Conception objet de haut-niveau avec UML 2

Durée: 2 jours
Public cible: Professionnel TI

À qui s'adresse cette formation ?

Analystes, concepteurs, architectes, programmeurs et gestionnaires impliqués en développement.

Objectifs de la formation

- Utiliser de façon pratique la notation standard UML2.
- Maintenir facilement vos diagrammes dans vos projets.

Description sommaire

L'importance de la vision de haut niveau d'une architecture. Ce cours vise à utiliser de façon pratique la notation standard UML2 et de maintenir facilement vos diagrammes dans vos projets.

Contenu du plan de cours

- Buts, impacts et importance d'UML dans le développement objet et agile.
- Modèle statique : diagrammes de classes, classes, associations, agrégations, propriétés, stéréotypes, etc.
- Modèle opératoire : préconditions, postconditions et invariants.
- Modèle dynamique : diagrammes de séquence, de communication, d'états et d'activités.
- Analyse systémique (packages et spécification des classes) et conception (définition de l'architecture et diagrammes de composants et de déploiement).
- Réalisation : règles de traduction des modèles vers un langage objet.
- Présentation des différentes vues et des meilleures pratiques dans l'utilisation des diagrammes UML.
- Étude de cas et exercices.
- Lignes directrices et conseils pratiques pour un usage pragmatique et facilement maintenable dans les projets.

Approche et méthode pédagogique

- Exposés
- Démonstrations
- Exercices dirigés et individuels

Prérequis

- Cours Introduction au développement orienté-objet ou la connaissance des concepts objet.
- Cours Programmation Java ou Programmation C# ou expérience d'un langage objet.

Recommandations

- Connaissance préalable d’un langage orienté objet (Java, C#, Python, etc.) pour faciliter la compréhension
- Expérience en conception ou en architecture (atout)
- Même informelle ou acquise en projet
- Être à l’aise avec la représentation visuelle de concepts
- Schémas, diagrammes, modélisation Intérêt pour les bonnes pratiques de conception SOLID, modularité, séparation des responsabilités
- Avoir un cas concret ou un projet en tête (optionnel)
- Permet de mieux ancrer les apprentissages