Introduction aux concepts Objet et UML

Le mode de pensée Objet est aujourd'hui omniprésent. Quels que soient les langages orientés objet sont présents dans tous les types de développements et tous les langages. La pensée objet n'est pas une révolution mais une évolution, mais c'est une évolution obligatoire. UML est la solution standard pour représenter et comprendre tout systèmes objet

Objectifs

- Apprendre à penser Objet
- Comprendre les concepts de la programmation orientée Objet (classe, objet, héritage, encapsulation, composant...)
- Savoir représenter tous les concepts de l'analyse Objet à l'aide d'UML

Public concerné

- Développeurs souhaitant acquérir des compétences UML
- Directeurs techniques souhaitant mesurer l'impact et comprendre l'enjeu du passage à l'Objet
- Responsables informatique qui souhaitent avoir une vision précise du développement Objet

Pré requis

- Connaissances de base en informatique.

Une formation d'une journée

Caractéristiques
Tarif : 570 € HT par personne
Numéro de formateur : 11753687675
Nombre d'heures : 7
Référence : OB200
Contact : Patrick LE GOFF
Telephone : 01.76.60.66.10
Email : contact@kaptive.com

Description des modules

num	Module
1	**Évolution des concepts de programmation**
Détails	- Des langages machines aux langages Objet - La programmation orientée Objet, une évolution n'est pas une révolution
2	**Pourquoi s'orienter vers le modèle Objet pour les développements d'applications logicielles ?**
Détails	- Quels sont les apports du modèle Objet ? - Une analyse plus simple, facilitée par une forte similitude avec le monde réel - Des concepts puissants : abstraction, encapsulation, héritage, polymorphisme - Vers une plus grande flexibilité
3	**Présentation d'UML**
Détails	- Les origines d'UML - UML : un standard incontournable - Intérêts et limites
4	**Comment concevoir efficacement en Objet ?**
Détails	- Définir les diagrammes de cas d'utilisation - Identifier les acteurs et les scénarii - Gérer le pilotage des tests
5	**Les principaux concepts de la programmation orientée Objet**
Détails	- Les classes - Les objets - L'héritage - L'encapsulation
6	**Les paquetages**
Détails	- Une vision synthétique du système - Des éléments d'architecture
7	**Les diagrammes de structure statique**
Détails	- Les diagrammes de classes - Les diagrammes d'objets - Les relations dans le système
8	**Les diagrammes d'interaction**
Détails	- Mise en évidence de la dynamique du système - Les diagrammes de séquences - Les diagrammes de collaborations - Création de diagrammes de séquences et de collaborations, pour compléter le modèle des besoins ou pour montrer la dynamique du système
9	**Les diagrammes d'états**
Détails	- Une autre vision dynamique du système - Les diagrammes d'états-transitions - Les diagrammes d'activités - Création de diagrammes d'état-transitions et de diagrammes d'activités
10	**Les diagrammes de déploiement**
Détails	- Les composants du système - Les noeuds du système