

Concepts essentiels en développement objet

Durée: 0,5 jour

Public cible: Professionnel TI

À qui s'adresse cette formation ?

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

Objectifs de la formation

Cette formation vise à exposer tous les termes, fondements, concepts et meilleures pratiques sur l'objet. Elle explique clairement avec des exemples concrets comment on intègre efficacement ces notions dans les projets. À la fin de cette formation, vous serez en mesure de mieux comprendre le jargon orienté objet et de promouvoir de meilleurs échanges et discussions dans votre équipe.

Description sommaire

Cette formation s'adresse à tous ceux qui souhaitent apprendre ou clarifier définitivement tous les concepts orientés objet. Elle présente également des exemples détaillés en Java et C# ainsi que des exercices et des quiz pour vous assurer que vous appliquez correctement ces concepts dans vos projets. Puisque les processus agiles favorisent le chevauchement des rôles en développement, cette formation est une excellente opportunité d'exposer tous les termes, fondements, concepts et meilleures pratiques sur les objets. Démystifier et mieux comprendre ce jargon encourage de meilleurs échanges dans les deux sens entre les membres et les gestionnaires de vos équipes.

Contenu du plan de cours

- Évolution et amélioration continue en développement logiciel
- Couverture complète des concepts : objets, classes, composants, instances, méthodes, propriétés, encapsulation, héritage, surcharge (overload), polymorphisme, redéfinition (override) et composition
- Différentes formes de l'héritage
- Mécanismes de réutilisation : héritage vs composition, interface, sous-classification, compatibilité, règles est-un et a-un
-

Capturer la connaissance métier avec UML2 Classes et objets Membres d'instance et de classe
Relations : dépendance, généralisation et association Navigation et cas particuliers d'associations
Contraintes et invariant d'une classe Importance du contexte Visibilités, propriétés et stéréotypes
Classes et méthodes abstraites Interfaces et packages

- Créer un document avec contexte métier
- Présenter les caractéristiques pour une architecture propre
- Appliquer les règles pour une architecture propre • Avantages et bénéfices de l'approche orientée objet

Approche et méthode pédagogique

Présentation simple des concepts de base Apprentissage étape par étape Explications claires avec des exemples concrets Lien avec des situations réelles de développement

Objectif : comprendre et appliquer facilement

Prérequis

- Connaissances de base en informatique
- Avoir déjà utilisé un langage de programmation (niveau débutant)
- Comprendre les notions simples comme variables, conditions et boucles

Recommandations

- Connaître un langage orienté objet (atout)
- Avoir fait quelques exercices ou petits projets de programmation
- Être intéressé par la structuration et l'organisation du code