

LE CATALOGUE DES FORMATIONS DE COGITEC

COGITEC vous propose ici toute l'étendue de ses compétences à travers son catalogue.

C/C++

- + Développer des services Web en C++ - page 4 -
- + Développer en langage C - page 5 -
- + Développer en langage C++ - page 6 -
- + Langage C++, aspects avancés - page 7 -

Client léger

- + Les technologies du client léger - page 9 -
- + Mise en oeuvre d'AJAX en environnement JEE - page 10 -

Informatique générale

- + Les bases de l'informatique - page 12 -
- + Technologies Objet : les standards actuels - page 13 -

Java

- + Développer des clients riches en Java - page 15 -
- + Java pour développeurs d'applications stand-alone - page 16 -
- + Java pour développeurs d'applications Web - page 17 -
- + Java pour développeurs objet - page 18 -
- + Java, programmation avancée et outillage de référence - page 19 -
- + Programmation Java avec Eclipse - page 20 -

JEE (J2EE)

- + Conception d'une application Web en J2EE - page 23 -
- + Développement JEE à l'aide des frameworks Hibernate, Spring et Struts2 - page 24 -
- + Développer des applications JEE multi-niveaux - page 26 -
- + Développer des applications Web à l'aide de STRUTS 1.x - page 27 -
- + Développer des applications Web avec STRUTS 2.x - page 28 -
- + Développer des applications Web en SERVLET/JSP - page 30 -
- + Développer des services Web en Java - page 31 -
- + Développer une couche de persistance avec Hibernate - page 32 -
- + Persistance Java avec l'API JDBC - page 34 -
- + Plate-forme JEE, état de l'art - page 35 -

Langages de script

- + Développer en JavaScript - page 38 -
- + Développer en langage Perl - page 39 -

Méthodes et langage UML

- + Analyse et conception d'applications logicielles en UML - page 41 -
- + Conception UML-C++, utilisation des Design Patterns - page 43 -
- + Conception UML-Java, utilisation des Design Patterns - page 44 -
- + Introduction à l'agilité - page 45 -
- + Introduction à un processus de développement UML - page 47 -
- + Notation UML et concepts OBJET - page 48 -
- + UML pour la formalisation des besoins - page 49 -

Technologies XML

- + Développer des applications XML en Java - page 51 -

- + Présentation des technologies XML
- + Utiliser XML dans le cadre d'applications Microsoft .Net
- + XML, concepts et techniques de mise en oeuvre

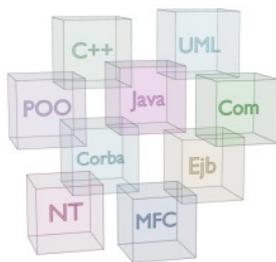
- page 53 -
- page 54 -
- page 55 -

Unix/Linux

- + Linux, utilisation d'une station de travail

- page 57 -

C/C++



ENVIRONNEMENTS UTILISES

Toolkit gSOAP ou Apache Axis C++, IDE Eclipse 3.2 et plug-in CDT

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de concevoir et implémenter une architecture orientée services basée sur les services Web lors de développements d'applications distribuées en C++ :

- Identifier les besoins justifiant l'utilisation de services Web
- Spécifier, développer et déployer un service Web
- Rechercher et exploiter un service Web sur Internet
- Sécuriser l'utilisation de services Web

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Construire des applications C++ maintenables et évolutives en utilisant l'héritage et le polymorphisme

METHODE

Les aspects théoriques sont concrétisés par le développement d'une véritable application distribuée à base de Services Web. Un poste de travail est fourni par stagiaire.

PROGRAMME

INTRODUCTION AUX SERVICES WEB

Rappel sur l'architecture n-tiers
Services Web, pour quel besoin
Typologie de services Web : SOAP et XML-RPC
Architectures cibles des Services Web
Développement, administration et orchestrations
Aspects commerciaux et évolutions futures

XML, LE LANGAGE DES SERVICES WEB

Présentation
Standards dérivés de la norme XML (XML namespace, XML schéma)
Messages XML entre applications et services Web

DESCRIPTION D'UN SERVICE WEB

Introduction à WSDL
Les étapes de création
Structure d'un document WSDL
Éléments du langage
Création d'un fichier WSDL décrivant un service Web

CREATION D'UN SERVICE WEB

La chaîne de production gSOAP ou Apache Axis
Parser de fichiers WSDL : wsdl2h
Génération du stub et du skeleton : soapcpp2
Implémentation du client et du serveur
Déploiement du service web

INVOCATION D'UN SERVICE WEB (SOAP)

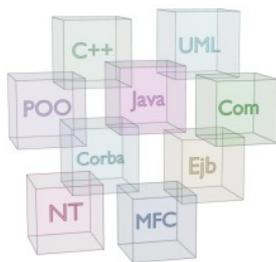
Présentation
Structure des messages SOAP
SOAP avec pièces jointes
SOAP via http
Exemples de requêtes et de réponses

DEPLOIEMENT DE SERVICES WEB SUR INTERNET (UDDI)

Objectifs de l'annuaire UDDI
Publication d'un service Web

SECURITE ET GESTION D'INTEGRITE

Intégrité des messages
Intégrité des transactions
Authentification d'accès



ENVIRONNEMENTS UTILISES

Chaîne de production gcc sous l'IDE Eclipse 3.2 et plug-in CDT.

PRIX

Inter-entreprise : 2010.0 € HT par stagiaire.

Intra-entreprise : 6633.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des programmes en langage C et en particulier de :

- Utiliser une chaîne de production C ANSI (éditeur, compilateur, linkeur, débogueur)
- Implémenter des algorithmes à l'aide des structures de contrôles C
- Développer des applications modulaires par décomposition fonctionnelle
- Manipuler les tableaux et les chaînes de caractères
- Accéder aux données par l'intermédiaire de pointeurs
- Créer ses propres structures de données
- Sauvegarder et restaurer des données à l'aide de fichier

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Chaque aspect théorique est illustré par un exemple concret de mise en œuvre puis exploité dans la réalisation incrémentale d'une véritable application en mode console. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

HISTORIQUE ET STRUCTURE D'UN PROGRAMME C

- Historique et caractéristiques
- Portabilité et normalisation (norme ANSI)
- Eléments lexicaux, les fichiers d'en-tête

TYPES DE BASE, OPERATEURS ET EXPRESSIONS

- Les types prédéfinis et leur taille
- Conversion de types
- Opérateurs, constantes et expressions
- Introduction aux entrées-sorties

LES STRUCTURES DE CONTROLE

- Alternative et sélecteur de cas
- Les structures itératives while
- Les ruptures de séquences

LES FONCTIONS

- Définition de fonction
- Prototypage de fonctions
- Passage d'arguments par valeur
- La récursivité

LA MODULARITE

- Structure d'un module
- Les fichiers headers
- Les directives pré processeur
- Les différentes classes d'allocation mémoire

TABLEAUX ET CHAINE DE CARACTERES

- Déclaration de tableaux
- Manipulation de tableaux
- Passage de tableaux comme arguments de fonctions
- Chaînes de caractères et fonctions de manipulations
- Les tableaux multidimensionnels

LES POINTEURS

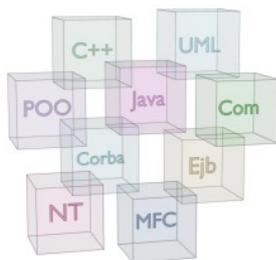
- définition et intérêt des pointeurs
- opérations sur les pointeurs
- Les tableaux de pointeurs
- Les pointeurs de fonctions
- passage d'arguments dans la fonction main
- allocation dynamique de mémoire

LES STRUCTURES

- Définition et opérations sur les structures
- Tableaux de structure
- Les unions
- Les champs de bits
- Les énumérations

LES FICHIERS

- Ouverture/fermeture de fichiers
- Lecture/écriture dans un fichier
- Les accès directs



OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications basées sur les mécanismes objets du langage C++ :

- Utiliser les aspects fortement typés du langage C++
- Développer des classes C++
- Construire des applications C++ maintenables et évolutives en utilisant l'héritage et le polymorphisme
- Exploiter les classes d'entrée/sortie C++ de la librairie standard
- Développer des templates et traiter les cas d'erreurs à l'aide des exceptions

ENVIRONNEMENTS UTILISES

Chaîne de production gcc sous l'IDE Eclipse 3.2 et plug-in CDT.

PRIX

Inter-entreprise : 2008.0 € HT par stagiaire.

Intra-entreprise : 6626.0 € HT (8 stagiaires).

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Implémenter des algorithmes à l'aide des structures de contrôles C
- Donner une définition des concepts objets et les bénéfices induits par leur utilisation

METHODE

Chaque aspect théorique est illustré par un exemple concret de mise en œuvre puis exploité dans la réalisation incrémentale d'une véritable application en mode console. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

LES ASPECTS FORTEMENT TYPES DU LANGAGE

- Référence et transparence d'appel
- Valeur par défaut des paramètres de fonctions
- Surcharge de fonction

LES CLASSES C++

- Définition d'attributs et de méthodes,
- Encapsulation : parties publiques, privées,
- Constructeurs et destructeurs,
- Techniques d'instanciation (pile et tas),
- Attributs et méthodes de classes (static)
- Les fonctions et classes amies (friend)

LA SURCHARGE D'OPERATEUR

- Principe de base,
- Opérateurs unaires et opérateurs binaire,
- Surcharge d'affectation,
- Surcharge des opérateurs de conversions

L'HERITAGE

- Classes de base et classes dérivées
- Constructeur et destructeur
- La partie protected
- Héritage multiple

LE POLYMORPHISME

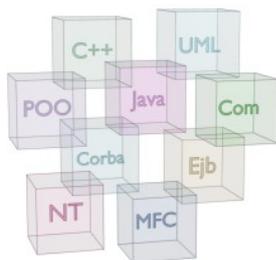
- Méthodes virtuelles,
- Infrastructure d'exécution,
- Destructeur virtuel,
- Classe abstraite et genericité
- Hiérarchie de classes

LES ENTREES SORTIES C++

- La hiérarchie de classe de la librairie iostream,
- Opérations sur les streams,
- Surcharge des opérateurs <<, >>
- Manipulation de fichiers

INTRODUCTION AUX ASPECTS AVANCES DU C++

- Les templates de classe,
- Les templates de fonctions,
- Découverte de la STL,
- Exceptions et stratégies de traitement des erreurs
- Mise en œuvre des exceptions C++
- Présentation des namespace et des RTTI



ENVIRONNEMENTS UTILISES

Chaîne de production gcc sous l'IDE Eclipse 3.2 et plug-in CDT.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de mettre en oeuvre les mécanismes les plus avancés du langage C++ :

- Développer et utiliser des classes génériques à l'aide des templates
- Mettre en oeuvre la STL (Standard Template Library)
- Structurer la gestion des erreurs en utilisant les exceptions
- Mettre en place une gestion du nommage efficace grâce au namespace
- Effectuer des contrôles de types dynamiques en utilisant les services du RTTI (Real Time Type Information)
- Développer des solutions C++ à base de Design Patterns

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes C++
- Construire des applications C++ maintenables et évolutives en utilisant l'héritage et le polymorphisme

METHODE

Chaque aspect théorique est illustré par un exemple concret de mise en oeuvre puis exploité dans la réalisation incrémentale d'une véritable application en mode console. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

LES TEMPLATES

- La généricité statique
- Spécification d'un template de classe
- Instanciation d'un template
- Contraintes de mise en oeuvre
- Templates de fonction
- Template de classe, template de fonction : utilisation conjointe
- Templates et héritage
- Templates versus polymorphisme

INTRODUCTION A LA STL (STANDARD TEMPLATE LIBRARY)

- Origines et présentation de la STL
- Principales collections : vector, liste, deque, set, map
- Les iterators
- La classe string
- Les algorithmes génériques courants

LES EXCEPTIONS C++

- Exceptions et stratégies de traitement des erreurs
- Mise en oeuvre des exceptions C++
- Spécification d'exceptions dans les prototypes des méthodes et propagation d'exception
- Les blocs try et catch
- Construire une hiérarchie d'exceptions
- Fonctions terminate et unexpected

LES NAMESPACES

- Les conflits de nommage et objectifs des namespaces
- Définition d'un namespace
- Utilisation des namespaces : directive et instruction using
- Instruction using et héritage et polymorphisme

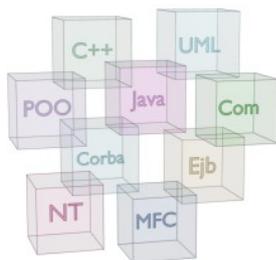
LE REALTIME TYPE INFORMATION (RTTI)

- Le besoin d'effectuer un contrôle de type rigoureux à l'exécution
- Opérateur typeid et la classe typeid_info
- Exceptions liées à l'opérateur typeid
- Les opérateurs static_cast et dynamic_cast

LES DESIGN PATTERNS

- Typologie des Design Patterns (Gamma)
- Singleton
- Façade et Adaptateur
- Observateur
- State
- Patterns liés à la gestion de la mémoire : pointeurs automatiques et intelligents

Client léger



ENVIRONNEMENTS UTILISES

Navigateur et toolkit Mozilla Firefox 1.5, IDE Eclipse 3.2 et plugins. Environnements spécifiques sur demande

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des interfaces Web répondant aux caractéristiques d'un " client léger " :

- Définir le cadre d'utilisation des technologies HTML/CSS/XSL
- Maîtriser le HTML généré à partir d'éditeurs graphiques
- Maîtriser la production de pages HTML à l'aide de transformations XSL et de feuilles de styles
- Identifier les situations justifiant l'usage de JavaScript et utiliser ce langage pour écrire des procédures de contrôle de saisies client
- Manipuler des pages HTML via leur arbre DOM

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation issus de d'applications réelles. 50 % du temps est consacré à la réalisation de ces exercices. Chaque stagiaire possède son poste de travail.

PROGRAMME

INTRODUCTION AUX LANGAGES DE BALISES

- HTML : historique, définitions et caractéristiques
- Les limites du HTML
- Objectifs des feuilles de style CSS
- Introduction à la technologie XML et panel des utilisations
- Objectifs du XSL
- Positionnement des technologies HTML/CSS/XSL dans le développement d'un " client léger "
- Exemple de toolkit dédié " Client Léger " : Mozilla

MISE EN OEUVRE DU HTML

- La syntaxe HTML
- Basiles graphiques de base : liens hypertexte, tableaux, images, ...
- Les formulaires : différents types de champs de saisie, les boutons, les listes, ...
- Les balises organisationnelles : frame, div, span, ...
- Les différentes propriétés supportées par les balises

MISE EN OEUVRE DE CSS

- Syntaxe spécifique d'une feuille de style CSS
- Adressage des éléments HTML
- Organisation des différentes feuilles de styles
- Règles de priorité et cascade de styles

MISE EN OEUVRE DE XSL

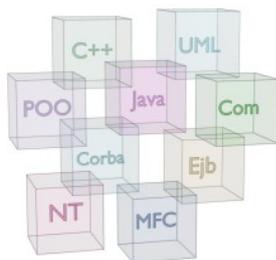
- Principes de fonctionnement
- Transformation coté client versus transformation coté serveur
- Le langage de transformation XSLT
- Les balises prédéfinies
- Utilisation de Xpath
- Processus de génération de pages HTML
- Organisation en cascade des feuilles de transformation

LE LANGAGE JAVA SCRIPT

- Caractéristiques du langage
- Possibilités et positionnement par rapport à Java
- Quand utiliser JavaScript
- Eléments syntaxiques
- Les fonctions et les objets
- DHTML, le dynamique HTML
- Norme et implantation

MANIPULATION DE PAGES HTML A L'AIDE DE LEUR ARBRE DOM

- Représentation DOM d'une page HTML
- Dépendance des navigateurs
- Modification de la valeur d'un nœud en JavaScript
- Création, insertion, suppression de nœuds en JavaScript
- Introduction à Ajax



OBJECTIFS

A l'issue du stage, les participants seront capables de développer des clients Web riches améliorant l'interactivité avec les utilisateurs, et ceci de manière rigoureuse et productive dans un environnement JEE :

- Identifier les situations justifiant le recours à Ajax
- Mettre en œuvre les principes et technologies sous-jacentes à Ajax
- Utiliser les bonnes pratiques du développement Ajax, connaître les principaux frameworks Ajax et les critères devant guider un choix
- Découvrir les possibilités du framework ZK

ENVIRONNEMENTS UTILISES

Plate-forme Java JEE, IDE Eclipse 3.2 et plugins, navigateur Mozilla Firefox 1.5, framework ZK 2.2. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Décrire une architecture Web
- Développer des utilitaires JavaScript en utilisant une démarche procédurale

METHODE

Chaque aspect théorique est illustré par des exemples concrets puis mis en œuvre lors de la réalisation d'une véritable application Web. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

PRESENTATION D'AJAX

Problématiques des clients Web riches
Motivations et objectifs d'AJAX
Normes et technologies sous-jacentes
Principes de fonctionnement
Exemples : Netvibes, GMail, Google Suggest, ...
Contraintes d'utilisation d'AJAX

RAPPELS SUR LES TECHNOLOGIES EN PRESENCE

Pages HTML et arbre DOM
Feuille de styles CSS
JavaScript
Flux XML et parseurs

MISE EN ŒUVRE

Typologie d'événements : requêtes planifiées ou sur action utilisateur
L'objet XMLHttpRequest
Dépendance vis-à-vis des navigateurs
Emission d'une requête, définition du callback
Traitement coté serveur
Activation du callback et mise à jour de la page

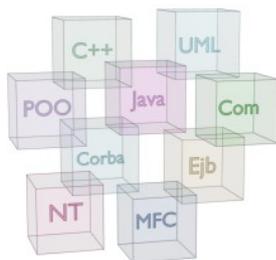
DEVELOPPEMENT PRODUCTIF AVEC AJAX

Panorama des frameworks AJAX : Dojo Toolkit, Rico, Rialto, Atlas...
Impact d'AJAX sur l'architecture logicielle
Utilisation de patterns

PRESENTATION DU FRAMEWORK ZK (AJAX BUT NO JAVASCRIPT)

Positionnement du framework ZK
Le modèle AJAX de ZK
Les composants XUL du framework
Le langage de balises ZUML
Mixage ZUML et HTML

Informatique générale



OBJECTIFS

A l'issue de ce stage, les participants seront capables de décrire les éléments participants à l'infrastructure informatique d'une entreprise et maîtriserons les bases de la programmation structurée. Et en particulier, ils seront capables de :

- Décrire les principaux constituants d'un ordinateur
- Décrire les tâches incombant à un système d'exploitation
- Développer une application logicielle via un langage évolué
- Organiser la solution d'un problème en procédures et fonctions
- Utiliser un processus de développement logiciel

ENVIRONNEMENTS UTILISES

Visual C de Microsoft sous Windows Turbo
Pascal de Borland sous Windows

PRIX

Inter-entreprise : 1608.0 € HT par stagiaire.

Intra-entreprise : 5306.0 € HT (8 stagiaires).

PREREQUIS

Aucun prérequis n'est nécessaire pour suivre cette formation.

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation en pseudo code. La solution des exercices sera aussi proposée en langage Pascal et langage C. L'atteinte des objectifs est contrôlée au fur et à mesure du déroulement du stage.

PROGRAMME

STRUCTURE D'UN ORDINATEUR

- Historique de l'informatique
- Constituants et fonctionnement d'un ordinateur
- Système de numérisation et représentation de l'information
- Les micros ordinateurs d'aujourd'hui
- Les réseaux d'entreprise et internet

LES SYSTEMES D'EXPLOITATION

- Tâches d'un système d'exploitation
- Principaux systèmes d'exploitation
- Rôle d'un interpréteur de commandes et principes de fonctionnement
- Organisation en couches

LES LOGICIELS

- Qu'est-ce qu'un programme
- La chaîne de production de programme
- les logiciels industriels et bureautiques
- Les principaux langages de programmation et classification

ALGORITHMIQUE : CONCEPTS DE BASE

- Variables et types
- Expressions et opérateurs
- Actions élémentaires
- Alternative et choix multiples
- Structures itératives

PROCEDURES ET FONCTIONS

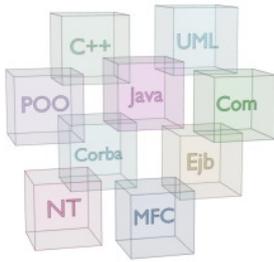
- Introduction à l'analyse descendante
- Définition d'une procédure, d'une fonction
- Passage d'arguments et restitution de résultats
- variables locales

TYPES COMPOSES ET ADRESSES

- Les tableaux : accès et algorithmes de base
- Les structures de données hétérogènes
- Allocation dynamique en mémoire et adresse
- Les pointeurs : définition et utilisation
- Exemples de structure de données dynamiques (liste chaînée, pile, arbre binaire, ...)

PROCESSUS DE DEVELOPPEMENT LOGICIEL

- Les différentes phases
- Objectifs du cahier des charges
- Objectifs de l'analyse, de la conception
- Techniques de tests
- Validation du logiciel et recette client



ENVIRONNEMENTS UTILISES

IDE Eclipse Visual Studio .Net Outils XML
AGL UML

PRIX

Inter-entreprise : 402.0 € HT par stagiaire.

Intra-entreprise : 1327.0 € HT (8 stagiaires).

OBJECTIFS

Ce séminaire s'adresse aux responsables informatiques, chefs de projet, ingénieurs logiciels souhaitant faire un tour d'horizon sur les techniques de développement objet dominantes et émergentes :

- Expliquer les raisons de l'émergence des nouvelles technologies de développement logiciel
- Comprendre l'architecture d'une application Java multi-niveaux
- Comprendre l'architecture d'une application .Net multi-niveaux
- Identifier les situations justifiant l'usage d'XML
- Énoncer les principales caractéristiques d'un processus de développement objet

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Donner une définition des concepts objets et les bénéfices induits par leur utilisation

METHODE

Les aspects théoriques sont concrétisés par des démonstrations d'environnements de développement et d'AGL, et une étude comparée de différentes technologies (Java et .Net) utilisées lors de la réalisation d'une même application.

PROGRAMME

OBJECTIFS DU DEVELOPPEMENT LOGICIEL

- Satisfaire les besoins des utilisateurs
- Produire plus vite et moins cher
- Produire maintenable et évolutif
- Réutiliser les infrastructures existantes
- Avec quelles technologies, quelles compétences

TECHNOLOGIES JAVA

- Présentation des différentes plateformes et typologies d'applications visées
- Caractéristiques du langage
- Composants d'architecture multi-niveaux : Servlet/JSP, JDBC, EJB et services Web
- Place de JavaScript
- Outils et environnements de développement

TECHNOLOGIES .NET

- Modèle d'architecture et organisation du framework
- Caractéristiques des langages supportés : Visual Basic, C#, ...
- Composants d'architecture multi-niveaux : pages et composants .Net, ADO.Net, .Net Remoting et services Web
- Outils et environnements de développement

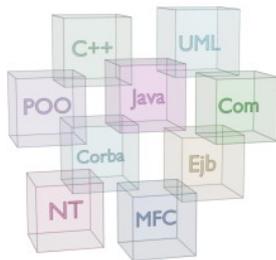
TECHNOLOGIES XML

- Les langages à balises : SGML, HTML, XML, XHTML, apports et positionnements
- Composition d'un document XML
- XML et la publication multi-formats
- XML et l'interopérabilité
- XML et les bases de données
- Outils disponibles

PROCESSUS DE DEVELOPPEMENT OBJET ET TECHNIQUES DE SPECIFICATION

- Objectifs d'un processus de développement logiciel
- Cycle de développement et activités de spécification
- Les différentes utilisations du langage UML
- Les AGL UML et l'intégration avec les environnements de développement

Java



ENVIRONNEMENTS UTILISES

IDE Eclipse 3.2 et le plugin Visual Editor.
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

Fournir aux participants une connaissance approfondie de la bibliothèque des Swing et les règles de conception nécessaires à la réalisation de clients riches maintenables et évolutifs en Java :

- Maîtriser la hiérarchie des composants Swing et le modèle événementiel
- Organiser une application desktop en exploitant les possibilités des frames et des boîtes de dialogue
- Enrichir une interface graphique à l'aide de menus, d'icônes, de toolbars et de tooltips
- Mettre en oeuvre des composants avancés tels que les JTable, JTree, JSplitPane et JTabbedPane
- Respecter les règles de conception préservant la performance et l'évolutivité
- Déployer des applications Swing à l'aide de Java Web Start

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes Java

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation complété par la réalisation incrémentale d'une véritable application en client swing. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

FONDATION DES SWING

- Portée des JFC
- AWT : la librairie graphique d'origine
- Justification des Swing et tour d'horizon
- Modèle MCV et UI-delegate
- Gestion du look and feel

ELEMENTS FONDAMENTAUX

- Composants, conteneurs et menus
- Les layout managers
- Le modèle événementiel
- Listeners et adaptateurs

LES COMPOSANTS DE BASE

- JButton et JLabel
- JCheckBox et JRadioButton
- JTextField et JTextArea
- JList et JComboBox

LES CONTENEURS DE BASE

- JFrame, JPanel et bordures
- Agencement et gestion des composants
- Gestion modale et non modale d'un JDialog
- Boîtes de dialogue standards : JColorChooser, JFileChooser et JOptionPane

LES MENUS

- Menu déroulant : JMenuBar et JMenuItem
- Pop up menu : JPopupMenu
- Icônes et toolbars : JToolBar
- Création de tooltips

MECANISMES AVANCES

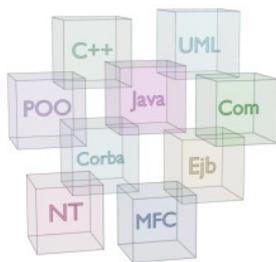
- Utilisation avancée des JList
- JTable et JTree
- Décomposition d'un panel : JTabbedPane et JSplitPane
- Gestion du drag and drop

CONCEPTION DE CLIENTS RICHES : LES BONNES PRATIQUES

- Inversion des dépendances entre les couches Présentation et Métier
- Enrichissement de composants existants
- Création et utilisation de JavaBeans graphiques
- Utilisation du multi-threading et des timers
- Internationalisation : localisation et ResourceBundle

DEPLOIEMENT D'UN CLIENT RICHE AVEC JAVA WEB START

- Objectifs de Java Web Start
- Packager et diffuser une application Java Web Start
- Mécanismes de sécurité
- Exploitation chez le client



ENVIRONNEMENTS UTILISES

Plate-forme J2EE, environnement de développement Eclipse 3.2 et plugin Visual Editor Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 2010.0 € HT par stagiaire.

Intra-entreprise : 6633.0 € HT (8 stagiaires).

OBJECTIFS

Apprendre le langage Java et les API de base nécessaires au développement d'applications auto suffisantes en client riche (Swing) :

- Décrire une architecture d'application " client riche "
- Développer des classes Java
- Utiliser l'héritage et le polymorphisme
- Organiser une application Java à l'aide des packages
- Concevoir une interface graphique basée sur la librairie Swing de Java
- Exploiter une base de données via JDBC
- Diffuser une application via Java Web Start

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Chaque aspect théorique est illustré par un exemple concret de mise en oeuvre puis exploité dans la réalisation incrémentale d'une véritable application « client riche ». 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

PRESENTATION D'UNE ARCHITECTURE D'APPLICATION " CLIENT RICHE "

- Typologies d'architecture
- Les applications " client riche " et leurs différents constituants
- Caractéristiques du langage Java

LES ELEMENTS DE BASE DU LANGAGE

- Structure d'un programme
- Les types de bases, opérateurs
- Structures de contrôle
- Introduction aux entrées/sorties

LES CLASSES ET L'ENCAPSULATION

- Définition d'attributs et de méthodes
- Attributs d'accès : public, private
- Les constructeurs et techniques d'instanciation
- Le garbage collector et méthode finalize

L'HERITAGE ET LE POLYMORPHISME

- Classes de base et classes dérivées
- Attribut d'accès protected
- Classe Object et hiérarchie unique
- Classes abstraites et interfaces

LES AUTRES CONCEPTS DU LANGAGE JAVA

- Organisation des classes en packages
- Exceptions et stratégies de traitement des erreurs
- Les collections Java

LA LIBRAIRIE SWING : ELEMENTS FONDAMENTAUX

- Composants, conteneurs et menus
- Les layout managers
- Le modèle événementiel, listeners et adaptateurs

LES COMPOSANTS DE BASE

- JButton et JLabel
- JTextField et JTextArea
- JList et JComboBox

LES CONTENEURS DE BASE

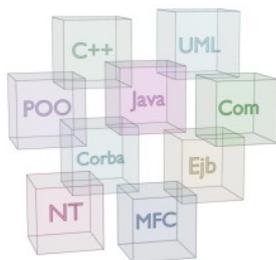
- JFrame et JPanel
- Agencement et gestion des composants
- Gestion modale et non modale d'un JDialog
- Mise en oeuvre de menus

JDBC

- Présentation de JDBC et des pilotes
- Connexion à une base de données
- Exécution de requêtes SQL et exploitation des résultats

PRESENTATION DE JAVA WEB START

- Objectifs de Java Web Start
- Packager et diffuser une application Java Web Start
- Exploitation chez le client



ENVIRONNEMENTS UTILISES

Plate-forme JEE, environnement de développement Eclipse 3.2 et moteur de Servlet Tomcat 5.5. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 2010.0 € HT par stagiaire.

Intra-entreprise : 6633.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications Web en utilisant le langage Java :

- Décrire une architecture Web
- Développer des classes Java
- Utiliser l'héritage et le polymorphisme
- Organiser une application Java à l'aide des packages
- Créer des Servlets et des Jsp
- Concevoir une application Web en respectant un modèle MVC
- Exploiter une base de données via JDBC

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Chaque aspect théorique est illustré par un exemple concret de mise en oeuvre puis exploité dans la réalisation incrémentale d'une véritable application Web. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

PRESENTATION D'UNE ARCHITECTURE MULTI-NIVEAUX

- Typologies d'architecture
- Les applications Web et leurs différents constituants
- Caractéristiques du langage Java

LES ELEMENTS DE BASE DU LANGAGE

- Structure d'un programme
- Les types de base, opérateurs
- Structures de contrôle
- Introduction aux entrées/sorties

LES CLASSES ET L'ENCAPSULATION

- Définition d'attributs et de méthodes
- Attributs d'accès: public, private
- Les constructeurs
- Technique d'instanciation
- Le garbage collector et méthode finalize

L'HERITAGE ET LE POLYMORPHISME

- Classes de base et classes dérivées
- Attribut d'accès protected
- Classe Object et hiérarchie unique
- Classes abstraites et interfaces

LES AUTRES CONCEPTS DU LANGAGE JAVA

- Organisation des classes en packages
- Exceptions et stratégies de traitement des erreurs
- Les collections Java

LES SERVLETS

- Principe de fonctionnement
- Création et déploiement d'un Servlet HTTP
- Récupération des données : HttpServletRequest
- Cycle de vie et suivi de sessions
- Le chaînage de Servlets

LES JSP

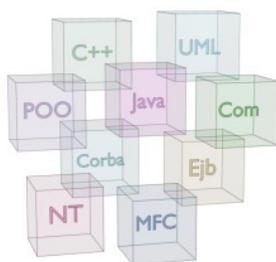
- Concept de base
- Création et lancement d'une Jsp
- Les scriptlets
- Déclarations, expressions et directives Jsp

LE MODELE MVC

- Présentation de l'architecture
- Définition des rôles
- Utilisation des java beans
- L'architecture MVC2
- Présentation de Struts

UTILISATION DE JDBC

- Présentation de JDBC et des pilotes
- Connexion à une base de données
- Exécution de requêtes SQL
- Exploitation des résultats d'une requête
- Support des transactions



ENVIRONNEMENTS UTILISES

Plate-forme J2SE 5.0 et environnement de développement Eclipse 3.2.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

Destiné à des développeurs possédant une première expérience dans un langage orienté objet, cette formation d'un contenu identique à "Développer en langage Java" permet de couvrir l'ensemble des mécanismes de base d'une plateforme Java et ce sur une durée optimisée de trois jours :

- Décrire les domaines couverts par la plate-forme J2SE
- Programmer en langage Java
- Construire des applications Java maintenables et évolutives en utilisant l'héritage et le polymorphisme
- Implémenter des associations en utilisant les collections Java
- Traiter les cas d'erreurs à l'aide des exceptions Java
- Exploiter la sérialisation et les nouveautés du JDK 1.5

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer du logiciel via un langage orienté objet (C++, Delphi, VB6 ou autres)

METHODE

Chaque aspect théorique est illustré par un exemple concret de mise en oeuvre puis exploité dans la réalisation incrémentale d'une véritable application en mode console. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

PRESENTATION DU LANGAGE

- Historique et caractéristiques
- Présentation des différentes plate formes Java : J2SE, J2EE, J2ME
- Panorama des environnements de développement
- Les concepts objets

LES ELEMENTS DE BASE

- Structure d'un programme
- Les types de bases, opérateurs,
- Structures de contrôle
- Introduction aux entrées/sorties

LES CLASSES ET L'ENCAPSULATION

- Définition d'attributs et de méthodes,
- Attributs d'accès : public, private,
- Les constructeurs
- Technique d'instanciation
- Le garbage collector et méthode finalize
- Organisation des classes en packages

L'HERITAGE ET LE POLYMORPHISME

- Classes de base et classes dérivées
- Attribut d'accès protected
- Classe Object et hiérarchie unique
- Classes abstraites et interfaces

LES COLLECTIONS JAVA

- Présentation de la hiérarchie de classes de java.util
- Conteneurs séquentiels : ArrayList, LinkedList, ...
- Conteneurs associatifs : Hashtable, HashMap, ...
- Parcours des collections à l'aide d'itérateurs

LES EXCEPTIONS JAVA

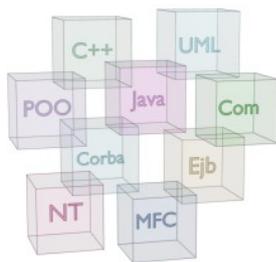
- Exceptions et stratégies de traitement des erreurs
- Hiérarchie des exceptions Java
- Traiter une exception : try et catch
- Lever ses propres exceptions : throw
- L'instruction final

ASPECTS AVANCES

- Persistance par sérialisation
- Introspection et JavaBean
- Développement d'application Java multi- threads

LES APPORTS DU JDK1.5

- Autoboxing/unboxing
- Les generics
- Les énumérations
- Méthodes avec nombres d'arguments variables
- Parcours de collection avec la boucle for



OBJECTIFS

A l'issue du stage, les participants seront capables de concevoir des applications Java répondant aux exigences d'un développement industriels :

- Concevoir une application Java multi-threads
- Mettre en oeuvre des techniques de distribution
- Internationaliser une application
- Exploiter l'API de réflexion
- Mettre en oeuvre une stratégie de tests unitaires et de logs
- Exploiter Ant pour produire, packager et diffuser une application Java

ENVIRONNEMENTS UTILISES

Plate-forme J2SE 5.0, environnement de développement Eclipse 3.2 et frameworks Junit, Log4J et Ant.

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes Java

METHODE

Chaque aspect théorique est illustré par un ou plusieurs exemples de mise en oeuvre, le tout consolidé par la réalisation d'une véritable application Java. Un poste de travail est fourni par groupe de deux stagiaires.

PROGRAMME

LE MULTI-THREADING EN JAVA

- La classe Thread et l'interface Runnable
- Gestion des Threads et cycle de vie
- Synchronisation et partage de données
- Fonctionnalités du package java.util.concurrent

COMMUNICATION EN JAVA

- Typologies d'application distribuée
- Communication " bas niveau " par socket
- Communication par appels de méthodes distantes : RMI
- Communication en milieu hétérogène : CORBA
- Comparaison des différentes techniques, critères de choix
- Autre techniques : EJB, Service Web, JMS

INTERNATIONALISATION D'UNE APPLICATION

- Principes de localisation : l'objet Locale
- Mise en oeuvre d'objets ResourceBundle
- Formatage des dates et des nombres en fonction de la localisation

L'INTROSPECTION EN JAVA

- Méta données et l'utilitaire javap
- Objectifs de l'API de réflexion
- Présentation du package java.util.reflect
- Instanciation et invocation de méthodes
- Exemples concrets d'utilisation

TEST UNITAIRE EN JAVA : JUNIT

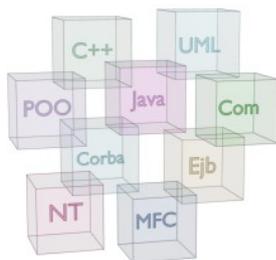
- Les classes TestCase et TestSuite
- Mise en oeuvre des tests : TestRunner
- Stratégies de test : les bonnes pratiques

JOURNALISATION AVEC LOG4J

- Principes généraux
- Configuration des Appenders et des Loggers
- Fichier de propriétés et fichier XML

PRODUCTION D'UN BUILD ET DIFFUSION : ANT

- Objectifs et concepts fondamentaux
- Structure des fichiers build.xml : targets et tasks
- Paramétrage et propriétés
- Packager une application avec Ant



ENVIRONNEMENTS UTILISES

Plate-forme JSE 6.0 et environnement de développement Eclipse 3.6.

PRIX

Inter-entreprise : 1520.0 € HT par stagiaire.

Intra-entreprise : 5016.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants maîtriseront les principes fondamentaux de l'approche objet et leur mise en oeuvre à l'aide du langage Java. Le cours introduira aussi la problématique de conception orientée objet en appuyant sur la notation UML :

- Modéliser un domaine métier à l'aide de diagrammes de classe UML
- Développer des classes Java
- Organiser une application Java à l'aide des packages
- Construire des applications maintenables et évolutives en utilisant l'héritage et le polymorphisme
- Implémenter des associations en utilisant les collections
- Traiter les cas d'erreurs à l'aide des exceptions
- Implémenter la persistance à l'aide de la sérialisation
- Utiliser efficacement l'IDE Eclipse pour le développement d'applications Java « Stand-alone »

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Chaque aspect théorique est illustré par un exemple concret puis exploité dans la réalisation incrémentale d'une véritable application en mode console. Les exercices pratiques permettent d'appréhender tous les concepts objet et leur mise en oeuvre en Java de manière progressive.

PROGRAMME

LE PARADIGME « OBJET »

- Les principes généraux de l'approche objet : l'abstraction, l'encapsulation, les classes et les objets, l'héritage, le polymorphisme, les interfaces
- Illustration UML de tous les concepts Objet (diagramme de classe et de séquence)

LES ELEMENTS DE BASE DU LANGAGE

- Les différentes plate formes Java : JSE, JEE, JME
- Eléments de base : types primitifs, tableaux, opérateurs, structures de contrôle, fonction main
- Introduction aux entrées/sorties
- Présentation de l'IDE Eclipse

LES CLASSES ET L'ENCAPSULATION

- Les attributs, les méthodes et leurs attributs d'accès
- Les constructeurs et les règles d'instanciation
- L'auto référence et les propriétés statiques
- Le garbage collector et méthode finalize
- Organisation des classes en packages

HERITAGE, INTERFACE ET POLYMORPHISME

- La généralisation/spécialisation
- Classes de base et dérivées : la relation extends
- Constructeurs de classes dérivées
- Redéfinition de méthodes et polymorphisme
- Le down casting, utilisation de l'opérateur instanceof
- Présentation de la classe Object
- Les classes abstraites
- Notion de contrat, de client et de fournisseur
- Définition d'interfaces et réalisation d'interfaces
- Héritage et interface : utilisation conjointe

LES COLLECTIONS JAVA

- Présentation de la hiérarchie de classes de java.util
- Conteneurs séquentiels : ArrayList, LinkedList, ...
- Conteneurs associatifs : Hashtable, HashMap, ...
- Parcours des collections à l'aide d'itérateurs

LES EXCEPTIONS JAVA

- Principes de traitement des erreurs à l'exécution
- La hiérarchie des exceptions Java : exceptions surveillées et non surveillées
- Génération, propagation et traitement d'une exception : throw, try, catch et finally
- Définir ses propres classes d'exception

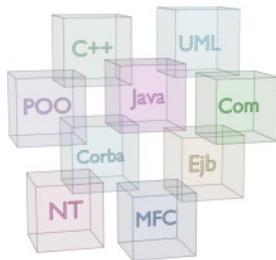
LA PROGRAMMATION DES ENTREES/SORTIES

- La hiérarchie des classes d'entrée/sortie
- Classes d'entrées/sortie en mode byte et caractère
- La sérialisation d'objets

LES APPORTS DE LA VERSION 5

- Autoboxing/unboxing
- Les génériques
- Les énumérations
- Parcours de collection avec la boucle for
- Les annotations standards

JEE (J2EE)



ENVIRONNEMENTS UTILISES

Plate-forme Java J2EE, Eclipse 3.1 et plugins, moteur de Servlet/Jsp Tomcat 5.5. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications Web sécurisées en utilisant des composants d'architecture Java répondant à la norme J2EE de Sun :

- Créer des Servlets et des Jsp
- Concevoir une application Web en respectant un modèle MVC
- Développer une partie cliente et mettre en place des contrôles de validation des formulaires
- Exploiter une base de données via JDBC
- Sécuriser une application Web
- Déployer une application J2EE sous Tomcat

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes Java

METHODE

Chaque aspect théorique est illustré par des exemples concrets puis mis en oeuvre par le stagiaire lors de la réalisation d'une véritable application Web. Un poste de travail est fourni par groupe de deux stagiaires.

PROGRAMME

J2EE ET LES ARCHITECTURES DISTRIBUEES

- Typologie d'une architecture Web
- Positionnement des différents composants de J2EE
- Présentation des infrastructures d'exécutions : Apache, Tomcat, ...
- Présentation des environnements de développements

LES SERVLETS

- Principes de fonctionnement
- Création et lancement d'un Servlet HTTP
- Récupération des données : HttpServletRequest
- Initialisation des Servlets
- Paramétrage via le fichier web.xml
- Cycle de vie et suivi de session
- Le chaînage de Servlets

LES JSP

- Concepts de base
- Création et lancement d'une Jsp
- Les Scriptlets
- Déclarations, expressions et directives Jsp

LE MODELE MVC

- Présentation de l'architecture
- Définition des rôles
- Utilisation des Java Beans
- L'architecture MVC2
- Présentation de Struts

CONCEPTION DE PARTIE CLIENTE AVEC HTML ET JAVA SCRIPT

- Les Balises de base : liens, tableaux, images, ...
- Création et gestion de formulaires
- Accès aux différents éléments d'une page
- Gestion des événements en JavaScript

UTILISATION DE JDBC

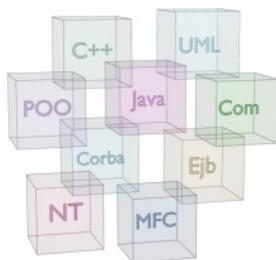
- Présentation de JDBC et des pilotes
- Connexion à une base de données
- Exécution de requêtes SQL
- Exploitation des résultats d'une requête
- Support des transactions

SECURISATION D'APPLICATIONS WEB

- Principes de base
- Mécanisme d'authentification
- Gestion de la confidentialité avec SSL

DEPLOIEMENT D'UNE APPLICATION J2EE

- Structure arborescente d'un fichier .war
- Création d'un fichier de déploiement web.xml
- Génération d'un fichier .war sous Eclipse
- Déploiement d'un fichier .war sous Tomcat



ENVIRONNEMENTS UTILISES

Frameworks Maven2, Hibernate 3.6, Spring 2.5, Struts 2.1, serveur web tomcat 6.0 et environnement de développement Eclipse 3.6.

PRIX

Inter-entreprise : 1600.0 € HT par stagiaire.

Intra-entreprise : 5280.0 € HT (8 stagiaires).

OBJECTIFS

Ce cours vous permettra de maîtriser la réalisation d'applications 3-tiers web basées sur les frameworks « phares » de l'écosystème Java :

- Organiser le développement d'un projet Java avec Maven2
- Développer une couche de persistance à l'aide d'Hibernate
- Développer une couche de présentation web avec Struts2
- Utiliser le conteneur d'IOC de Spring pour préserver l'indépendance inter couches
- Contrôler les transactions applicatives à l'aide des techniques AOP de Spring

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- La connaissance du langage java est impérative,
- Une première expérience dans le développement d'applications web Java est un plus.

METHODE

Les différents frameworks seront progressivement étudiés à partir des problématiques d'architecture pour lesquels ils apportent aujourd'hui des solutions de référence. L'implémentation d'une étude de cas « fils rouge » servira de cadre à leur mise en oeuvre conjointe.

PROGRAMME

PRESENTATION DE L'ECOSYSTEME JEE

- Typologies d'application 2-tiers, 3-tiers, N-tiers et le modèle en couches
- Architecture JEE : conteneurs, composants et services transverses
- Positionnement des frameworks Hibernate, Spring et Struts2 dans une architecture 3-tiers Java

INDUSTRIALISATION D'UN PROJET JAVA AVEC MAVEN2

- Cycle de vie d'un projet et ses phases
- Archétype d'un projet, arborescence standard
- les plugins standards et leurs goals
- Dépôts local et distant
- La gestion des dépendances et des ressources
- Plugin d'intégration à Eclipse (m2eclipse)

LES BASES DU FRAMEWORK SPRING

- L'organisation modulaire de Spring
- Architecture du conteneur d'IOC
- Gestion du cycle de vie des beans
- Configuration XML des injections
- Les annotations comme alternative à la configuration XML

DEVELOPPEMENT D'UNE COUCHE DE PERSISTANCE HIBERNATE

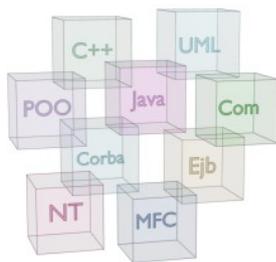
- Problématique de l'ORM et présentation d'Hibernate
- Architecture générale
- POJO transient et persistant, configuration, SessionFactory et Session
- Configuration du framework : hibernate.cfg.xml
- Définition du mapping : les fichiers .hbm
- Gestion des sessions et API de persistance
- Requêtes HQL et gestion des transactions
- Intégration du framework Hibernate à Spring
- Configuration, support DAO et services génériques

MISE EN OEUVRE DE SPRING AOP

- Objectifs de l'AOP et le vocabulaire : joinPoint, pointcut, advice, aspect, target et weaver
- Les différentes mises en oeuvre : par configuration ou par annotation
- Principes de la gestion des transactions avec Spring
- Mise en oeuvre de transactions déclaratives via AOP

DEVELOPPEMENT D'UNE COUCHE DE PRESENTATION STRUTS2

- Objectifs et architecture générale de Struts2
- Principe du traitement d'une requête
- Les principaux fichiers de configuration
- Mise en oeuvre du modèle MVC Struts2
- Piles d'intercepteurs, développement d'actions et de vues (struts-tags)
- Configuration des actions
- Externalisation des libellés et internationalisation
- Validation déclarative et par programmation
- Intégration Spring
- Découplage de la couche présentation avec la couche des services métier



ENVIRONNEMENTS UTILISES

Plate-forme Java JEE, IDE Eclipse 3.2 et plugins, moteur de Servlet/Jsp Tomcat 5.5, serveur d'applications JBoss 4.0
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 2010.0 € HT par stagiaire.

Intra-entreprise : 6633.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables d'exploiter de manière pertinente les principaux services techniques de la plate-forme JEE lors d'un développement d'application JAVA multi-niveaux :

- Caractériser les composants d'architecture JEE
- Mettre en oeuvre des Servlets et des Jsp
- Concevoir une application Web en respectant un modèle MVC2 à l'aide du framework Struts ou JSF
- Gérer la persistance avec Hibernate
- Développer des applications distribuées à l'aide des EJB et des Services Web

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes Java

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation et la réalisation d'une véritable application J2EE. Un poste de travail est fourni par groupe de deux stagiaires.

PROGRAMME

JEE ET LES ARCHITECTURES DISTRIBUEES

- Typologie d'une architecture d'application JEE
- Positionnement des différents composants de JEE
- Présentation des infrastructures d'exécution : JBoss, Tomcat, Geronimo, ...
- Présentation d'environnements de développement

SERVLETS ET JSP

- Principe de fonctionnement
- Création et déploiement d'un Servlet HTTP
- Récupération des données : HttpServletRequest
- Cycle de vie et suivi de sessions des servlets
- Création et déploiement d'une Jsp
- Scriptlets, déclarations, expressions et directives Jsp
- Patterns MVC et MVC2

MISE EN OEUVRE DU FRAMEWORK STRUTS

- Architecture du framework
- Présentation du Contrôleur Struts
- Création et configuration de classes Action
- Création et gestion des vues
- Présentation des bibliothèques de tags de Struts
- Gestion des erreurs

AUTRES FRAMEWORKS WEB

- Présentation du framework JSF
- Gestion de formulaires et principes de navigation entre pages
- Le modèle événementiel et de validation
- Introduction au framework Spring

TECHNIQUES D'ACCES AUX DONNEES

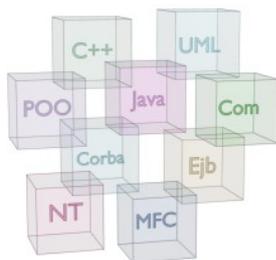
- Présentation de JDBC et des pilotes
- Connexion à une base de données
- Exécution de requêtes SQL
- Présentation et configuration d'Hibernate
- Les classes persistantes (POJO)
- Gestion du mapping
- Création, ajout et suppression d'objets
- Le langage HQL

METTRE EN OEUVRE DES EJB 3

- Architecture globale
- Serveurs, conteneurs, composants et connecteurs
- Typologie de Beans : session, message, entity
- Développement et déploiement d'EJB
- Cycles de vie des Beans
- Les services standards
- Localisation des EJB : JNDI
- Exploiter un EJB dans la partie cliente

EXPLOITER DES WEB SERVICES

- Services Web , pour quels besoins
- Typologie de Services Web et architectures cibles
- Spécification de services Web : WSDL
- Création d'un service Web avec JAX-RPC
- Invocation de services Web : SOAP
- Déploiement sur internet : UDDI



ENVIRONNEMENTS UTILISES

Plate-forme Java JEE, IDE Eclipse 3.2 et plugins, framework Struts 1.3.5 et moteur de Servlet/JSP tomcat 5.5. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications Web de qualité professionnelle en exploitant les possibilités du framework Struts :

- Concevoir des applications Web exploitant le modèle MVC de Struts
- Faciliter le développement des JSP tout en les rendant plus maintenables à l'aide des bibliothèques de tags Struts
- Déployer une application Struts sous Tomcat
- Traiter les cas d'erreurs à l'aide des classes Struts
- Définir des règles de validation, coté client comme coté serveur, en exploitant le moteur Validator
- Réaliser des modèles de présentation avec les Tiles

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Créer des Servlets et des Jsp

METHODE

Les aspects théoriques sont concrétisés par la réalisation incrémentale d'une véritable application Web exploitant tous les mécanismes fondamentaux de Struts. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

PRESENTATION DU FRAMEWORK STRUTS

- Problématique d'architecture des applications Web
- Présentation et limites du modèle MVC
- Le modèle MVC2 et les principaux frameworks Web
- Le modèle d'architecture de Struts : cinématique et principaux constituants
- Introduction à la configuration Struts : le fichier struts-config.xml
- Réalisation d'une première application Struts

LA COUCHE CONTROLEUR

- Présentation du contrôleur frontal : ActionServlet
- Création et configuration des Actions Struts
- La méthode execute
- Factorisation des Actions : DispatchAction
- Paramétrage des vues avec l'ActionMapping
- Définition de forwards

LA GESTION DE FORMULAIRE ET L' ECHANGE DE DONNEES

- Création de formulaires avec la bibliothèque Struts-html
- Principe d'automatisation de gestion de formulaire
- Création et configuration des ActionForm
- Coopération entre ActionForm et Action

LA COUCHE PRESENTATION

- Création d'une vue
- Coopération entre ActionForm et JSP
- Récupération et présentation des données avec la librairie Struts-beans
- La bibliothèque de tags Struts Logic

DEPLOIEMENT D'UNE APPLICATION STRUTS

- Configuration du fichier Web.xml
- Réalisation du fichier Struts-config.xml
- Génération de fichier .war
- Création et configuration de plugins
- Externalisation des messages

GESTION DES ERREURS

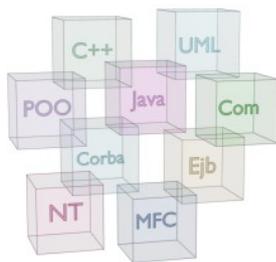
- Les classes de traitement d'erreurs
- Externalisation des messages
- Configuration des traitements d'erreurs
- Le tag <html:errors>

LE MOTEUR DE VALIDATION

- Présentation du plugin Validator
- Configuration par fichiers XML
- Définition des règles de contrôles

LES TILES

- Le concept de template
- Présentation des Tiles
- Les layouts
- Les principales balises de Tiles



ENVIRONNEMENTS UTILISES

Plateforme Java JEE, IDE Eclipse 3.6 et plugins, framework Struts 2.1.x et moteur de Servlet/JSP tomcat 6.0

PRIX

Inter-entreprise : 1200.0 € HT par stagiaire.

Intra-entreprise : 3960.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications Web de qualité professionnelle en exploitant les possibilités du framework Struts 2 :

- Concevoir des applications web exploitant rigoureusement le modèle MVC2 de Struts 2
- Optimiser le workflow d'une requête par l'utilisation pertinente des intercepteurs
- Développer des pages JSP à l'aide de la librairie de tags « struts-tags »
- Simplifier l'écriture des JSP par une bonne utilisation d'OGNL
- Internationaliser une application Struts 2
- Traiter les cas d'erreurs à l'aide des mécanismes de validation
- Homogénéiser la mise en page des JSP avec Tiles

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Mettre en oeuvre des Servlets et des Jsp

METHODE

Les aspects théoriques sont concrétisés par la réalisation incrémentale d'une véritable application Web exploitant les principaux mécanismes de Struts 2. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

PRESENTATION DU FRAMEWORK STRUTS 2

- Problématique d'architecture des applications web
- Le modèle d'architecture de Struts 1 : limites et besoins d'évolution
- Le modèle d'architecture de Struts 2 : cinématique d'ensemble et principaux constituants
- Introduction à la configuration Struts 2 : le fichier struts.xml
- Réalisation d'une première application Struts 2

LA COUCHE CONTROL

- La configuration : les fichiers web.xml, struts.xml et struts.properties
- Le filtre « FilterDispatcher »
- Les actions Struts 2 et la méthode « execute »
- Les différentes implémentations d'une action Struts 2 : POJO ou extension de la classe « ActionSupport »
- Principes des « intercepteurs »
- Configuration et mise en oeuvre des « intercepteurs »
- Mapping des paramètres de requête sur les attributs de l'action
- Types de résultats d'une action
- Pile d'objets et langage de navigation « OGNL »
- Conversions de types

LA COUCHE PRESENTATION

- Architecture d'une vue Struts 2
- Les différentes technologies disponibles : JSP, Velocity et Freemarker
- Organisation de la librairie de tags « struts-tags »
- Étude systématique des tags des trois catégories : «Control Tags », «Data Tags» et «Form Tags»
- Mise en oeuvre des expressions OGNL dans des JSP
- Customisation du rendu des vues par l'utilisation de « themes »

INTERNATIONALISATION D'UNE APPLICATION STRUTS 2

VALIDATION DES FORMULAIRES

- Validation coté serveur et/ou coté client
- Validation par programme et/ou déclarative
- Les intercepteurs « validation » et « workflow »
- L'interface « Valideable » et l'implémentation par défaut de la classe « ActionSupport »
- Définition de validations déclaratives : les fichiers Action-validation.xml
- Les « validateurs » prédéfinis
- Affichage des messages d'erreur : les tags « actionmessage », « actionerror » et « fielderror »
- Validation côté " client " : attribut « validate » du formulaire, Javascript généré

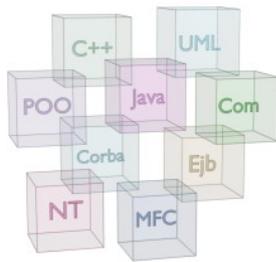
INTEGRATION DE TILES 2.0

- Le plugin « struts2-tiles-plugin »
- Le concept de template de pages (layout)
- Configuration d'une action utilisant Tiles
- Les principales balises de la librairie Tiles
- Définition de pages par programmation et de manière déclarative

AUTRES ASPECTS DU FRAMEWORK

- Mise en oeuvre bas niveau d'Ajax : configuration de l'action, traitement de la requête asynchrone, traitement de la réponse en Javascript.
- La librairie de tags « ajax » et exemples de librairies tiers.
- Traitement des erreurs de conversion.
- Configuration par annotation.

- L'intercepteur « i18n »
- Convention de nommage et organisation des « resource bundles »
- Le paramètre « request_locale »
- Les « form ui tags » exploitant l'internationalisation
- l'attribut « key »
- Les messages paramétrés et les différentes méthodes « getText() »
- Facilités offertes par la classe « ActionSupport »



ENVIRONNEMENTS UTILISES

Plate-forme Java JEE, Eclipse 3.2 et plugins, moteur de Servlet/JSP Tomcat 5.5, Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications Web sécurisées en utilisant des composants standards d'architecture Java répondant à la norme JEE de Sun :

- Créer des Servlets et des Jsp
- Concevoir une application Web en respectant un modèle MVC
- Utiliser les "Expressions Languages" et les taglibs JSTL
- Développer une partie cliente et mettre en place des contrôles de validation des formulaires
- Utiliser les fonctionnalités Eclipse WTP
- Déployer une application Web JEE sous Tomcat

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Utiliser le HTML et le protocole HTTP
- Développer des classes Java

METHODE

Chaque aspect théorique est illustré par des exemples concrets puis mis en oeuvre par le stagiaire lors de la réalisation d'une véritable application Web. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

ARCHITECTURES D'APPLICATIONS WEB JEE

- Typologie d'architectures d'applications Web
- Positionnement des différents composants Web de JEE
- Présentation d'infrastructures d'exécutions : Apache, Tomcat, Jetty, Resin, ...

MISE EN OEUVRE DES SERVLETS

- Principes de fonctionnement
- Création et lancement d'une Servlet HTTP
- Récupération des paramètres de la requête
- Initialisation des Servlets
- Paramétrage via le fichier web.xml
- Cycle de vie d'une servlet
- Suivi de session et gestion des cookies

SERVLET : ASPECTS AVANCES

- Traitement des exception et gestion des erreurs HTTP
- Modèle single et multi threads
- Chainage de Servlets et modèle SSI
- Gestion du "Context Root"
- Création de filtres et de listeners

LES JSP, PRINCIPES DE BASE

- Justification des JSP
- Création d'une JSP et processus de transformation
- L'objet PageContext et les objets implicites d'une JSP
- Les scriptlets
- Déclarations, expressions et directives JSP
- Règles de chaînage Servlet/JSP
- Les paramètres de page : "errorPage" et "isErrorPage"
- Contrôle de surface d'un formulaire en JavaScript

MISE EN OEUVRE DU MODELE MVC

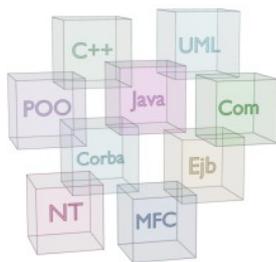
- Répartition des rôles entre Servlet et JSP
- Utilisation de Java Beans et gestion de leur portée
- Les actions standards JSP : <jsp:xxx>
- Utilisation des "Expressions Languages" (EL)
- Limite du modèle MVC et introduction au MVC2
- Présentation d'un framework Web MVC2 : Struts

UTILISATION DE TAGLIBS

- Motivation des TagLibs et structure
- Présentation de la JSTL
- Utilisation de la JSTL core
- Exemples de TagLibs "on the shelf"
- Utilisation de la TagLib "displaytag"

DEPLOIEMENT D'UNE APPLICATION J2EE

- Structure arborescente d'un fichier .war
- Création d'un fichier de déploiement web.xml
- Génération d'un fichier .war sous Eclipse
- Déploiement d'un fichier .war sous Tomcat
- Fonctions d'administration de Tomcat



ENVIRONNEMENTS UTILISES

Framework JEE et Axis 1.6, environnement de développement Eclipse 3.2 et plugin dédiés Services Web, Tomcat 5.5

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables d'utiliser de manière pertinentes les Services Web lors de développements d'applications distribuées en Java:

- Identifier les besoins de Services Web
- Spécifier, développer et déployer un Service Web
- Rechercher et exploiter un Service Web sur Internet
- Sécuriser l'utilisation de Services Web
- Utiliser des outils facilitant le développement et l'exploitation de Services Web

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Construire des applications maintenables et évolutives en utilisant l'héritage et le polymorphisme

METHODE

Les aspects théoriques sont concrétisés par le développement d'une véritable application distribuée à base de Services Web. Un poste de travail est fourni par groupe de deux stagiaires.

PROGRAMME

INTRODUCTION AUX SERVICES WEB

- Rappel sur l'architecture n-tiers
- Services Web , pour quel besoin
- Typologie de Services Web : SOAP et XML-RPC
- Architectures cibles des Services Web
- Développement, administration et orchestrations
- Aspects commerciaux et évolutions futures

XML, LE LANGAGE DES SERVICES WEB

- Présentation
- Standards dérivés de la norme XML (XML namespace, XML schéma)
- Messages XML entre applications et Services Web

DESCRIPTION D'UN SERVICE WEB

- Introduction à WSDL
- Les étapes de création
- Structure d'un document WSDL
- Eléments du langage
- Création d'un fichier WSDL décrivant un Service Web

CREATION D'UN SERVICE WEB AVEC AXIS

- Création des classes Java associées à un Service Web (côté serveur)
- Création du descriptif de déploiement
- Déploiement d'un Service Web
- Test du Service Web (via le navigateur)
- Création de classes clientes
- Test du Service Web (via le client)

INVOCATION D'UN SERVICE WEB (SOAP)

- Présentation
- Structure des messages SOAP
- SOAP avec pieces jointes
- SOAP via http
- Exemples de requêtes et de réponses

DEPLOIEMENT DE SERVICES WEB SUR INTERNET (UDDI)

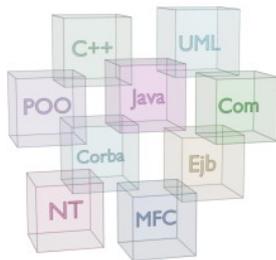
- Objectifs de l'annuaire UDDI
- Publication d'un Service Web
- Cas de COM+ et des " Serviced components "

SECURITE ET GESTION D'INTEGRITE

- Intégrité des messages
- Intégrité des transactions
- Authentification d'accès

LES OUTILS

- Outils de développement
- Outils de test et socle d'exécution
- Outils d'assemblage et d'orchestration de Services Web
- Outils de déploiement, de surveillance et d'administration



ENVIRONNEMENTS UTILISES

Plate-forme Java JEE, IDE Eclipse 3.4 et plugins, base de données H2 et Hibernate Core 3.3

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de mettre en œuvre une couche de persistance Java en s'appuyant sur le framework Hibernate :

- Positionner Hibernate parmi les différentes possibilités de persistance objet
- Réaliser des requêtes SQL via l'API JDBC
- Rendre persistant les objets Java avec hibernate
- Rendre persistant tout type de relation : association, composition et héritage
- Utiliser les différentes techniques de requêtes proposées par Hibernate
- Gérer efficacement les transactions et les accès concurrents

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- La connaissance du SQL est un plus
- Développer des classes Java

METHODE

Les aspects théoriques sont appuyés par de nombreux exemples de configuration/programmation, le tout consolidé par la réalisation d'une application en mode console exploitant un modèle complexe de données. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

INTRODUCTION A LA PERSISTENCE OBJET

- Présentation de la problématique
- Typologie de solutions (sérialisation, SGBDOO, SGBDR, autres)
- Mapping modèle Objet/modèle Relationnel (Impedance mismatch)
- les différentes solutions pour un support relationnel : JDBC, EJB, JPA, ...

MISE EN ŒUVRE DE L'API JDBC

- Présentation de JDBC et des pilotes
- Connexion à une base de données
- Exécution de requêtes SQL
- Exploitation des résultats d'une requête
- Support des transactions

PRESENTATION HIBERNATE

- Positionnement d'Hibernate, présentation de l'architecture
- Installation d'Hibernate : rôle des principales bibliothèques
- Configuration de base : hibernate.cfg.xml
- Définition d'une classe persistante (POJO) et de son mapping sur une table
- Notions de session et de transaction
- Premier exemple de persistance d'objets

MAPPING DES OBJETS

- Unicité des objets, les différents générateurs d'id
- Définition des propriétés de l'objet, les types Hibernate
- Gestion des sessions : la SessionFactory
- Cycle de vie d'un objet Hibernate
- Utilisation des transactions pour créer, modifier supprimer des objets persistants
- Traitement des exceptions Hibernate

MAPPING DES RELATIONS

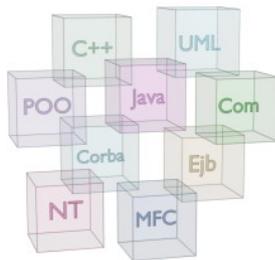
- Gestion des associations 1-1, 1-N, N-1 et N-N
- Traitement de la navigation : mono et bidirectionnelle
- Prise en compte de l'intégrité référentielle
- Gestion de la composition
- Les opérations cascadiées
- Gestion de l'héritage : les trois stratégies possibles
- Critères de choix
- Analyse des opérations SQL engendrées et optimisation

FONCTIONNALITES DE REQUETAGE

- Typologie des techniques disponibles : HQL, API Criteria ou SQL directe
- Requêtes HQL, les clauses from, select, where
- Association et jointure en HQL
- Requêtes HQL polymorphiques
- Les sous requêtes HQL
- Requêtes par critères : utilisation de l'API Criteria
- SQL natif, utilisation de l'interface SQLQuery

LES ASPECTS AVANCEES D'HIBERNATE

- Initialisation paresseuse : le lazy loading
- Gestion des caches et des pools de connexions
- Transactions JTA
- Les « long life sessions »
- Mapping par annotations



ENVIRONNEMENTS UTILISES

Plate-forme Java JEE, IDE Eclipse 3.2 et plugins, base de données H2 ou autres. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 402.0 € HT par stagiaire.

Intra-entreprise : 1327.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de mettre en œuvre une persistance d'objets Java s'appuyant sur des bases de données relationnelles :

- Choisir un pilote JDBC adapté au contexte du projet
- Etablir une connexion avec une base de données
- Exécuter des requêtes DDL, DML et DQL
- Gérer des transactions
- Optimiser l'usage de JDBC fonction du contexte projet

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Construire des applications maintenables et évolutives en utilisant l'héritage et le polymorphisme

METHODE

Les aspects théoriques sont concrétisés par la mise en œuvre concrète de l'API JDBC dans le contexte d'un projet en mode console. 50 % du temps est consacré à la réalisation de cette application. Chaque stagiaire possède son poste de travail.

PROGRAMME

INTRODUCTION A JDBC

- Les différents types de persistance Java (sérialisation, requêtes sur SGBDR et persistance d'objets)
- Positionnement de JDBC et constituants de base
- Typologies de pilotes JDBC et critères de choix
- Rappels SQL (create, delete, insert, update, select, ...)

ETABLIR UNE CONNEXION A UNE BASE DE DONNEES

- Chargement des pilotes JDBC
- Structure d'une URL de connexion et création d'une connexion
- Traitement des exceptions SQLException
- Fermeture de connexions

EXECUTION DE REQUETES

- L'interface Statement
- Requêtes DDL et DML, et les méthodes « execute » et « executeUpdate »
- Requêtes DQL et la méthode « executeQuery »
- Les requêtes pré compilées : interface PreparedStatement
- Paramétrage de l'objet Statement
- Accès au méta données d'une base de données

EXPLOITATION DES RESULTATS

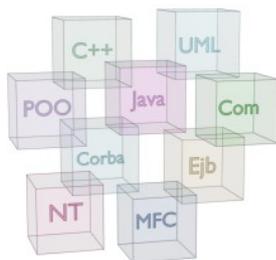
- L'interface ResultSet
- Gestion du curseur d'un ResultSet
- Méthodes d'accès aux valeurs des colonnes
- Modification de la base à partir d'un ResultSet
- Accès au méta données d'un ResultSet

MISE EN OEUVRE DES TRANSACTIONS

- Le mode auto-commit
- Commit et rollback
- Gestion des niveaux d'isolation

ASPECTS AVANCES

- Appel de procédures stockées
- Gestion d'un pool de connexions
- Accès à une data source via JNDI
- Introduction aux frameworks de persistance objet (JDO-JPOX, Hibernate)



ENVIRONNEMENTS UTILISES

Non applicable à ce séminaire.

PRIX

Inter-entreprise : 402.0 € HT par stagiaire.

Intra-entreprise : 1327.0 € HT (8 stagiaires).

OBJECTIFS

La plateforme JEE constitue aujourd'hui une réponse globale aux besoins des entreprises pour le développement d'architectures multi-niveaux. La clé du succès réside dans la capacité à définir une solution adaptée aux besoins de l'entreprise et la garantie de pouvoir faire évoluer cette solution pour s'adapter aux changements structurels et organisationnels de l'entreprise. En une journée, ce séminaire destiné à des directeurs informatiques, chefs de projets et ingénieurs commercial expose l'ensemble des concepts et connaissances indispensables à tout informaticien en charge de gérer ou de participer à des projets Java d'entreprise :

- Quels sont les constituants de la plate-forme JEE ?
- Quelles sont les architectures applicatives types pour ce type de plate-forme ?
- Quel est le rôle des frameworks et comment faire son choix ?
- Quelles méthodes et quels outils choisir pour le développement d'applications JEE ?
- Quelle plate-forme d'exploitation choisir, avec quels outils d'audit et de reporting ?

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Donner une définition des concepts objets et les bénéfices induits par leur utilisation

METHODE

Les aspects théoriques seront illustrés par de nombreux exemples issus de véritables applications JEE et des démonstrations d'outils.

PROGRAMME

FONDEMENTS DE LA TECHNOLOGIE JAVA

- Historique, objectifs et infrastructure de base
- Les différentes plates-formes et leurs cibles respectives
- Processus de décision des évolutions de Java
- Java et l'Open Source, les principaux contributeurs
- Java versus offres propriétaires

JEE ET LES APPLICATIONS WEB

- Les principaux composants techniques (servlet, JSP, JMS et autres)
- Modèle d'architecture
- Positionnement des principaux frameworks Web (Struts, Spring, Mentawai, ...)
- L'alternative JSF
- Les infrastructures d'exécution (Tomcat, Jetty, Resin, ...)
- Problématique des clients Web riches, utilisation d'Ajax
- Comparatif des différentes solutions techniques et critères de choix

JEE ET LA COUCHE D'ACCES AUX DONNEES

- Persistance : typologie de solutions (sérialisation, SGBDOO, SGBDR et autres)
- Les différentes solutions pour un support relationnel : JDBC, JDO, EJB entity, ...
- Les frameworks de persistance objet (JDO-JPOX, Hibernate, ...)
- Rôle du JNDI
- Connecteurs JCA pour accéder à d'autres sources de données
- Comparatif des différentes solutions techniques et critères de choix

JEE ET LES ARCHITECTURE DISTRIBUEES

- Problématiques des architectures applicatives distribuées
- Le modèle de composants Java distribués : les EJB
- Les apports de la spécification EJB 3
- Les architectures orientées services (SOA) et ses différentes solutions : RMI/CORBA, Web Services (SOAP et WSDL)
- Les architecture orientées messages : JMS
- Support du transactionnel avec JTA
- Gestion de la sécurité avec JCE, JSSE, et JAAS
- Les principales offres commerciales : IBM Websphere, BEA WebLogic, Borland AppServer, Oracle iAS, Sun One App Server, ...
- Briques Open Source : JBoss, JOnAS, GlassFsh, Apache Axis, ...
- Le projet ObjectWeb
- Comparatif des différentes solutions techniques et critères de choix

DEPLOIEMENT ET MISE EN EXPLOITATION

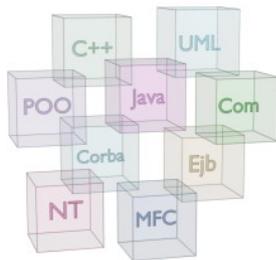
- Les formats de déploiement d'une application JEE : fichiers WAR et EAR
- Administration d'applications : la spécification JMX
- Outils JMX de référence
- Autres outils Open Source de surveillance applicative (nagios, munin, ...) et critères de choix

LES ENVIRONNEMENTS DE DEVELOPPEMENT

- Constituants du Java Development Toolkit (JDK)
- IDE : offre Open Source (Eclipse, NetBeans) et propriétaire (IBM RSA/RAD, Borland JBuilder/Together)
- Place de la modélisation UML dans le développement JEE, intégration d'AGL UML dans Eclipse
- Développement agile et outils Open Source associés : Ant Maven, Log4J, Junit, ...

- Outils analyse de la qualité du code source
- Organisation d'un projet JEE

Langages de script



ENVIRONNEMENTS UTILISES

Eclipse sous linux ou Windows Internet explorer 6.0 ou + Netscape version 7.0 ou +

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des scripts basés sur la norme ECMAScript :

- Identifier les contextes justifiant l'usage de JavaScript
- Développer des utilitaires JavaScript en utilisant une démarche procédurale
- Optimiser une solution à l'aide de l'héritage et du polymorphisme
- Traiter les cas d'erreurs à l'aide des exceptions
- Exploiter les objets prédéfinis de la norme

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué
- Organiser la solution d'un problème en procédures et fonctions

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation. Un poste de travail est fourni pour 2 stagiaires. L'atteinte des objectifs est contrôlée au fur et à mesure du déroulement du stage.

PROGRAMME

PRESENTATION DE JAVASCRIPT

- Historique et caractéristiques du langage
- Javascript versus Java
- Quand utiliser Javascript

LES ELEMENTS SYNTAXIQUES DU LANGAGES

- Les types
- Les opérations
- Les conversions de type
- Les structures de contrôles

LES FONCTIONS JAVASCRIPT

- Définition de fonctions
- Les paramètres d'appel et valeur de retour
- La variable argument
- Principales fonctions prédéfinies

LE MODELE OBJET

- Classes et instances
- Envoi de messages
- Héritage et polymorphisme

LES OBJETS EN JAVASCRIPT

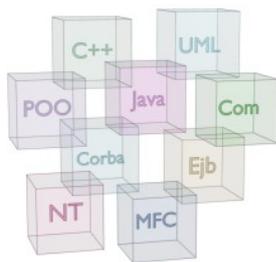
- Les concepts objet couverts par JavaScript
- Opérateurs d'instanciation et de destruction
- Ajout dynamique de propriétés
- Accès aux propriétés
- Classe et constructeur
- Auto référence this
- Héritage en JavaScript

LES EXCEPTIONS EN JAVASCRIPT

- Traiter une exception : try et catch
- Lever ses propres exceptions : throw
- L'instruction finally

LES OBJETS PRÉDÉFINIS

- Object
- Function
- Array
- String
- Number
- Date
- RegExp



ENVIRONNEMENTS UTILISES

Interpréteur Perl sous Windows ou Linux,
Internet Explorer 6.0

PRIX

Inter-entreprise : 1608.0 € HT par stagiaire.

Intra-entreprise : 5306.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications et/ou des utilitaires basés sur les mécanismes du langage PERL :

- Développer des scripts PERL structurés et maintenables
- Effectuer des traitements élaborés sur des fichiers
- Elaborer des expressions complexes
- Organiser une application PERL en packages
- Construire de petites applications internet en PERL/CGI

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation. Un poste de travail est fourni par groupe de deux stagiaires

PROGRAMME

PRESENTATION DU LANGAGE

- Programme PERL et module PERL
- Références internet
- Lancement d'un programme PERL

CONSTITUANTS D'UN PROGRAMME PERL

- Les variables : scalaires, tableaux, ...
- Variables spéciales et éléments de syntaxe
- Appel de commandes à partir d'un programme PERL
- Structures de contrôle : if, unless, while, until, for, next, foreach, last, redo, écrire un switch en PERL
- Fonctions standards de manipulation de variables
- Définition de fonctions
- Déclaration de variables locales : my et local

ACCES AUX FICHIERS

- Ouverture, fermeture, lecture, écriture, déplacement.
- Propriétés d'un fichier: fonction stat
- Opérateurs de tests sur les fichiers
- Présentation d'autres fonctions de gestion de fichiers
- Répertoires, liens physiques et symboliques
- Récupérer le contenu d'un fichier dans un hachage

LES EXPRESSIONS

- Opérateurs traditionnels et spécifiques à PERL
- Les contextes d'exécution : scalaire, liste, ...
- Expressions régulières : méta caractères
- Recherche de correspondance et substitutions
- Transformer une ligne d'un fichier en tableau

LES REFERENCES

- Référence sur scalaire, tableau, hachage, handle, ...
- Compositeurs de tableaux/hachages/sous-programmes anonymes.
- Constructeurs d'objets et opérateur -> et ref.
- tableaux multidimensionnels et enregistrements élaborés.

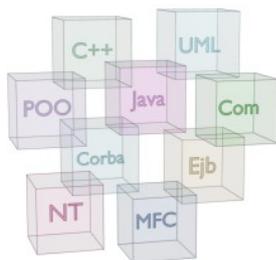
LES PACKAGES

- Définition et objectifs, directives use et require
- Blocs BEGIN et END
- Packages orientés objet : classe, instanciation, constructeur, destructeur, héritage

INTRODUCTION A LA PROGRAMMATION CGI

- Objectifs du module CGI.pm
- Principe de la page HTML dynamique
- Utilisation classique et orientée objet

Méthodes et langage UML



ENVIRONNEMENTS UTILISES

AGL UML StarUML ou Enterprise Architect
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue de ce stage, les participants seront capables de mener une analyse et une conception orientée objet en UML dans le cadre d'une démarche « Unified Process » :

- Définir les champs d'utilisation d'UML
- Exploiter une spécification des besoins réalisée à l'aide des cas d'utilisation
- Organiser les activités liées à une analyse orientée objet
- Conduire une analyse et une conception dans le cadre d'une démarche Unified Process (UP)
- Elaborer une architecture logicielle robuste et favorisant l'évolutivité
- Transformer un modèle d'analyse en un modèle de conception dans le respect des principes fondamentaux de la conception objet
- Concevoir une base de données en UML

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- une première expérience dans l'utilisation de la notation UML.

METHODE

Les aspects théoriques sont illustrés par de nombreux exemples concrets de modélisation UML, le tout consolidé par une étude de cas réalisée à l'aide d'un AGL UML.

PROGRAMME

CONTEXTE D'UN PROJET LOGICIEL

- Cycle de vie d'un projet logiciel
- Cycle en cascade versus démarche itérative
- Rappels sur les concepts objets et la notation UML

QU'EST QUE L'ANALYSE ORIENTEE OBJET

- Objectifs et limite d'une analyse orientée objet
- Conduite d'une analyse orientée objet : réalisation de cas d'utilisation et gestion des itérations
- Quels diagrammes UML utilisés pour l'analyse

PROCESSUS METHODOLOGIQUE D'UNE ANALYSE ORIENTEE OBJET

- Identification des objets candidats et élaboration des diagrammes de classes initiaux
- Recherche des responsabilités à l'aide des diagrammes de séquences
- Raffinement des diagrammes de classes
- Règles de structuration en packages
- Construction des diagrammes d'états pour les classes le nécessitant
- Contrôles de cohérence intra et inter modèles
- Itération et optimisation
- Finalisation de l'architecture logique
- Transition entre analyse et conception

L'ARCHITECTURE LOGICIELLE

- Typologies d'architectures, organisation en couches et décomposition en sous-système
- Modélisation de l'architecture en paquetages et relations inter paquetages
- Modèle d'implémentation : diagramme de composants et de déploiement

PRINCIPES FONDAMENTAUX DE CONCEPTION

- Ouverture/fermeture et substitution de Liskov
- Inversion des dépendances
- Séparation des interfaces
- Équivalence livraison /réutilisation
- Dépendances acycliques
- Les principaux Design Patterns : Singleton, Façade, Fabrique abstraite, Observateur, Adaptateur, ...

CONCEVOIR LES SOUS-SYSTEMES

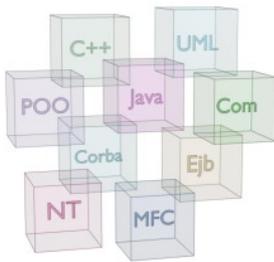
- Mapping des objets d'analyse sur les sous-systèmes
- Optimisation de la solution à l'aide de l'héritage
- Spécification des interfaces entre sous-systèmes
- Réorganisation des objets d'analyse en fonction des principes de conception et des design patterns
- Modélisation UML des sous-systèmes à l'aide des diagrammes de classe et de séquence

CONCEVOIR LES CLASSES

- Responsabilités et interfaces
- Héritage versus composition
- Conception des associations et choix des collections
- Implantation des contraintes sur les associations (agrégation, composition)
- Transformation du modèle de conception dans un langage objet
- Enrichissement manuel des sources et re-synchronisation avec le modèle (Round Trip)

MODELISATION DES DONNEES (DATA MODELING)

- Le profil « Data Modeling »
- Élaboration un modèle Entité-Relation (ER) en UML
- Transformation d'un modèle ER en un modèle Relationnel
- Règles d'optimisation du modèle relationnel
- Production de scripts DDL de création de tables



ENVIRONNEMENTS UTILISES

AGL UML Entreprise Architect ou
IBM-Rational Rose Enterprise.
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables d'élaborer des modèles de conception UML favorisant la maintenabilité et l'évolutivité et de les implémenter en C++ à partir de composants techniques génériques :

- Mettre en oeuvre un processus de conception rigoureux
- Définir et modéliser en UML une architecture système
- Organiser les choix de conception autour de formes idiomatiques (Design Patterns)
- Elaborer des modèles UML de conception détaillée
- Implanter ces modèles en langage C++

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes C++

METHODE

Chaque aspect théorique est systématiquement concrétisé par un ou plusieurs exemples de modèles de conception et son implantation en langage C++. Le tout consolidé par la réalisation d'une véritable application.

PROGRAMME

QU'EST QUE LA CONCEPTION ORIENTEE OBJET

- Objectifs de la conception
- Eléments d'entrée : modèle logique de la solution et architecture système
- Utilisation d'UML en phase de conception
- Rôles de l'AGL UML : élaboration du modèle de conception, génération de code, de documentation, reverse engineering, ...

RAPPELS SUR LA NOTATION UML

- Diagramme de classe et notation associée
- Diagramme de séquence et d'état, notation associée
- Diagramme de composant et de déploiement
- Structuration du modèle en packages

PROCESSUS METHODOLOGIQUE D'UNE CONCEPTION ORIENTEE OBJET

- Analyser les besoins non fonctionnels et identification des solutions " sur étagères "
- Bâtir l'architecture générale en s'appuyant sur un modèle en couches
- Identifier les sous-systèmes
- Modélisation UML de l'architecture à l'aide des diagrammes de package, de composant et de déploiement
- Concevoir les sous-systèmes par itération en implantant les classes d'analyse

PRINCIPES FONDAMENTAUX DE CONCEPTION

- Ouverture/fermeture et substitution de Liskov
- Inversion des dépendances
- Séparation des interfaces
- Equivalence livraison /réutilisation
- Dépendances acycliques

LES DESIGN PATTERNS

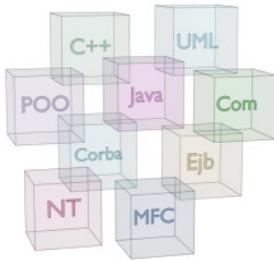
- Objectif d'un idiome (pattern) de conception
- Présentation du catalogue de patterns de GAMMA
- Singleton et Fabrique abstraite
- Façade, Composite, Adaptateur
- Observateur, Itérateur, Automate et Patron de méthode
- Instanciation des Patterns à partir d'un AGL UML

CONCEVOIR LES SOUS-SYSTEMES

- Mapping des objets d'analyse sur les sous-systèmes
- Optimisation de la solution à l'aide de l'héritage
- Spécification des interfaces entre sous-systèmes
- Réorganisation des objets d'analyse en fonction des principes de conception et des design patterns
- Modélisation UML des sous-systèmes à l'aide des diagrammes de classe et de séquence

CONCEPTION DES CLASSES

- Responsabilités et interfaces
- Choix de l'héritage " public " ou " private "
- Héritage versus composition
- Conception des associations et choix des collections C++ (bibliothèque STL)
- Implantation des contraintes sur les associations (agrégation, composition)
- Génération de code à partir du modèle de conception UML
- Enrichissement manuel des sources générés et re-synchronisation avec le modèle (Round Trip)



ENVIRONNEMENTS UTILISES

AGL UML Enterprise Architect ou IBM-Rational Rose, Plate-forme J2SE 5.0 et environnement de développement Eclipse 3.1. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1206.0 € HT par stagiaire.

Intra-entreprise : 3980.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables d'élaborer des modèles de conception UML favorisant la maintenabilité et l'évolutivité et de les implémenter en Java à partir de composants techniques génériques :

- Mettre en oeuvre un processus de conception rigoureux
- Définir et modéliser en UML une architecture système
- Organiser les choix de conception autour de formes idiomatiques (Design Patterns)
- Elaborer des modèles UML de conception détaillée
- Implanter ces modèles en langage Java

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer des classes Java

METHODE

Chaque aspect théorique est systématiquement concrétisé par un ou plusieurs exemples de modèles de conception et son implantation en langage Java. Le tout consolidé par la réalisation d'une véritable application.

PROGRAMME

QU'EST QUE LA CONCEPTION ORIENTEE OBJET

- Objectifs de la conception
- Eléments d'entrée : modèle logique de la solution et architecture système
- Utilisation d'UML en phase de conception
- Rôles de l'AGL UML : élaboration du modèle de conception, génération de code, de documentation, reverse engineering, ...

RAPPELS SUR LA NOTATION UML

- Diagramme de classe et notation associée
- Diagramme de séquence et d'état, notation associée
- Diagramme de composant et de déploiement
- Structuration du modèle en packages

PROCESSUS METHODOLOGIQUE D'UNE CONCEPTION ORIENTEE OBJET

- Analyser les besoins non fonctionnels et identification des solutions " sur étagères "
- Bâtir l'architecture générale en s'appuyant sur un modèle en couches
- Identifier les sous-systèmes
- Modélisation UML de l'architecture à l'aide des diagrammes de package, de composant et de déploiement
- Concevoir les sous-systèmes par itération en implantant les classes d'analyse

PRINCIPES FONDAMENTAUX DE CONCEPTION

- Ouverture/fermeture et substitution de Liskov
- Inversion des dépendances
- Séparation des interfaces
- Equivalence livraison /réutilisation
- Dépendances acycliques

LES DESIGN PATTERNS

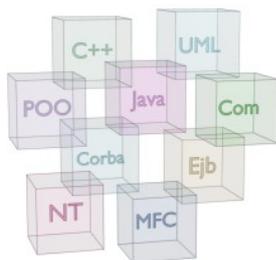
- Objectif d'un idiome (pattern) de conception
- Présentation du catalogue de patterns de GAMMA
- Singleton et Fabrique abstraite
- Façade, Composite, Adaptateur
- Observateur, Itérateur, Automate et Patron de méthode
- Instanciation des Patterns à partir d'un AGL UML

CONCEVOIR LES SOUS-SYSTEMES

- Mapping des objets d'analyse sur les sous-systèmes
- Optimisation de la solution à l'aide de l'héritage
- Spécification des interfaces entre sous-systèmes
- Réorganisation des objets d'analyse en fonction des principes de conception et des design patterns
- Modélisation UML des sous-systèmes à l'aide des diagrammes de classe et de séquence

CONCEPTION DES CLASSES

- Responsabilités et interfaces
- Héritage versus composition
- Conception des associations et choix des collections Java
- Implantation des contraintes sur les associations (agrégation, composition)
- Génération de code Java à partir du modèle UML
- Enrichissement manuel des sources et re-synchronisation avec le modèle (Round Trip)



ENVIRONNEMENTS UTILISES

Utilisation d'outils spécifiques Google Wave, IceScrum,

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

OBJECTIFS

Qu'est ce que les méthodes agiles ? faire ce que l'on veut, il n'y a plus de chef, plus de doc et enfin le programmeur est roi. Si cela n'est pas antinomique de certaines méthodes (comme Crystal), il n'en est pas de même pour les autres (XP, SCRUM) qui imposent au contraire une très forte discipline, mais apporte par contre une forte motivation. Cette introduction à l'agilité permet de faire le point sur ce sujet et fournit aux participants les critères pertinents sur lesquels doit s'effectuer un choix parmi les méthodes proposées aujourd'hui.

- Comprendre les différences fondamentales entre un projet classique et un projet agile
- Connaitre les principales méthodes agiles et leurs différences
- Donner un sens précis à la terminologie consacrée
- Mesurer les impacts produits par l'introduction d'une méthode agile sur les pratiques existantes

PREREQUIS

Aucun prérequis n'est nécessaire pour suivre cette formation.

METHODE

Formation basée sur une participation active des stagiaires et dont le déroulement s'appuie sur les principes de l'agilité. Qu'il soit développeur logiciel ou client, le stagiaire est constamment invité à exprimer ses idées (ce qu'on ne peut pas faire simplement par la lecture du Web ou de livres), et est acteur de son propre apprentissage. Les aspects conceptuels sont accompagnés de vidéos d'experts et de mises en situation (XP-game, tests, planning game, stand up, bilan, ...)

PROGRAMME

GESTION CLASSIQUE D'UN PROJET

- Cycle en V (l'histoire des Royce)
- Les techniques classiques (Gantt, Cocomo, Fibonacci, ...)
- UP et RUP
- UML

INTRODUCTION A L'AGILITÉ

- Lean software development (Muda, Kanban)
- Le manifeste agile
- Les principes de base
- Apports des méthodes agiles

PANORAMA DES MÉTHODES AGILES

Pour chaque méthode, sont étudiés : les rôles, les livrables, les spécificités, les options et le cycle proposé

- XP
- SCRUM
- Crystal
- DSDM

LES DIFFÉRENTES PHASES D'UN CYCLE (mini projet)

- Phase de définition des besoins (Use case, User story, product backlog)
- Phase de planification (planning game)
- L'itération

DÉROULEMENT D'UNE ITÉRATION (mini projet)

- Sprint backlog
- UML agile
- Découpe et distribution des tâches
- Binômes
- Stand up meeting
- Simplicité du codage
- Tests d'acceptation
- Fin d'une itération
- Les BurnDown charts
- Le bilan

OUTILS ADAPTÉS A L'AGILITÉ

Le minimum

- Script de génération automatique (Make, Ant/Maven)
- Gestion de configuration (Fusion, CVS/SVN)

Le nécessaire

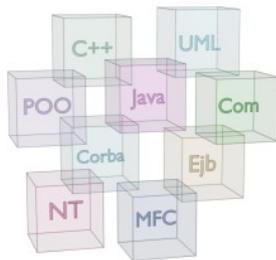
- Outils de test automatisés
- Notion de spécification exécutable
- TU avec X-Unit
- Charge avec Chess
- IHM avec Selenium
- Outils de dev avec du Refactoring

Facultatif

- Gestion de projet (XtremPlanner, iceScrum)
- Gestion des demandes (Bugzilla)

COMMENT MIGRER VERS L'AGILITÉ

- Mise en place de l'équipe, des locaux
- Rôle du Scrum master ou coach
- Définition de la méthode (les indicateurs, ...)
- Migration douce ou forte
- Différentes situations : développement au forfait, équipe distribuée, offshore, ...
- Des exemples (Google, Snfc, Projets critiques)



OBJECTIFS

A l'issue de ce stage, les participants auront appris l'importance que prend la spécification UML dans les activités de développement de logiciels industriels :

- Comprendre les raisons conduisant à spécifier des logiciels en UML
- Définir les caractéristiques fondamentales d'un processus de développement objet
- Connaître les principaux diagrammes UML et leurs utilisations
- Estimer les bénéfices induits par l'industrialisation d'un processus objet mais aussi les coûts
- Evaluer un AGL UML en fonction de critères pertinents

ENVIRONNEMENTS UTILISES

AGL UML Enterprise Architect et autres outils.

PRIX

Inter-entreprise : 402.0 € HT par stagiaire.

Intra-entreprise : 1327.0 € HT (8 stagiaires).

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Utiliser un processus de développement logiciel

METHODE

Les aspects théoriques sont illustrés par de nombreux exemples concrets de modélisation UML et la présentation de différents outils.

PROGRAMME

PRESENTATION GENERALE DU LANGAGE UML

- Modéliser : pourquoi faire ?
- Origines et objectifs du langage UML
- Présentation des principaux diagrammes UML
- Limites d'UML et apport d'une démarche méthodologique
- AGL UML et différents niveaux d'utilisation

INTRODUCTION AU PROCESSUS UNIFIE (UP)

- Caractéristiques fondamentales d'UP et finalités
- Positionnement d'UP par rapport aux démarches traditionnelles
- Les processus dérivés d'UP : RUP, YUP et Méthodes Agiles

LES DIFFERENTES PHASES D'UN PROCESSUS UP

- Formalisation des besoins fonctionnels
- Analyse et conception objet d'une solution
- La gestion des itérations
- Exemple de mise en oeuvre des différentes phases et diagrammes UML associés

INDUSTRIALISATION D'UN PROCESSUS

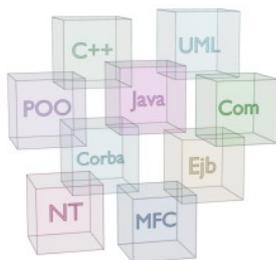
- Model Driver Development
- Model Driven Architecture
- L'importance de l'outillage

LES APPORTS DE UML 2.0

- Enrichissement de la notation
- Nouveaux diagrammes
- Ce qui est important et ce qui l'est moins

L'OFFRE AGL UML

- Les AGL UML "stand alone"
- Les plugins UML pour Eclipse
- Positionnement/prix de différentes offres (IBM Rational, Objecteering, Sparx Systems, No Magic, ...)



ENVIRONNEMENTS UTILISES

AGL UML StarUML ou Enterprise Architect.
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue de ce stage, les participants maîtriseront les principales techniques de modélisation UML employées lors des différentes phases d'un cycle de vie d'un projet logiciel :

- Modéliser les aspects fonctionnels d'un système à l'aide de diagrammes de cas d'utilisation
- Modéliser les aspects statiques d'un système à l'aide de diagramme de classes et d'objets
- Modéliser les aspects dynamiques d'un système à l'aide de diagrammes de séquence, d'activités et d'états

PREREQUIS

Aucun prérequis n'est nécessaire pour suivre cette formation.

METHODE

Les aspects théoriques sont illustrés par de nombreux exemples concrets de modélisation UML et la réalisation d'exercices de modélisation couvrant tous les aspects de la notation.

PROGRAMME

INTRODUCTION A UML

- Origines d'UML
- La modélisation UML
- UML et les processus de développement
- Premières conclusions

LE MODELE OBJET

- Encapsulation, types abstraits
- Classes et instances
- Héritage de structure et d'interface
- Redéfinition et polymorphisme
- Généricité
- Illustration UML de tous les concepts Objet (diagramme de classe et de séquence)

NOTATION UML POUR LA CAPTURE DES ASPECTS FONCTIONNELS

- Cas d'utilisation et acteurs
- Diagramme de cas d'utilisation
- Décomposition des cas d'utilisation : relations « include », « extend » et « generalize »

NOTATION UML POUR LA CAPTURE DES ASPECTS STATIQUES

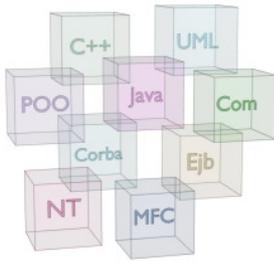
- Diagrammes de classe et de contexte
- Classes, attributs, opérations
- Association, agrégation, composition, cardinalité, navigation, rôle et qualificatif
- Attributs de lien et classe d'association
- Généralisation et interface
- Contraintes statiques (textuelle ou OCL)
- Organisation du modèle en packages

NOTATION UML POUR LA CAPTURE DES ASPECTS DYNAMIQUES

- Diagrammes de séquence et de collaboration
- Éléments syntaxiques : garde, message, itération, ...
- Contraintes temporelles
- Concepts d'état, d'événement et de transition
- Diagramme d'activités
- Diagramme d'états et notation de base
- Techniques de factorisation : entry, exit, états emboîtés

AUTRES ELEMENTS DE MODELISATION UML

- Diagrammes de composants
- Diagrammes de déploiement
- Les apports d'UML 2.0



ENVIRONNEMENTS UTILISES

AGL UML StarUML ou Enterprise Architect.
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 804.0 € HT par stagiaire.

Intra-entreprise : 2653.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue de ce stage, les participants seront capables de modéliser les processus métier de l'entreprise et de formaliser une expression de besoins fonctionnels à l'aide de modèles UML :

- Modéliser les processus métier et les workflows d'une organisation
- Structurer un besoin logiciel en catégories d'exigences
- Identifier les cas d'utilisation et les acteurs
- Modéliser les aspects statiques et dynamiques des cas d'utilisation
- Définir les éléments d'estimation de charge, de suivi de projet et de traçabilité
- Conduire une analyse des besoins dans le cadre d'une démarche Unified Process en appliquant des principes méthodologies éprouvés

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- une première expérience dans l'utilisation de la notation UML.

METHODE

Les aspects théoriques sont illustrés par de nombreux exemples concrets de modélisation UML et la réalisation d'une étude de cas issue d'un véritable projet.

PROGRAMME

CONTEXTE D'UN PROJET LOGICIEL

- Cycle de vie d'un projet logiciel
- Cycle en cascade versus démarche itérative
- Rappels sur les concepts objets et la notation UML

MODELISATION DES PROCESSUS METIER

- Typologie des processus d'une entreprise
- Modélisation UML des processus et des acteurs,
- Modélisation des workflows par des diagrammes d'activité et des diagrammes d'interactions
- Description textuelle des processus métier

SPECIFICATION DES EXIGENCES

- Critères de succès d'un projet logiciel
- Classification des exigences
- Règle d'écriture
- Outils de gestion d'exigences

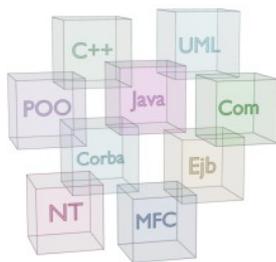
FORMALISER LES BESOINS FONCTIONNELS

- Définition du périmètre du système
- Identification des acteurs et des cas d'utilisation : définitions des rôles et ébauche du diagramme de cas d'utilisation
- Description textuelle d'un use case : canevas documentaire, scénarios nominal, alternatif, et d'exception
- Capture des entités fondamentales du domaine : ébauche du dictionnaire de données
- Règles méthodologiques d'écriture des cas d'utilisation

MODELISER EN UML LES BESOINS FONCTIONNELS

- Raffinement du diagramme des cas d'utilisation par décomposition et spécialisation
- Modéliser un scénario à l'aide de diagrammes de séquence système
- Modéliser le comportement d'un cas d'utilisation à l'aide d'un diagramme d'activité
- Cinématique de l'application et enchaînement des écrans
- Modéliser les entités du domaine à l'aide d'un diagramme de classe
- Structuration du modèle des besoins en packages
- Établissement d'une matrice de traçabilité
- Définition des contextes justifiant ces activités de modélisation

Technologies XML



ENVIRONNEMENTS UTILISES

Environnement Eclipse 3.4 et plugins associés, parser Xerces, outil de transformation Xalan, runtime Axis. Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1608.0 € HT par stagiaire.

Intra-entreprise : 5306.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue de cette formation, les participants auront acquis une vision globale sur les technologies XML et les contextes de leur utilisation:

- Décrire les domaines couverts par XML et les technologies associées
- Définir un vocabulaire XML métier
- Concevoir des documents XML à l'aide de DTD et XSD
- Concevoir des feuilles de styles XSL
- Mettre en œuvre différentes techniques de parsing de flux XML
- Stocker, transformer, véhiculer et restituer des flux de données XML
- Définir des web services, les implémenter et les mettre en œuvre

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Programmer en langage Java

METHODE

Les aspects théoriques seront concrétisés par la réalisation d'un document XML et son exploitation par les différentes technologies XML. Un poste de travail est fourni par stagiaire. L'atteinte des objectifs est contrôlée au fur et à mesure du déroulement du stage.

PROGRAMME

INTRODUCTION A LA TECHNOLOGIE XML

- Langages de balises
- Historique, SGML, HTML
- Présentation du panel des utilisations d'XML

CONSTITUANTS DU LANGAGE XML

- Principes de base : éléments, attributs, entités
- Déclarations XML
- Présentation des espaces de nommage
- Documents valides et documents bien formés

VALIDATION DE DOCUMENT XML

- Présentation et mise en œuvre des DTD
- Relation Document/DTD
- Limites et inconvénients des DTD
- Présentation et mise en œuvre des schémas XSD
- DTD versus XSD, avantages et contraintes

MODELES DE PARSING

- Les différents types de parsers XML
- Utilisation du scripting
- Le Document Object Model (DOM)
- Typage, namespaces et mécanismes de liaison (XLink)

UTILISATION DE JAXP POUR LE MODELE SAX

- Gestion des événements SAX : ContentHandler
- Localisation des erreurs : ErrorHandler
- Entités externes : EntityResolver
- Configuration de l'analyseur SAX et traitement des exceptions

UTILISATION DE JAXP POUR LE MODELE DOM

- Syntaxe de DOM pour Java
- Chargement de l'analyseur DOM
- Manipulation des nœuds DOM
- Validation de DOM et traitement des exceptions

TRANSFORMATION DE FLUX XML

- Introduction au langage XSL et principaux constituants
- Préprocesseur XSLT
- Adressage de la structure interne d'un document avec Xlink / Xpointer
- Recherche et filtrage de données avec XPath
- Introduction à de XSL-FO
- Les schématrons

PERSISTANCE XML

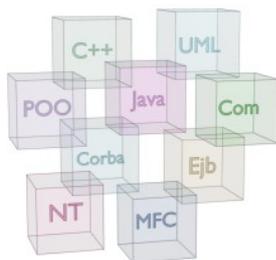
- Sérialisation XML via l'API 1.4
- Frameworks de sérialisation : XStream et autres
- Le Data Binding XML - JAVA
- Mise en œuvre via le framework Castor

INTRODUCTION AUX SERVICES WEB

- Rappel sur l'architecture N-Tiers
- Typologie de Services Web : SOAP et XML-RPC
- Description d'un service web en WSDL

CREATION D'UN SERVICE WEB AVEC AXIS

- Création d'un fichier WSDL décrivant un Service Web
- Création des classes Java associées à un Service Web (côté serveur)
- Création du descriptif de déploiement
- Déploiement du Service Web et test via un navigateur
- Création de classes clientes et test du Service Web via le client



OBJECTIFS

A l'issue de ce séminaire, les participants auront acquis une vision globale sur les technologies XML et les contextes de leur utilisation:

- Décrire les domaines couverts par XML et les technologies associées
- Définir un vocabulaire XML métier
- Décrire et positionner les différentes technologies connexes à XML
- Décrire les différentes techniques de parsing d'un fichier XML
- Décrire la logique de production de services Web
- Citer les principaux d'outils d'édition XML

ENVIRONNEMENTS UTILISES

Non applicable pour ce séminaire.

PRIX

Inter-entreprise : 406.0 € HT par stagiaire.

Intra-entreprise : 1340.0 € HT (8 stagiaires).

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Développer une application logicielle via un langage évolué

METHODE

Les aspects théoriques seront illustrés par de nombreux exemples issus d'applications réelles et des démonstrations d'outils.

PROGRAMME

INTRODUCTION A LA TECHNOLOGIE XML

- Langages de balises
- Historique, SGML, HTML
- Présentation du panel des utilisations d'XML

CONSTITUANTS DU LANGAGE XML

- Principes de base : éléments, attributs, entités
- Déclarations XML
- Présentation des espaces de nommage
- Documents valides et documents bien formés

VALIDATION DE DOCUMENT XML

- Présentation et mis en œuvre des DTD
- Relation Document/DTD
- Limites et inconvénients des DTD
- Présentation des schémas : XSD

PRESENTATION DE XSL

- Objectifs d'XSL
- Transformation à l'aide de XSLT
- Adressage de la structure interne d'un document avec Xlink / Xpointer
- Recherche et filtrage de donnée avec Xpath

LE PARSING XML

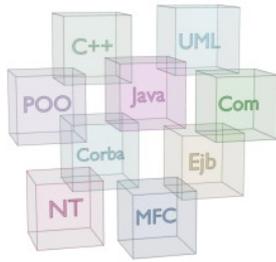
- Les différents parseurs
- Traitement de données avec SAX
- Manipulation des données avec DOM

INTRODUCTION A SOAP ET AUX SERVICES WEB

- Services Web, pour quels besoins
- Chaîne de production de services Web
- Rôle du protocole SOAP et infrastructure de mise en oeuvre

Outils XML

- Les éditeurs xml (XmlSpy, XMLWriter, Liquid XML)
- Les environnements Java/XML (plug-in eclipse, ...)
- AGL UML



ENVIRONNEMENTS UTILISES

Framework .Net et Visual Studio .Net.
Langage C#. MSDE ou SQL Server.
Environnements spécifiques sur demande.

PRIX

Inter-entreprise : 1608.0 € HT par stagiaire.

Intra-entreprise : 5306.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables de développer des applications .Net exploitant les techniques XML :

- Créer des documents XML bien formés et valides
- Manipuler des Datasets ADO .Net en mode XML
- Transformer des documents XML en HTML ou XML
- Utiliser le parser XML de .Net

PREREQUIS

Aucun prérequis n'est nécessaire pour suivre cette formation.

METHODE

Les aspects théoriques sont concrétisés par de nombreux exercices pratiques de programmation en langage C#. L'atteinte des objectifs est contrôlée au fur et à mesure du déroulement du stage.

PROGRAMME

INTRODUCTION A LA TECHNOLOGIE XML

- Langages à balises
- Historique, SGML, HTML
- Présentation du panel des utilisations d'XML : publication de données, stockage de données, communication, informatique répartie et services web (SOAP)
- Autres technologies : SVG, SMIL, XMI, DOCBOOK, ...

CONSTITUANTS DU LANGAGE XML

- Principes de base : éléments, attributs, entités
- Déclarations XML
- Structure et syntaxe
- Documents valides et documents bien formés
- Rôle des parseurs (DOM et SAX)

DOCUMENT TYPE DEFINITIONS ET SCHÉMAS XSD

- Spécification et définition des éléments et attributs
- Définition des données PCDATA et CDATA
- Traitement des répétitions de données et attributs
- DTD versus schéma XML

SERIALISATION ET ACCÈS AUX DONNÉES

- Sérialisation générique (SOAP formatter, XML serializer)
- Attributs de sérialisation
- Extraction de données sous forme de documents XML
- XML Data readers
- Sérialisation des DataSets
- Sérialisation binaire
- Diffgrams

MISE EN OEUVRE DES CONCEPTS - APPLICATION

- Support de XML dans Visual Studio .Net
- Parsing de document
- Manipulation du DOM
- Schémas, typage et Namespaces
- Récupération des données et sérialisation
- Feuilles de style / XSL

TRANSFORMATION DE XML

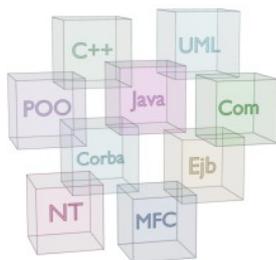
- Le langage de transformation XSLT
- Le processeur XSLT de .Net et XslTransform
- Découverte de XSL-FO
- Création de XSL pour conversion de XML en XML ou HTML
- Objets d'extension de XSLT

CLASSES XML FONDAMENTALES DANS .NET

- XML Readers
- Gestion des exceptions XML
- Émulation de SAX avec les outils .Net
- XML Reader personnalisé
- Classe XmlValidatingReader
- Validation des données (DTDs)
- Validation des données (schémas XSD)
- XML Writers
- XML Writer personnalisé

UTILISATION DE DOM EN .NET

- Classes DOM dans .Net
- Chargement et sauvegarde des documents
- Manipulation de l'arborescence DOM
- XPath (XPathNavigator & XPathNodeIterator)



OBJECTIFS

A l'issue du stage, les participants disposeront d'une connaissance globale des technologies XML et de ses standards. Ils seront en mesure d'élaborer des solutions logicielles n-tiers performantes basées sur ces technologies :

- Décrire les domaines couverts par XML et les technologies associées
- Définir un vocabulaire XML métier
- Publier des documents XML
- Stocker, transformer, véhiculer et restituer des flux de données XML
- Mettre en oeuvre des services web

ENVIRONNEMENTS UTILISES

Moteur XSL-FO FOP Base de données XML
Xindice Toolkit Apache

PRIX

Inter-entreprise : 1480.0 € HT par stagiaire.

Intra-entreprise : 4884.0 € HT (8 stagiaires).

PREREQUIS

Aucun prérequis n'est nécessaire pour suivre cette formation.

METHODE

Chaque aspect théorique est concrétisé par une mise en oeuvre effective dans le cadre d'un atelier. Un poste de travail est fourni par stagiaire.

PROGRAMME

XML ET LES TECHNOLOGIES ASSOCIEES

- Les langages à balises: SGML, HTML, XML, XHTML, apports et positionnements
- Documents XML bien formés, valides
- Les documents de validation : DTD et schémas
- Les parseurs XML
- Domaines d'application, exemples (Docbook, SVG, XML, XSL-FO, ...)
- Les namespaces
- Langages de transformations : CSS, XSLT et XPATH
- XLink et Xpointer

COMPOSITION D'UN FLUX XML

- Structure et syntaxe d'un document XML
- Déclaration XML, éléments et attributs
- Entités, blocs non parsés, instructions et notations
- DTD et schéma par l'exemple
- Stratégie et exemple d'utilisation des espaces de nommage
- Atelier : Spécification d'un vocabulaire XML multi-espaces

XML ET LA PUBLICATION MULTI-FORMATS

- Cas d'utilisation et approches
- Approche client léger : Rendu par un navigateur
- Utilisation de feuilles de styles CSS
- Transformations coté client avec XSLT
- Approche serveur et clients lourds : génération PDF et RTF
- Spécification et moteurs XSL-FO
- Atelier : Création des feuilles de styles CSS et XSLT pour la publication web et pour XSL-FO. Transformations à l'aide du moteur XSL-FO FOP

TRAITEMENTS DES FLUX XML

- Panel des approches, contextes
- SAX : L'approche événementielle
- DOM : Le standard de manipulation des données
- Les APIs Java JDOM et JPP3
- Transformations coté serveur avec XSLT : restructurations, filtres, tris et projections
- Atelier : Traitements de flux XML en Java et JavaScript

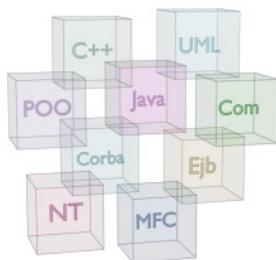
LE STOCKAGE DES DONNEES XML

- Problématique de persistance de données XML : approches data-centric et document-centric
- Les différentes spécifications et APIs de stockage
- Les langages de requêtes
- Les solutions du marché
- Atelier : exploitation de la base de données XML Xindice

LES SERVICES WEB

- Architectures n-tiers et flux XML
- Pourquoi les services web oe
- XML-RPC et SOAP, WSDL et UDDI
- Atelier : développement d'une application client-serveur basée sur des flux XML. Migration de cette application sur un service web à l'aide du toolkit Apache

Unix/Linux



ENVIRONNEMENTS UTILISES

Système Linux Red-Hat 7.3

PRIX

Inter-entreprise : 1608.0 € HT par stagiaire.

Intra-entreprise : 5306.0 € HT (8 stagiaires).

OBJECTIFS

A l'issue du stage, les participants seront capables d'exploiter une machine Linux en tant que station de travail pour développeur :

- Configurer leur environnement de travail
- Gérer leurs fichiers et travailler sur leur contenu
- Réaliser des utilitaires en shell script
- Evoluer au sein d'un réseau
- Connaître les principaux outils standards de développement
- Ajouter/supprimer des utilisateurs et configurer leur espace de travail

PREREQUIS

Cette formation s'adresse aux personnes sachant déjà :

- Décrire les tâches incombant à un système d'exploitation
- Développer une application logicielle via un langage évolué

METHODE

Les exposés théoriques sont concrétisés par de nombreux exercices pratiques sur le système LINUX. Un poste de travail est fourni par stagiaire. L'atteinte des objectifs est contrôlée au fur et à mesure du stage.

PROGRAMME

HISTORIQUE, CONNEXION ET COMMANDES DE BASE

- L'origine du système LINUX
- Notions d'utilisateur, de super-utilisateur, de groupe
- Paramétrage du terminal
- Interface graphique GNOME et utilitaires standards
- Compatibilité avec Windows

ORGANISATION DES FICHIERS

- L'arborescence des fichiers
- Commandes de manipulation des fichiers
- Modification des droits d'accès à un fichier

INTERPRETEUR DE COMMANDES

- L'interpréteur de commandes et les métacaractères
- Les redirections d'entrées-sorties
- Les tubes de communication
- Les variables d'environnement et variables utilisateur

OPERATION SUR LE CONTENU DES FICHIERS

- L'éditeurs de texte vi, emacs, xemacs
- Manipulation sur le contenu des fichiers : sort, uniq, grep, awk, ...
- Sauvegarde et restauration de fichiers (tar, cpio, ...)

PROGRAMMATION EN LANGAGE SHELL (BASH)

- Les paramètres positionnels
- Expressions arithmétiques : commande expr
- La commande test, et les structures de contrôle
- Structures de contrôle : if, case, while, until, for, ...
- Saisie clavier (commande read) et Shell script

LINUX ET LES RESEAUX

- Réseau Ethernet et adressage Internet
- Introduction au protocole TCP/IP
- Les remote commandes: rlogin, rsh, rcp, ftp, telnet, ...

OUTILS DE DEVELOPPEMENT SOUS LINUX

- Compilateurs et interpréteurs (gcc, egcs, perl, java, ...)
- IDE et autres outils (kdevelop, glade, kyllix, eclipse, ...)
- Chaîne de compilation (autoconf, automake, ...)
- Libraries libres et licences

GESTION DES UTILISATEURS

- Les droits d'accès et la sécurité
- Les fichiers /etc/passwd et /etc/group
- Ajout/suppression d'un utilisateur
- Configuration des espaces de travail des utilisateurs