApplicationContext
- La définition des beans
- Les techniques d'injection : setter et annotation
- La gestion des dépendances
- L'autowiring
- L'héritage de beans
- Les interfaces et méthodes de callback
- Les post-processeurs

Utiliser Spring pour les données persistantes

- Présentation des techniques de persistances
- Développer une DAO avec Spring JDBC
- Développer une DAO avec Spring / Hibernate

Gérer les transactions

- Rappel sur le concept de transaction
- Les transactions avec Spring
- Les transactions gérées par programmation
- Les transactions déclaratives
- La configuration pour Hibernate

Développer une application web avec Spring Framework

- La programmation Struts avec Spring
- L'intégration de Spring avec JSF
- Présentation des frameworks Spring MVC et Spring Web Flow

Créer des composants évolués avec Spring AOP et le module de sécurité

- Présentation du module Spring AOP
- Sécuriser une application avec le module de sécurité Spring Security

Appels distants avec Spring

- Présentation des techniques de Remoting (RMI, HttpInvoker,...)
- Le développement JMS avec Spring Framework
- Les services Web avec Spring Framework

Tester une application Spring

- Les bonnes pratiques de conception pour les tests
- Utiliser des ressources autonomes et des objets de mock
- L'intégration avec JUnit et TestNG

Spring : les bases

Le framework Spring offre de nombreuses possibilités au développeur. Acquérir des connaissances sur l'ensemble des fonctionnalités peut s'avérer long et coûteux. C'est pour cela que cette formation se concentre sur l'essentiel des fonctionnalités de Spring, pour plus d'efficacité.

A l'issue de cette formation, vous serez en mesure de développer des applications JavaEE basées sur Spring.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

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
- Les fonctionnalités du framework Spring

Premiers pas avec Spring

- L'implémentation dans une classe
- La configuration du conteneur
- L'accès au bean

Manipulation les beans Spring

- La définition des beans et les méthodes de fabrique
- L'injection des dépendances
- La portée des beans
- La configuration XML ou par annotations
- L'externalisation par properties

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 / JSF

- Rappels sur la programmation avec JSF
- L'intégration des beans JSF

Test de composants Spring

- L'intégration avec JUnit
- L'ApplicationContext dans les tests (mode dirty)
- Les transactions dans les tests

Sécurité avec Spring Security

- Présentation du module Spring Security
- La gestion des autorisations
- La gestion de l'authentification

Mapping O/R avec Hibernate

Objectifs :

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

3 jours

Tarif

- intra : 3870 euros HT
(maxi 6 participants)
- inter : 1770 euros HT

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

Développement Web avec Vert.X

La légende prétend que Eclipse Vert.x serait issu d'un croisement entre Node.JS et Java. Ce qui est certain, c'est que c'est une boite à outil qui permet entre autres de développer des applications Web modernes, sans serveur d'application.

Dans cette formation, vous apprendrez à utiliser les API de Vert.X pour le développement Web. Puis en étudiant ses principales fonctionnalités, vous arriverez à l'utiliser pour mettre en place une architecture réactive.

3 jours

Tarif

- intra : 3870 euros HT (maxi 6 participants)
- inter : 1770 euros HT

Introduction

- Avantages de la programmation non-bloquante
- Techniques de programmation asynchrones en Java : callback, (completable) future, Rx
- Architecture par thread pool contre architecture par event-loop
- Champs d'utilisation : Web, IoT, µServices

Premiers pas

- Définition d'un verticle
- Démarrage d'un serveur Web
- Gestion des erreurs
- Interactions avec l'event bus

Architecture d'une application

- Threads et exploitation des processeurs
- Installation et configuration d'un verticle
- Verticles et découpage métier
- Intégration de services bloquants avec les service workers
- Simplification des interactions avec les service proxies
- Déploiement local ou déploiement distribué

RxJava

- Programmation par callback contre programmation réactive
- Principes de l'API RxJava
- Amélioration de la qualité de service : timeout et retry

Développement Web

- Serveur HTTP, options avancées
- Organisation des routes
- Client HTTP

Intégration

- Interaction avec le système de fichiers
- Principaux protocoles : HTTP/1, HTTP/2, WebSocket, MQTT,&
- Base de données relationnelles avec JDBC
- Cas particuliers de PosgreSQL et MySQL
- MongoDB, Kafka

Tests

- Test unitaire et mock
- Test d'intégration, par service

Préparation au déploiement

- Health check
- Publication de métriques avec Vert.x metrics et Prometheus
- Service discovery, circuit breaker
- Déploiement avec Docker, prise en compte des contraintes mémoire et processeur