Technologies Objet, synthèse des modèles aux architectures

★★★★★

MR-68 **3 Jours (21 Heures)**

Description

Ce séminaire vous propose une synthèse claire et précise de l'ensemble des technologies Objet. Après une présentation des principes, il vous donne un premier aperçu de la méthode de conception UML et vous montre comment l'appliquer avec succès. Il vous présente les caractéristiques des langages Objet et de leurs environnements. Enfin, avec l'analyse des architectures à objets JEE et .NET, il vous permet d'appréhender les différentes approches en matière de développement de systèmes distribués et d'objets métier.

À qui s'adresse cette formation ?

Pour qui

Responsables informatiques, ingénieurs d'études et ingénieurs système souhaitant appréhender d'une manière claire les concepts Objet et la manière de les utiliser.

Prérequis

Aucun

Les objectifs de la formation

- Comprendre les concepts de base de l'appoche objet
- Découvrir le formalisme UML de représentation des objets
- Connaître les caractéristiques des principaux langages objets
- Identifier et comparer les principales plateformes objet du marché
- Appréhender les différentes approches en matière de développement de systèmes distribués et d'objets métier

Programme de la formation

L'émergence de l'approche Objet

- Les problèmes sur les projets de développement.
- L'émergence des concepts Objet et leur impact.
- Les qualités attendues d'un développement Objet.

Les concepts de base

- Les ressemblances et différences entre l'objet au sens commun et l'objet informatique.
- Les notions de classes, d'encapsulation, d'héritage, d'abstraction, de polymorphisme.
- Les objets, propriétés, opérations et liaisons.
- La séparation des interfaces et des implémentations.
- Les avantages : extensibilité, réutilisabilité, rapidité de conception, mythe ou réalité ?

Analyse et conception par objets, UML

- Rappels sur le cycle de vie du logiciel.
- L'objet et l'approche itérative.
- La modélisation, le développement, les acteurs et les rôles.
- Historique des méthodes Objet.
- Comparaison.
- Nécessité d'un formalisme universel de représentation des concepts.
- La genèse d'UML.
- Les caractéristiques essentielles.
- La présentation du processus unifié.
- L'analyse des spécifications.
- Les cas d'utilisation.
- Les scénarios et les diagrammes de séquences.
- L'analyse du domaine.
- Diagrammes de classes, d'états-transitions et de collaborations.
- La conception.
- L'algorithmique vue par les diagrammes d'activités.
- La réalisation avec des langages objets.
- L'architecture.
- Diagrammes de composants et de déploiement.
- Une comparaison synthétique entre Merise et UML.

Les principes des modélisations réussies

- La réification ? Pourquoi et quand mettre des informations sous forme d'objets ? Comment traduire des concepts métiers sous forme d'objets ? Les objets comme entités autonomes.
- L'interaction entre objets.
- Les interfaces.
- L'abstraction à partir d'une analyse.
- L'extensibilité et l'adaptabilité des conceptions abstraites.
- La réutilisation.
- La réalisation par les classes concrètes.

L'objet en programmation

- Les grands langages objets.
- Quel langage choisir ? Les caractéristiques fondamentales des langages.
- Comparaison : C++, Java, etc.
- Les approches de ces langages objets.
- L'impact des modes d'exécution.
- Les outils de développement, le marché, les acteurs, les catégories et les tendances.
- Les caractéristiques du langage Java.
- L'intérêt d'une machine virtuelle.
- L'importance des bibliothèques de classes.
- L'organisation d'un projet Java.
- Le " tout Java ".
- De l'intranet à la carte à puce, des téléphones mobiles à la station de travail.
- Stratégies autour de Java.
- Quelle attitude adopter ?

L'organisation de la réutilisation avec les Design Patterns

- Favoriser la réutilisation par l'industrialisation du processus de conception.
- Mise en place de solutions types réutilisables : les Design Patterns.
- Les travaux du GOF (Gang Of Four) et les grandes catégories de Design Patterns.

Objets métiers, frameworks

- Qu'est-ce qu'un framework, comment l'utiliser ? Rapport avec les composants logiciels.
- Les pièges à éviter lors de la conception de frameworks.
- Différences entre Design Patterns et frameworks.

Les clients-serveurs à base d'objets

- Les architectures à base d'objets répartis.
- CORBA, Microsoft COM-DCOM, Java RMI.
- Apports et limites.
- Prise en charge des services techniques afin de tendre vers un assemblage d'objets métiers.

Les objets métiers, serveurs d'applications et architectures n-tiers

- Les limites du 2-tiers en matière de modularité, d'évolutivité et de capacité à accompagner une montée en charge.
- Les apports des architectures multiniveaux.
- L'ouverture sur Internet.
- Sécurité.
- Composants métiers.
- Les offres : JEE, .
- NET, Corba Component Model.
- Le standard JEE.
- Extension des notions de composants JavaBeans aux architectures distribuées.
- Les acteurs du marché des serveurs JEE, de Sun à JBoss.
- L'intégration.
- Le Mapping objet/relationnel.
- Les différents types d'EJB : session, entité, message.
- L'architecture .
- NET.
- Portabilité et interopérabilité.
- Evolution de COM à .
- NET.
- C#, un nouveau langage Objet orienté composants.
- Comparaison avec Java.
- L'infrastructure CLR.
- Les classes de base de .
- NET, ADO.
- NET, les WebServices.
- L'approche Model Driven Architecture.
- Les concepts.
- L'outillage.
- Profils et métamodèle.

Les infrastructures Web à base d'objets

- Les architectures à base de services Web, le fonctionnement, les constituants.
- SOAP, WSDL, UDDI.
- SOA (Service Oriented Architecture), les concepts.
- Standards de gestion de processus métier.
- Les offres en présence.