

UML Objet

Concepts objets et introduction à UML

Les langages de dernière génération -Java, C# 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. L'approche pratique de l'objet y est abordée à travers un formalisme papier et UML afin de ne pas être perturbé par les contraintes techniques des langages de programmation.

Détails

- **Code** : MM-UM1
- **Durée** : 1 jour (7 heures)
- **Public**
 - Décideurs
 - Développeurs
- **Pré-requis**
 - Connaissance d'un langage de programmation

Objectifs

- Assimiler et mettre en oeuvre les concepts de la Programmation Orientée Objet
- Connaître les normes et standards, les langages de développement et de modélisation, les méthodes
- Acquérir les principes de bases de la représentation objet avec UML pour développeur

Programme

Etat de l'art des technologies objet

- L'évolution des langages de programmation
- Les motivations des technologies objet
- Panorama des langages : Smalltalk, C++, Java, C#, PHP, ...

Les concepts de base

- Encapsulation : attributs et méthodes
- Notion d'état et de comportement
- Communication et collaboration entre les objets (Message)
- Concepts de classe et d'instances
- Représentation UML avec le diagramme de classe

Les concepts avancés

- Comprendre le polymorphisme
- Relations entre classes
- Association, Agrégation et Composition
- Dépendance
- Généralisation et spécialisation, principe d'héritage
- Abstraction et réutilisation

- Concept de Type : typage dynamique vs typage statique
- Représentation UML

Concepts architecturaux

- Programmation par contrat de service, notion d'interface
- Notions d'architecture et de composants
- Couplage des composants
- Organisation en couches applicatives d'un système

Modélisation Objet

- Représentation naturelle des entités métiers
- Approche systémique : organisation, responsabilités
- Bonne pratiques et Design Patterns
- Frameworks : introduction, utilisation

Présentation d'UML pour le développeur objet

- L'importance de la modélisation
- Les vues statiques d'un système : les diagrammes de classe, de packages, de cas d'utilisation
- Les vues dynamiques : les diagrammes de séquence, de communication, d'activités et d'états

Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse

- [Validation](#) : Exercices de validation – Attestation de stages