

Catalogue Formation

2007

- Méthodes
- Management et Organisation
- Usine Logicielle
- Architectures
- Java/JEE
- XML/XSL
- .Net
- PhP

OXiane

L'informatique mature

Usine logicielle, UML 2, UP, Processus, MDA, SOA, Java EE, .Net, XML, PhP

Editorial	3
Présentation d'Oxiane	4
Moyens pédagogiques	5
Calendrier inter-entreprises 2007	6-7
Filière Méthodes	8-21
Filière Management et organisation	22-27
Filière Usine Logicielle	28-39
Filière Architecture	40-52
Filière Java – Java EE	53-66
Filière XML/XSL	67-72
Filière .Net	73-76
Filière PhP	77-80
Index	81
Conditions générales de vente	82
Bulletin d'inscription	83



L'informatique est –une fois de plus me direz-vous– en pleine mutation. S'agit-il d'un mouvement de fond ? De techno-buzz alimenté par le marketing et la presse ? D'une rupture tranquille, pour reprendre un terme à la mode ?

En tous les cas, "agilité", "adaptation à la stratégie d'entreprise", "pilotage par les métiers", "transversalités", "performance", "garanties de services", "contrat utilisateur", "productivité des développements" ne vous sont aujourd'hui pas étrangers si vous êtes acteur de près ou de loin des mutations de votre Système Informatique. Peut être même peut-on parler de "Système d'Information", car l'utilisateur, le processus métier, la stratégie d'entreprise figurent désormais –dans la volonté au moins si ce n'est encore dans la pratique– au centre des préoccupations.

Face à ces aspirations, les techniciens ont des idées, et même des solutions ... UML 2, UP, SOA, ESB, urbanisation et cartographie, usine logicielle, MDA, démarche processus, Web Services, Objets et Composants, Web 2.0, référentiel qualité ITIL ou CMM ...

Un jargon certes technique mais qu'il est nécessaire de décrypter. Car, oui, l'informatique amorce à travers les concepts sous-jacents de formidables mutations tant technologiques que méthodologiques.

L'architecture orientée services (SOA) n'est pas un effet de mode. Ce n'est pas non plus une technologie en soi mais l'arrivée à maturité de concepts et démarches éprouvés capables d'organiser « in fine » le S.I. en briques réellement indépendantes, pilotables par les processus (métiers), transverses aux acteurs tant internes qu'externes. Cerise sur le gâteau, l'approche n'impose pas le big-bang mais permet d'intégrer au fur et à mesure un existant et faire évoluer globalement le S.I. par retouches successives (on parle alors d'urbanisation).

Le développement piloté par les modèles (MDA) qui s'appuie entre autres sur UML, sera sans nul doute une révolution dans la façon de conduire le développement d'applications informatiques. Il ouvre la voie à l'industrialisation des développements par **l'usine logicielle**, la "software factory" tant attendue. Process qualité, référentiel partagé, intégration continue, génération automatisée ... produiront de plus en plus vite les services qui enrichiront une architecture "agile", type SOA ... N'est ce pas là une nouvelle révolution industrielle ?

Anticiper, veiller, comprendre, vous **accompagner** dans ces mutations technologiques et méthodologiques, vous entraîner vers une "Informatique mature", tels sont notre défi et notre vocation.

Pour compléter votre savoir faire, nous vous proposons cette année, en collaboration avec **Jean Planet Conseil**, cabinet spécialiste des RH, management et organisation, une filière qui semble bien éloignée de nos habituelles préoccupations.

Et pourtant, la démarche processus à l'usage des métiers est sans nul doute le meilleur préalable à la construction d'un S.I. « agile » ; la conduite de projet, l'animation d'une équipe dans une perspective de changements, une réalité quotidienne peu maîtrisée par nos Chefs de Projets ; le réseau individuel une source de valeur additionnelle du salarié et des organisations ... encore faut-il savoir l'entretenir, l'utiliser, le développer.

Stéphane LIETARD
Président

L'équipe

Oxiane est un cabinet de conseil et de formation technologique. Fondée en février 2000, la société est animée par une équipe d'une dizaine de consultants-formateurs de grande expérience (5 à 16 ans).

Les plus anciens d'entre nous ont commencé à aborder l'objet et les architectures distribuées avant l'avènement du web et de java. Nous parlions alors par exemple de Smalltalk, C++ ou encore Corba.

La société est basée à Boulogne-Billancourt et la majeure partie de notre activité se déroule en Ile-de-France ... mais pas uniquement puisque nous sommes capable de venir animer vos sessions de formations en intra-entreprise en Province et à l'Etranger !

Nos activités

- Les sessions de formations en inter et intra entreprises représentent plus de la moitié de notre activité
- Les missions de coaching et conseil technologique représentent un peu moins d'1/4 de notre activité. Il s'agit de missions très ponctuelles, menées sur une base forfaitaire et qui peuvent prendre diverses formes :
 - accompagnement à la définition d'une architecture,
 - aide aux choix techniques,
 - réalisation de prototype et « *proof of concept* »
 - suivi d'une équipe à la suite d'un cursus de formation
 - urbanisation, cartographie
 - expression des besoins avec UML
- Enfin, nous développons des logiciels de gestion et nous accompagnons leur intégration chez nos clients. C'est une activité passionnante mais qui reste confidentielle.

Nos consultants-formateurs interviennent en mission environ 70% de leur temps. Le temps restant est important ; il est consacré à la veille technologique et à la communication, à la mise au point de nouveaux supports et enfin à la participation à des projets open-source.

Cette façon de travailler construit notre densité technologique ce qui assure, nous en sommes convaincus, notre plus-value et votre retour sur investissement.

Ils nous font confiance

Voici une liste non exhaustive de certains de nos clients parmi les plus fidèles mais également de partenaires de qualité avec qui nous co-investissons ... Merci à tous !.

Atos Origin	Ippon Technologies
Beijaflore	Les Petits Frères Des Pauvres
CVF	Ministère de la Défense
Demos	Prodixia
GFI Informatique	Vianova

Locaux :

A proximité de la porte de Saint-Cloud et du nouveau grand chantier de Paris, l'Île Seguin, les locaux d'Oxiane sont situés dans un cadre exceptionnel pour votre formation aux portes de Paris !

Nos locaux sont agréables et faciles d'accès par les transports en commun : métro, nombreux bus et tramway.

- La station Billancourt, métro ligne 9 se trouve à 50 mètres du centre de formation
- Le tram, ligne T2 est à moins de 5mn à pied et met le centre de formation Oxiane à 20 minutes de La Défense

Par des investissements réguliers, nous offrons un parc de machines récentes et performantes, complètement dédiées à la formation qui permettent aux participants d'avoir une vision concrète de la diversité des IDE, des outils et des plates-formes dès les formations de premier niveau.

Oxiane dispose de deux salles de formation de 10 à 12 places climatisées.

Formations, Supports de cours et Outils pédagogiques

Oxiane est spécialisée dans la conception et la réalisation de supports pédagogiques sur les dernières technologies. Les supports de cours sont réalisés en interne et sont régulièrement actualisés, à l'image des formations elles-mêmes. Ils sont généralement constitués de trois parties :

- Un classeur de formation : ce support est nécessaire pour suivre l'exposé théorique du formateur. La personne qui suit la formation peut également y consigner des notes personnelles dans les zones prévues à cet effet. Des renvois réguliers vers le cahier de travaux pratiques (voir ci-dessous) sont annotés afin de signaler les nombreuses alternances exposés-travaux pratiques ;
- Un cahier de travaux pratiques : dans un format plus réduit que le classeur de formation, ce cahier contient les énoncés des exercices et représente, lorsque cela est adapté, la progression des travaux réalisés. Ainsi, nous utilisons fréquemment un formalisme UML simplifié pour représenter à chaque étape des travaux pratiques les évolutions de l'application en cours de construction (diagramme de classes UML par exemple). Nos sessions mettent l'accent sur les travaux pratiques qui sont effectués au fur et à mesure de la présentation des thèmes abordés ;
- Un cahier de solutions : la mise en œuvre d'un cahier de solutions n'est pas systématique (en fonction de la nature de la formation) mais pour des formations techniques intégrant la création progressive d'une application, nous systématisons la fourniture de ce type de document. Ainsi, après chaque exercice, nous mettons en évidence le code qui doit être ajouté/modifié. Généralement, ce cahier n'est pas fourni lors de la formation mais après celle-ci, ce qui évite aux stagiaires d'être trop influencés par une solution standard.

Nos stages regroupent délibérément un maximum de 10 participants.

Les cours inter-entreprises organisés dans les locaux d'Oxiane comprennent les pauses café et les repas du midi.

Les Consultants/Formateurs

Les intervenants d'Oxiane sont des experts reconnus. Spécialisés par filière, ils bénéficient d'un accès privilégié aux informations techniques des éditeurs et constructeurs. Ils interviennent régulièrement dans le cadre d'études, de conseils et de projets de développement.

Ils possèdent une culture pédagogique. La qualité de leur enseignement est fondée notamment sur l'interaction entre la mise en œuvre et la communication des techniques. Leur expertise et leur expérience, en environnement réel, font de leurs modules une source d'enrichissement théorique, concrète et opérationnelle.

Les consultants/formateurs d'Oxiane analysent les demandes en terme d'évolution informatique et d'évolution de compétences. Ainsi chaque action de formation s'appuie sur un audit de la population à former, de ses connaissances, de son métier, de ses outils informatiques et des finalités opérationnelles recherchées.

INTITULE DU STAGE	Durée	Réf.	Janv.	Fév.	Mars	Avril	Mai	Juin	Juillet	Août	Sept.	Oct.	Nov.	Déc.	Tarif €
Filière : Méthodes															
Concepts Objet	1 j	UM1	8	5	5	2	2	1	2	6	3	1	5	3	400
Notation UML 2	1 j	UM2	9	6	6	3	3	2	3	7	4	2	6	4	400
Modélisation métier avec UML 2.0	2 j	UM3	10-11	7-8	7-8	4-5	4-5	3-4	4-5	8-9	5-6	3-4	7-8	5-6	800
Modélisation des processus métier avec BPMN au travers d'une étude de cas	2 j	UM4	18-19		15-16		24-25	14-15		23-24		11-12		6-7	800
Modélisation des exigences avec UML 2 au travers d'une étude de cas	2 j	UM5		7-8		4-5		3-4	4-5		5-6		7-8		800
Analyse, conception et Design Patterns avec UML 2 avec étude de cas	5 j	UM6		12-16		2-6		11-15		20-24		8-12		3-7	2 000
Les apports d'UML2 : la synthèse	1 j	UM7	12	9	9	6	6	5	6	10	7	5	9	7	400
Design Patterns	3 j	UM8	15-17		12-14		21-23	11-13		20-22		8-10		3-5	1 200
Java/J2EE : Architecture Pilotée par les Composants	2 j	UM9	18-19		15-16		24-25	14-15		23-24		11-12		6-7	800
Conduire un projet avec le Processus Unifié et UML 2	2 j	UM10		7-8		4-5		3-4	4-5		5-6		7-8		800
Etude de cas complète JEE5 – UML 2 – UP	5 j	UM11	15-19		12-16		21-25		9-13		10-14		5-9		2 000
SysML	1 j	UM12		6		3		2		7		2		4	400
Urbanisation du Système d'Information	2 j	UM13		7-8		4-5		3-4		8-9		3-4		5-6	800

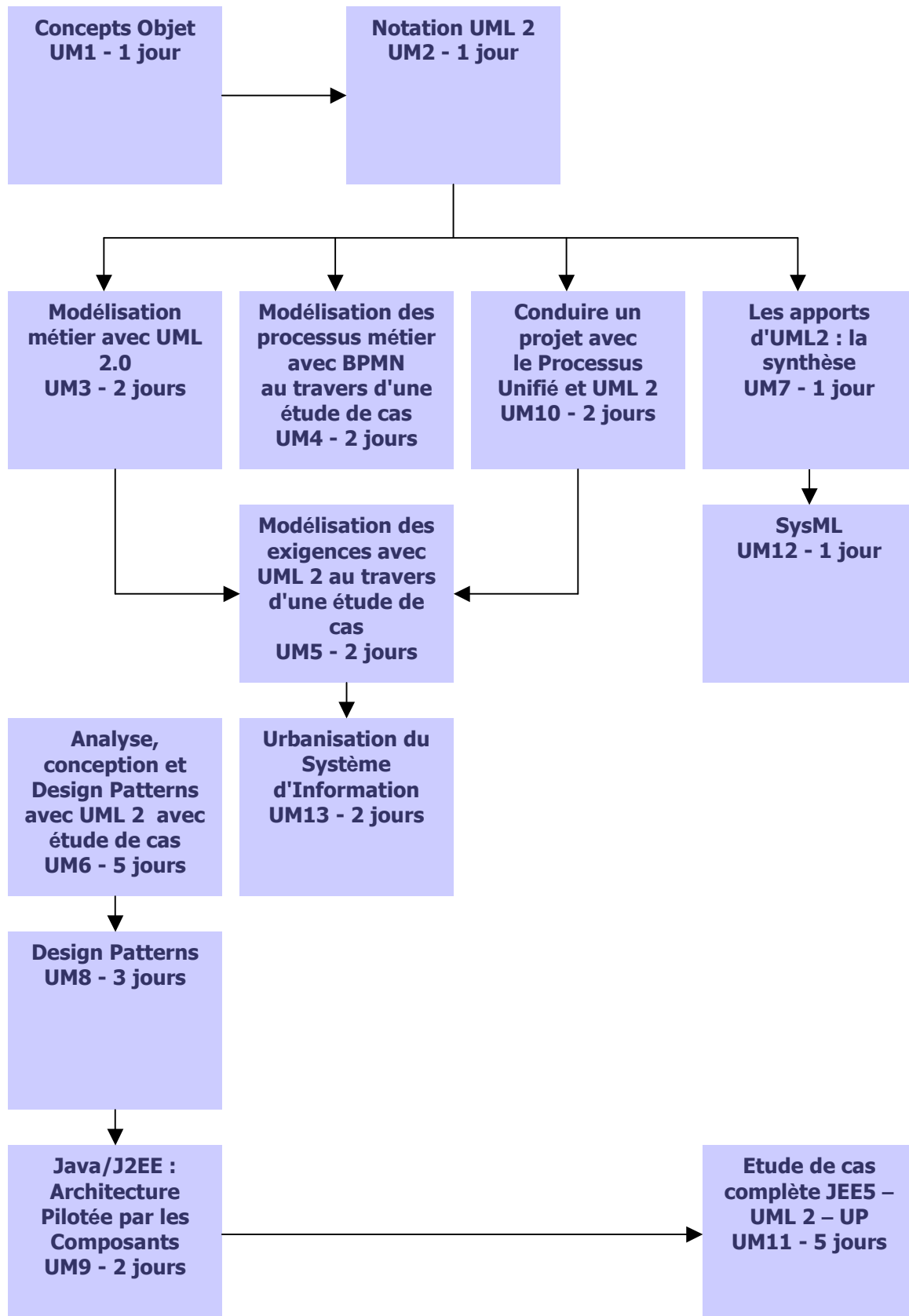
Filière : Management & Organisation															
Conduire le changement dans un projet	1 j	MO1			14	24		20				9		6	800
Identifier et formaliser vos processus afin de préparer la mise en place d'un système d'information	1 j	MO2			20	19		21				11		4	800
Identifier et formaliser vos processus en vue d'améliorer la qualité de vos prestations	1 j	MO3			14	24		20				9		6	800
Animer une équipe	1 j	MO4		14		25		25			25		21		800
Le réseau : L'Individu au sein de la dynamique collective	1 j	MO5		15		26		26			26		22		800

Filière : Usine logicielle															
L'usine logicielle : concepts et mise en oeuvre	1 j	MD0	8	5	5	2	2	1	2	6	3	1	5	3	400
Développement Pilotée par les Modèles (MDA)	2 j	MD1	9-10	6-7	6-7	3-4	3-4	4-5	3-4	7-8	4-5	2-3	6-7	4-5	800
MDA : concepts et mise en oeuvre avec Borland Together	3 j	MD2	8-10	5-7	5-7	2-4	2-4	4-6	2-4	1-3	3-5	1-3	5-7	4-6	1 200
MDA : concepts et mise en oeuvre avec AndroMDA	3 j	MD3	8-10	5-7	5-7	2-4	2-4	4-6	2-4	1-3	3-5	1-3	5-7	4-6	1 200
MDA : concepts et mise en oeuvre avec AOW	3 j	MD4	8-10	5-7	5-7	2-4	2-4	4-6	2-4	1-3	3-5	1-3	5-7	4-6	1 200
MDA : concepts et mise en oeuvre avec RSA	3 j	MD5	8-10	5-7	5-7	2-4	2-4	4-6	2-4	1-3	3-5	1-3	5-7	4-6	1 200
Travail en groupe avec Subversion	1 j	TW1	8		5		2		2		3		5		400
Subversion avancé	1 j	TW2	9		6		3		3		4		6		400
Intégration en Continue avec Ant / Maven / Cruise Control	1 j	IC1		7		4		3		8		3		6	400
Tests unitaires avec Junit et Cactus	1 j	TDD1		5		2		1		6		1		3	400
Tests fonctionnel avec JMeter	1 j	TDD2		6		3		2		7		2		4	400

Filière Architecture															
SOA (Architecture Orientée Services)	3 j	AR1	8-10		5-7		2-4		2-4		3-5		5-7		1 200
Web Services : concepts et mise en oeuvre avec Java	3 j	AR2		5-7		2-4		4-6		1-3		1-3	28-30		1 200
Web Services : concepts et mise en oeuvre avec .net	3 j	AR3	15-17		12-14		21-23	11-13		20-22		8-10		3-5	1 200
Web Services : concepts et mise en oeuvre avec PHP	3 j	AR4		5-7		2-4		4-6		1-3		1-3	28-30		1 200
Développer des applications avec Struts	3 j	AR5	29-31		26-28		9-11		23-25		24-26		26-28		1 200
Persistence objet avec Hibernate	3 j	AR6		26-28		10-12		25-27		29-31		22-24		17-19	1 200
Développer des applications avec Spring	3 j	AR7	29-31		26-28		9-11		23-25		24-26		26-28		1 200
Développer des composants EJB 3	3 j	AR8		26-28		10-12		25-27		29-31		22-24		17-19	1 200
Applications Web2.0 avec Ajax	3 j	AR9	15-17		12-14		21-23	11-13		20-22		8-10		3-5	1 200
Développer des applications avec Tapestry	3 j	AR10		12-14		02-04			9-11		10-12		5-7		1 200
Administration JBoss	2 j	AR11		15-16		05-06			12-13		13-14		8-9		800
Administration JOnAS	2 j	AR12	18-19		15-16		24-25	14-15		23-24		11-12		6-7	800

INTITULE DU STAGE	Durée	Réf.	Janv.	Fév.	Mars	Avril	Mai	Juin	Juillet	Août	Sept.	Oct.	Nov.	Déc.	Tarif €
Filière : Java/JEE															
Java – Les bonnes pratiques de développement avec Eclipse	5 j	JA1	15-19	12-16	12-16	2-6	21-25	11-15	9-13	20-24	10-14	8-12	5-9	3-7	2 000
Java – Les bonnes pratiques de développement avec WSAD	5 j	JA2	15-19	12-16	12-16	2-6	21-25	11-15	9-13	20-24	10-14	8-12	5-9	3-7	2 000
Java – Les bonnes pratiques de développement avec RAD	5 j	JA3	15-19	12-16	12-16	2-6	21-25	11-15	9-13	20-24	10-14	8-12	5-9	3-7	2 000
Java – Les bonnes pratiques de développement avec NetBeans	5 j	JA4	15-19	12-16	12-16	2-6	21-25	11-15	9-13	20-24	10-14	8-12	5-9	3-7	2 000
Java/JEE – Avancé	3 j	JA5	10-12		8-9		19-20	24-25	28-29		27-28			10-11	1 200
Création d'application Client riches avec Swing	2 j	JA6	8-9		5-7		16-18	21-23	25-27		24-26			7-9	800
Création d'application Client riches avec Eclipse RCP et SWT	3 j	JA7		26-28		10-12		25-27		29-31		22-24		17-19	1 200
Réaliser des applications Web avec JEE et Tomcat	4 j	JA8	22-25	19-22	19-22	10-13	29-1er	18-22	16-19	27-30	17-20	15-18	12-15	10-13	1 600
Réaliser des applications Web avec JEE et WebSphere	4 j	JA9	22-25	19-22	19-22	10-13	29-1er	18-22	16-19	27-30	17-20	15-18	12-15	10-13	1 600
Réaliser des applications Web avec JEE et JBoss	4 j	JA10	22-25	19-22	19-22	10-13	29-1er	18-22	16-19	27-30	17-20	15-18	12-15	10-13	1 600
Réaliser des applications Web avec JEE et JOnAS	4 j	JA11	22-25	19-22	19-22	10-13	29-1er	18-22	16-19	27-30	17-20	15-18	12-15	10-13	1 600
Développer des applications avec JSF	3 j	JA12		12-14		02-04			9-11		10-12		5-7		1 200
Mustang, les Apports de Java 6	1 j	JA13			7		4		3		8		3		400
Filière : XML/XSL															
HTML, DHTML et Javascript	3 j	XM1	15-17		12-14		21-23		9-11		10-12		5-7		1 200
Les concepts de XML	2 j	XM2	18-19		15-16		24-25		12-13		13-14		8-9		800
Manipuler du XML avec les APIs Java/JEE	2 j	XM3		15-16		05-06		14-15		23-24		11-12		6-7	800
Développer des applications avec XSLT	2 j	XM4	8-9		5-7		16-18	21-23	25-27		24-26			7-9	800
XSLT avancé	3 j	XM5	10-12		8-9		19-20	24-25	28-29		27-28			10-11	1 200
Filière .NET															
Programmation en C#	5j	NE1	15-19	12-16	12-16	2-6	21-25	11-15	9-13	20-24	10-14	8-12	5-9	3-7	2 000
Programmation avec ADO .NET	2 j	NE2		15-16		05-06			12-13		13-14		8-9		800
Développement d'applications web avec ASP .NET	5 j	NE3	8-12		5-9		16-20	21-25	25-29		24-28			7-11	2 000
Filière Php															
PHP - Les bonnes pratiques de développement	3 j	PH1	8-10		5-7		2-4		2-4		3-5		5-7		1 200
PHP - Optimisation et expertise	3 j	PH2		5-7		2-4		4-6		1-3		1-3	28-30		1 200
Symphony : Concepts et Mise en œuvre du framework	3 j	PH3	15-17		12-14		21-23	11-13		20-22		8-10		3-5	1 200





CODE

UM1

DURÉE

1 jour

OBJECTIFS

Pouvoir aborder une problématique sous l'angle de la Programmation Orientée Objet

Assimiler et mettre en œuvre les concepts

Comprendre la réutilisation de composants

Connaître les normes et standards, les langages de développement et de modélisation, les méthodes et les outils

Acquérir les principes de bases de la modélisation UML

PUBLIC

Informaticiens et décideurs issus d'une culture procédurale

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

400 € H.T. par personne

Maîtriser les Concepts Objet et UML 2.0

Les langages de dernière génération –C++, Java, .net mais aussi PhP et tant d'autres– ont rendu la Programmation Orientée Objet incontournable. Développer une application Internet/Intranet, un service ou un composant dans un environnement distribué implique de réfléchir, concevoir et modéliser des « Objets ».

Pour l'informaticien habitué à aborder une problématique sous l'angle procédural, « passer à l'objet » est un changement radical qui se fera dans la durée.

Ce cours est pour nous le préalable nécessaire à toute formation aux nouvelles technologies des Systèmes d'Information.

Programme

Etat de l'art des technologies objet

- L'évolution des langages de programmation
- Comparaison entre l'approche fonctionnelle et l'approche objet
- Etat de l'art concernant les langages et technologies objet actuelles : C++, Java, C#, PHP
- Le langage de modélisation UML 2.0
- Les bonnes pratiques de gestion de projet objet : le Processus Unifié (UP)

Les concepts fondamentaux

- Abstraction, objets et classes
- Encapsulation, attributs et méthodes
- Généralisation et spécialisation, classification, principe d'héritage
- Relations entre les classes, association, agrégation, composition et dépendance
- Communication et collaboration entre les objets
- Programmation par contrat, les interfaces
- Polymorphisme
- Les composants interchangeables

Les avantages de l'objet dans un projet

- Représentation naturelle des entités métier
- Organisation et compréhension du système
- Système évolutif à base de composants interchangeables et réutilisables
- Réduction du couplage entre les composants
- Les Design Patterns de conception, des solutions optimisées pour les problèmes récurrents

Modélisation objet avec UML 2.0

- L'importance de la modélisation
- La vue statique d'un système, les diagrammes de classe
- La vue dynamique, les diagrammes de séquence, de communication, d'état et d'activité

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8	5	5	2	2	1	2	6	3	1	5	3

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

Maîtriser la notation UML 2

CODE

UM2

DURÉE

1 jour

OBJECTIFS

Connaître les diagrammes UML pour savoir les interpréter,

Savoir utiliser les diagrammes UML pour modéliser un système

Connaître les enjeux et bonnes pratiques d'UML 2

PUBLIC

Concepteurs/développeurs,

Analystes,

Chefs de projets,

Equipe projet,

Architectes technique

PRE-REQUIS

Connaissances équivalentes au stage UM1 – Maîtriser les concepts Objet

TARIF D'INSCRIPTION

400 € H.T. par personne

En 10 ans, UML s'est imposé comme le standard en matière de langage de modélisation des systèmes informatiques.

De l'expression des besoins au déploiement, en passant par la conception, utiliser ou tout simplement comprendre UML est un incontournable de tout acteur du S.I., qu'il soit maître d'ouvrage ou maître d'œuvre.

Enfin, la nouvelle mouture d'UML se veut plus productive et devient le support premier d'un développement orienté MDA (Model Development Architecture) : de contemplatifs, les modèles deviennent productifs.

Ce cours permet de comprendre la notation, les enjeux de chaque type de modèle et ouvre les perspectives apportées par UML 2.0 : l'usine logicielle et MDA.

Programme

Les concepts fondamentaux

- Historiques d'UML
- Objectifs de la modélisation
- Les bonnes pratiques UML

Le diagramme de classe

- Classe, attributs, méthodes, encapsulation
- Classe abstraite, interface
- Relations : héritage, implémentation, association, agrégation, composition, dépendance
- Exercices d'application

Le diagramme de package

- Définition
- Encapsulation
- Relations : inport, access
- Fusion de package (merge)
- Exercices d'application

Le diagramme de structure composite

- Définition
- Collaboration
- Exercices d'application

Le diagramme de composant

- Définition
- Composant
- Exercices d'application

Le diagramme de déploiement

- Artefacts, nœuds
- Déploiement
- Exercices d'application

Le diagramme de cas d'utilisation

- Objectifs
- Introduction à la discipline « Exigences fonctionnelles »
- Définition d'un cas d'utilisation
- Les bonnes pratiques
- Exercices d'application

Le diagramme d'état

- Machine d'états
- Evènements
- Exercices d'application

Le diagramme d'activité

- Activités et actions
- Jeton et nœuds d'activité
- Exercices d'application

Les diagrammes d'interactions : séquence, communication, vue d'ensemble des interactions et timing

- Acteur d'une interaction
- Message
- Occurrence d'exécution
- Invariant d'état
- Les fragments combinés
- Chronométrage de séquence
- Exercices d'application

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
9	6	6	3	3	2	3	7	4	2	6	4

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM3

DURÉE

2 jours

OBJECTIFS

Comprendre les enjeux d'une modélisation d'une entreprise ou d'une organisation

Comprendre les objectifs de la discipline « Modélisation métier » dans la méthode de développement logiciel UP (Unified Process)

Savoir modéliser les processus métier avec UML et les extensions standards

Savoir modéliser les entités métier avec UML

PUBLIC

Analystes, chefs de projets MOA/MOE, experts métier, rédacteur de cahier des charges

PRE-REQUIS

Connaissances équivalentes au UM1 - Concepts Objet

TARIF D'INSCRIPTION

800 € H.T. par personne

Mettre en oeuvre la modélisation métier avec UML 2.0

Les entreprises qui souhaitent s'appuyer sur les nouvelles technologies de l'information pour développer leurs avantages concurrentiels placeront naturellement la connaissance de leur métier au cœur de leur stratégie de développement du S.I.

Dès lors, il importe que les experts métiers s'impliquent dans la modélisation de leur savoir faire ... afin de le faire savoir.

Ce cours s'adresse à tous ceux qui accompagnent de près ou de loin la mutation du S.I. de leur entreprise.

Programme

Présentation du Processus Unifié (UP)

- Objectifs
- Les grands principes d'UP
- Les disciplines et les phases d'UP

Le Processus méthodologique du Business Modeling

- Objectifs, terminologie et méta-modèles
- Méthodes et outils de réalisation de glossaire métier
- Comment modéliser les règles métier ?
- Comment modéliser la cartographie de l'organisation ?
- Modèle de contexte statique (diagramme de classe) et modèle de contexte dynamique (diagramme de communication)
- Acteurs métier et Use Case métier (Cas d'Utilisation métier), utilisation de stéréotypes UML
- Modèle Use Case métier
- Description sommaire des Use Case métier
- Méthode et fiche type pour rédiger en détail et efficacement les Use Case métier
- Affiner et structurer le modèle de Use Case métier avec les packages et les relations normalisées include, extend et l'héritage
- Définition d'un processus métier et méta-modèles UML
- Modéliser les processus métier avec le profil UML Eriksson Penker
- Diagramme d'activités UML
- Activité et action
- Flux d'information
- Synchronisation
- Les branchements conditionnels
- Emission et réception de signal
- Travailleurs métier et réalisation des Processus métier
- Cartographie des processus
- Les processus « client », « support » et « interne »
- Les règles d'usage et les bonnes pratiques pour la modélisation de processus métier
- Méthodes et techniques pour identifier les objets métier
- Modélisation des objets métier (Business Object Model) avec le diagramme de classe UML
- Affiner le Business Object Model avec les concepts avancés du diagramme de classe
- Modéliser les états des objets métier et les transitions entre états avec le diagramme d'état UML (automate d'états)
- Propriétés des états
- Les événements
- Etats composites
- Règles de cohérence
- Relations entre les processus métier et les automates d'états
- Optimisation avec les Business Patterns
- Composants métier, classe primaire et classe secondaire, règles de regroupement
- Architecture métier
- Comment préparer la discipline UP suivante : Exigences et expression des besoins

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
10 et 11	7 et 8	7 et 8	4 et 5	4 et 5	3 et 4	4 et 5	8 et 9	5 et 6	3 et 4	7 et 8	5 et 6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM4

DURÉE

2 jours

OBJECTIFS

Comprendre les enjeux de la modélisation des processus

Comprendre les objectifs de la discipline « Modélisation métier » dans la méthode de développement logiciel UP (Unified Process)

Savoir modéliser les processus métier avec BPMN (Business Process Modeling Notation)

PUBLIC

Consultants en stratégie d'entreprise,

Analystes métier,

Urbanistes,

Chefs de Projets MOA/MOE,

Experts métier

PRE-REQUIS

Connaissances équivalentes au stage UM1 – Maîtriser les concepts Objet

TARIF D'INSCRIPTION

800 € H.T. par personne

Modélisation des processus métier avec BPMN au travers d'une étude de cas

Le système d'information s'organise désormais de plus en plus autour des processus de l'entreprise. Il s'agit là d'une innovation considérable. Ainsi l'identification et la modélisation des processus métiers devient une étape cruciale de la mise en forme d'un Système d'Information dit "agile" (ou encore « orienté service », SOA).

Pour que cette mutation soit réussie, il faut que les métiers s'impliquent activement dans la mise en forme et la maîtrise de leurs processus, qu'ils adhèrent à la démarche de formalisation des processus et enfin qu'ils disposent d'un langage de représentation commun.

Programme

Présentation du Processus Unifié (UP)

- Objectifs
- Les grands principes d'UP
- Les disciplines et les phases d'UP

Modélisation métier (Business Modeling)

- Objectifs, terminologie et méta-modèles
- Méthodes et outils de réalisation de glossaire métier
- Comment modéliser les règles métier ?
- Comment modéliser la cartographie de l'organisation ?
- Modèle de contexte statique (diagramme de classe) et modèle de contexte dynamique (diagramme de communication)
- Acteurs métier et Use Case métier (Cas d'Utilisation métier), utilisation de stéréotypes UML
- Modèle Use Case métier
- Description sommaire des Use Case métier
- Méthode et fiche type pour rédiger en détail et efficacement les Use Case métier
- Affiner et structurer le modèle de Use Case métier avec les packages et les relations normalisées include, extend et l'héritage
- Définition d'un processus métier et méta-modèles UML

BPMN (Business Process Modeling Notation)

- Les concepts
- Les éléments de modélisation
- Activités
- Taches

- Evènements (début, intermédiaire et fin)
- Passerelles (inclusives, exclusives, complexes, parallèles)
- Connecteurs
- Flot de séquences
- Flot de message
- Association
- Swinlanes, pools, lanes
- Artefacts
- Annotations textuelles
- Objets données
- Groupes
- Méthodologie de modélisation de processus métier avec BPMN
- Guideline de modélisation
- Orchestration et chorégraphie des processus métier

Le modèle structurel avec UML 2

- Méthodes et techniques pour identifier les objets métier
- Modélisation des objets métier (Business Object Model) avec le diagramme de classe UML
- Affiner le Business Object Model avec les concepts avancés du diagramme de classe
- Modéliser les états des objets métier et les transitions entre états avec le diagramme d'état UML
- Composants métier, classe primaire et classe secondaire, règles de regroupement
- Comment préparer la discipline UP suivante : Exigences et expression des besoins

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
18 au 19		15 au 16		24 au 25	14 et 15		23 au 24		11 au 12		6 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM5

DURÉE

2 jours

OBJECTIFS

Comprendre les objectifs de la discipline « Exigences et expression des besoins » dans la méthode de développement logiciel objet UP (Unified Process)

Savoir utiliser la discipline « Modélisation métier » pour réaliser la discipline « Exigences »

Savoir mettre en oeuvre les techniques pour identifier les acteurs et les cas d'utilisation du futur système informatique

Etre capable d'utiliser UML pour décrire les besoins

PUBLIC

Analystes, chefs de projets MOA/MOE, experts métier, rédacteur de cahier des charges

PRE-REQUIS

Connaissances équivalentes au stage UM1 – Maîtriser les concepts Objet

TARIF D'INSCRIPTION

800 € H.T. par personne

Modélisation des exigences avec UML 2 au travers d'une étude de cas

La formalisation des exigences est un bon préalable à la conduite réussie d'un projet informatique. Pour la maîtrise d'ouvrage, savoir exprimer toutes ses exigences, tant implicites qu'explicites, c'est l'assurance d'avoir décrit de manière complète ce qu'on attend. Pour la maîtrise d'œuvre, au delà de l'intérêt d'un contrat clair (qu'elle aura peut être contribué à construire) avec sa maîtrise d'ouvrage, le référentiel d'exigences est un outil puissant de traçabilité tout au long des phases du projet.

Programme

Présentation du Processus Unifié (UP)

- Objectifs
- Les grands principes d'UP
- Les disciplines et les phases d'UP

Processus méthodologique de la discipline Exigences (Requirements)

- Définir le périmètre du système
- Elaboration du dictionnaire des termes métier
- Comment élaborer un dictionnaire efficace et complet intégrant la traçabilité dans le SI ?
- Elaboration de la liste des exigences fonctionnelles et non fonctionnelles
- Liste des règles de gestion
- Modélisation statique du contexte
- Modélisation dynamique du contexte
- Comment identifier les acteurs et les Use Case ?
- Quelle doit être la granularité d'un Use Case ?
- Décomposition : processus, activités, use case, scénarios, interactions
- Acteur principal et secondaire
- Diagramme UML de Use Case UML, le Use Case Model
- Affiner le Use Case Model avec les packages, les relations include, extend et l'héritage
- Techniques et fiche type pour rédiger efficacement les Use Case
- Modélisation des scénarios de Use Case avec les diagrammes de séquence UML 2, intégrant les fragments (loop, alt, ...)
- Réalisation d'un Use Case avec le diagramme de séquence et les stéréotypes UML (boundary, controller et entity)
- Modélisation des objets métier (Business Object Model) avec le diagramme de classe UML
- Affiner le Business Object Model avec les concepts avancés du diagramme de classe
- Modéliser les états des objets métier et les transitions entre états avec le diagramme d'état UML
- Optimisation avec les Business Patterns
- Composants métier, classe primaire et classe secondaire, règles de regroupement
- Architecture métier
- Passage de la modélisation métier aux exigences
- Relations entre le modèle métier et le modèle des besoins
- Modèle métier comme entrée des 2 premières couches du système
- Modèle métier et acteurs du futur système informatique
- Passage des Use case métier aux Use Case
- Correspondance Modélisation métier et modélisation des besoins
- Automatisation des travailleurs métiers
- Passage des entités métiers aux classes d'analyse
- Gérer la traçabilité des exigences
- Présentation de la discipline suivante : « Analyse et Conception » (Analysis & Design)

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	7 et 8		4 et 5		3 et 4	4 et 5		5 et 6		7 et 8	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM6

DURÉE

5 jours

OBJECTIFS

Comprendre les enjeux de l'analyse et de la conception

Savoir utiliser les 13 diagrammes de la nouvelle version UML 2.0

Maîtriser les principaux Design Pattern

Comprendre les principes de la démarche d'ingénierie logicielle UP (Unified Process)

PUBLIC

Chefs de projets MOA/MOE,

Experts métier,

Analystes,

Equipe projet

PRE-REQUIS

Connaissances équivalentes au UM1 - Concepts Objet

TARIF D'INSCRIPTION

2 000 € H.T. par personne

Analyse, conception avec UML 2 et UP avec étude de cas

Programme

Les fondamentaux de l'orienté objet

- Rappels

Présentation des architectures

- Architectures à objets distribués, architectures n-tiers, modèle en couches
- Présentation des architectures J2EE et .NET
- Architecture SOA
- Services web

Présentation du Processus Unifié (UP)

- Historique et Objectifs
- Le développement itératif
- Cascade vs Itérative
- Les 6 bonnes pratiques UP
- Processus de génie logiciel, structure dynamique et statique
- Les 9 disciplines : Modélisation métier, Exigences, Analyse et Conception, Implémentation, Test, Déploiement, Gestion de la Configuration et du Changement, Gestion de Projet, Environnement
- Les 4 phases : Initialisation, Elaboration, Construction et Transition
- Les rôles

Présentation d'UML 2.0

- Présentation des 13 diagrammes
- Compréhension du méta-modèle

Modéliser les exigences et les besoins

- Objectifs de la discipline
- Comment identifier les acteurs et les Use Case ?
- Découverte de la granularité
- Décomposition : processus, activité, use case, scénario, interaction
- Diagramme d'Activités
- Diagramme de Use Case
- Techniques et fiche type pour rédiger efficacement les Use Case
- Modélisation des scénarios avec les diagrammes de Séquences

Analyse et Conception objet

- Vue d'ensemble de la démarche
- Définir une architecture candidate
- Découvrir les Use Case
- Modèle du domaine, diagramme de classes, entités du domaine et attributs
- Les relations : héritage, réalisation, association, agrégation, composition, dépendance, ...
- Rôles et multiplicité
- Collaboration
- Diagramme de structure composite
- Diagramme de packages
- Analyser le comportement du système
- Diagramme d'états
- Evènements et messages système
- Diagrammes de séquence
- Stéréotypes : Boundary, Controller et Entity
- Etats du système
- Définition de responsabilités des classes
- Couplage faible
- Les diagrammes d'interaction : séquence et communication
- Diagrammes de classes de conception
- Classes de programmation
- Navigabilité et méthodes, encapsulation
- Le langage de contraintes OCL
- Le paradigme composant
- Diagramme de Composants
- Diagramme de Déploiement
- Diagramme de Timing
- Les profils: stéréotype, tagged value et contraintes en OCL

Les principaux Design Patterns

- Etude détaillée de fabrique, façade, itérateur, visiteur, état, IoC

De la conception à l'implémentation

- Traduction des classes de conception dans un langage cible
- Obtenir les corps des méthodes
- Outils de génération de code

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	12 au 16		2 au 6		11 au 15		20 au 24		8 au 12		3 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM7

DURÉE

1 jour

OBJECTIFS

Comprendre les enjeux de la nouvelle version majeure d'UML

Comprendre les impacts des évolutions d'UML sur les modèles existants

Comprendre l'intérêt pour l'architecture pilotée par les modèles (MDA) et les usines logicielles

Savoir mettre en pratique les nouveaux concepts et diagrammes UML 2

Connaître les outils UML 2

PUBLIC

Concepteurs/développeurs, chefs de projets MOA/MOE, équipe projet, architectes technique, analystes, experts métier

PRE-REQUIS

Connaissances équivalentes au UM1 - Concepts Objet et avoir pratiqué UML 1.x

TARIF D'INSCRIPTION

400 € H.T. par personne

Les apports d'UML2 : la synthèse

En 10 ans, UML s'est imposé comme le standard en matière de langage de modélisation des systèmes informatiques.

De l'expression des besoins au déploiement, en passant par la conception, utiliser ou tout simplement comprendre UML est un incontournable de tout acteur du S.I., qu'il soit maître d'ouvrage ou maître d'œuvre.

La nouvelle mouture d'UML se veut plus productive et devient le support premier d'un développement orienté MDA (Model Driven Architecture) : de contemplatifs, les modèles deviennent productifs.

Ce cours permet de comprendre la notation, les enjeux de chaque type de modèle et ouvre les perspectives apportées par UML 2.0, l'usine logicielle et MDA.

Programme

Présentation

- Objectifs de la nouvelle version : MDA, temps réel et levée des ambiguïtés de UML 1.x
- Synoptique des nouveautés
- Organisation : infrastructure et superstructure
- L'OMG intègre enfin une notion de conformité

Diagramme de structure composite, diagramme de classe, diagramme de composant et diagramme de déploiement

- Objectifs
- Notion de « part », « connector » et « port »
- Communication entre les structures
- Interfaces « required » et « provided »
- Les artefacts

Diagramme de séquence et diagramme d'interaction overview

- Objectifs
- Fragments
- Opérateurs : loop, alt, par, ...

Diagramme de communication (anciennement collaboration)

- Objectifs
- Nouveautés

Diagramme de timing

- Objectifs
- Modélisation des systèmes temps réel

Diagrammes d'états

- Nouveautés
- Automates de protocoles et de comportements

Diagrammes d'activités

- Nouveautés
- Décompositions d'activités, InputPin / OutputPin, flots, connecteurs

MDA (Model Driven Architecture)

- Objectifs
- Les apports d'UML 2 dans MDA
- Les nouvelles versions d'OCL (Object Constraint Language), MOF (Meta Object Facility) et XMI (XML Metadata Interchange)

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
12	9	9	6	6	5	6	10	7	5	9	7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM8

DURÉE

3 jours

OBJECTIFS

Comprendre les Design Patterns

Savoir trouver le (ou les) Design Pattern(s) répondant à une problématique donnée

Savoir les utiliser et les modifier pour répondre à un besoin spécifique.

PUBLIC

Concepteurs/développeurs, chefs de projets MOA/MOE, équipe projet, architectes technique, analystes, experts métier

PRE-REQUIS

Connaissances équivalentes à :

UM1 - Concepts Objet,

UM2 - Notation UML 2,

UM6 - Analyse, conception et Design Patterns avec UML 2 avec étude de cas

JA1 - Java – Les bonnes pratiques de développement avec Eclipse

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Design Patterns

Maîtriser les bonnes pratiques de conception, c'est dans un premier temps l'assurance de savoir mieux appréhender un environnement technique qui les utilise.

C'est aussi pouvoir échanger plus facilement avec des experts techniques, sur les bases d'un langage commun.

C'est enfin la capacité à mettre en œuvre une pratique éprouvée face à une problématique type que l'on reconnaît, sans avoir à chercher.

Les designs patterns sont la boîte à outil au quotidien du développeur-expert objet.

Programme

Les fondamentaux de l'orienté objet et UML

- Notion d'objet, de classe, d'encapsulation
- Les relations d'héritage, d'implémentation d'interface et les associations
- Les diagrammes UML

L'origine des Design Patterns

- La genèse
- Leur rôle: avoir des solutions « génériques »
- L'organisation du catalogue des Design Patterns

L'utilisation des Designs Patterns

- Identifier le problème
- Trouver le pattern générique répondant au problème
- Créer sa solution en réutilisant la solution générique

Les Patterns créateurs

- Singleton
- Fabrique Abstraite
- Fabrication

Les Patterns structuraux

- Composite
- Adaptateur
- Décorateur
- Façade
- Proxy

Les Patterns comportementaux

- Itérateur
- Etat
- Observateur
- Commande
- Visiteur
- Chaîne de responsabilité

Les GRASP patterns (General Responsibility Assignment Software Patterns)

- Les meilleures pratiques pour affecter les responsabilités aux classes
- Contrôleur, Créateur, Expert, Loi de Demeter, Faible couplage/Forte cohésion, Polymorphisme, Pure Fabrication, « ne parle pas aux inconnus »

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 17		12 au 14		21 au 23	11 au 13	20 au 22			8 au 10		3 au 5

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM9

DURÉE

2 jours

OBJECTIFS

Acquérir les principes de la conception d'un composant

Disposer des arguments sur les apports des Design Pattern et les avantages d'une conception par composants ainsi que des modalités de mise en oeuvre

PUBLIC

Concepteurs/développeurs, chefs de projets MOA/MOE, équipe projet, architectes technique, analystes, experts métier

PRE-REQUIS

Connaissances équivalentes à :

UM1 - Concepts Objet,

UM2 - Notation UML 2

JA1 - Java – Les bonnes pratiques de développement

TARIF D'INSCRIPTION

800 € H.T. par personne

Java EE : Architecture Pilotée par les Composants

Le terme de « composant » est choisi par analogie à l'électronique et c'est bien là l'équivalence recherchée : un composant logiciel se doit d'être testable individuellement, puis « enfichable » et immédiatement opérationnel dans une entité logicielle plus complexe.

Ce cours montre comment réaliser des composants logiciels interchangeable dans un monde Java.

Programme

Les raisons de la conception par composants

- Les 6 niveaux d'une architecture moderne
- Structure d'une application web
- Architecture à objets distribués
- Nécessité des frameworks techniques
- Méthodologie de développement logiciel : UP et XP basées sur les composants
- Structure d'un composant

Concevoir des composants

- Le modèle métier
- Design Pattern d'analyse
- Classe primaire et secondaires
- Amélioration de la cohérence sémantique
- Patterns de réduction de couplage
- Design Pattern de conception
- Avantages des interfaces
- Intégration des patterns Singleton, Façade et Factory au sein d'un composant
- Communication entre composants
- Conception générique
- Conception détaillée

Techniques de composants distribués

- Pattern proxy
- Classes techniques de type Stub ou Skeleton
- Illustration au travers de CORBA et J2EE-EJB

Persistance des composants

- Mapping Objet-Relationnel
- Les techniques de persistance avec les EJB
- Exemple de frameworks ORM : Hibernate

Les composants transactionnels

- Rappel sur les transactions
- Mécanisme du support transactionnel dans les EJB

Etude de cas complète

- Présentation de l'architecture, des produits et outils utilisés
- Du modèle métier vers la conception détaillée

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
18 au 19		15 au 16		24 au 25	14 et 15		23 au 24		11 au 12		6 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM10

DURÉE

2 jours

OBJECTIFS

Comprendre les principes de la démarche d'ingénierie logicielle UP (Unified Process)

Connaître les apports d'une démarche agile et être capable d'obtenir l'adhésion de tous les acteurs d'un projet

Etre capable de mettre en oeuvre un processus itératif, incrémental et piloté par les Use Case

Savoir exploiter le formalisme UML 2.0 dans toutes les disciplines de la méthode

Savoir estimer un projet piloté par les Use Case

Savoir gérer les risques

PUBLIC

Chefs de projets MOA/MOE,

Responsables méthodes et qualité

PRE-REQUIS

Avoir suivi le stage UM1 « Concepts Objet et UML 2.0 » ou avoir l'expérience équivalente.

Connaître les bases de la gestion de projet (les outils de planification Pert, Gantt et l'organisation d'un projet)

TARIF D'INSCRIPTION

800 € H.T. par personne

Conduire un projet avec UP et UML 2

Les changements fréquents d'exigences et les évolutions techniques ne permettent plus de conduire un projet de manière classique. C'est en partant de ce constat que les pères fondateurs d'UML ont créé la méthode UP (Unified Process ou Processus Unifié).

Pour prendre en compte les changements, le chef de projet moderne doit adopter une démarche itérative et y adapter sa planification.

Ce module décrit le Processus Unifié permettant de prendre en compte les modifications des exigences et de mieux estimer un projet grâce à la méthode des points de Cas d'Utilisation.

Programme

Historique

- Processus en cascade (linéaire), ses points forts et ses faiblesses
- Nombreux échecs, quelles raisons ?

Le développement itératif

- La solution aux problèmes rencontrés par les processus classiques
- Organisation d'un cycle, caractéristiques d'une itération
- Mise en place des tests dès le début du processus de développement
- Intégration précoce et en continu
- Planification prédictive contre planification adaptative
- Comment choisir ?

Les 6 bonnes pratiques UP et mise en oeuvre du processus

- Gestion des exigences et des besoins
- Mise en place d'une architecture pilotée par les composants ou par les modèles (MDA, Model Driven Architecture)
- Modélisation avec UML
- Vérification en continu de la qualité
- Gestion des changements
- Identifier les risques et établir un plan de réductions de risques

Présentation d'UML 2.0

- Objectifs
- Rappels des concepts objet
- Les axes de modélisation
- Présentation des 13 diagrammes de la version 2.x

La phase d'initialisation

- Les acteurs et les Use Case essentiels
- Revue de projet, le jalon LCO (Lifecycle objectives milestone)

La phase d'élaboration

- Comprendre en détail les exigences
- Concevoir, implémenter, valider l'architecture et en établir une version de référence
- Revue de projet, le jalon LCA (Lifecycle architecture milestone)

La phase de construction

- Paralléliser les développements, minimiser les coûts
- Revue de projet, le jalon IOC (Initial operational capability milestone)

La phase de transition

- Exécuter les bêtas tests
- Préparer le déploiement et récupérer les données
- Recette du projet
- Revue de projet, le jalon PR (Product release milestone)

Méthodes d'estimation

- Méthode des points de cas d'utilisation
- Tableaux de paramètres d'ajustements
- Outils d'estimations

Planifier un projet UP

- Le plan de projet
- Les plans d'itération

Autres méthodes agiles

- 2TUP (2 Tracks Unified Process)
- XP (eXtreme Programming)
- Synthèse, récapitulatif et comparatif des méthodes

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8	5	5	2	2	1	2	6	3	1	5	3

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM11

DURÉE

5 jours

OBJECTIFS

Reproduire l'ensemble du processus de réalisation d'une application Java EE

Comprendre les étapes d'un développement logiciel dans le cadre de la méthodologie UP (Unified Process)

Comprendre une architecture n tiers, « in situ »

Savoir intégrer les frameworks Servlet/JSP, EJB, JSF, Struts, Hibernate et Spring

PUBLIC

Concepteurs/développeurs, chefs de projets MOE

PRE-REQUIS

Connaissances équivalentes : UM1 - Concepts Objet,

UM3 - Modélisation métier avec UML 2.0,

UM5 - Modélisation des exigences avec UML 2,

JA1 - Java – Les bonnes pratiques de développement

JA10 - Réaliser des applications Web avec Java EE

AR8 - Développer des composants EJB 3,

AR6 - Persistance objet avec Hibernate,

AR2 – Web Services - concepts et mise en oeuvre avec Java,

TARIF D'INSCRIPTION

2 000 € H.T. par personne

Etude de cas complète JEE5 – UML 2 – UP

Ce cours aborde une étude de cas traitée dans un contexte opérationnel similaire à celui que vit une équipe projet.

Toutes les étapes de réalisation d'une application sont abordées, de l'expression des besoins à la recette.

Le participant utilisera les principales technologies Java EE couplées aux frameworks les plus éprouvés (Struts, Hibernate, Spring).

Programme

Objectifs

- Objectif : réalisation du système informatique d'une SSII
- Appropriation de la description de l'organisation, des Interviews des collaborateurs et du cahier des charges
- Gestion de projet
- Gestion de la configuration et du changement
- Choix des Use Case les plus risqués et les plus représentatifs de l'architecture à valider

La phase d'initialisation

- La vision du projet
- Modélisation métier
- Exigences
- Environnement
- Gestion de projet
- Les différentes architectures candidates
- Etude d'opportunité et de faisabilité
- Etudes des risques
- Les fonctionnalités essentielles
- Revue de projet

La phase d'élaboration

- Business Modeling
- Exigences
- Analyse et Conception
- Implémentation
- Tests
- Environnement

Les autres phases UP

- La phase de construction, déploiement
- La phase de transition, recette de l'application

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 19		12 au 16		21 au 25		9 au 13		10 au 14		5 au 9	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM12

DURÉE

1 jour

OBJECTIFS

Comprendre les concepts de modélisation technique d'un système informatique

Savoir spécifier les exigences techniques, la structure d'un logiciel, son comportement, les allocations de ressources système et les contraintes non fonctionnelles.

Savoir modéliser les entités métier avec UML

PUBLIC

Ingénieurs système,

Architectes technique

PRE-REQUIS

Connaissances équivalentes à UM2 - Notation UML 2

TARIF D'INSCRIPTION

400 € H.T. par personne

SysML

SysML (Systems Modeling Language) est un ensemble de profils UML 2 normalisés par l'OMG spécialisés dans la modélisation des systèmes complexes.

Utilisé en ingénierie de systèmes, SysML permet d'appliquer une approche objet et la formalisation des exigences à ce domaine.

Programme

Présentation

- Architecture du langage
- Mécanismes d'extension (profil UML)
- Les diagrammes SysML
- Relations avec UML 2
- Les 4 catégories de diagrammes : structurel, contraintes, comportemental et exigences

Les diagrammes structurels

- Diagramme de package
- Diagramme de « Blocks »
- Les « Blocks » internes
- Les « parts », « ports », « connectors » et les « flows »
- Les références à des propriétés
- Les « blocks » de contraintes

Les diagrammes de contraintes

- Utilisation de formules mathématiques reliant des propriétés
- Exemple : équations de la cinématique et dynamique d'un véhicule

Les diagrammes comportementaux

- Diagramme d'activités
- Utilisation de « pins » et d' « object node »
- Utilisation des « swimlanes »
- Le profil EFFBD (Enhanced Functional Flow Block Diagram)
- Diagramme d'interactions, diagramme de séquence
- Diagramme d'état
- Diagramme de Use cases
- Diagramme d'allocations

Les diagrammes d'exigences

- Définitions et caractéristiques
- Relations entre les exigences
- Stéréotypes et modélisation des exigences

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	6		3		2		7		2		4

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

UM13

DURÉE

2 jours

OBJECTIFS

Connaître les enjeux de l'urbanisation

Comprendre le principe, la méthode et les outils de modélisation

Savoir cartographier le SI suivant les 4 couches : métier, fonctionnelle, applicative et technique

PUBLIC

Chefs de projets,

Responsable méthodes et qualité,

Urbaniste SI,

Architectes technique

PRE-REQUIS

Connaissance d'ensemble du Système d'Information

TARIF D'INSCRIPTION

800 € H.T. par personne

Urbanisation du Système d'Information

La mise en place d'un Système d'Information dit « agile » passe sans doute par sa mutation réussie vers une SOA (Services Oriented Architecture). Cette démarche impose d'exprimer en amont ses « processus métier », indépendamment de toute considération informatique. Au delà des intérêts du métier (identification des processus de façon formelle, échanges, optimisation), cette étape fera apparaître –pour l'informaticien– la granularité des « services » nécessaires au S.I.

Sauf opération de « big-bang » qui consisterait à tout reconstruire, cette démarche préalable ne va pas sans un indispensable état des lieux précis (la cartographie) de l'existant construit par strates technologiques successives, du mainframe aux Internet/Intranet en passant par les applications client/serveur des années 90.

Si ces deux approches semblent de prime abord contradictoires, organiser leur convergence de façon itérative est l'un des enjeux de l'urbanisation. On parle d'ailleurs d'urbanisation par métaphore avec la cité : si celle-ci se transforme considérablement au fil du temps, c'est par touches successives où toute nouvelle modification s'intégrera à l'existant local sans bouleverser l'environnement plus général.

L'urbanisation aide à définir les priorités, à piloter, à mettre en œuvre un « plan » structuré qui fixe le cap et sera partagé par l'ensemble des acteurs.

Programme

Les enjeux de l'urbanisation

- Rationaliser le système d'information (SI)
- Rendre le SI plus modulaire et plus réactif
- Améliorer la communication entre les composantes du SI
- La métaphore de la cité
- Outil de base : la cartographie du SI

Les concepts fondamentaux

- Le métamodèle et les concepts
- Les 4 couches : métier, fonctionnelle, applicative et technique
- Les règles appliquées à chaque niveau
- Identification de la stratégie
- Modélisation de la stratégie
- Cartographie du SI
- Urbanisme et processus métier
- Modélisation des processus métier et la relation avec la stratégie
- Les bonnes pratiques pour la modélisation des processus métier

- Evaluation et amélioration des processus
- Urbanisme et architecture fonctionnelle
- Relation entre l'architecture métier et fonctionnelle
- Les bonnes pratiques pour l'architecture fonctionnelle
- Urbanisme et architecture applicative
- Relation entre l'architecture fonctionnelle et applicative
- Les bonnes pratiques pour l'architecture applicative
- Transition de l'architecture applicative à l'architecture technique
- L'architecture technique logique n-tiers

Le processus méthodologique

- Recherche des axes stratégiques
- Formalisation de l'existant
- Elaboration de la stratégie
- Mise au point du plan de convergence

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	7 et 8		4 et 5		3 et 4		8 et 9		3 et 4		5 et 6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter



Le Cabinet JEAN PLANET CONSEIL, spécialiste des questions relatives à **l'organisation**, aux **ressources humaines**, à la **conduite du changement et de projet**, travaille depuis de nombreuses années en partenariat avec les fondateurs d'Oxiane.

Ses formations, assurées par ses consultants, reposent sur l'expertise acquise sur le terrain lors des interventions de conseil auprès d'entreprises, d'établissements publics et d'administrations.

Elles poursuivent un double objectif :

- **ouvrir** les spécialistes des secteurs à haute technicité, tel que les SI, aux thématiques organisationnelles et managériales concernant l'ensemble des professionnels d'une organisation ;
- fournir des **pratiques de travail communes** facilitant la compréhension et la conduite de projets transversaux entre les DSI et les directions utilisatrices (DRH, Direction financière...)

CODE

MO1

DURÉE

1 journée

OBJECTIFS

Acquérir les concepts, les méthodes et les outils pour une action de changement efficace au sein d'une conduite de projet ;

Lier la décision stratégique à la mise en œuvre d'un projet : mesurer l'importance du facteur humain afin d'évaluer les pièges et les potentialités.

PUBLIC

Décideurs, chefs de projets, analystes, concepteurs, développeurs

TARIF D'INSCRIPTION

800 € H.T. par personne

Conduire le changement dans un projet

L'accompagnement du changement lors de la conduite d'un projet est indispensable dans un contexte organisationnel en perpétuelle transformation. Plus que jamais, les projets, décidés en fonction d'une analyse stratégique et économique, doivent être conduits de façon à accompagner les évolutions socio-organisationnelles qu'ils induisent. L'ensemble de ces mutations a rendu plus complexe le métier de chef de projet. Si sa mission première reste le respect des délais, des coûts et de la qualité des résultats, il devient l'acteur incontournable par lequel transite l'évolution des modes de fonctionnement individuels et collectifs.

Cette formation permet aux participants de s'approprier une méthodologie de conduite d'un projet tout en explorant les possibilités et les facteurs du changement.

Programme

Les principes de base du changement

- Les trois niveaux : la personne, l'équipe et la structure
- Les attitudes face au changement : modes de pensée et d'action des acteurs
- L'articulation entre vision, stratégie et projets
- L'identification des leviers et des résistances à la conduite du changement
- Les principes de base de la communication
- La prise en compte de la dimension temporelle dans la conduite du changement : donner du temps au temps

Piloter une action de changement : conduire un projet

- Les impacts d'un projet sur son environnement : compréhension du contexte et de la réalité de l'organisation
- Respecter les étapes fondamentales d'un management de projet :
 - sensibilisation : connaître les schéma mentaux des principaux acteurs, donner du "sens" au projet, identifier et gérer les impacts...
 - préparation : identifier les objectifs, anticiper les risques et les freins, annoncer le changement, établir le plan de communication...

- déroulement : piloter les actions, développer la dynamique, mobiliser, responsabiliser, concevoir de nouveaux modes de fonctionnement
- clôture : évaluer les résultats et capitaliser

Rôles et responsabilités des décideurs

- Mettre en place un système de délégation et de contrôle
- Faire remonter les bonnes et les mauvaises informations
- Prévenir les conflits d'intérêts et les risques socio-professionnels

Mobiliser : le rôle du chef de projet

- Favoriser la coopération entre les acteurs : les attitudes face au projet
- Développer les synergies : prendre conscience des fonctionnements individuels
- Savoir organiser le travail collaboratif pour valoriser individuellement le travail collectif
- Savoir mobiliser : techniques et méthodes pour établir un climat de confiance
- Accroître sa capacité à être force de propositions

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
		14	24		20				9		6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MO2

DUREE

1 journée

OBJECTIFS

Comprendre et analyser le fonctionnement de l'équipe dans son environnement.

Appréhender les différents leviers de la motivation.

Apprendre à s'affirmer dans les situations difficiles

PUBLIC

Opérationnels et encadrants intermédiaires chargés d'animer tous types d'équipe (permanente, temporaire)

TARIF D'INSCRIPTION

800 € H.T. par personne

Animer une équipe

Une somme d'individus compétents ne donne pas forcément naissance à une équipe performante.

L'adéquation n'est possible que si l'animateur de l'équipe sait tirer la plus grande efficacité des volontés et des compétences de ses collaborateurs dans le but d'accomplir le projet ou la mission donnée à l'équipe.

Un animateur qui n'a pas conscience des difficultés inhérentes à la gestion de l'humain, matière mouvante et insaisissable par excellence, s'expose aux résistances, aux conflits..., autant d'éléments contre productifs perturbant la réalisation des objectifs .

Loin d'être inné, le management d'une équipe est une des conditions de la performance d'un service ou d'un projet. Il demande un réel savoir-être et savoir-faire.

Aucune règle stricte ne peut être valable. Il est juste possible de définir quelques points de repères, quelques propositions méthodologiques sur lesquelles s'appuyer en fonction de la conjoncture.

Programme

Les préalables à l'animation de son équipe

- Mobiliser l'équipe autour des enjeux et des valeurs communes
- Analyser les sources de résistance
- Clarifier les **missions, les rôles et les objectifs de ses collaborateurs**

Adapter le style de management à son équipe

- Comprendre les types d'autorité, les styles de management et les utiliser en fonction des collaborateurs
- Les dix erreurs à éviter dans le management d'équipe

Connaître et motiver les membres de son équipe

- Identifier les motivations, compétences et le potentiel de chacun.
- Les différents ressorts de motivation, délégation et implication (les enjeux de la délégation, avantages et risques, l'échec du collaborateur, analyse des causes ...)
- Motiver par les relations (conduire un entretien d'évaluation et d'écoute, donner des signes de reconnaissance...)
- Evaluer et valoriser les performances de ses collaborateurs

Communiquer avec l'équipe

- Connaître les différents niveaux de communication et les outils de la communication
- Comprendre (écoute active) et être compris, être convaincant
- Organiser les communications inter-métier

Prévoir et résoudre les conflits

- Adapter son attitude en fonction des différents types de conflits
- Méthodes et techniques pour prévenir les conflits
- Méthodes et techniques pour résoudre les conflits (arbitrage, collaboration, négociation, médiation...)

Chaque thème donne lieu à un exercice de simulation d'une situation à laquelle peut se retrouver confronté tout animateur d'équipe.

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
		20	19		21				11		4

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MO3

DUREE

1 journée

OBJECTIFS

Savoir analyser, dans son environnement de travail, les réseaux en action;

Etre capable de comprendre, de participer, de créer et d'animer un tel type d'organisation, dans le cadre d'un projet faisant appel à une diversité de compétences et de comportements.

PUBLIC

Décideur, manager, expert, tout salarié en situation de responsabilité au sein d'une équipe engagée dans un projet

TARIF D'INSCRIPTION

800 € H.T. par personne

Le réseau : l'individu au cœur de la dynamique collective

La notion de réseau, prise au plan social et organisationnel, désigne l'entrelacement des liens psychologiques et sociaux qui unissent les individus au sein d'un groupe. Ce concept - par le développement des TIC - prend aujourd'hui une ampleur nouvelle, car les atouts organisationnels se conjuguent avec une approche renouvelée du management

Le réseau permet en effet :

- *d'inventer une nouvelle forme d'efficacité en introduisant de la souplesse par la constitution de relations professionnelles et de liens personnels entre agents ;*
- *d'améliorer les performances par une meilleure coordination entre les unités de travail.*

Agir ensemble signifie développer l'apprentissage collectif et partager les informations et les connaissances afin d'atteindre efficacement les objectifs fixés ;

- *d'encourager l'anticipation et l'innovation en favorisant un processus de décentralisation ;*
- *de gérer la complexité et de gagner en réactivité;*
- *de répondre aux aspirations des agents. Les réseaux augmentent le niveau de compréhension et de motivation du personnel et facilitent la conduite du changement.*

Programme

Concept / finalité du réseau

- Une démarche individuelle inscrite dans la recherche d'un résultat collectif
- Une forme d'organisation
- Réseau divisionnel et réseau relationnel
- "vrai" et "faux réseau" : critères de différenciation

Fonctionnement du réseau

- Conditions de succès :
 - valeurs et culture partagées
 - le principe "d'égalité"
 - un minimum de règles communes
 - une répartition des rôles
 - la communication au cœur du dispositif
- Le réseau dans son environnement
- L'interpénétration des réseaux

Types et structures de réseaux

- Réseau formel / réseau informel
- Réseau "de confort" / réseau "de conquête"
- Réseau permanent / réseau temporaire
- Réseau et hiérarchie

L'appartenance à un réseau

- Identifier son ou ses réseaux d'appartenance
- Intégrer un réseau
- Bien "utiliser" un réseau :
 - qualités exigées
 - contraintes
 - attentes
- Créer un réseau

Les limites du réseau

- Comment évaluer les apports de chaque membre
- Comment durer
- Les risques de déviation

Les apports conceptuels sont nourris en permanence par des exemples et illustrations de l'intervenant, tirés de notre environnement social et professionnel, et d'une analyse "à chaud" des expériences des participants.

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
		14	24		20				9		6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MO4

DUREE

1 journée

OBJECTIFS

Acquérir la terminologie de l'analyse par processus ;

Disposer des concepts, méthodes et outils pour identifier, décrire et améliorer ses processus ;

Comprendre l'intérêt de la clarification des processus pour la mise en place d'un système informatique

PUBLIC

Responsables opérationnels (RH, marketing, financier...) ayant un projet de développement informatique en cours ou en prévision.

TARIF D'INSCRIPTION

800 € H.T. par personne

Identifier et formaliser vos processus afin de préparer la mise en place d'un système d'information

Le système d'information s'organise désormais de plus en plus autour des processus de l'entreprise. Il s'agit là d'une innovation considérable. Ainsi l'identification et la modélisation des processus métiers devient l'étape première et cruciale de la mise en forme d'un système d'information dit "agile" (ou encore « orienté service », SOA). Il en résulte une articulation précise du système d'information aux modes de travail de chaque métier, mais également de l'interaction entre tous les acteurs (processus transverses).

Pour que cette articulation soit réussie, il faut que les métiers s'impliquent activement dans la mise en forme et la maîtrise de leurs processus et qu'ils adhèrent à la démarche de formalisation des processus. Si la maîtrise de ses processus facilite la mise en place d'un système d'information capable de mieux faire inter-agir ses utilisateurs (processus métiers transverses), elle permet également d'améliorer ses performances, de créer plus de valeur pour le client, de mieux travailler ensemble... Ce cursus présente les principes essentiels, les étapes et principaux outils permettant d'identifier et de modéliser ses processus.

Programme

Les processus : une notion clé

- Approche systémique de l'organisation
- Qu'est ce qu'un processus ?
- Intérêt de l'approche processus
- Quelles notions derrière quels mots ?

L'identification des processus

- Les processus clés : définition et méthode d'identification
- Élaboration d'une cartographie des processus

La méthode de formalisation des processus

- Principes de la méthode
- Identifier et décrire un processus – les informations qui permettent de délimiter le périmètre d'action du processus, les caractéristiques intrinsèques du processus
- Définir les ressources (données, compétences)
- Modéliser le déroulement d'un processus : le logigramme
- Cas pratique de description d'un processus

La méthode d'amélioration des processus

- Principes de la méthode
- Mise en application
- Validation des résultats obtenus
- Mise en œuvre des améliorations
- Mesure de l'efficacité des améliorations

Conclusion : du processus à l'infrastructure des TIC

- Nature des liens entre les processus et la mise en place d'un système informatique
- La modélisation informatique des processus métiers : savoir décrypter les diagramme UML.

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	14		25		25			25		21	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MO5

DUREE

1 journée

OBJECTIFS

Acquérir la terminologie de l'analyse par processus ;

Disposer des concepts, méthodes et outils pour identifier, décrire et améliorer ses processus ;

Comprendre la place de l'analyse des processus dans l'élaboration d'un système qualité.

PUBLIC

Responsables opérationnels (RH, marketing, financier, informatique...)

TARIF D'INSCRIPTION

800 € H.T. par personne

Identifier et formaliser vos processus en vue d'améliorer la qualité de vos prestations

Assurer la qualité de ses produits et de ses prestations est devenu une des préoccupations majeures des entreprises. Pour cela, des systèmes qualité aboutissant souvent à une certification sont mis en place. La première - et plus importante - étape de ces projets souvent longs et complexes est d'identifier, de modéliser et d'améliorer ses processus métiers.

Ce cursus présente les principes essentiels, les étapes et principaux outils permettant de réaliser cette formalisation de ses processus et ainsi d'enclencher une démarche qualité au sein de son organisation.

Programme

Les processus : une notion clé

- Approche systémique de l'organisation
- Qu'est-ce qu'un processus ?
- Intérêt de l'approche processus
- Quelles notions derrière quels mots ?

L'identification des processus

- Les processus clés : définition et méthode d'identification
- Élaboration d'une cartographie des processus

La méthode de formalisation des processus

- Principes de la méthode
- Identifier et décrire un processus – les informations qui permettent de délimiter le périmètre d'action du processus, les caractéristiques intrinsèques du processus
- Définir les ressources (données, compétences)
- Modéliser le déroulement d'un processus : le logigramme
- Cas pratique de description d'un processus

La méthode d'amélioration des processus

- Principes de la méthode
- Mise en application
- Validation des résultats obtenus
- Mise en œuvre des améliorations
- Mesure de l'efficacité des améliorations

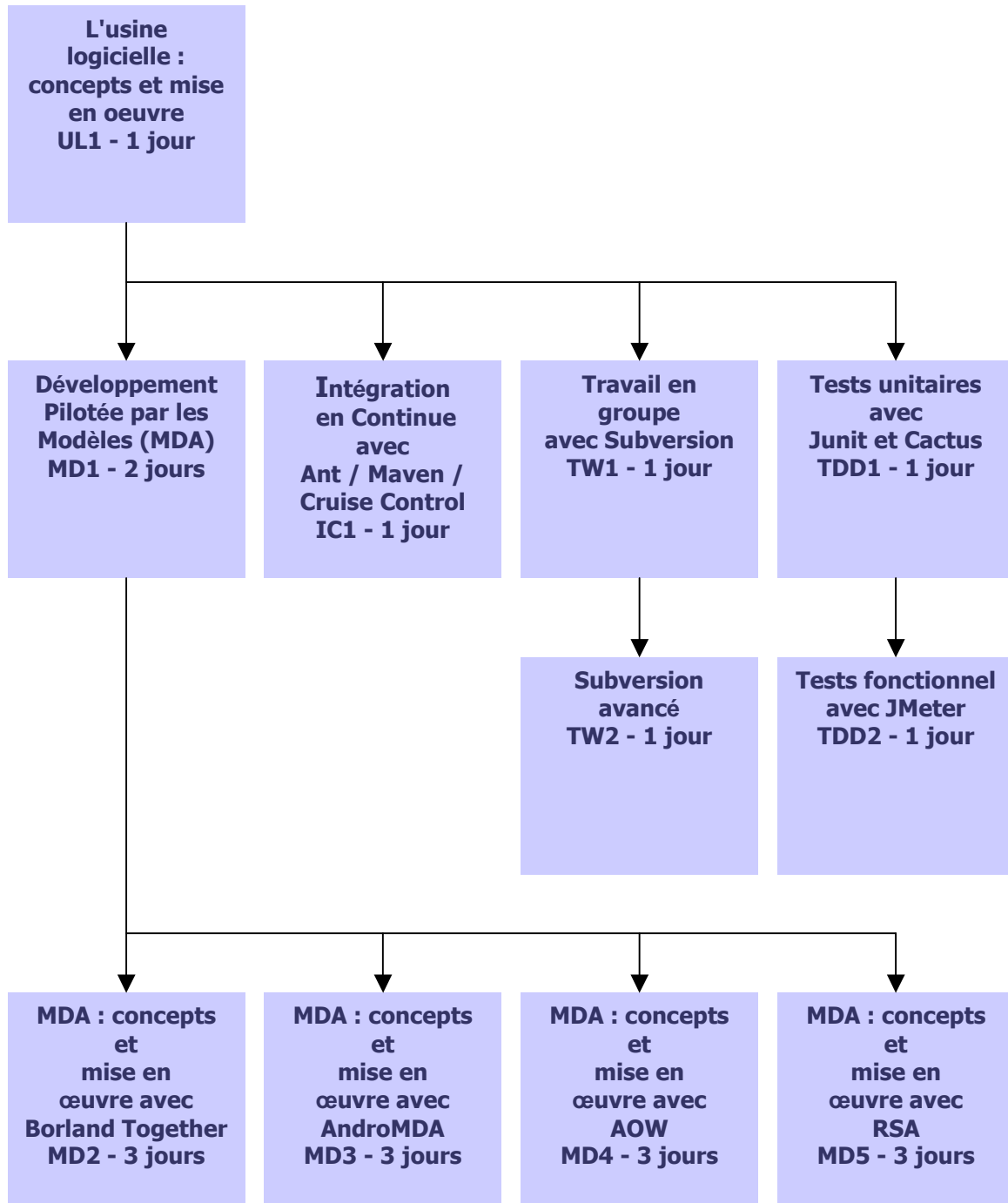
Conclusion : du processus au système qualité

- Typologie proposée par la norme ISO 9001 V2000
- Les tableaux de bord de la qualité
- Réussir et faire vivre une démarche qualité

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	15		26		26			26		22	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter



CODE

UL1

DUREE

1 journée

OBJECTIFS

Comprendre les enjeux d'une fabrique logicielle ;

Comprendre les processus et mécanismes d'automatisation du développement logiciel ;

Connaître les outils du marché.

PUBLIC

Analystes, urbanistes, chefs de projets, concepteurs, architectes

PRE-REQUIS

Connaissance de XML et UML.

TARIF D'INSCRIPTION

400 € H.T par personne

L'usine logicielle : concepts et mise en œuvre

Le concept d « usine logicielle » est une métaphore qui vise à passer d'une production artisanale de logiciels (une application de gestion par exemple) à une production « industrialisée » de ce même logiciel. Cette approche permet à terme une montée en productivité et en qualité, une meilleure maîtrise (coût, réutilisation, sous-traitants) des éléments produits ou constituants de la production.

Ce module présente les enjeux de l'usine logicielle, les différents secteurs d'industrialisation du logiciel (modèles productifs, pilotage par les tests, référentiels communs, intégration continue, etc.) et un panorama des technologies et outils pouvant répondre à ces besoins.

Programme

Méthodes de développement logiciel

- Processus Unifié, RUP (Rational Unified Process), 2TUP (2 Tracks Unified Process), XP (eXtreme Programming), EUP (Extended Unified Process)
- CMM (Capability Maturity Model), ITIL
- L'accroissement de l'évolutivité, réutilisabilité, maintenabilité, efficacité et rentabilité des logiciels

Les concepts fondamentaux

- Limitation des pratiques courantes (redondance de code, système figé, test insuffisant, ...)
- Limite des IDE (Environnement de Développement Intégré)
- Analogie entre les processus industriels couvrant tout le cycle de la conception à l'exploitation
- Concept de référentiel unifié et de chaîne d'ateliers

Les outils

- Les offres IBM, Borland, Microsoft
- Les normes de l'OMG : UML 2, OCL, MOF, XMI, MDA, QVT
- La solution Microsoft : DSL (Domain Specific Language)

Les règles à respecter

- Les étapes du processus
- Les erreurs à éviter
- Retours d'expérience

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8	5	5	2	2	1	2	6	3	1	5	3

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MD1

DUREE

2 jours

OBJECTIFS

Acquérir la terminologie de la technologie MDA (Model Driven Architecture) ;

Disposer des arguments sur les apports de MDA et les avantages en terme d'organisation, de génération de code et de documentation ainsi que des modalités de mise en œuvre.

PUBLIC

Décideurs, architectes, chefs de projets, analystes, concepteurs, développeurs

PRE-REQUIS

Pratique d'un langage de programmation.

Connaissance de XML et UML.

TARIF D'INSCRIPTION

800 € H.T. par personne

Développement Piloté par les Modèles (MDA)

Le concept d' « usine logicielle » est une métaphore qui vise à passer d'une production artisanale de logiciel (une application de gestion par exemple) à une production « industrialisée » de ce même logiciel. Cette approche permet à terme une montée en productivité et en qualité, une meilleure maîtrise (coût, réutilisation, sous-traitants) des éléments produits ou constituants de la production.

Ce module présente l'un des axes majeurs de l'usine logicielle : le développement piloté par les modèles, c'est-à-dire la production de tout ou partie des éléments d'un logiciel (code, configuration, documentation, etc.) à partir de modèles dits « productifs ».

Programme

Les apports de MDA

- Positionnement et intérêts de MDA (Model Driven Architecture)
- Le modèle métier et d'exigences (CIM Computation Independent Model)
- Le modèle d'analyse et de conception abstraite (PIM Platform Independent Model)
- Le modèle de code ou de conception concrète (PSM Platform Specific Model)
- Processus de développement par transformations de modèles
- Indépendance des choix des architectures techniques
- Amélioration de l'évolutivité et de l'organisation du projet

Les techniques de base de MDA

- La métamodélisation MOF (Meta Object Facility)
- Langage de modélisation UML 2.0 (Unified Modeling Language)
- Langage de règles OCL (Object Constraint Language)
- Format XML de modélisation XMI (XML Metadata Interchange) et DI (Diagram Interchange)
- Exemple de framework opensource : EMF (Eclipse Modeling Framework)

Les transformations de modèles

- Réalisation des règles de transformation avec EMF
- Modèle de transformation de modèles : le méta-modèle MOF2.0 QVT (Query View Transformation)

Le langage normalisé OCL (Object Constraint Language)

- Introduction
- Relation avec le méta-modèle UML
- Les bases
- Objets et propriétés
- Opérations sur les Collections
- Messages
- Résolutions de propriétés

Le langage normalisé QVT (Query View Transformation)

- L'implémentation de la partie impérative
- Les transformations
- Les types de méthodes : mappings et queries
- Eléments syntaxiques du langage (object expression, resolve operations, inout parameter, transformation import, virtual methods, query libraries, traceability, OCL support)
- Le « Hello word » pour QVT : transformation d'un modèle métier en un modèle de conception avec encapsulation privée des attributs et génération des méthodes getter et setter.

Mise en oeuvre de MDA

- Panorama des principaux outils MDA

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
9 et 10	6 et 7	6 et 7	3 et 4	3 et 4	4 et 5	3 et 4	7 et 8	4 et 5	2 et 3	6 et 7	4 et 5

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MD2

DUREE

3 jours

OBJECTIFS

Acquérir la terminologie de la technologie MDA (Model Driven Architecture)

Comprendre les apports de MDA en terme d'organisation, de génération de code et de documentation

Maîtriser sa mise en œuvre avec Borland Together

PUBLIC

Architectes, chefs de projets, analystes, concepteurs, développeurs

PRE-REQUIS

Maîtrise du langage Java.

Connaissance de XML et UML.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

MDA concepts et mise en œuvre avec Borland Together

Le concept d'« usine logicielle » est une métaphore qui vise à passer d'une production artisanale de logiciel (une application de gestion par exemple) à une production « industrialisée » de ce même logiciel. Cette approche permet à terme une montée en productivité et en qualité, une meilleure maîtrise (coût, réutilisation, sous-traitants) des éléments produits ou constituants de la production.

Ce module présente l'un des axes majeurs de l'usine logicielle : le développement piloté par les modèles, c'est-à-dire la production de tout ou partie des éléments d'un logiciel (code, configuration, documentation, etc.) à partir de modèles dits « productifs ».

La pratique est abordée à travers l'outil Borland Together.

Programme

Les apports de MDA

- Positionnement et intérêts de MDA (Model Driven Architecture)
- Le modèle métier et d'exigences (CIM Computation Independant Model)
- Le modèle d'analyse et de conception abstraite (PIM Platform Independant Model)
- Le modèle de code ou de conception concrète (PSM Platform Specific Model)
- Processus de développement par transformations de modèles
- Indépendance des choix des architectures techniques
- Amélioration de l'évolutivité et de l'organisation du projet

Langage de modélisation UML 2 (Unified Modeling Language)

- Les diagrammes de structure : classe, objet, structure composite, composant, déploiement
- Les diagrammes comportementaux : séquence, communication, activité, état, vue d'ensemble des interactions, timing, use case

Le langage OCL (Object Constraint Language)

- Introduction
- Relation avec le métamodèle UML
- Les bases
- Objets et propriétés

- Opérations sur les Collections
- Messages
- Résolutions de propriétés

Les techniques de base de MDA

- La métamodélisation MOF (Meta Object Facility)
- Les profils UML
- Les stéréotypes
- Les tagged-value
- Les contraintes OCL
- Format XML de modélisation XMI (XML Metadata Interchange) et DI (Diagram Interchange)
- Exemple de framework opensource : EMF (Eclipse Modeling Framework)

Les transformations de modèles

- Réalisation des règles de transformation avec EMF
- Modèle de transformation de modèles : le métamodèle MOF2.0 QVT (Query View Transformation)

Mise en œuvre de MDA avec Together Architect

- Création de projet de transformation MDA
- Les vues MDA
- Création de transformation (de modèle à modèle) avec le langage QVT
- Appliquer une transformation de modèle
- Déployer une transformation

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 10	5 au 7	5 au 7	2 au 4	2 au 4	4 au 6	2 au 4	1 au 3	3 au 5	1 au 3	5 au 7	4 au 6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MD3

DUREE

3 jours

OBJECTIFS

Acquérir la terminologie de la technologie MDA (Model Driven Architecture)

Comprendre les apports de MDA en terme d'organisation, de génération de code et de documentation

Maîtriser sa mise en œuvre avec AndroMDA

PUBLIC

Architectes, chefs de projets, analystes, concepteurs, développeurs

PRE-REQUIS

Maîtrise du langage Java.

Connaissance de XML et UML.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

MDA : concepts et mise en œuvre avec AndroMDA

Le concept d' « usine logicielle » est une métaphore qui vise à passer d'une production artisanale de logiciel (une application de gestion par exemple) à une production « industrialisée » de ce même logiciel. Cette approche permet à terme une montée en productivité et en qualité, une meilleure maîtrise (coût, réutilisation, sous-traitants) des éléments produits ou constituants de la production.

Ce module présente l'un des axes majeurs de l'usine logicielle : le développement piloté par les modèles, c'est-à-dire la production de tout ou partie des éléments d'un logiciel (code, configuration, documentation, etc.) à partir de modèles dits « productifs ».

La pratique est abordée à travers l'outil open-source AndroMDA.

Programme

Les apports de MDA

- Positionnement et intérêts de MDA (Model Driven Architecture)
- Le modèle métier et d'exigences (CIM Computation Independent Model)
- Le modèle d'analyse et de conception abstraite (PIM Platform Independent Model)
- Le modèle de code ou de conception concrète (PSM Platform Specific Model)
- Processus de développement par transformations de modèles
- Indépendance des choix des architectures techniques
- Amélioration de l'évolutivité et de l'organisation du projet

Langage de modélisation UML 2 (Unified Modeling Language)

- Les diagrammes de structure : classe, objet, structure composite, composant, déploiement
- Les diagrammes comportementaux : séquence, communication, activité, état, vue d'ensemble des interactions, timing, use case

Le langage OCL (Object Constraint Language)

- Introduction
- Relation avec le métamodèle UML
- Les bases
- Objets et propriétés
- Opérations sur les Collections
- Messages
- Résolutions de propriétés

Les techniques de base de MDA

- La métamodélisation MOF (Meta Object Facility)
- Les profils UML
- Les stéréotypes
- Les tagged-value
- Les contraintes OCL
- Format XML de modélisation XMI (XML Metadata Interchange) et DI (Diagram Interchange)
- Exemple de framework opensource : EMF (Eclipse Modeling Framework)

Les transformations de modèles

- Réalisation des règles de transformation avec EMF
- Modèle de transformation de modèles : le métamodèle MOF2.0 QVT (Query View Transformation)

Mise en œuvre de MDA avec AndroMDA

- Architecture d'une application
- Architecture d'une application générée par AndroMDA
- Mapping objet – SQL avec Hibernate
- Utilisation de Maven comme outil de gestion de projet open source
- Utilisation d'un AGL UML open source : ArgoUML ou MagicDraw
- Concept de cartridge
- Panorama de cartridges
- Création d'une application jee complète avec les cartridges existants
- Ecrire un cartridge en 10 étapes

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 10	5 au 7	5 au 7	2 au 4	2 au 4	4 au 6	2 au 4	1 au 3	3 au 5	1 au 3	5 au 7	4 au 6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MD4

DUREE

3 jours

OBJECTIFS

Acquérir la terminologie de la technologie MDA (Model Driven Architecture)

Comprendre les apports de MDA en terme d'organisation, de génération de code et de documentation

Maîtriser sa mise en oeuvre avec oAW (openArchitectureWare)

PUBLIC

Architectes,

Chefs de projets,

Analystes,

Concepteurs/développeurs

PRE-REQUIS

Maîtrise du langage Java, connaissance de XML et UML

TARIF D'INSCRIPTION

1 200 € H.T. par personne

MDA : concepts et mise en œuvre avec Eclipse oAW

Le concept d' « usine logicielle » est une métaphore qui vise à passer d'une production artisanale de logiciel (une application de gestion par exemple) à une production « industrialisée » de ce même logiciel. Cette approche permet à terme une montée en productivité et en qualité, une meilleure maîtrise (coût, réutilisation, sous-traitants) des éléments produits ou constituants de la production.

Ce cursus présente l'un des axes majeurs de l'usine logicielle : le développement piloté par les modèles, c'est-à-dire la production de tout ou partie des éléments d'un logiciel (code, configuration, documentation, etc.) à partir de modèles dits « productifs ». La pratique est abordée à travers l'outil open-source oAW (openArchitectureWare) de la fondation Eclipse.

Programme

Les apports de MDA

- Positionnement et intérêts de MDA (Model Driven Architecture)
- Le modèle métier et d'exigences (CIM Computation Independant Model)
- Le modèle d'analyse et de conception abstraite (PIM Platform Independant Model)
- Le modèle de code ou de conception concrète (PSM Platform Specific Model)
- Processus de développement par transformations de modèles
- Indépendance des choix des architectures techniques
- Amélioration de l'évolutivité et de l'organisation du projet

Langage de modélisation UML 2 (Unified Modeling Language)

- Les diagrammes de structure : classe, objet, structure composite, composant, déploiement
- Les diagrammes comportementaux : séquence, communication, activité, état, vue d'ensemble des interactions, timing, use case

Le langage OCL (Object Constraint Language)

- Relation avec le méta-modèle UML
- Les bases
- Objets et propriétés
- Opérations sur les Collections
- Messages
- Résolutions de propriétés

Les techniques de base de MDA

- La méta-modélisation MOF (Meta Object Facility)
- Les profils UML
- Les stéréotypes
- Les tagged-value
- Les contraintes OCL
- Format XML de modélisation XMI (XML Metadata Interchange) et DI (Diagram Interchange)
- Exemple de framework opensource : EMF (Eclipse Modeling Framework)

Mise en oeuvre de MDA avec oAW

- Installation des APIs et du plugin Eclipse
- Vue d'ensemble des composants de oAW
- Le moteur de workflow
- Configuration et invocation du workflow
- Le langage d'expression et système de type
- Valider un modèle avec le DSL (Domain Specific Language) « Check »
- Transformer et modifier des modèles avec le langage « Wombat »
- Gérer le code généré en sortie avec le langage « Xpand »
- Définir des bibliothèques de transformation avec le langage « Extend »
- Personnaliser le code généré avec le plugin Recipe

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 10	5 au 7	5 au 7	2 au 4	2 au 4	4 au 6	2 au 4	1 au 3	3 au 5	1 au 3	5 au 7	4 au 6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

MD5

DUREE

3 jours

OBJECTIFS

Acquérir la terminologie de la technologie MDA (Model Driven Architecture)

Comprendre les apports de MDA en terme d'organisation, de génération de code et de documentation

Maîtriser sa mise en œuvre avec Rational Software Architect

PUBLIC

Architectes, chefs de projets, analystes, concepteurs, développeurs

PRE-REQUIS

Maîtrise du langage Java.

Connaissance de XML et UML.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

MDA : concepts et mise en œuvre avec RSA

Le concept d' « usine logicielle » est une métaphore qui vise à passer d'une production artisanale de logiciel (une application de gestion par exemple) à une production « industrialisée » de ce même logiciel. Cette approche permet à terme une montée en productivité et en qualité, une meilleure maîtrise (coût, réutilisation, sous-traitants) des éléments produits ou constituants de la production.

Ce module présente l'un des axes majeurs de l'usine logicielle : le développement piloté par les modèles, c'est-à-dire la production de tout ou partie des éléments d'un logiciel (code, configuration, documentation, etc.) à partir de modèles dits « productifs ».

La pratique est abordée ici à travers l'outil d'IBM Rational Software Architect.

Programme

Les apports de MDA

- Positionnement et intérêts de MDA (Model Driven Architecture)
- Le modèle métier et d'exigences (CIM Computation Independent Model)
- Le modèle d'analyse et de conception abstraite (PIM Platform Independent Model)
- Le modèle de code ou de conception concrète (PSM Platform Specific Model)
- Processus de développement par transformations de modèles
- Indépendance des choix des architectures techniques
- Amélioration de l'évolutivité et de l'organisation du projet

Langage de modélisation UML 2 (Unified Modeling Language)

- Les diagrammes de structure : classe, objet, structure composite, composant, déploiement
- Les diagrammes comportementaux : séquence, communication, activité, état, vue d'ensemble des interactions, timing, use case

Le langage OCL (Object Constraint Language)

- Introduction
- Relation avec le métamodèle UML
- Les bases
- Objets et propriétés
- Opérations sur les Collections
- Messages
- Résolutions de propriétés

Les techniques de base de MDA

- La métamodélisation MOF (Meta Object Facility)
- Les profils UML
- Les stéréotypes
- Les tagged-value
- Les contraintes OCL
- Format XML de modélisation XMI (XML Metadata Interchange) et DI (Diagram Interchange)
- Exemple de framework opensource : EMF (Eclipse Modeling Framework)

Les transformations de modèles

- Réalisation des règles de transformation avec EMF
- Modèle de transformation de modèles : le métamodèle MOF2.0 QVT (Query View Transformation)

Mise en oeuvre de MDA avec Rational Software Architect

- Architecture d'une application
- Les niveaux du développement logiciel
- Création d'une transformation de modèle
- Génération de code

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 10	5 au 7	5 au 7	2 au 4	2 au 4	4 au 6	2 au 4	1 au 3	3 au 5	1 au 3	5 au 7	4 au 6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

TW1

DUREE

1 journée

OBJECTIFS

Être capable d'utiliser l'outil collaboratif Subversion

Développer une application informatique en équipe

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet

PRE-REQUIS

Aucun

TARIF D'INSCRIPTION

400 € H.T. par personne

Travailler en équipe avec Subversion

Subversion est un outil de gestion de versions open source destiné à remplacer CVS.

Si les concepts ne remettent pas en cause fondamentalement l'héritage de CVS, en revanche Subversion comble un certain nombre de lacunes, apporte de nouvelles fonctionnalités et pratiques qu'il convient de comprendre et maîtriser.

Un outil de ce type a une place centrale dans une usine logicielle : il permet de gérer les versions et évolutions des éléments de configuration (code, documentation, ressources, etc.) au sein d'un référentiel unique, partagé par tous les acteurs de la production logicielle.

Programme

Les principes de gestion de version de projets informatiques

Objectifs et positionnement par rapport à CVS

Intégration avec Eclipse

Installation d'un client Subversion: TortoiseSVN

Utilisation de Subversion

- Récupérer des sources informatiques
- Déverser les modifications vers le serveur
- Récupérer les modifications faites par les autres développeurs
- Ajouter, enlever des fichiers
- Déplacer, renommer des fichiers
- Création et application de patches
- Gérer les conflits

Les branches de développement

- Le principe des branches
- Changer de branche de développement
- Transférer les développements d'une branche vers une autre branche

Visualiser les modifications

- Savoir qui a fait quoi et quand
- Voir le synoptique des modifications
- Voir le graphe des modifications

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8		5		2		2		3		5	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

Subversion avancé

CODE

TW2

DUREE

1 journée

OBJECTIFS

Être capable d'administrer, de configurer l'outil collaboratif Subversion

Être capable d'utiliser Subversion de manière avancée

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet,

Architectes technique, administrateurs de GCL

PRE-REQUIS

Connaissances de base de Subversion.

TARIF D'INSCRIPTION

400 € H.T. par personne

Subversion est un outil de gestion de versions open source destiné à remplacer CVS.

Si les concepts ne remettent pas en cause fondamentalement l'héritage de CVS, en revanche Subversion comble un certain nombre de lacunes, apporte de nouvelles fonctionnalités et pratiques qu'il convient de comprendre et maîtriser.

Un outil de ce type a une place centrale dans une usine logicielle : il permet de gérer les versions et évolutions des éléments de configuration (code, documentation, ressources, etc.) au sein d'un référentiel unique, partagé par tous les acteurs de la production logicielle.

Programme

Installation d'un serveur Subversion

- Installation locale
- Installation à partir d'un serveur Web

Création d'un repository SVN accessible via le web

- Restriction au protocole https
- Gestion des accès au système: lecture seule, lecture/écriture
- Gestion des droits au système: administration des utilisateurs

Les URLs employées dans les différents protocoles

Les actions pré- et post-commit: notification par mail

Migrer de CVS vers Subversion

Les branches de développement

- Rappels
- Création, suppression d'une branche

Intégration de projets

- Utilisation des tags
- Travailler sur des fichiers taggués

Utilisation avancée

- La gestion des numéros de version
- Les propriétés
- Relocalisation de projet

Que faire en cas de problèmes ?

- Les mises à jour
- Libérer des verrous
- Nettoyer l'espace local
- Revenir dans un espace propre

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
9		6		3		3		4		6	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

IC1

DUREE

1 journée

OBJECTIFS

Être capable de développer une application Java/JEE en utilisant les outils de gestion de version, d'intégration et de gestion de projet Java.

PUBLIC

Concepteurs/développeurs,
Chefs de projets,
Architectes technique.

PRE-REQUIS

Pratique du langage Java.

TARIF D'INSCRIPTION

400 € H.T. par personne

Intégration Continue avec Ant / Maven / Cruise Control

La complexité croissante des projets applicatifs actuels implique la mise en place d'usines logicielle.

Ces "usines" permettent d'automatiser les tâches répétitives pour les membres d'une équipe projet. Outre le gain en productivité, cette approche apporte fiabilité et indicateurs mis à jour pour tous les acteurs du projet.

Plus concrètement, certains projets impliquent des dizaines d'acteurs et la complexité de l'application produite amène parfois des heures d'assemblage régulier.

Un processus d'intégration continue va permettre de réaliser cet assemblage automatiquement et très régulièrement, permettant ainsi de mettre à disposition –quasiment « en continu »– une version assemblée « de référence », vérifiée, testée et utilisable.

Programme

Objectifs

- L'architecture d'intégration
- Processus et méthodes
- Notion de builds
- Le gestionnaire de version
- L'automatisation des tests et de la production logicielle

Ant

- Objectif
- L'automatisation des activités de développement d'une application
- Installation, utilisation et exécution
- Les tâches standards et étendues
- Les bonnes pratiques

Maven

- Présentation du framework
- Positionnement par rapport à Ant
- Installation et configuration
- La notion de plug-in
- Référentiels locaux et distants
- Le project Object Model
- Les builds automatisés avec Maven

Cruise Control

- Mise en oeuvre
- Architecture et installation
- La configuration à base de plug-ins
- L'accès à un gestionnaire de versions
- Le déclenchement du build
- La génération du rapport de build
- La notification des développeurs

Subversion

- Objectifs et positionnement par rapport à CVS
- Installation d'un serveur Subversion
- Création d'un repository SVN
- Gestion des droits au système
- La notification de commit par mël
- Installation et utilisation d'un client Subversion

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	7		4		3		8		3		6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

TDD1

DUREE

1 journée

OBJECTIFS

Comprendre les principes de développement par les tests

Être capable de développer une application Java/J2EE en utilisant les outils de tests de projet Java

PUBLIC

Concepteurs/développeurs, Chefs de projets, architectes technique.

PRE-REQUIS

Pratique du langage Java.

TARIF D'INSCRIPTION

400 € H.T. par personne

Tests unitaires avec JUnit et Cactus

L'industrialisation des processus de développement logiciel, ce que nous appelons « usine logicielle » passe par une approche systématique, re-jouable et automatisable des tests devant servir à valider le logiciel produit.

Le développement piloté par les test (Test Driven Development) est l'une des pratiques préconisées par la méthode agile XP (eXtreme Programming). Cette pratique est issue d'un principe simple : « puisque nous n'avons jamais le temps de tester une application produite, commençons par écrire les tests auxquels l'application à réaliser devra se conformer ».

Au niveau des fonctionnalités « basiques », le test unitaire vérifie alors qu'un code réalise bien ce qui a été prévu lors de la conception détaillée d'un programme. Il est écrit avant le programme lui même.

Les outils open source Junit, Cactus et Ant permettent d'aider l'équipe de développement à mettre en œuvre ces pratiques.

Programme

Objectifs

- Les concepts du développement piloté par les tests (TDD)
- L'architecture d'intégration
- Processus et méthodes
- Notion de builds
- L'automatisation des tests

JUnit

- Objectif des tests unitaires
- Les concepts : test case, fixture, test suite, test runner
- Les bibliothèques complémentaires
- Exemple de framework opensource : EMF (Eclipse Modeling Framework)

Cactus

- Objectifs et positionnement par rapport à JUnit
- Les Mock Objects
- Les tests unitaires
- Les tests d'intégration
- Les tests fonctionnels

Ant

- Objectif
- L'automatisation des activités de développement d'une application
- Installation, utilisation et exécution
- Les taches standards et étendues
- Les bonnes pratiques

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	5		2		1		6		1		3

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

TDD2

DUREE

1 journée

OBJECTIFS

Mettre en place une stratégie de développement par les tests en s'appuyant sur les fonctionnalités de JMeter.

PUBLIC

Concepteurs/développeurs, Chefs de projets, architectes technique.

PRE-REQUIS

Pratique du langage Java.

TAIRF D'INSCRIPTION

400 € H.T. par personne

Tests fonctionnels avec JMeter

L'industrialisation des processus de développement logiciel, ce que nous appelons « usine logicielle » passe par une approche systématique, re-jouable et automatisable des tests devant servir à valider le logiciel produit.

Au niveau des fonctionnalités « basiques », le test unitaire vérifie qu'un code réalise bien ce qui a été prévu lors de la conception détaillée d'un programme. A un niveau fonctionnel, le scénario de test vérifie qu'une fonctionnalité se comporte telle qu'elle a été souhaitée lors de l'expression des besoins par les utilisateurs (cas d'utilisation, exigences).

Piloter ses développements par les tests (Test Driven Development) signifie d'une part de définir et formaliser les scénarios de tests et d'autre part d'être capable de re-jouer systématiquement ces scénarios sur une application. Ceci permet de maîtriser la non régression lors des évolutions successives du code mais également de mesurer la performance d'une application à travers ses évolutions.

L'outil open source JMeter permet d'aider l'équipe de développement à mettre en œuvre ces pratiques.

Programme

Objectifs

- Les concepts de base du développement piloté par les tests
- Notions de scénarios fonctionnels
- Procédure de recette, jeux de données, scénarios de tests fonctionnels

Etude de cas

- Description de l'application à tester

Installation de JMeter

- Compatibilité et version de Java
- Construction d'un plan de tests

Développer un plan de tests

- « Samplers »
- « Listeners »
- « Timers »
- « Logic Controllers »
- « Configuration elements »
- « Assertions »

Enregistrer un Use case**Audit de performance**

- Mesurer les temps de réponse d'un service web

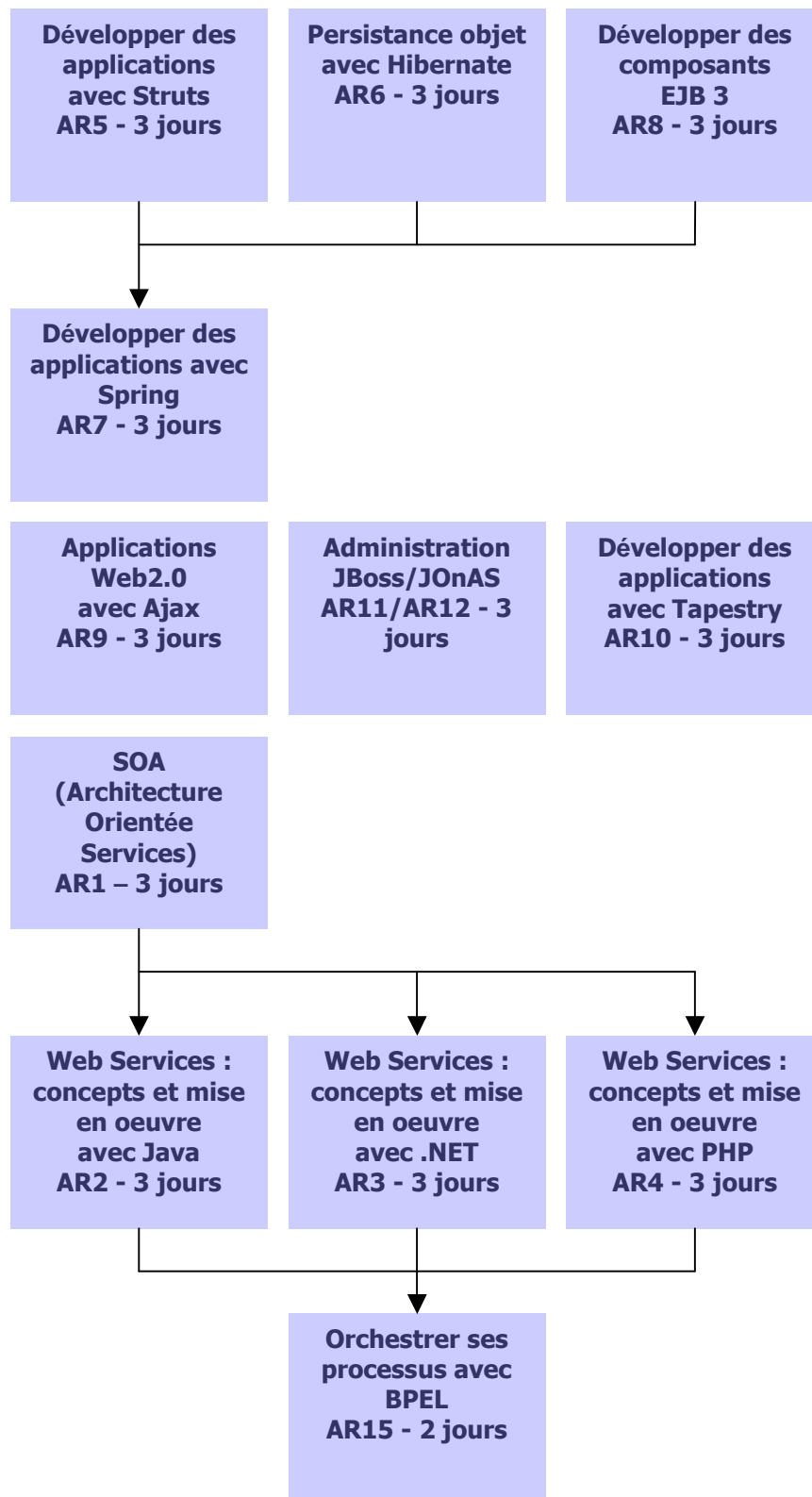
Pratique

- Mise en pratique avec une architecture plus complexe utilisant Struts

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	6		3		2		7		2		4

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter



CODE

AR1

DURÉE

3 jours

OBJECTIFS

Comprendre les avantages de SOA par rapport aux autres architectures

Connaître la démarche SOAJUnit Savoir appliquer la démarche SOA

PUBLIC

Chefs de projet, architectes techniques et fonctionnels, analystes et décideurs informatiques.

PRE-REQUIS

Tout public.

TARIF D'INSCRIPTION

1 200 € H.T. par personne.

SOA (Architecture Orientée Services)

Agilité, réactivité : telles sont les exigences nouvelles qu'exprime l'entreprise pour son S.I. L'entreprise « Agile » souhaitera aligner son S.I. avec sa stratégie, ses métiers.

Dès lors, il est nécessaire de savoir organiser un S.I. supportant les processus métiers par nature transverses.

En amont de cette démarche, la cartographie applicative, l'urbanisation du S.I. et l'approche processus sont nécessaires.

Techniquement, SOA est une réponse qui apporte méthodologie et concepts éprouvés pour répondre à ces enjeux.

Vous apprendrez les bonnes pratiques SOA et les technologies qui permettent leur mise en oeuvre.

Programme

Historique

- Les architectures traditionnelles
- Introduction à SOA
- Historique de SOA
- Les évolutions des architectures

Enjeux de SOA

- Modèle conceptuel et architecture
- Les avantages
- Exemples d'applications
- Définition des services
- Les contrats, interfaces
- L'identification, la qualification et la classification des services
- Bus de Service (ESB)
- Cycle de vie d'un service : composition, publication et référencement
- Les architectures événementielles (EDA Event Driven Architecture)
- Urbanisation des SI

Les bonnes pratiques de SOA

- Faible couplage
- Communication asynchrone
- Granularité et intégralité

Les normes

- BPMN
- BPEL
- Les Services Web

Méthodologie

- Urbanisation et cartographie, prendre en compte l'existant
- Approche processus, l'approche par les métiers de l'entreprise
- Démarche itérative, du choix d'une technologie au déploiement

Les outils

- Les principaux acteurs : IBM, BEA, Microsoft

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 10		5 au 7		2 au 4		2 au 4		3 au 5		5 au 7	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR2

DUREE

3 jours

OBJECTIFS

Comprendre la portée et les buts des Services Web.
Assimiler les techniques permettant la mise en oeuvre de Services Web dans une architecture Internet/Intranet.

Concevoir l'architecture et l'urbanisation des systèmes avec les Services Web.

Analyser l'architectures J2EE.

Réaliser et publier des services web et montrer l'interopérabilité

PUBLIC

Développeurs, chefs de projet, architectes techniques.

PRE-REQUIS

Avoir participé ou posséder les connaissances équivalentes aux formations Java - Les bonnes pratiques de développement.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Web Services - concepts et mise en oeuvre avec Java

Avec l'approche SOA (Services Oriented Architecture), l'interopérabilité est en passe de devenir aujourd'hui une réalité. Les Web Services en proposent une implémentation fiable et relativement aisée à mettre en oeuvre.

Ce cours apprendra à développer, déployer, consommer et publier des services web avec la plate-forme Java EE.

Programme

Les champs d'application

- Principe et objectifs des Services Web
- Positionnement par rapport à l'EAI
- Intégration de l'existant : connectivité des Services Web avec les autres technologies
- Facteurs d'augmentation de l'agilité de l'entreprise
- Architecture n tiers et applications distribuées : accès aux applications depuis le Web, les clients riches et clients légers
- Les techniques et les protocoles : XML, SOAP, WSDL, UDDI

Conception d'un Service Web

- Architecture des Services Web
- Urbanisation d'applications à base de Web Services : accès aux données, communication entre Services Web
- Techniques de gestion de la sécurité côté Services Web

Le protocole SOAP

- Les principe généraux et l'architecture
- Les messages SOAP, les messages d'erreur et encodage

Description des Services Web : le langage WSDL

- Les types de données
- Définition des messages et protocoles d'échange associés
- Interface et localisation du service

Publication et référencement de services web : UDDI

- Annuaire UDDI pour référencer des Services Web (Intranet et Internet)
- Mise en oeuvre des annuaires UDDI par les applications : découverte dynamique des services, gestion du catalogue des services

Mise en oeuvre des APIs JEE et Apache

- APIs JEE 5 : JAXP, JAX-RPC, JAXM et JAXR
- Les bibliothèques Apache : AXIS, XML-RPC

Développement, déploiement et consommation des Services Web

- Recherche et invocation de services côté client
- Gestion des transactions
- Utilisation avec Eclipse, Tomcat, JBoss

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	5 au 7		2 au 4		4 au 6		1 au 3		1 au 3	28 au 30	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR3

DUREE

3 jours

OBJECTIFS

Comprendre la portée et les buts des Services Web.
Assimiler les techniques permettant la mise en oeuvre de Services Web dans une architecture Internet/Intranet.

Concevoir l'architecture et l'urbanisation des systèmes avec les Services Web.

Analyser l'architectures .NET.

Réaliser et publier des service web et montrer l'interopérabilité (ex. JEE et .NET).

PUBLIC

Développeurs, chefs de projet, architectes techniques.

PRE-REQUIS

Connaissances équivalentes au NE1 - Programmation en C#.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Web Services - concepts et mise en oeuvre avec .NET

Avec l'approche SOA (Services Oriented Architecture), l'interopérabilité est en passe de devenir aujourd'hui une réalité. Les Web Services en proposent une implémentation fiable et relativement aisée à mettre en oeuvre.

Ce cours apprendra à développer, déployer, consommer et publier des services web avec

La plate-forme .NET et le langage C#.

Programme

Les champs d'application

- Principe et objectifs des Services Web
- Positionnement par rapport à l'EAI
- Intégration de l'existant : connectivité des Services Web avec les autres technologies
- Facteurs d'augmentation de l'agilité de l'entreprise
- Architecture n tiers et applications distribuées : accès aux applications depuis le Web, les clients riches et clients légers
- Les techniques et les protocoles : XML, SOAP, WSDL, UDDI

Conception d'un Service Web

- Architecture des Services Web
- Urbanisation d'applications à base de Web Services : accès aux données, communication entre Services Web
- Techniques de gestion de la sécurité côté Services Web

Le protocole SOAP

- Les principe généraux et l'architecture
- Les messages SOAP, les messages d'erreur et encodage

Description des Services Web : le langage WSDL

- Les types de données
- Définition des messages et protocoles d'échange associés
- Interface et localisation du service

Publication et référencement de services web : UDDI

- Annuaire UDDI pour référencer des Services Web (Intranet et Internet)
- Mise en oeuvre des annuaires UDDI par les applications : découverte dynamique des services, gestion du catalogue des services

Mise en oeuvre des APIs .NET

- Authentification intégrée, personnalisée (en-têtes SOAP)
- Autorisation : Sécurité basée sur des rôles et sécurité d'accès au code
- Cryptage
- Contraintes, évolutivité, performance, fiabilité, gestion des versions
- Fournisseur de services Internet et déploiement ASP
- Développement, déploiement et consommation des Services Web
- Recherche et invocation de services côté client
- Gestion des transactions

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 17		12 au 14		21 au 23	11 au 13		20 au 22		8 au 10		3 au 5

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR4

DUREE

3 jours

OBJECTIFS

Comprendre la portée et les buts des Services Web. Assimiler les techniques permettant la mise en oeuvre de Services Web dans une architecture Internet/Intranet.

Concevoir l'architecture et l'urbanisation des systèmes avec les Services Web.

Analyser l'architectures PHP.

Réaliser et publier des service web et montrer l'interopérabilité.

PUBLIC

Développeurs, chefs de projet, architectes techniques.

PRE-REQUIS

Connaissances équivalentes au module NE1 - Programmation en C#.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Web Services - concepts et mise en oeuvre avec PHP

Avec l'approche SOA (Services Oriented Architecture), l'interopérabilité est en passe de devenir aujourd'hui une réalité. Les Web Services en proposent une implémentation fiable et relativement aisée à mettre en oeuvre.

Ce cours apprendra à développer, déployer, consommer et publier des services web avec la plate-forme et les outils PHP.

Programme

Les champs d'application

- Principe et objectifs des Services Web
- Positionnement par rapport à l'EAI
- Intégration de l'existant : connectivité des Services Web avec les autres technologies
- Facteurs d'augmentation de l'agilité de l'entreprise
- Architecture n tiers et applications distribuées : accès aux applications depuis le Web, les clients riches et clients légers
- Les techniques et les protocoles : XML, SOAP, WSDL, UDDI

Conception d'un Service Web

- Architecture des Services Web
- Urbanisation d'applications à base de Web Services : accès aux données, communication entre Services Web
- Techniques de gestion de la sécurité côté Services Web

Le protocole SOAP

- Les principe généraux et l'architecture
- Les messages SOAP, les messages d'erreur et encodage

Description des Services Web : le langage WSDL

- Les types de données
- Définition des messages et protocoles d'échange associés
- Interface et localisation du service

Publication et référencement de services web : UDDI

- Annuaire UDDI pour référencer des Services Web (Intranet et Internet)
- Mise en oeuvre des annuaires UDDI par les applications : découverte dynamique des services, gestion du catalogue des services

Mise en oeuvre des APIs PHP

- Développement, déploiement et consommation des Services Web
- Recherche et invocation de services côté client
- Gestion des transactions

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	5 au 7		2 au 4		4 au 6		1 au 3		1 au 3	28 au 30	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR5

DUREE

3 jours

OBJECTIFS

Connaître les principes du modèle MVC2

Etre capable de développer une application MVC2 avec le framework Struts

Maîtriser les bonnes pratiques de développement Struts

Mettre en place l'internationalisation et la sécurité

Connaître les bibliothèques et les outils graphiques associés à Struts.

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet, architectes technique

PRE-REQUIS

Connaissances équivalentes au JA1 - Java - Les bonnes pratiques de développement avec Eclipse.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Développer des applications avec Struts

La maintenabilité d'une application web est fortement améliorée avec l'intégration d'une architecture qui implémente le modèle MVC2. Il s'agit de distinguer trois couches logiques : la présentation, le contrôleur et le modèle.

Le framework open source Struts est aujourd'hui l'une des solutions éprouvées qui permet de mettre en place une telle architecture.

Programme

Introduction

- Historique
- Présentation de Struts
- Intérêts de Struts
- Architecture n tiers
- Architecture MVC2

L'architecture de Struts

- Une première application
- Application du MVC2
- Le composant Vue : le formulaire HTML et le bean formulaire
- Les fichiers MessageResources et Application.properties
- Le bean formulaire HelloForm
- Validation des données et ActionErrors
- Automatisation des contrôles de saisies : DynaForm et Validator
- Formulaire dynamique
- Le composant Contrôleur HelloAction
- Le composant Modèle HelloModel
- Transmission de données à la Vue via une classe Constantes
- Assemblage de l'application : le fichier struts-config.xml

Les balises Struts

- Rappels sur le fonctionnement des tag lib personnalisés et la JSTL
- Vue d'ensemble des balises Struts
- Comparaison avec la JSTL
- Les balises Struts pour HTML
- Les éléments de base
- Traitement des formulaires
- Validation des saisies
- Téléchargement d'un fichier
- La bibliothèque Bean : stockage et transmission des données
- Les balises Logic
- Les balises d'imbrication et de modèles
- La bibliothèque Tiles

Configuration

- Analyse de la structure du fichier struts-config.xml
- Modèle de struts-config.xml (application struts-blank.war)
- Les DataSource
- Les beans formulaires
- Gestion des exceptions
- Transferts globaux
- Associations aux actions
- Le Contrôleur
- Ressource de messages

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
29 au 31		26 au 28		9 au 11		23 au 25		24 au 26		26 au 28	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR6

DUREE

3 jours

OBJECTIFS

Etre capable de développer une application Java EE (web, EJB) en réalisant la persistance des objets avec le framework Hibernate

Connaître et assimiler un framework ORM (Object Relational Mapping) dans une architecture n-tiers

Maîtriser les concepts d'Hibernate

Savoir utiliser les bonnes pratiques de développement Hibernate

Savoir utiliser les annotations, les outils Hibernate.

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissances équivalentes au JA1 - Java - Les bonnes pratiques de développement avec Eclipse.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Persistance objet avec Hibernate

L'intérêt d'implémenter une couche de mapping objet-relationnel entre un modèle métier « objet » et une base de données relationnelle n'est probablement plus à démontrer : le code qui contient les requêtes SQL est isolé dans un endroit unique, permettant une adhérence réduite au schéma de la base ; le modèle métier peut respecter les principes de la programmation objet, utiliser l'héritage et la navigation.

Hibernate est un framework qui permet d'automatiser la persistance des objets dans une base de données relationnelle. Il est fiable et performant pour peu que l'on maîtrise ses différentes stratégies de chargement et gestion de caches. Il s'agit également d'un outil open source plébiscité et très actif. La nouvelle spécification EJB3 de Sun reprend en grande partie, pour ce qui est de la persistance, les « best practices » développés par Hibernate.

Programme

Objectifs

- Définition de la persistance objet
- Classes, fichier de mapping, configuration, cycle de vie des objets
- Mapper des associations unidirectionnelle/bidirectionnelle et les collections

L'architecture d'Hibernate

- Vue en couche
- Cycle de vie des instances
- Intégration JNDI, JDBC, JTA
- Gestion des sessions

Configuration

- APIs permettant la configuration par programmation
- Obtenir une session
- Gestion des connexions JDBC
- Dialectes SQL, cache de second niveau et cache de requêtes
- Fichier de configuration XML
- Intégration à un serveur d'application J2EE

Réalisation de classes persistances

- Les Plain Ordinary Java Object (POJO)
- Mise en oeuvre de l'héritage
- Redéfinition des méthodes equals et hashCode

Les bases du mapping objet - relationnel

- Les déclarations de mapping
- Les types utilisés par Hibernate
- Plusieurs mappings pour une même classe
- Les annotations Java 5

Etablir la persistance des collections

- Les types de collections persistantes
- Les clés étrangères, les éléments et les index d'une collection
- Association * - * et 1 - *

Etablir la persistance des associations

- Associations unidirectionnelles
- Associations bidirectionnelles
- Mappings complexes

Etablir la persistance des relations d'héritage

- Différentes stratégies
- Utilisation d'un discriminant
- Gestion du polymorphisme
- Comparaison des méthodes, avantages et inconvénients, dans quels cas les utiliser ?

Hibernate Query Language (HQL) et gestion des objets persistants

- Stocker et charger un objet
- Syntaxe de HQL, les clauses from, select, where, order by, group by
- Jointures, agrégation, requêtes polymorphiques
- Les requêtes imbriqués
- Les bonnes pratiques
- Externalisation de requêtes nommées

Hibernate Query Language (HQL) et gestion des objets persistants

- Utilisation des associations et des propriétés chargées tardivement
- Gestion du cache de second niveau
- Le cache de requêtes

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	26 au 28		10 au 12		25 au 27		29 au 31		22 au 24		17 au 19

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR7

DUREE

3 jours

OBJECTIFS

Etre capable de développer une application Java EE en utilisant Spring

Connaître et assimiler les concepts d'un conteneur « léger »

Connaître les apports de la Programmation Orientée Aspect (AOP)

Savoir utiliser les bonnes pratiques de développement Spring

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissances équivalentes au JA1 - Java - Les bonnes pratiques de développement avec Eclipse.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Développer des applications avec Spring

Spring est un framework qui permet de gérer les technologies de présentation, de persistance ou services tout en limitant fortement l'adhérence aux technologies d'implémentation. La clé en est l'utilisation du pattern IoC (Inversion of Control). Ce type de solution est également dénommé « conteneur léger ».

Il s'agit également d'un outil open source plébiscité et très actif. La nouvelle spécification EJB3 de Sun reprend en grande partie les « best practices » développés par Spring.

Programme

Objectifs

- Concepts de conteneur léger
- Vue d'ensemble et exemples d'utilisation
- Pattern « Inversion de Contrôle (IoC) / Injection de dépendance »

Mise en oeuvre

- Les Beans, BeanFactory et ApplicationContext
- Modes singleton ou normal
- Gestion des propriétés, « collaborators »
- Méthodes d'injection de dépendance
- Configuration de Beans spécifiques à Spring, cycle de vie
- Définitions de bean abstrait et héritage
- APIs d'accès aux ressources
- Techniques de validations (PropertyEditors, DataBinder, BeanWrapper)

Programmation Orientée Aspect (AOP)

- Les concepts
- La terminologie utilisée : Aspect, Joinpoint, Advice, Pointcut, Introduction, Target object, AOP Proxy, Weaving
- Les différents types « advice » : Around, Before, Throws, After
- Les objectifs et fonctionnalités AOP de Spring
- Intégration avec le framework AspectJ

Gestion des transactions

- La couche d'abstraction Spring
- Les stratégies de transaction

Pattern DAO, Accès aux données avec JDBC

- Classes abstraites pour le support DAO
- Utilisation de JDBC via les APIs d'encapsulation Spring

Accès aux données avec Hibernate comme framework de persistance objet

- Gestion des ressources
- Configuration de SessionFactory dans le contexte d'application Spring
- IoC : HibernateTemplate et HibernateCallback
- Implémentation DAO avec les APIs Hibernate
- Démarcation de transactions par programmation et déclaration

Framework MVC Web spécifique à Spring

- Les fonctionnalités de Spring
- DispatcherServlet
- Les contrôleurs
- Gestion des vues
- Techniques de présentation : Velocity et FreeMarker
- Intégration avec JSF et Struts

Utilisation de Spring avec RMI et les Web Services

- Concepts
- Accéder aux EJBs
- Utilisation des classes Spring

Intégration avec les EJBs

- Utilisation des associations et des propriétés chargées tardivement
- Gestion du cache de second niveau
- Le cache de requêtes

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
29 au 31		26 au 28		9 au 11		23 au 25		24 au 26		26 au 28	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR8

DUREE

3 jours

OBJECTIFS

Etre capable de développer des composants métier EJB3

Connaître et assimiler les concepts d'un conteneur EJB3

Mettre en oeuvre les différents types d'EJB : session, entity et MDB (Message Driven Bean)

Savoir utiliser les bonnes pratiques et design patterns EJB3

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissances équivalentes au JA1 - Java - Les bonnes pratiques de développement avec Eclipse.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Développer des composants EJB 3

Les *Entreprise JavaBeans (EJB)* n'ont que peu de rapport avec les *JavaBeans standards*.

Les *EJB* sont des composants serveur qui prennent en charge la sécurité, la réplication, les transactions et la persistance.

La spécification *EJB3* cherche à revoir largement *EJB2*, jugé trop complexe pour les développements quotidiens. Elle emprunte d'ailleurs assez largement aux solutions éprouvées qui se sont développées avec succès en alternative à *EJB2*, essentiellement *Spring* et *Hibernate*.

Programme

Objectifs

- Architecture n tiers
- Les couches présentation, logique applicative, logique métier, données métier, technique/middleware, stockage
- Composants et clients J2EE
- Les conteneurs J2EE et les services offerts : gestion de ressources, distribution, persistance, transaction et sécurité
- Vue d'ensemble de l'architecture J2EE et J2SE
- Les serveurs GlassFish et JBoss
- Les IDE NetBeans et Eclipse

JNDI (Java Naming Directory Interface)

- Objectifs
- DataSource et pool de connexions
- Connexions aux base de données
- Mise en oeuvre
- Spécifier la référence d'une ressource

Caractéristiques des EJB3

- Les règles à suivre pour assurer l'évolutivité et la pérennité des développements
- Interfaces métier, intérêts des niveaux d'abstraction
- Les annotations
- Les types d'EJB3 : session, entity et Message Driven Bean
- Déploiement des EJB3
- Localisation des EJB3 avec JNDI
- Injection de dépendance
- Les clients standalone et web
- Mise en oeuvre du MVC avec EJB3

Session Bean

- Principes des design pattern Factory et Façade
- Session stateless
- Session stateful
- Le service de planification pour work flow (Timer Service)

Entity

- Mapping objet relationnel
- L'Entity Manager
- Le cycle de vie des EJB3
- Gestion des relations
- Les stratégies d'implémentation de l'héritage

Enterprise Java Bean Query Language (EJB QL)

- Syntaxe, les clauses from, where, select, order by
- Exemples des possibilités d'EQL

JMS et les Message Driven Bean

- Java Message API (JMS), présentation et objectifs
- Les types de transfert de messages : Publish/Suscribe et Point à Point.
- Les Message Driven Bean (MDB) définition et mise oeuvre

Les transactions

- Transactions gérées par le conteneur
- Les attributs de transaction
- Roll back d'une transaction
- Java Transaction API (JTA)

Sécurité

- Gestion des utilisateurs et des groupes
- Configuration des rôles de sécurité
- Sécurité déclarative ou programmée sur les EJB3

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	26 au 28		10 au 12		25 au 27		29 au 31		22 au 24		17 au 19

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

Applications Web2.0 avec Ajax

CODE

AR9

DUREE

3 jours

OBJECTIFS

Apprendre à manipuler le modèle DOM

Tirer parti des possibilités offertes par les CSS

Se familiariser à la programmation objet en Javascript

Comprendre le fonctionnement de l'objet XMLHttpRequest

Utiliser un framework AJAX

PUBLIC

Concepteurs/développeurs, chefs de projets, architectes technique

PRE-REQUIS

Connaissance d'un langage de programmation

TARIF D'INSCRIPTION

1 200 € H.T. par personne

AJAX n'est pas une technologie en elle-même, mais un terme qui évoque l'utilisation conjointe d'un ensemble de technologies couramment utilisées sur le Web :

HTML (ou XHTML) pour la structure sémantique des informations ; CSS pour la présentation des informations ; DOM et JavaScript pour afficher et interagir dynamiquement avec l'information présentée ; l'objet XMLHttpRequest pour échanger et manipuler les données de manière asynchrone avec le serveur Web ; et enfin XML et XSLT.

Le Web 2.0 désigne alors les application web qui utilisent intelligemment ces technologies combinées pour offrir des interfaces riches au sein d'un navigateur web.

Programme

Les concepts d'AJAX

- Introduction
- L'objet XMLHttpRequest
- Exemple de site utilisant AJAX
- Le protocole HTTP pour Ajax

JavaScript

- Introduction
- Syntaxe
- Les objets Javascript
- Présentation de framework Ajax

Modèle DOM

- Les concepts DOM
- Manipulation du modèle à partir de Javascript
- Les feuilles de styles CSS

Elaboration d'un modèle de bloc

- Modèle HTML
- Ajout de feuilles CSS au bloc
- Intégration de fonctionnalités avec DOM et Javascript
- Connexion Ajax avec le serveur

Fonctions sophistiquées

- Intégrer du contenu
- Editer du texte en ligne
- Déplacer des blocs
- Connexion des blocs à des services

Etude de cas

- Maquette HTML de l'application
- Réalisation des blocs par défaut
- Mode connecté (utilisation d'un annuaire)
- Implémentation de la procédure d'identification (stockage en base de donnée)

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 17		12 au 14		14 au 16	20 au 22			8 au 10			3 au 5

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR10

DUREE

3 jours

OBJECTIFS

Etre capable de développer une application web en architecture MVC avec le framework Tapestry

Connaître et assimiler un framework de présentation dans une architecture n-tiers web

Mettre en oeuvre les différentes classes du framework

Savoir utiliser les bonnes pratiques spécifiques au cadre de développement Tapestry

Connaître les bibliothèques et les outils graphiques associés à Tapestry

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet, architectes technique

PRE-REQUIS

Connaissances équivalentes au JA1 - Java - Les bonnes pratiques de développement avec Eclipse et au JA8 réaliser des applications Web avec JEE et Tomcat

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Développer des applications avec Tapestry

Tapestry est un framework de présentation mettant en oeuvre les « best practices » objet.

C'est une alternative élégante et puissante aux technologies Swing ou SWT.

Programme

Introduction

- Vue d'ensemble du framework
- Les pages et les composants
- Les moteurs et les services
- Le langage objet de navigation dans les graphes
- Avenir de Tapestry

Les templates

- Définition, rappel XML
- Localisation, contenu
- Place des composants dans les templates

Configuration des composants

- Les différents types de liens
- Bibliothèque

Les listeners

- Le principe de call back
- Type de retour et les paramètres
- Invocation de listener

Les valideurs de saisie

- FieldLabel
- Field validation
- Extension de validation

Création de composants Tapestry

- Les interfaces de base
- Spécifications des composants
- Coder un composant
- Les paramètres d'un composant
- Les bibliothèques de composants

Injection de propriétés

- Meta injection
- Object et page injection
- Les événements sur la page

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	12 au 14		02 au 04			9 au 11		10 au 12		5 au 7	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

Administration JBoss

CODE

AR11

DUREE

2 jours

OBJECTIFS

Savoir installer et configurer JBoss.

Savoir intégrer une application JEE

Savoir assurer la mise en production, l'exploitation

PUBLIC

Concepteurs/développeurs, chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissances système

TARIF D'INSCRIPTION

800 € H.T. par personne

Ce module montre l'installation, les optimisations de déploiement, la surveillance du serveur et son administration. Il sera utile aussi bien aux administrateurs qu'aux développeurs qui ont souvent besoin de savoir comment fonctionne le serveur d'application.

Programme

Introduction

- Architecture : noyau
- Services
- Présentation
- Applications
- Le framework JMX
- Le concept de Mbean
- Configuration et installation du serveur

Architecture JEE

- Application Web, container, sources de données, pilote JDBC, hôtes virtuels
- Utilisation de la console de JBoss
- Configuration des services JBoss

Moteur Web : Tomcat

- Intégration de Tomcat et paramétrage
- Activation du service Tomcat
- Description du fichier de configuration : server.xml

Déployer des applications JEE

- Les fichiers EAR
- Les EJB
- Les applications web (WAR)
- Regrouper des composants unitaires

Exploitation du serveur

- La console JMX
- Mise en cluster
- Configurer un cluster
- Concept de Netboot
- Gérer les performances
- Les logs
- Résoudre des incidents

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	15 au 16		05 au 06			12 au 13		13 au 14		8 au 9	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

Administration JOnAS

CODE

AR12

DUREE

2 jours

OBJECTIFS

Savoir installer et configurer JOnAS.

Savoir intégrer une application JEE

Savoir assurer la mise en production, l'exploitation

PUBLIC

Concepteurs/développeurs, chefs de projets, architectes technique.

PRE-REQUIS

Connaissances système

TARIF D'INSCRIPTION

800 € H.T. par personne

Ce module montre l'installation, les optimisations de déploiement, la surveillance du serveur JOnAS et son administration. Il sera utile aussi bien aux administrateurs qu'aux développeurs qui ont souvent besoin de savoir comment fonctionne le serveur d'application.

Programme

Introduction

- Architecture
- Configuration et installation du serveur

Architecture JEE

- Application Web, container, sources de données, pilote JDBC
- Utilisation de la console d'administration jonasadmin et autres outils d'administration JOnAS
- Activation et configuration de services JOnAS

Déployer des applications JEE

- Les fichiers EAR
- Les EJB
- Les applications web (WAR)

La sécurité dans JOnAS

- Les risques.
- JAAS et la sécurisation des applications JOnAS.
- Propagation de la sécurité.

Exploitation du serveur

- Gérer les performances
- Les logs
- Résoudre des incidents

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
18 au 19		15 au 16		24 au 25	14 au 15		23 au 24		11 au 12		6 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

AR13

DUREE

2 jours

OBJECTIFS

Assimiler les concepts appliqués de SOA : service, processus métier, workflow humain, orchestration automatique,

Comprendre visuellement un processus BPEL,

Savoir créer, modifier et maintenir un processus BPEL

Connaître les éléments de la notation WS-BPEL 2.0

Mettre en œuvre une orchestration de web services avec BPEL, dans un moteur BPEL Java

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissance du langage XML,

Connaissance du langage Java et XSL,

Connaissance des concepts SOA et de l'implémentation Web-Services

PRIX

1 000 € H.T.

Orchestrer ses processus avec BPEL

Avec l'avènement des « Web-Services », implémentation particulière d'une Architecture Orientée Services (SOA), il est désormais possible de définir et piloter l'orchestration, c'est à dire l'enchaînement automatisé, de web-services entre eux.

Dès lors, il est possible de définir entièrement un processus métier qui fait inter-agir des services issus de systèmes différents sans intervenir sur les « programmes » mais en paramétrant les dits processus.

Le langage BPEL, ou plus exactement la norme WS-BPEL est promue par le consortium OASIS dans cet objectif. Elle est en passe de s'imposer comme langage commun des moteurs d'orchestration.

Programme

Rappels sur SOA

- Les limitation de l'objet classique
- Le concept d'architecture orientée services (SOA), définitions
 - Service, Contrat, Couplage faible
 - Annuaire,
 - Transport & communication
- Implémentation de SOA avec les « Web-Services »
- Rôle et place de BPEL dans une architecture SOA
- Rappels techniques des principales technologies utilisées : XML, WSDL, XSL, outils nécessaires pour WS-BPEL

Orchestration

- Objectifs, Définitions
- Workflow « people centric » vs « computer centric » (orchestration) : les limites de BPEL
- Principales problématiques
 - Conservation de contexte
 - Transformations
 - Transactions

Description simple d'un processus avec BPEL

- Modèle graphique de processus BPEL
- Enchaînements : séquentiel, conditionnel, parallèle
- Invocation de services : flux d'entrée, flux de sorties
- Gestion des évènements
- Activités BPEL
- Transformations et mappings

TP : Création et mise en œuvre d'un processus BPEL avec l'éditeur graphique intégré de NetBeans 5.5

WS-BPEL avancé

- Revue de détail de la spécification WS-BPEL 1.1
- Création et manipulation d'un process BPEL sous sa forme native XML
- Transformations successives entres flux d'entrées/sorties des services
- Problématiques de transformations complexes : utilisation de XSLT
- Passage de contexte aux services, les limites de BPEL
- Gestion d'erreurs
- Gestion de transactions

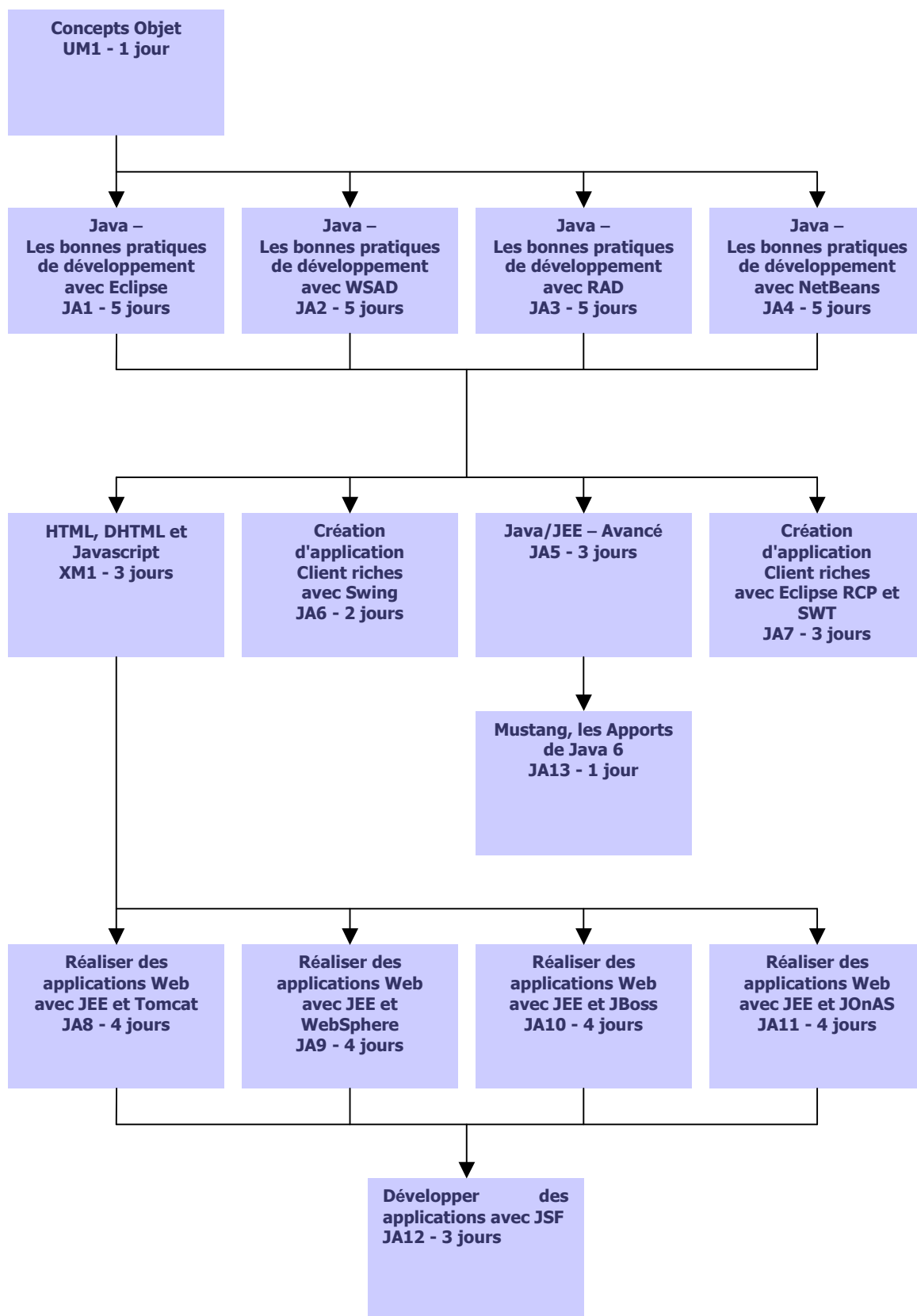
TP : Manipulation d'un process BPEL sous Apache ODE avec Eclipse.

Panorama des solutions, pour aller plus loin

- Les principaux moteurs BPEL
- La norme WS-BPEL 2.0 (non publiée) : ce qui est nouveau
- Les serveurs d'application et Bus de Services (ESB)
- JBPEL, JEE5, Sun GlassFish

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
				24 au 25	14 au 15		23 au 24		11 au 12		6 au 7



CODE

JA1

DURÉE

5 jours

OBJECTIFS

Etre capable de développer une application optimisée en Java

Savoir installer, configurer, utiliser et étendre Eclipse

Assimiler les concepts Objet par la pratique de Java

Savoir gérer les accès concurrents

Utiliser une base de données avec Java

Connaître les caractéristiques des différentes plate-formes J2SE, J2EE et J2ME

Savoir traduire un modèle de conception UML en langage Java

Acquérir les bonnes pratiques de conception et de programmation

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

2 000 € H.T par personne

Programmation Java : Les bonnes pratiques de développement avec Eclipse

Programme

Architecture de la plate-forme Java

- Les caractéristiques du langage
- La machine virtuelle
- Les outils standards
- Projet, compilation, exécution, déboguage, génération de documentations avec Eclipse
- Conception de scripts d'exploitation avec Ant
- Les évolutions et les versions

Les éléments syntaxique du langage et les packages de base

- Structure d'une application
- Les types primitifs, les classes
- « Autoboxing/unboxing »
- Les structures de contrôle
- Mise en oeuvre des concepts objet
- Classe, attributs et méthodes
- Instancier des objets, garbage collector
- Optimiser la JVM
- Optimisation du JIT
- Pools d'objet
- Encapsulation, intégrité, évolutivité
- Les tableaux
- Les package
- Mise en oeuvre de l'héritage
- Polymorphisme
- Les interfaces
- Gestion des erreurs, les exceptions

- Les collections
- Les collections paramétrées
- Les annotations
- La programmation asynchrone, les threads

Présentation des framework AWT et Swing pour les clients riches

- Le pattern Modèle Vue Contrôleur
- Gestion des fenêtres
- Les principaux composants de base
- La Gestion évènementiel

Accéder à une base de donnée relationnelle avec JDBC

- Architecture JDBC
- Récupérer une connexion à la base
- Exécuter des requêtes SQL
- Exploitation des résultats
- Persistance objet

Les Design Patterns

- Singleton
- Fabrique
- Façade

Mise en oeuvre d'Eclipse

- Les concepts d'Eclipse
- L'environnement, gestion de projet, les perspectives
- Compiler, exécuter et mettre au point un programme Java
- Installer des plugins dans Eclipse
- Les plugins indispensables

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 19	12 au 16	12 au 16	2 au 6	21 au 25	11 au 15	9 au 13	20 au 24	10 au 14	8 au 12	5 au 9	3 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA2

DUREE

5 jours

OBJECTIFS

Etre capable de développer une application optimisée en Java

Savoir installer, configurer, utiliser et étendre WSAD

Assimiler les concepts Objet par la pratique de Java

Savoir gérer les accès concurrents

Utiliser une base de données avec Java

Connaître les caractéristiques des différentes plate-formes J2SE, J2EE et J2ME

Savoir traduire un modèle de conception UML en langage Java

Acquérir les bonnes pratiques de conception et de programmation

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

2 000 € H.T par personne

Programmation Java : les bonnes pratiques de développement avec WSAD

Programme

Architecture de la plate-forme Java

- Les caractéristiques du langage
- La machine virtuelle
- Les outils standards
- Projet, compilation, exécution, déboguage, génération de documentations avec Eclipse
- Conception de scripts d'exploitation avec Ant
- Les évolutions et les versions

Les éléments syntaxique du langage et les packages de base

- Structure d'une application
- Les types primitifs, les classes
- « Autoboxing/unboxing »
- Les structures de contrôle
- Mise en oeuvre des concepts objet
- Classe, attributs et méthodes
- Instancier des objets, garbage collector
- Optimiser la JVM
- Optimisation du JIT
- Pools d'objet
- Encapsulation, intégrité, évolutivité
- Les tableaux
- Les package
- Mise en oeuvre de l'héritage
- Polymorphisme
- Les interfaces
- Gestion des erreurs, les exceptions

- Les collections
- Les collections paramétrées
- Les annotations
- La programmation asynchrone, les threads

Présentation des framework AWT et Swing pour les clients riches

- Le pattern Modèle Vue Contrôleur
- Gestion des fenêtres
- Les principaux composants de base
- La Gestion évènementiel

Accéder à une base de donnée relationnelle avec JDBC

- Architecture JDBC
- Récupérer une connexion à la base
- Exécuter des requêtes SQL
- Exploitation des résultats
- Persistance objet

Les Design Patterns

- Singleton
- Fabrique
- Façade

Mise en oeuvre avec WSAD

- Installation de WSAD
- Les perspectives Données de WSAD pour faciliter l'accès et le visualisation des données
- Les outils WSAD pour créer de nouvelles bases, tables et requêtes SQL

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 19	12 au 16	12 au 16	2 au 6	21 au 25	11 au 15	9 au 13	20 au 24	10 au 14	8 au 12	5 au 9	3 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA3

DUREE

5 jours

OBJECTIFS

Etre capable de développer une application optimisée en Java

Savoir installer, configurer, utiliser et étendre RAD

Assimiler les concepts Objet par la pratique de Java

Savoir gérer les accès concurrents

Utiliser une base de données avec Java

Connaître les caractéristiques des différentes plate-formes J2SE, J2EE et J2ME

Savoir traduire un modèle de conception UML en langage Java

Acquérir les bonnes pratiques de conception et de programmation

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

2 000 € H.T par personne

Programmation Java : les bonnes pratiques de développement avec RAD

Programme

Architecture de la plate-forme Java

- Les caractéristiques du langage
- La machine virtuelle
- Les outils standards
- Projet, compilation, exécution, déboguage, génération de documentations avec Eclipse
- Conception de scripts d'exploitation avec Ant
- Les évolutions et les versions

Les éléments syntaxique du langage et les packages de base

- Structure d'une application
- Les types primitifs, les classes
- « Autoboxing/unboxing »
- Les structures de contrôle
- Mise en oeuvre des concepts objet
- Classe, attributs et méthodes
- Instancier des objets, garbage collector
- Optimiser la JVM
- Optimisation du JIT
- Pools d'objet
- Encapsulation, intégrité, évolutivité
- Les tableaux
- Les package
- Mise en oeuvre de l'héritage
- Polymorphisme
- Les interfaces
- Gestion des erreurs, les exceptions

- Les collections
- Les collections paramétrées
- Les annotations
- La programmation asynchrone, les threads

Présentation des framework AWT et Swing pour les clients riches

- Le pattern Modèle Vue Contrôleur
- Gestion des fenêtres
- Les principaux composants de base
- La Gestion événementielle

Accéder à une base de donnée relationnelle avec JDBC

- Architecture JDBC
- Récupérer une connexion à la base
- Exécuter des requêtes SQL
- Exploitation des résultats
- Persistance objet

Les Design Patterns

- Singleton
- Fabrique
- Façade

Mise en oeuvre avec RAD

- Installation
- Les outils de mise au point de programmes Java
- Exemple de modèle UML et démonstration de génération de code Java

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 19	12 au 16	12 au 16	2 au 6	21 au 25	11 au 15	9 au 13	20 au 24	10 au 14	8 au 12	5 au 9	3 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA4

DUREE

5 jours

OBJECTIFS

Etre capable de développer une application optimisée en Java

Savoir installer, configurer, utiliser et étendre NetBeans

Assimiler les concepts Objet par la pratique de Java

Savoir gérer les accès concurrents

Utiliser une base de données avec Java

Connaître les caractéristiques des différentes plate-formes J2SE, J2EE et J2ME

Savoir traduire un modèle de conception UML en langage Java

Acquérir les bonnes pratiques de conception et de programmation

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

2 000 € H.T par personne

Programmation Java : les bonnes pratiques de développement avec NetBeans

Programme

Architecture de la plate-forme Java

- Les caractéristiques du langage
- La machine virtuelle
- Les outils standards
- Projet, compilation, exécution, déboguage, génération de documentations avec Eclipse
- Conception de scripts d'exploitation avec Ant
- Les évolutions et les versions

Les éléments syntaxique du langage et les packages de base

- Structure d'une application
- Les types primitifs, les classes
- « Autoboxing/unboxing »
- Les structures de contrôle
- Mise en oeuvre des concepts objet
- Classe, attributs et méthodes
- Instancier des objets, garbage collector
- Optimiser la JVM
- Optimisation du JIT
- Pools d'objet
- Encapsulation, intégrité, évolutivité
- Les tableaux
- Les package
- Mise en oeuvre de l'héritage
- Polymorphisme
- Les interfaces
- Gestion des erreurs, les exceptions

- Les collections
- Les collections paramétrées
- Les annotations
- La programmation asynchrone, les threads

Présentation des framework AWT et Swing pour les clients riches

- Le pattern Modèle Vue Contrôleur
- Gestion des fenêtres
- Les principaux composants de base
- La Gestion évènementiel

Accéder à une base de donnée relationnelle avec JDBC

- Architecture JDBC
- Récupérer une connexion à la base
- Exécuter des requêtes SQL
- Exploitation des résultats
- Persistance objet

Les Design Patterns

- Singleton
- Fabrique
- Façade

Mise en oeuvre avec NetBeans

- Les concepts de NetBeans, de qui le différencie d'Eclipse
- Les projets NetBeans basés sur l'outil Ant
- La mise au point de programmes Java
- Le concepteur d'applications graphiques Swing

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 19	12 au 16	12 au 16	2 au 6	21 au 25	11 au 15	9 au 13	20 au 24	10 au 14	8 au 12	5 au 9	3 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA5

DUREE

3 jours

OBJECTIFS

Etre capable de développer une application optimisée en Java

Assimiler les concepts avancés de Java et JEE à travers de nombreux exemples et travaux pratiques

Mettre en oeuvre les annotations

Savoir utiliser le framework d'objets distribués RMI

Connaître les caractéristiques des différentes plate-formes J2SE, J2EE et J2ME

Connaître les diagrammes de UML 2

Acquérir les bonnes pratiques de conception et de programmation

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi la formation JA1 « Java : les bonnes pratiques de développement » ou avoir l'expérience équivalente.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Programmation Java/JEE – Avancé

Ce cours aide les développeurs à améliorer leur pratique de Java. Il introduit la modélisation UML, les design patterns ainsi que les bases d'une architecture à objets distribués.

Programme

Concepts et syntaxes avancés du langage

- Les annotations (metadata)
- Les types génériques
- Autoboxing et auto-unboxing des types primitifs
- Amélioration des boucles avec les collections
- Les énumérations
- Les « import static »
- Méthodes à nombre d'arguments variable

Les diagrammes de modélisation UML 2

- Diagramme de classe
- Diagramme de séquence

Principes des architectures n-tiers, pattern MVC et Design Patterns

- Les couches logiques
- Singleton
- Factory
- Façade
- Observateur
- Proxy

Gérer les accès concurrents

- Définition d'un thread
- Créer et manipuler les threads
- Les bonnes pratiques

Accéder à une base de donnée relationnelle avec JDBC

- Architecture et types de pilotes JDBC
- Charger le pilote et récupérer une connexion à la base de données
- Exécuter des requêtes SQL, requêtes paramétrées
- Exploitation des résultats
- Persistance objet, Object Relational Mapping, Hibernate

Architecture à objets distribués : RMI (Remote Method Invocation)

- Présentation
- Appel de méthodes d'un objet distant (TCP-IP)
- Appel de méthodes d'un objet distant (RMI)
- Annuaire d'objets : RMIRRegistry
- Les packages de l'API RMI
- Mise en route
- Interface Remote
- Classe implémentée
- Application côté serveur
- La sécurité : le SecurityManager
- Le client RMI
- API Java 1.4 et 5 (1.5)

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
10 au 12		8 au 9		19 au 20	24 au 25	28 au 29		27 au 28			10 au 11

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA6

DUREE

2 jours

OBJECTIFS

Etre capable de développer une application graphique avec Java Swing

Assimiler les bonnes pratiques de développement de client riche Java

Savoir éviter les pièges des composants Java Swing

Mettre en oeuvre les outils pour concevoir une interface graphique ergonomique

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi la formation JA1 « Java : les bonnes pratiques de développement » ou avoir l'expérience équivalente.

TARIF D'INSCRIPTION

800 € H.T. par personne

Création d'applications Client riches avec Swing

Programme

Présentation de Java Swing

- L'architecture applicative (logique) n tiers et le pattern MVC
- Les concepts de Swing
- Les APIs Swing
- Prototypage de l'architecture logique avec UML 2
- Use case
- Diagramme de séquence et diagramme de classe
- Maquettage de l'IHM

Les composants Swing

- Le framework AWT (Abstract Window Toolkit)
- Le framework JFC (Java Foundation Class), son positionnement par rapport à AWT
- Les concepts de Swing
- Panorama des principaux composants
- Classification des composants : les conteneurs, les contrôles simples et les composants d'affichage
- La classe racine de l'arborescence : JComponent
- Les bases : JButton, JTextField, JTextArea
- Les conteneurs : JPanel, JScrollPane, JTabbedPane
- Le Look & Feel
 - Personnalisation du Look & Feel
- Les gestionnaires de disposition (Layout Manager)
- Configuration de la position et les dimensions à l'intérieur d'un container
- Les principaux Layout : BorderLayout, FlowLayout, GridLayout, GridBagLayout
- Création d'un Layout
- Le modèle évènementiel par délégation
- Le design pattern du GoF « Observateur », la notification des changements
- Les Listeners et les Events
- Implémentation de Listener, héritage d'Adapter et les inner classes
- Les différents types de listeners (MouseListener, KeyboardListener, ...)
- Intégration des threads avec Swing
- Les techniques single-thread et multi-thread
- Les menus
- Les onglets (JTabbedPane)
- Les composants avancés, les tables (JTable) et les arbres (JTree)
- Déploiement d'applications avec Java Web Start

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 9		5 au 7	16 au 18	21 au 23	25 au 27			24 au 26			7 au 9

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA7

Durée

4 jours

Objectifs

Comprendre l'architecture d'Eclipse et d'Eclipse RCP.

Savoir utiliser Eclipse et le PDE (Plug-in Development Environment) pour développer, tester et déployer des applications Eclipse RCP.

Public

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

Pré-requis

Avoir suivi la formation JA1 « Java : les bonnes pratiques de développement » ou avoir l'expérience équivalente.

Tarif d'inscription

1 600 € H.T. par personne

Création d'applications Client riches avec Eclipse RCP

Programme

Présentation

- Historique d'Eclipse et présentation des principaux projets de la fondation Eclipse.
- Eclipse RCP et l'architecture d'Eclipse : modularité et extensibilité.
- Les notions de plugin et de bundle. Apports d'OSGi.
- Le PDE (Plug-in Development Environment).
- Aperçu du cycle de développement d'une application Eclipse RCP.

Le socle technique

- Structure des plugins. Le fichier MANIFEST.MF.
- Les relations entre plugins : dépendances et contribution.
- La notion de points d'extension.
- Utiliser le PDE pour contribuer à un point d'extension.
- Développer et tester un plugin.
- Création pas à pas d'une application Eclipse RCP.
- Définir de nouveaux point d'extension.

Le cadre graphique d'Eclipse

- Présentation de la notion de Worbench.
- Les vues, les éditeurs et les perspectives.
- Les menus (menu principal, barre de boutons et menus contextuels).
- Les pages de préférences.
- Les assistants (Wizards).
- La gestion de l'aide.

Les bibliothèques graphiques d'Eclipse : SWT et JFace

- Caractéristiques de SWT.
- Les composants graphiques de SWT.
- Les différents 'Layouts' proposés.
- Utilisation du 'Visual Editor' pour la construction des interfaces graphiques.
- Présentation de JFace.
- Savoir utiliser les 'Viewers' JFace.
- Lien entre composants graphiques et modèle : l'API 'JFace Data Binding'.
- Simplifications apportées par JFace pour les pages de préférences.
- La librairie Eclipse Forms

Déploiement des applications Eclipse RCP.

- Internationalisation et gestion des fragments.
- Exporter une application Eclipse RCP : la notion de produit et l'outillage associé.
- Le système de gestion des mises à jour d'Eclipse. La Notion de 'Feature'.
- Déploiement par Java WebStart.

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	26 au 28		10 au 12		25 au 27		29 au 31		22 au 24		17 au 19

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA8

DUREE

4 jours

OBJECTIFS

Etre capable de développer une application web avec le framework Java EE

Assimiler les concepts de Servlets et de JSP à travers de nombreux exemples et travaux pratiques

Mettre en oeuvre les bonnes pratiques du développement en client léger

Savoir utiliser un environnement de développement standard du marché

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi le JA1 « Java : Les bonnes pratiques du développement » ou avoir l'expérience équivalente.

Connaissance de base du langage HTML.

TARIF D'INSCRIPTION

1 600 € H.T. par personne

Réaliser des applications Web avec Java EE et Tomcat

Programme

Architecture Java EE

- Architecture web, serveur HTTP et client léger, les navigateurs
- Protocole HTTP, requête cliente et réponse serveur, protocole en mode déconnecté sans session
- Le pattern d'architecture Modèle Vue Contrôleur (MVC 2)
- Fonctionnement et rôle des servlets
- Fonctionnement et rôle des pages JSP
- Les serveurs d'application open source JAVA EE : Tomcat
- Structure standard d'une application web JAVA EE, norme de répertoires
- Le fichiers de configuration normalisé web.xml
- Déploiement des applications web JAVA EE, les fichiers WAR et EAR
- Les fichiers de configuration spécifiques aux serveurs
- Les descripteurs de déploiement

Mise en oeuvre des servlets

- Les APIs de base
- Le cycle de vie d'une servlet
- Les méthodes init, destroy, doPost, doGet
- Les objets request et response
- Partage des servlets pour l'ensemble des clients
- Chargement et déchargement des servlets
- Redirection vers d'autres ressources
- Gestion d'un formulaire HTML et récupération des données
- Gestion d'une session utilisateur par champs cachés HTML
- Gestion d'une session utilisateur par cookie non persistant et persistant
- Gestion d'une session utilisateur par l'API Session gérée par le serveur JAVA EE
- Accès à la session utilisateur, fermeture de la session
- Les bonnes pratiques d'utilisation de la session
- Les différents contextes : servlet, session, requête
- Cycle de vie des attributs
- Utilisation d'Eclipse Web Tool Platform pour la création et la mise au point de servlets
- Les listeners de servlet
- Mise en oeuvre de filtres de servlet

Conception de JSP

- Intégration de balises JSP dans une page HTML
- Lancement d'une page JSP
- Syntaxe des balises JSP : directives, scriptlets, et autres balises spécifiques
- Transmettre des données à une JSP, les patterns Data Transfert Object et Value Object
- Accès aux contexte d'application, de session et de requête
- Les objets prédéfinis
- Optimisation et utilisation de cache
- Gestion de la navigation, sécurité, pages d'erreurs

La bibliothèque de balise JSTL (JSP Standart Tag Library)

- Exemple d'utilisation
- Les balises pour les variables
- Les balises de structure de contrôle (tests, boucles, ...)
- Les balises pour l'internationalisation
- Les balises pour les requêtes SQL
- Réalisation de ses propres balises
- Conception étape par étape d'une balise personnalisée

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
22 au 25	19 au 22	19 au 22	10 au 13	29 mai au 1 ^{er} juin	18 au 22	16 au 19	27 au 30	17 au 20	15 au 18	12 au 15	10 au 13

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA9

DUREE

4 jours

OBJECTIFS

Etre capable de développer une application web avec le framework Java EE

Assimiler les concepts de Servlets et de JSP à travers de nombreux exemples et travaux pratiques

Mettre en oeuvre les bonnes pratiques du développement en client léger

Savoir utiliser un environnement de développement standard du marché

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi le JA1 « Java : Les bonnes pratiques du développement » ou avoir l'expérience équivalente.

Connaissance de base du langage HTML.

TARIF D'INSCRIPTION

1 600 € H.T. par personne

Réaliser des applications Web avec Java EE et WebSphere

Programme

Architecture JAVA EE

- Architecture web, serveur HTTP et client léger, les navigateurs
- Protocole HTTP, requête cliente et réponse serveur, protocole en mode déconnecté sans session
- Le pattern d'architecture Modèle Vue Contrôleur (MVC 2)
- Fonctionnement et rôle des servlets
- Fonctionnement et rôle des pages JSP
- Le serveur d'application JAVA EE : WebSphere
- Structure standard d'une application web JAVA EE, norme de répertoires
- Le fichiers de configuration normalisé web.xml
- Déploiement des applications web JAVA EE, les fichiers WAR et EAR
- Les fichiers de configuration spécifiques aux serveurs
- Les descripteurs de déploiement
- Accès à la session utilisateur, fermeture de la session
- Les bonnes pratiques d'utilisation de la session
- Les différents contextes : servlet, session, requête
- Cycle de vie des attributs
- Utilisation d'Eclipse Web Tool Platform pour la création et la mise au point de servlets
- Les listeners de servlet
- Mise en oeuvre de filtres de servlet

Conception de JSP

- Intégration de balises JSP dans une page HTML
- Lancement d'une page JSP
- Syntaxe des balises JSP : directives, scriptlets, et autres balises spécifiques
- Transmettre des données à une JSP, les patterns Data Transfert Object et Value Object
- Accès aux contexte d'application, de session et de requête
- Les objets prédéfinis
- Optimisation et utilisation de cache
- Gestion de la navigation, sécurité, pages d'erreurs

La bibliothèque de balise JSTL (JSP Standard Tag Library)

- Le cycle de vie d'une servlet
- Les méthodes init, destroy, doPost, doGet
- Les objets request et response
- Partage des servlets pour l'ensemble des clients
- Chargement et déchargement des servlets
- Redirection vers d'autres ressources
- Gestion d'un formulaire HTML et récupération des données
- Gestion d'une session utilisateur par champs cachés HTML
- Gestion d'une session utilisateur par cookie non persistant et persistant
- Gestion d'une session utilisateur par l'API Session gérée par le serveur JAVA EE
- Exemple d'utilisation
- Les balises pour les variables
- Les balises de structure de contrôle (tests, boucles, ...)
- Les balises pour l'internationalisation
- Les balises pour les requêtes SQL
- Réalisation de ses propres balises
- Conception étape par étape d'une balise personnalisée

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
22 au 25	19 au 22	19 au 22	10 au 13	29 mai au 1 ^{er} juin	18 au 22	16 au 19	27 au 30	17 au 20	15 au 18	12 au 15	10 au 13

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA10

DURÉE

4 jours

OBJECTIFS

Etre capable de développer une application web avec le framework Java EE

Assimiler les concepts de Servlets et de JSP à travers de nombreux exemples et travaux pratiques

Mettre en oeuvre les bonnes pratiques du développement en client léger

Savoir utiliser un environnement de développement standard du marché

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi le JA1 « Java : Les bonnes pratiques du développement » ou avoir l'expérience équivalente.

Connaissance de base du langage HTML.

TARIF D'INSCRIPTION

1 600 € H.T. par personne

Réaliser des applications Web avec Java EE et JBoss

Programme

Architecture Java EE

- Architecture web, serveur HTTP et client léger, les navigateurs
- Protocole HTTP, requête cliente et réponse serveur, protocole en mode déconnecté sans session
- Le pattern d'architecture Modèle Vue Contrôleur (MVC 2)
- Fonctionnement et rôle des servlets
- Fonctionnement et rôle des pages JSP
- Le serveur d'application open source JAVA EE : JBoss
- Structure standard d'une application web JAVA EE, norme de répertoires
- Le fichiers de configuration normalisé web.xml
- Déploiement des applications web JAVA EE, les fichiers WAR et EAR
- Les fichiers de configuration spécifiques aux serveurs
- Les descripteurs de déploiement

Mise en oeuvre des servlets

- Le cycle de vie d'une servlet
- Les méthodes init, destroy, doPost, doGet
- Les objets request et response
- Partage des servlets pour l'ensemble des clients
- Chargement et déchargement des servlets
- Redirection vers d'autres ressources
- Gestion d'un formulaire HTML et récupération des données
- Gestion d'une session utilisateur par champs cachés HTML
- Gestion d'une session utilisateur par cookie non persistant et persistant
- Gestion d'une session utilisateur par l'API Session gérée par le serveur JAVA EE

- Accès à la session utilisateur, fermeture de la session
 - Les bonnes pratiques d'utilisation de la session
 - Les différents contextes : servlet, session, requête
 - Cycle de vie des attributs
 - Utilisation d'Eclipse Web Tool Platform pour la création et la mise au point de servlets
 - Les listeners de servlet
- Mise en oeuvre de filtres de servlet

Conception de JSP

- Intégration de balises JSP dans une page HTML
- Lancement d'une page JSP
- Syntaxe des balises JSP : directives, scriptlets, et autres balises spécifiques
- Transmettre des données à une JSP, les patterns Data Transfert Object et Value Object
- Accès aux contexte d'application, de session et de requête
- Les objets prédéfinis
- Optimisation et utilisation de cache
- Gestion de la navigation, sécurité, pages d'erreurs

La bibliothèque de balise JSTL (JSP Standart Tag Library)

- Exemple d'utilisation
- Les balises pour les variables
- Les balises de structure de contrôle (tests, boucles, ...)
- Les balises pour l'internationalisation
- Les balises pour les requêtes SQL
- Réalisation de ses propres balises
- Conception étape par étape d'une balise personnalisée

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
22 au 25	19 au 22	19 au 22	10 au 13	29 mai au 1 ^{er} juin	18 au 22	16 au 19	27 au 30	17 au 20	15 au 18	12 au 15	10 au 13

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA11

DUREE

4 jours

OBJECTIFS

Etre capable de développer une application web avec le framework Java EE

Assimiler les concepts de Servlets et de JSP à travers de nombreux exemples et travaux pratiques

Mettre en oeuvre les bonnes pratiques du développement en client léger

Savoir utiliser un environnement de développement standard du marché

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi le JA1 « Java : Les bonnes pratiques du développement » ou avoir l'expérience équivalente.

Connaissance de base du langage HTML.

TARIF D'INSCRIPTION

1 600 € H.T. par personne

Réaliser des applications Web avec Java EE et JOnAS

Programme

Architecture Java EE

- Architecture web, serveur HTTP et client léger, les navigateurs
- Protocole HTTP, requête cliente et réponse serveur, protocole en mode déconnecté sans session
- Le pattern d'architecture Modèle Vue Contrôleur (MVC 2)
- Fonctionnement et rôle des servlets
- Fonctionnement et rôle des pages JSP
- Le serveur d'application open source JAVA EE : JOnAS
- Structure standard d'une application web Java EE, norme de répertoires
- Le fichiers de configuration normalisé web.xml
- Déploiement des applications web JAVA EE, les fichiers WAR et EAR
- Les fichiers de configuration spécifiques aux serveurs
- Les descripteurs de déploiement

Mise en oeuvre des servlets

- Le cycle de vie d'une servlet
- Les méthodes init, destroy, doPost, doGet
- Les objets request et response
- Partage des servlets pour l'ensemble des clients
- Chargement et déchargement des servlets
- Redirection vers d'autres ressources
- Gestion d'un formulaire HTML et récupération des données
- Gestion d'une session utilisateur par champs cachés HTML
- Gestion d'une session utilisateur par cookie non persistant et persistant
- Gestion d'une session utilisateur par l'API Session gérée par le serveur Java EE

- Accès à la session utilisateur, fermeture de la session
- Les bonnes pratiques d'utilisation de la session
- Les différents contextes : servlet, session, requête
- Cycle de vie des attributs
- Utilisation d'Eclipse Web Tool Platform pour la création et la mise au point de servlets
- Les listeners de servlet
- Mise en oeuvre de filtres de servlet

Conception de JSP

- Intégration de balises JSP dans une page HTML
- Lancement d'une page JSP
- Syntaxe des balises JSP : directives, scriptlets, et autres balises spécifiques
- Transmettre des données à une JSP, les patterns Data Transfert Object et Value Object
- Accès aux contexte d'application, de session et de requête
- Les objets prédéfinis
- Optimisation et utilisation de cache
- Gestion de la navigation, sécurité, pages d'erreurs

La bibliothèque de balise JSTL (JSP Standard Tag Library)

- Exemple d'utilisation
- Les balises pour les variables
- Les balises de structure de contrôle (tests, boucles, ...)
- Les balises pour l'internationalisation
- Les balises pour les requêtes SQL
- Réalisation de ses propres balises
- Conception étape par étape d'une balise personnalisée

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
22 au 25	19 au 22	19 au 22	10 au 13	29 mai au 1 ^{er} juin	18 au 22	16 au 19	27 au 30	17 au 20	15 au 18	12 au 15	10 au 13

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA12

DURÉE

3 jours

OBJECTIFS

Etre capable de développer une application web en intégrant une interface utilisateur basée sur la technologie standard JSF (Java Server Faces) intégrée à Java EE

Assimiler les concepts de JSF

Mettre en oeuvre les composants graphiques JSF dans une page web

Savoir connecter les éléments graphiques JSF aux données

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Avoir suivi la formation JA1 « Java : les bonnes pratiques » ou avoir l'expérience équivalente.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Développer des applications avec JSF

Programme

La technologie JSF

- Présentation, les avantages de JSF
- Les éléments d'une application JSF
- Découverte du fonctionnement à travers une application de type « Hello world »
- Les étapes du développement
- Création des pages web
- Définition de la navigation
- Développement des Java Bean métier
- Ajout du gestionnaire de Java Bean
- Le modèle de composant de l'interface utilisateur
- Concept de « renderer »
- Le modèle évènementiel
- Validation de données
- Le modèle de navigation
- Gestion des « Backing Bean »
- Vue d'ensemble et intégrations des éléments JSF
- Cycle de vie d'une page JSF

Mise en oeuvre des pages JSP

- Configuration de la page
- La bibliothèque des balises de base (JSF Core Tags)
- Les balises de composants HTML
- Les composants : UIForm, UIColumn, UICommand, UIData, UIGraphic, UIInput et UIOutput, UIPanel, UISelectBoolean, UISelectMany, UIMessage, UISelectOne, UISelectItem(s)
- La gestion de l'internationalisation
- Les conversions de données
- Les évènements générés lors d'un changement sur un composant, gestion des listeners

- Les validateurs de données
- Connexion des composants à des sources de données
- Référence à une méthode de Backing Bean

Création de composants personnalisés d'interface utilisateur

- Choix entre un composant personnalisé et un « renderer »
- Utilisation d'un « Image Map »
- Les étapes pour créer un composant personnalisé
- Création du gestionnaire de balise du composant
- Définition de la balise du composant dans une bibliothèque de description de balises
- Création des classes du composant
- Gestion des évènements pour le composant

Configuration des applications JSF

- Le fichier de configuration de ressource : faces-config.xml
- Configuration des beans
- Gestion des messages personnalisés
- Utilisation d'un validateur personnalisé
- Configuration des règles de navigation
- Enregistrement d'un renderer personnalisé
- Enregistrement d'un composant personnalisé
- Configuration logicielle requise

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	12 au 14		02 au 04			9 au 11		10 au 12		5 au 7	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

JA13

DUREE

1 jour

OBJECTIFS

Assimiler les nouveautés de Java 6

Mettre en oeuvre JDBC 4

Savoir utiliser les nouvelles API XML StAX

PUBLIC

Concepteurs/développeurs,

Chefs de projets,

Equipe projet,

Architectes technique.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

400 € H.T par personne

Mustang les Apports de Java 6

Programme

Utilisation de langages de script

- Présentation du framework
- Appel de scripts depuis une application Java
- Appel d'une application Java depuis un script
- Rendre les objets Java utilisables depuis les scripts
- Exemple : invocation de l'interpréteur JavaScript

Utilisation d'un compilateur Java depuis une application

- Présentation de l'API
- Invoquer un compilateur Java dans une application Java
- Récupérer toutes les informations retournées par le compilateur

Pluggable Annotation-Processing API

- Présentation de l'API
- Traitement des annotations lors de la compilation des codes sources

Common Annotations

- Les nouvelles annotations : @Generated, @PostConstruct, @PreDestroy, @Resource

Le nouveau format des fichiers .class

- L'architecture « split verification »
- Amélioration des performances, gain en espace utilisé et temps de chargement

JDBC 4.0

- Chargement automatique des drivers JDBC
- Utilisation d'annotations
- Manipuler les tables facilement via un mapping object/relationnel
- Le type SQL ROWID
- Détection des connexions devenues invalides
- La nouvelle hiérarchie des exceptions SQL

XML et Web Services

- L'API : Streaming API for XML (StAX)
- Description des web services avec « Web-Services Metadata »

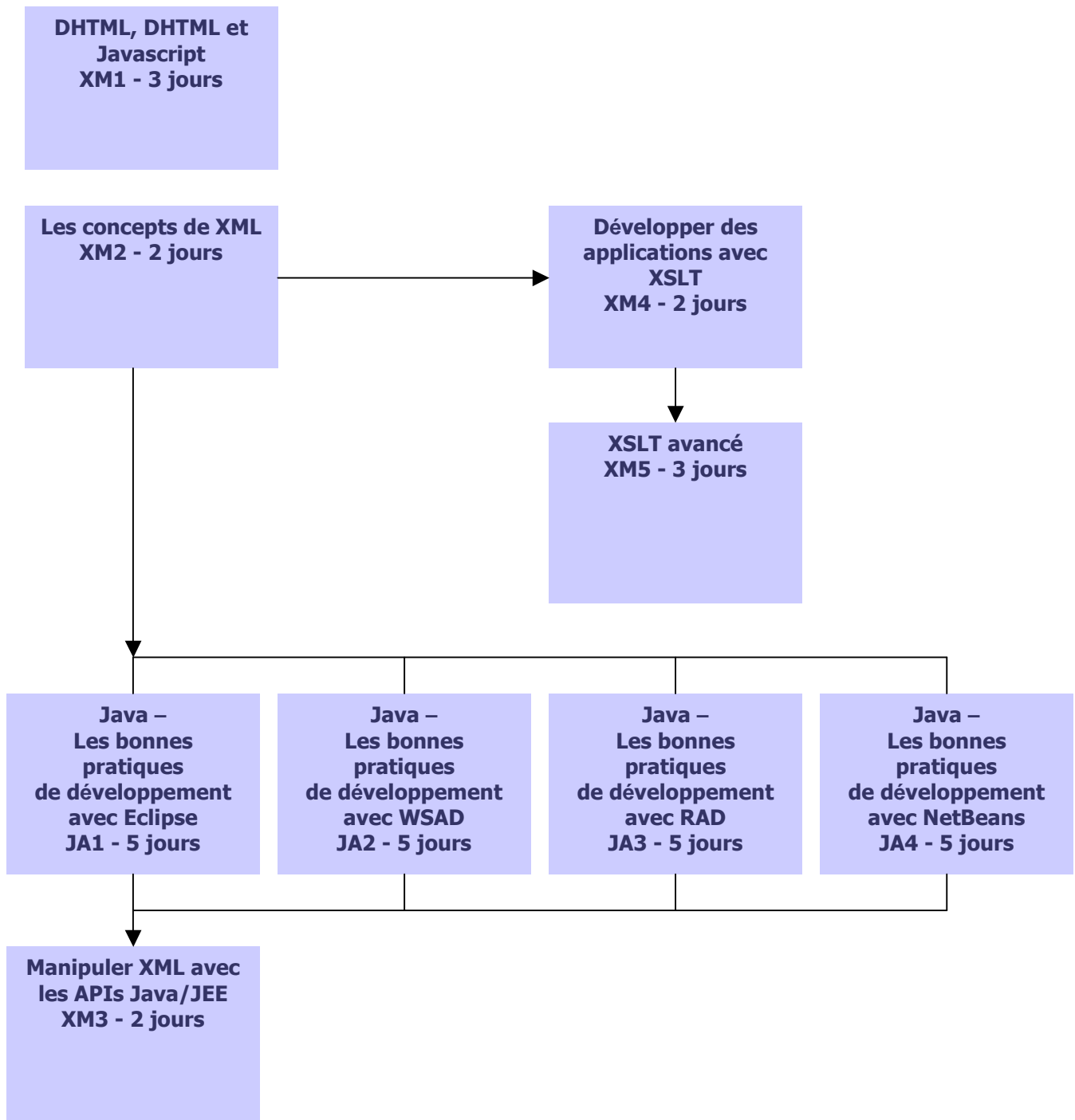
Les autres nouveautés

- Les améliorations de AWT et Swing
- Les nouvelles fonctionnalités de l'API Collection
- La JVMTI (JVM Tool Interface)

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	7		4		3		8		3		6

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter



HTML, DHTML et JavaScript

CODE

XM1

DUREE

3 jours

OBJECTIFS

Connaître les composants d'une application web

Etre capable de développer des documents XHTML. Explorer l'ensemble des possibilités offertes par XHTML.

Maîtriser la syntaxe du langage Javascript

Savoir intégrer HTML et JavaScript pour réaliser des menus déroulants, des animations, modifier dynamiquement les styles

Savoir gérer les événements utilisateurs

PUBLIC

Concepteurs/développeurs,

Chefs de projets,

Equipe projet,

Architectes technique

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Programme

Architecture web

- Architecture fonctionnelle : composants urbanisés
- Architecture applicative : les 5 tiers logiques
- Les composants d'une application web
- Les protocoles de base : TCP/IP, HTTP
- Concept de client/serveur HTTP
- Les serveurs d'applications web
- Les technologies : HTML, XHTML, CSS, Javascript, Flash, Java...

XHTML

- Portée du langage
- Le concept de balise
- Eléments syntaxiques
- Mise en forme du document
- Mise en forme du texte
- Les listes
- Les tableaux
- Les liens hyper média
- Insertions d'images
- Insertions d'éléments multimédia Flash, Real, Quicktime, Pdf et d'applets Java
- Les frames
- Les formulaires
- Les meta-informations
- Impact de Javascript sur l'accessibilité.
- Importance de XHTML sur l'utilisation dynamique de Javascript.
- Méthode de développement Javascript en équipe.
- Les outils de développement.
- Positionnement et illustration des technologies DHTML.

Javascript et DHTML

- Les limites du langage HTML
- Intégrer les scripts Javascript dans une page web
- Connaître les variables, constantes et opérateurs
- Connaître les fonctions les plus utiles sur des nombres, des chaînes de caractères
- Utiliser des structures de contrôle
- Réutiliser du code grâce aux fonctions et méthodes
- Dynamiser une page web
- Gérer les actions de l'utilisateur
- Préremplir et contrôler la saisie d'un formulaire
- Manipuler les objets prédéfinis de Javascript
- Manipuler les fenêtres et documents (objets window, document, frame)
- Créer des popups
- Créer des rollover
- Gérer la navigation entre les pages web
- Personnaliser une page Web avec des feuilles de style CSS
- Placer et manipuler les feuilles de styles CSS
- Animer des textes et des images dans la page : le DHTML
- Etudier les balises de calques (div, span, layer, nlayer) suivant le navigateur
- Animer des textes et des images dans la page

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 17		12 au 14		21 au 23		9 au 11		10 au 12		5 au 7	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

XM2

DUREE

2 jours

OBJECTIFS

Connaître et maîtriser le langage XML

Comprendre les intérêts de l'indépendance de la présentation et des données

Acquérir la terminologie XML

Utiliser un outil comme Altova XML Spy

PUBLIC

Concepteurs/développeurs

PRE-REQUIS

Connaissance d'un langage de programmation

TARIF D'INSCRIPTION

800 € H.T. par personne

Les concepts d'XML

Programme

Introduction

- Historique
- Les principaux domaines d'application d'XML
- Rappels succincts d'HTML
- Intérêts d'XML
- Comparaison avec HTML : différenciation données et présentation
- Les concepts de base
- Exemples de documents XML

Etude de cas complète

- Description et objectifs
- Illustration de tous les concepts abordés

Syntaxe

- Caractères, encodage
- Eléments, attributs
- Comment choisir entre éléments et attributs
- Définition d'un document bien formé
- Concept de grammaire
- Définition et syntaxe des DTD
- Exemples de DTD
- Définition et éléments de syntaxe de Schéma XML (XSD)
- Exemples de XSD
- Différences entre DTD et XSD
- Définition d'un document valide

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	15 au 16		05 au 06			12 au 13		13 au 14		8 au 9	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

XM3

DUREE

2 jours

OBJECTIFS

Assimiler les principales techniques de développement Java pour le traitement des documents XML

Savoir manipuler XML avec les analyseurs SAX et DOM

Acquérir les bonnes pratiques

Savoir implémenter du code indépendant de l'analyseur avec l'API JAXP

PUBLIC

Concepteurs/développeurs

PRE-REQUIS

Avoir suivi la formation XM2 « Les concepts d'XML » ou posséder des connaissances équivalentes.

TARIF D'INSCRIPTION

800 € H.T. par personne

Manipuler XML en Java

Programme

Analyse d'un document XML

Motivation

Définition d'un analyseur XML

Présentation des standards DOM (Document Object Model), SAX (Simple API for XML) et JDOM (Java DOM)

Transformation avec les APIs SAX

- Motivation
- Le modèle des événements
- Les interfaces Handler
- Construction d'un modèle objet s'appuyant sur SAX
- Comparaison avec DOM
- Implantations commerciales
- Exemples avec les standards Open Source Apache (Xerces)

Transformation avec les APIs DOM et JDOM

- Motivation
- Le modèle mémoire de document
- Les interfaces du modèle
- Les méthodes d'accès au document
- Chargement, modification, sauvegarde du document
- Limites du modèle

Etude de cas complète

- Description et objectifs
- Illustration de tous les concepts abordés

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	15 au 16		05 au 06		14 au 15		23 au 24		11 au 12		6 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

XM4

DUREE

2 jours

OBJECTIFS

Maîtriser le langage XSLT et XPath

Comprendre les enjeux des transformations XSLT

Savoir réaliser des feuilles de style XSLT

PUBLIC

Concepteurs/développeurs

PRE-REQUIS

Avoir suivi la formation XM2 « Les concepts d'XML » ou posséder des connaissances équivalentes.

TARIF D'INSCRIPTION

800 € H.T. par personne

Développer des applications avec XSLT

Programme

Présentation

- Objectifs
- Rappels sur CSS
- Limitations de CSS
- Présentation des concepts de base
- Présentation du langage de navigation : XPath
- Analyseur XML et processeur XSLT
- Intégration dans les navigateurs Web.
- Mise en oeuvre de XSLT

Recherche avec XPath

- Syntaxe et expressions XPath
- Se déplacer dans un document XML
- Concepts de recherche
- Requêtes
- Critères de recherche : éléments/nœuds, attributs
- Les opérateur Xpath
- Les fonctions standard

Transformation de documents XML avec XSLT

- Concepts fondamentaux
- Organisation d'une transformation XSLT
- Fonctionnement du processeur XSLT.
- La racine de l'arbre et la gestion des namespaces
- Sortie formatée en XML ou HTML
- Exemples d'expressions XPath.
- Manipulations de nœuds et ensemble de nœuds.
- Les templates.
- La syntaxe XSLT
- Les types de données
- Les structures de contrôle : tests, itérations
- Les variables
- Méthode pour créer des éléments et des attributs.
- Différence entre langage impératif et déclaratif
- Traitement impératif et déclaratif
- Récupérer des éléments et leurs contenus
- Méthode d'optimisation des traitements
- Gérer les erreurs.

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 9		5 au 7		16 au 18	21 au 23	25 au 27		24 au 26			7 au 9

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

XSLT avancé

CODE

XM5

DUREE

3 jours

OBJECTIFS

Maîtriser les concepts avancés du langage XSLT

Savoir générer des documents au format PDF avec le langage XSL FO

PUBLIC

Concepteurs/Développeurs

PRE-REQUIS

Avoir suivi la formation XM4 « Développer des applications avec XSLT » ou posséder des connaissances équivalentes.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Programme

Concepts avancés

- Expressions, tris, exceptions avec les version 2 de XSLT
- Les types supportés
- Créer des fonctions
- Récursivité des fonctions
- Les fonctions sur les chaînes de caractères, les numériques et les dates
- Organisation structurée des feuilles de style
- Les bonnes pratiques de conception des feuilles
- Importer des feuilles de style
- Transmettre des arguments entre feuilles de style
- Créer et réutiliser des bibliothèques
- Notions objet, héritage
- Utiliser des fichiers multiples
- Gérer les espaces de noms
- Utiliser des expressions régulières

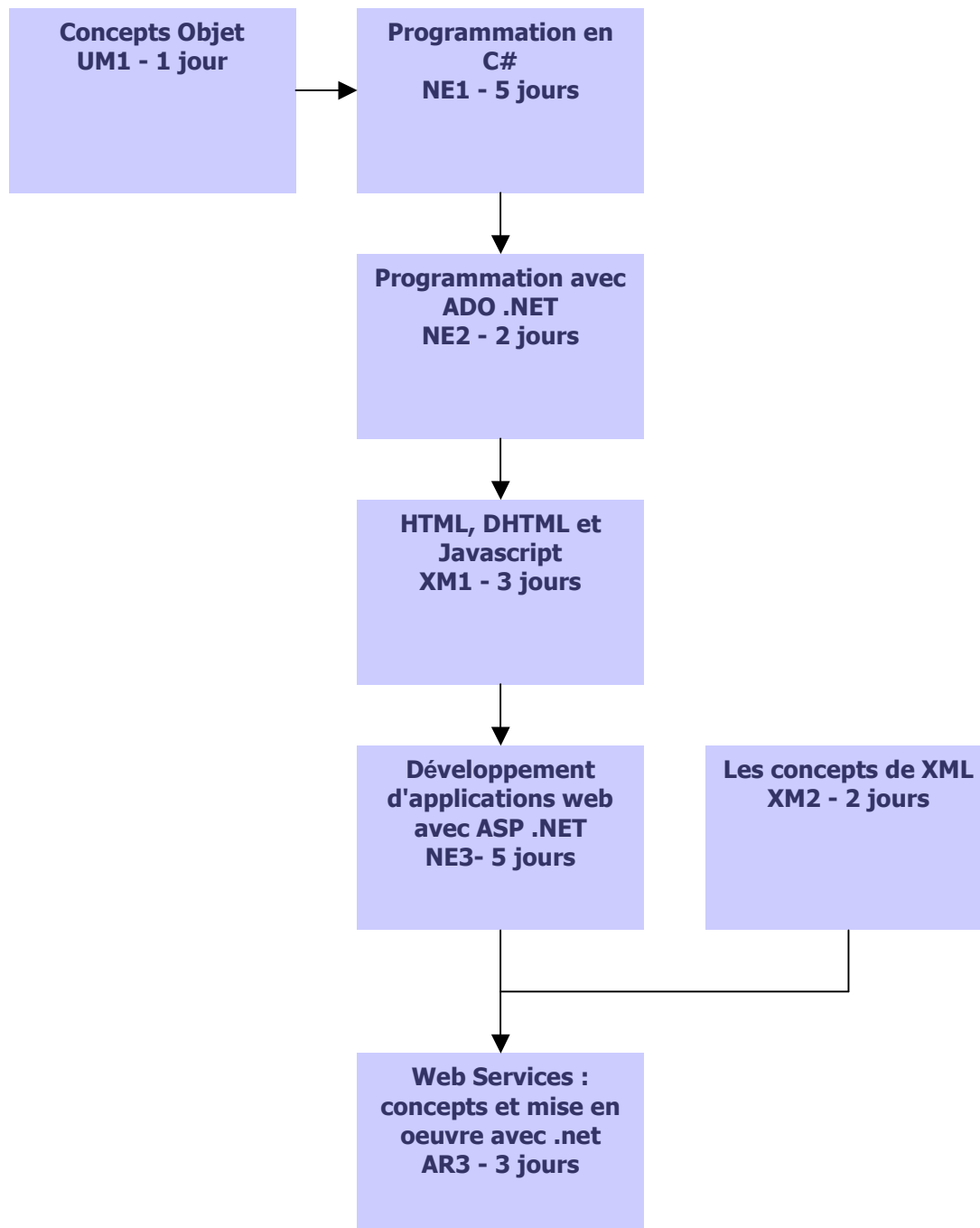
Transformer du XML en PDF avec XSLFO

- Présentation de FO (Formatting Objects)
- Objectifs
- Syntaxe de base du langage.
- Outils et implémentations
- Exemple de FOP d'Apache
- Définir des modèles de pages
- Les zones de contenu

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
10 au 12		8 au 9		19 au 20	24 au 25	28 au 29		27 au 28			10 au 11

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter



CODE

NE1

DUREE

5 jours

OBJECTIFS

Maîtriser la syntaxe du langage C#

Comprendre les concepts objets

Savoir appliquer les bonnes pratiques de programmation

Savoir utiliser l'IDE Visual Studio pour mettre au point un programme C#

PUBLIC

Concepteurs/développeurs,

Chefs de projets,

Equipe projet,

Architectes technique.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

2 000 € H.T. par personne

Programmation en C#

Programme

Plate-forme Microsoft .NET et Langage C#

- Introduction
- Panorama du framework .NET
- Les composants de .NET
- Les langages de .NET

Les concepts du langage C#

- Organisation d'un programme C#
- Compiler, exécuter et mettre au point un programme

Les variables et les types

- Les types de données
- Le Common Type System
- Les opérateurs sur les types primitifs
- Les types définis par le programmeur
- Conversion de types

Les instructions de contrôle

- Les tests
- Les boucles
- Les instructions de saut
- La gestion des exceptions

Les méthodes et transmission par paramètres

- Syntaxe des méthodes
- Transmission par paramètres
- Surcharge de méthode

Les tableaux

- Définition
- Création et manipulation de tableaux
- Les tableaux à n dimensions

Les concepts de la POO (Programmation Orientée Objet)

- Classe et objet
- Instanciation
- Encapsulation
- Héritage
- Polymorphisme

Les variables références

- Définition
- Manipulation de références
- Classification des objets
- Les espaces de noms
- Les classes d 'usage courant (String, StreamReader, StreamWriter, Regex)

Cycle de vie des objets

- Constructeur
- Garbage collector

L'héritage avec C#

- Super classe et sous classe
- Redéfinition de méthode
- Les interfaces
- Abstraction de classe et de méthode

Organisation et Namespace

- Encapsulation
- L'agrégation
- Les namespace
- Les assemblies et les modules

Programmation événementielle

- Les opérateurs
- Surcharge d'opérateurs
- Les délégués
- Création et utilisation d'évènements

Les propriétés et indexeurs

- Définition
- Utilisation
- Les attributs

Programmation avancée

- Les collections
- La sérialisation
- Les interfaces graphiques
- Les threads
- Introduction à la programmation réseau (TCP/IP)

Calendrier interentreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 19	12 au 16	12 au 16	2 au 6	21 au 2	11 au 15	9 au 13	20 au 24	10 au 14	8 au 12	5 au 9	3 au 7

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

NE2

DUREE

2 jours

OBJECTIFS

Maîtriser les accès aux SGBDR avec les APIs ADO .NET et C#

Optimiser les performances d'accès aux bases de données

PUBLIC

Concepteurs/développeurs,

Chefs de projets,

Equipe projet,

Architectes technique.

PRE-REQUIS

Connaissances équivalentes au NE1 - Programmation en C#

TARIF D'INSCRIPTION

800 € H.T. par personne

Programmation avec ADO .NET

.NET est l'architecture objet et n tiers de Microsoft. La couche d'accès aux données se fait par le framework ADO .NET que ce soit un SGBDR ou du XML.

L'outil Visual Studio permet d'intégrer très facilement ce framework dans toute application .NET et fera gagner aux développeurs beaucoup de temps.

Programme

Présentation de ADO .NET**Les connexions**

- Les fournisseurs d'accès aux données
- Les chaînes de connexion
- La gestion des connexions
- Les pools de connexion
- Les évènements

L'accès aux données

- Le mode connecté
- Les commandes
- Les paramètres
- Data Definition Language
- Data Manipulation Language
- Les transactions
- La gestion des exceptions

Les DataSets

- Le mode déconnecté
- L'objet DataSet
- Les contraintes
- L'auto-incrémentation
- Les champs calculés
- Les relations
- Ajout/Suppression/Modification de données

Recherches, tris et filtres Travailler avec une source de données

- L'objet DataAdapter
- La mise à jour des données

ADO .NET et XML

- Les schémas XSD
- Lecture d'un document XML
- Génération d'un document XML
- L'objet XmlDataDocument
- La liaison des données

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	15 au 16		05 au 06			12 au 13		13 au 14		8 au 9	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

NE3

DUREE

5 jours

OBJECTIFS

Savoir développer une application web avec .NET et C#

Optimiser les performances d'accès aux serveur web

Connaître les bonnes pratiques d'architecture

PUBLIC

Concepteurs/développeurs,

Chefs de projets,

Equipe projet,

Architectes technique.

PRE-REQUIS

Connaissances équivalentes au NE1 - Programmation en C#

TARIF D'INSCRIPTION

2 000 € H.T. par personne

Développement d'applications web avec ASP .NET

ASP .NET est le framework de développement web de la plate-forme .NET.

Grace à l'IDE très puissant Visual Studio et ASP .NET, le développeur peut concevoir un site web très rapidement.

Ce module met l'accent sur la présentation, la visualisation, le contrôle, la navigation ainsi que sur la mise au point et l'optimisation des performances d'une application web.

Programme

Introduction à ASP .Net

- Architecture
- Présentation d'ASP.NET
- Les ressources

L'environnement de développement Visual Studio .NET

- Présentation
- Création d'un projet d'application web ASP.NET
- Création d'un composant en C#

Gestion d'un formulaire web

- Création de formulaires web
- Utilisation des contrôles de serveur

Les traitements liés à un formulaire web

- Les pages code-behind
- Les procédures d'événement aux contrôles de serveur web
- Utilisation des événements de page

Mise au point des applications ASP.NET

- Suivre pas à pas le déroulement d'une application web
- Débogage à distance

La validation des saisies utilisateur

- Présentation
- Les contrôles de validation
- Valider la page

Personnalisation des contrôles

- Créer un contrôle
- Ajout de contrôles utilisateurs à un formulaire web

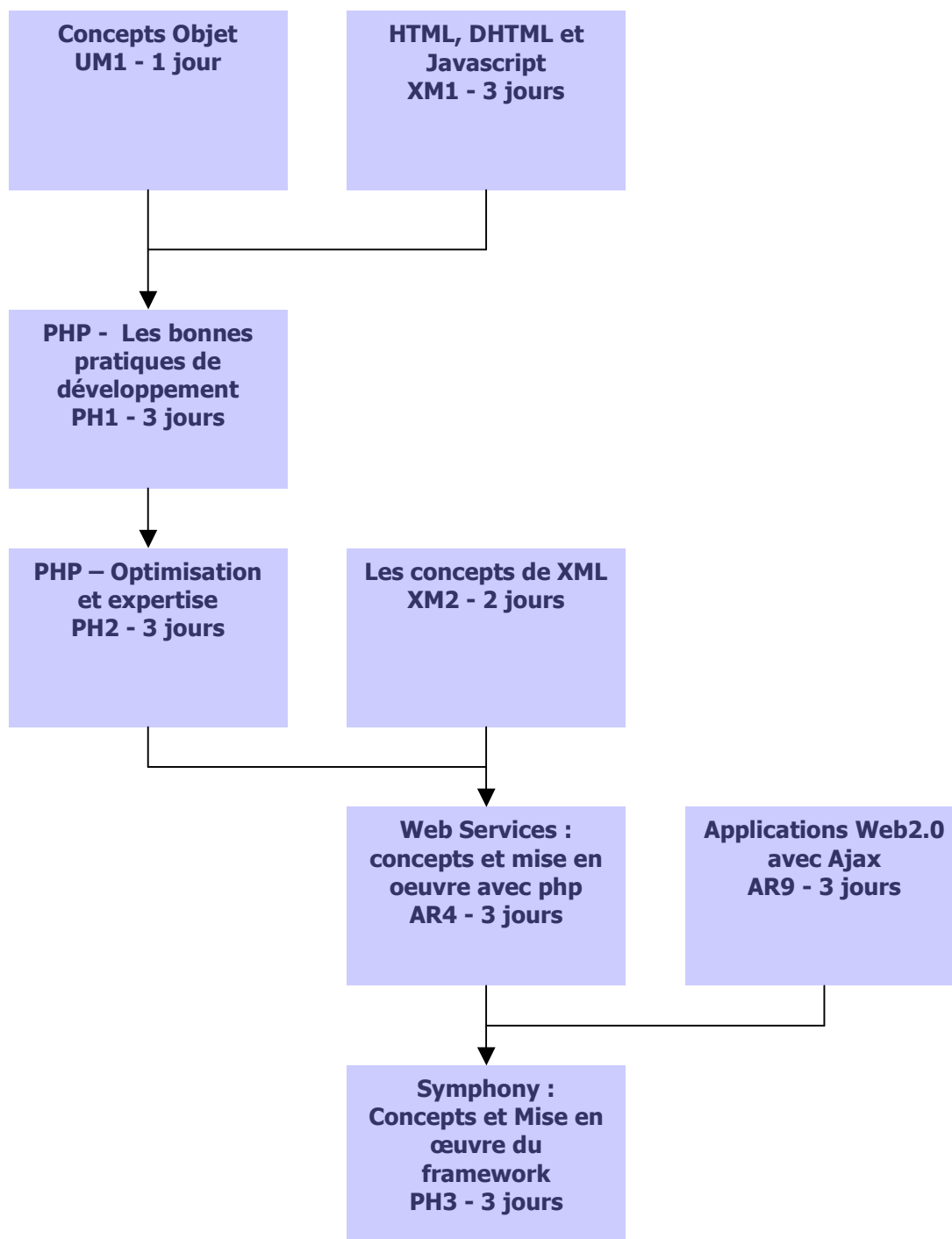
Notions avancées

- Les DataGrids
- L'accès aux données
-

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 12		5 au 9		16 au 20	21 au 25	25 au 29		24 au 28			7 au 11

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter



CODE

PH1

DUREE

3 jours

Objectifs

Être capable de développer une application en PHP

Mettre en oeuvre une connexion à une base de données, exécuter des requêtes et récupérer les résultats

Faire le lien entre la modélisation et l'implémentation en PHP

Savoir traduire un modèle de conception UML 2.0 en langage PHP

Savoir travailler en équipe projet, avec les outils de cycle de vie, de gestion de version et d'accès concurrents

PUBLIC

Concepteurs/développeurs,

Chefs de projets.

PRE-REQUIS

Connaissance d'un langage de programmation.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

PHP - Les bonnes pratiques de développement

Programme

Les Objets et leurs modèles

- Syntaxe du langage
- Organisation d'une page PHP/HTML
- Les types et les variables
- Les chaînes de caractères
- Les opérateurs
- Les structures de contrôle
- Les fonctions
- Les références
- Les tableaux
- Les expressions régulières
- Les instructions include et require
- Classes et instances
- Encapsulation
- La communication avec les Objets
- Travaux pratiques : définir et utiliser des classes en PHP 5

Spécialisation d'une classe

- L'héritage
- Travaux pratiques : utiliser l'héritage en PHP 5

Le polymorphisme

- Définition
- Travaux pratiques : exemple de polymorphisme en PHP 5

Principes de modélisation avec UML 2.0

- Aspect statique : diagramme de classe
- Aspect dynamique : diagramme de séquence
- Les bonnes pratiques de la programmation objet
- Notion de Design Pattern

PHP et le Web

- Gérer les fichiers
- Recevoir et envoyer des mès
- Récupérer les informations du formulaire
- Les méthodes POST et GET
- Construction de l'interface utilisateur
- Gestion des codes et pages d'erreurs

La gestion des sessions

- Fonctionnement des sessions
- Débuter une session et assigner des variables
- Suppression des variables de session
- Destruction d'une session

Les entrées / sorties en PHP

- Php - cli : command line interface
- Utilisation du buffer et compression des données

PHP et système de fichiers

- La sécurité sur les fichiers
- Les fonctions de manipulation de fichiers

Connectivité avec les SGBDR

- Les principaux SGBDR
- Les principaux objets de connectivité
- Transmettre des requêtes SQL
- Récupérer et traiter les résultats

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
8 au 10		5 au 7		2 au 4		2 au 4		3 au 5		5 au 7	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

PH2

DUREE

3 jours

OBJECTIFS

Être capable de faire du tuning de pages

Savoir manipuler du XML avec PHP

Savoir intégrer des services web avec PHP.

PUBLIC

Concepteurs/développeurs,

Chefs de projets,

Equipe projet,

Architectes technique.

PRE-REQUIS

Connaissances équivalentes à la formation PH1.

TARIF D'INSCRIPTION

1 200 € H.T. par personne

PHP - Optimisation et expertise

Programme

Manipulation de l'API pour étendre le langage

- L'API PHP
- Possibilités d'extension
- Arborescence des sources
- Création et utilisation d'extensions
- Macros de configuration

Gestion des images

- Présentation des librairies.
- Création d'images
- Gestion des polices et de l'écriture en mode image.
- Superposition de texte pour protection de droits.
- Intégration au site.
- Réalisation de graphiques statistiques avec JGraph
- Accessibilité, malvoyants
- Intégration des modules réalisés

Tuning de pages

- Intérêt des templates
- Utiliser les Templates
- Séparer les scripts PHP et le code HTML
- Améliorer les performances du serveur en utilisant des systèmes de cache
- Les outils de cache
- Les caches globaux
- Les caches de données utilisateur
- Les caches HTTP
- Les caches de page, d'Opcache, les optimiseurs
- Etude de Jpcache, de TurckMMCache

Intégrer entre différents systèmes grâce à XML

- Le langage XML
- Manipulation du XML
- Les différentes librairies XML sous PHP
- Lecture et requêtes avec SimpleXML
- Analyse de document XML avec SAX
- Créer et manipuler les fichiers XML grâce aux librairies DOM
- Effectuer des transformations XSLT
- Utiliser des Web Services et la technologie WDDX
- Les Web services (SAX, DOM, et principes de SOAP, WSDL, ...)

Concepts avancés

- Générer des fichiers PDF
- LDAP : accès aux annuaires et gestion des droits
- Librairies et sources publiques

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
	5 au 7		2 au 4		4 au 6		1 au 3		1 au 3	28 au 30	

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

CODE

PH3

DUREE

3 jours

OBJECTIFS

Maîtriser l'utilisation d'un framework orienté objet en PHP.

Maîtriser l'architecture en couches : Partie cliente Web 2.0 avec Ajax, MVC, mapping objet relationnel

Maîtriser les outils de développement et de déploiement

PUBLIC

Concepteurs/développeurs, Chefs de projets, équipe projet, architectes technique.

PRE-REQUIS

Concepts Objets (UM1) ;

Maîtrise de PHP (PH1) ;

HTML et le web (XM1) ;

XML (bases, XM2)

TARIF D'INSCRIPTION

1 200 € H.T. par personne

Symfony : Concepts et Mise en œuvre du framework

Symfony est un framework open-source orienté objet pour PHP5. Symfony propose une architecture de composants découpée en couches : séparation Modèle-Vue-Contrôleur, facilités Web2.0 avec Ajax, Mapping objet-relationnel avec cache mémoire. Symfony arrive également avec des outils d'aide au développement et au déploiement : génération complète d'un site d'administration, debug, tests unitaires, déploiement automatisé, etc.

Yahoo a récemment choisi ce framework pour la refonte de son portail et décidé de reverser ses développements à la communauté, conférant d'emblée une légitimité et une pérennité à cette solution.

Programme

Présentation des concepts

- Concepts objets et framework : rappels
- Architecture applicative : pattern MVC, mapping objet-relationnel, Web2.0
- Internationalisation

Mise en oeuvre

- Installation, configuration
- Présentation des outils d'aide au développement
- Modèle de déploiement Symfony

Couche présentation

- Modèle MVC (séparation présentation / services)
- Pattern Action et Contrôleur en pratique
- Architecture de services, XML et Web2.0 : concepts
- Moteur de « template » Symfony
- Intégration Ajax avec Symfony
- Internationalisation (i18n)
- Modèle de sécurité applicative

Couche Services, mapping avec la base de données

- Mapping Objet-Relationnel, concepts
- Notion de DAO, Services
- MDA : concepts
- Utilisation de l' « admin generator » de Symfony, orientation MDA simplifiée

Outils Symfony

- Outils de debug
- Outil de gestion des projets
- Génération d'un site d'administration CRUD
- Outils de déploiement : déployer son projet en production

Cycle de développement

- Travail en équipe : rappels
- Développement guidé par les tests (TDD) avec Symfony

Etendre Symfony

- Extensions spécifiques
- Développement de plug-ins

Calendrier inter-entreprises 2007

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
15 au 17		12 au 14		21 au 23	11 au 13	20 au 22			8 au 10		3 au 5

Organisation en formule intra-entreprise ou « sur-mesure » dans nos locaux ou sur site, nous consulter

A			
Administration.....	51, 52	Maven	37
ADO .NET.....	75	MDA	30, 31, 32, 33, 34
Ajax.....	49	Modélisation métier.....	11
AndroMDA.....	32	N	
Ant	37	NetBeans	57
Applications Web.....	61, 62, 63, 64	O	
ASP .NET	76	oAW	33
B		P	
Borland Together.....	31	Php	78, 79
BPMN	12	PHP.....	44
C		Processus	26, 27
C#	74	Processus métier	12
Cactus	38	Programmation Java.....	54, 55, 56, 57
Changement.....	23	Programmation Java/JEE	58
Client riches.....	59, 60	R	
Composants	17	RAD.....	56
Concepts Objet.....	9	Réseau.....	25
Cruise Control.....	37	RSA.....	34
D		S	
Design Patterns.....	16	SOA.....	41
DHTML	68	Spring.....	47
DotNET	43	Struts.....	45
E		Subversion	35
Eclipse	54	Subversion avancé	36
Eclipse RCP	60	Swing	59
EJB 3	48	Symfony	80
Equipe	24	Synthèse	15
Etude de cas.....	12, 13, 14, 19	SysML	20
Exigences	13	T	
H		Tapestry	50
Hibernate	46	Tests fonctionnels.....	39
HTML.....	68	Tests unitaires.....	38
I		U	
Intégration Continue	37	UML 2.....	9, 11, 10, 13, 14, 15, 19
J		UP	14, 18, 19
Java	42, 70	Urbanisation.....	21
Java 6	66	Usine logicielle.....	29
Java EE	17, 19	W	
Java EE et JBoss	63	Web Services en .Net	43
Java EE et JOnAS.....	64	Web Services en Java	42
Java EE et Tomcat.....	61	Web Services en Php.....	44
Java EE et WebSphere	62	Web 2.0.....	49
JavaScript.....	68	WSAD	55
JBoss.....	51	X	
JMeter.....	39	XML.....	69, 70
JOnAS	52	XSLT	71, 72
JSF	65		
JUnit	38		

Inscription

Des options peuvent être prises par téléphone au **0 810 818 424**

L'inscription est ferme et définitive uniquement après réception d'une confirmation écrite :

- par fax, au **0 810 818 425**
- par E-mail : formation@oxiane.com
- par courrier : OXIANE - 98, avenue du Général Leclerc – 92100 Boulogne-Billancourt

Horaires

Les formations ont lieu de 9h30 à 17h30

Validation de l'inscription

OXIANE retourne au client, une convocation de stage comportant toutes les indications pratiques sur la formation (horaires, programme, plan d'accès ...) environ dix jours avant le début de la session.

Conditions de participation

La société OXIANE pourra confier la conduite des actions de formation précisées à un autre organisme de formation avec lequel elle aura passé un contrat de sous-traitance.

Le stagiaire est placé par son entreprise et pour la durée de la formation en situation de salarié exécutant son contrat de travail. Il se conformera au règlement intérieur de l'organisme de formation.

Annulation

OXIANE se réserve le droit de reporter une session dans le cas d'un nombre insuffisant de participants.

Le client en est informé, au plus tard, une semaine avant la date prévue du stage et peut alors, soit choisir une autre date dans le calendrier des stages, soit annuler sa demande d'inscription.

Une annulation d'inscription doit être effectuée par courrier ou télécopie.

- Toute inscription annulée moins de deux semaines avant le début du stage fera l'objet d'une facturation de 30 % du montant de la formation.
- Toute inscription annulée moins d'une semaine avant le début du stage fera l'objet d'une facturation de 50 % du montant de la formation.
- Toute défection pendant le stage lui-même donnera lieu à une facturation intégrale.

OXIANE se réserve le droit de modifier les dates et prix des formations ainsi que les contenus des stages.

Prix et paiement

Tout stage ou cycle de formation commencé est dû dans son intégralité.

Nos prix s'entendent Hors Taxes et sont à majorer de la TVA ou autres taxes légales en vigueur au jour de la facturation.

Les repas et les supports de cours sont compris dans le prix du stage pour les sessions inter-entreprises.

Nos tarifs sont indiqués hors taxes et sont établis au départ de Paris.

Le coût des prestations se règle à réception de facture, tenant lieu de convention simplifiée de formation.

Celle-ci est accompagnée des attestations de présence.

Les tarifs n'incluent pas les frais de transport, repas, hébergement des participants.

Confidentialité

Les manuels et supports de cours font l'objet de la protection prévue par la Loi du 11 mars 1957.

Toute reproduction, modification ou divulgation à des tiers de tout ou partie de ces stages et/ou documents, sous quelque forme que ce soit, est interdite sans autorisation écrite d'OXIANE.

Les informations et le savoir-faire dispensés pendant les formations sont destinées à l'usage exclusif de l'auditeur.

Le Client s'engage à ne pas débaucher le personnel d'OXIANE avec qui il aurait eu contact à l'occasion d'un stage, et ce durant une période de 24 mois après le dernier stage suivi.

En cas de non respect de cette clause, le Client s'engage à dédommager OXIANE en lui versant une somme forfaitaire égale à la rémunération brute d'une année de ce collaborateur.

Droit applicable

Attribution de compétence

Les présentes Conditions Générales sont régies par la Loi française.

Tout litige né de l'interprétation ou de l'exécution des Conditions Générales de vente sera, sauf accord amiable, soumis à la seule compétence du tribunal de Commerce de Paris.

Merci de retourner ce bulletin d'inscription dûment complété par courrier ou par fax aux coordonnées suivantes :

OXIANE – Service Formation – 98, avenue du Général Leclerc – 92100 Boulogne-Billancourt

Renseignements par téléphone au : **0 810 818 424** – Fax : **0 810 818 425** - formation@oxiane.com

RCS Paris B 430112250 – Code APE : 722 C - Organisme de formation professionnelle n°11753388475

Dès réception, nous vous ferons parvenir, sous huitaine, une confirmation d'inscription.

Formation demandée :

Référence, module(s) :

Dates de session :

Tarif d'inscription :

€HT

Participant (s):

Mme/Mlle/M. - Prénom :

Nom : _____

Tél :

Mail : _____

Mme/Mlle/M. - Prénom :

Nom : _____

Tél :

Mail : _____

Mme/Mlle/M. - Prénom :

Nom : _____

Tél :

Mail : _____

Mme/Mlle/M. - Prénom :

Nom : _____

Tél :

Mail : _____

Responsable de l'inscription :

Mme-Mlle-M :

Raison sociale de l'entreprise :

Adresse :

N°de SIRET :

N°de TVA Intra Communautaire :

Code postal :

Ville :

Téléphone :

Fax :

Mail :

Adresse d'envoi des convocations si différente :

Facturation :

La facture est à adresser : à un OPCA (Fafiec...) à l'entreprise autre :

Si le destinataire n'est pas l'entreprise cliente, préciser :

Raison sociale :

Adresse :

Prénom et Nom du responsable du suivi administratif et financier :

Téléphone :

Fax :

Le responsable du participant reconnaît avoir pris connaissance et accepté les conditions générales de ventes. Ce formulaire complété est un Bon de commande.

Date :

Signature :

Cachet de la Société :

