

UML 2 : Les fondamentaux - 3 jours

formation 216

- Vous apprendrez à**
- Concevoir des systèmes logiciel et non-logiciel en utilisant UML 2
 - Récupérer et documenter les spécifications utilisateur en utilisant les use-cases
 - Générer et interpréter les diagrammes UML en utilisant la notation graphique
 - Utiliser des outils CASE pour concevoir et manipuler des modèles UML complets
 - Assurer cohérence et précision dans tous les diagrammes
 - Représenter des design patterns en UML

Objectif Le langage UML (Unified Modeling Language) est la notation standard de modélisation de systèmes. Lors de cette formation, vous apprendrez à générer et interpréter des modèles UML pour de nombreux domaines grâce aux extensions et améliorations d'UML 2. Vous mettrez en œuvre les compétences acquises grâce à l'outil CASE.

À qui s'adresse cette formation Aux concepteurs, programmeurs, chefs de projet et à tous ceux qui sont impliqués dans le développement de systèmes logiciels. Les utilisateurs d'UML qui souhaitent mettre à jour leurs connaissances avec UML 2 retireront également un avantage de cette formation. Une connaissance des techniques orientées objet est utile mais pas obligatoire.

Travaux pratiques Vous acquerrez une expérience pratique d'UML en utilisant des ateliers de génie logiciel. Les exercices comprennent :

- Modélisation des besoins système avec des use-cases
- Conception descendante et rétroconception entre des modèles UML et du code
- Représentation de la structure du système en utilisant des diagrammes d'objets et de classes
- Modélisation du comportement avec des diagrammes d'interaction, de machine d'état et d'activités
- Génération de documentation HTML et textuelle
- Découpage d'un système complexe en sous-diagrammes interdépendants

UML 2 : Les fondamentaux - 3 jours

formation 216

Introduction à UML

Parler un langage commun

- L'importance de la modélisation
- Une communication précise et concise
- L'évolution d'UML

Éléments d'UML

- Blocs de construction : entités, relations et diagrammes
- Vues d'un projet : spécification, conception, mise en œuvre, traitement et déploiement
- Niveaux de détail : visualisation, spécification et construction

Concepts des techniques orientées objet

- Objets et classes
- Liens et associations
- Héritage et polymorphisme

Modélisation de la structure d'un système

Spécification des classes

- Modélisation de types définis par l'utilisateur en tant que classes
- Représentation des informations sous forme d'attributs
- Représentation des fonctionnalités sous forme d'opérations

Identification des relations entre classes

- Dépendances
- Associations
- Agrégation et composition
- Généralisation

Diagrammes d'objets/classes : le cœur d'UML

- Représentation de classes et leurs relations
- Représentation d'un instantané du système avec les diagrammes d'objets
- Définition de modèles d'informations avec des diagrammes de classe

Modélisation du comportement d'un système

Diagrammes des cas d'utilisation : description des besoins utilisateurs

- Représentation des limites du système
- Acteurs et cas d'utilisation
- Notations complémentaires

Diagrammes de séquence et de communication : représentation de scénarios types

- Événements et signaux
- Chronologie des opérations

- Réalisation de cas d'utilisation dans l'optique du développement
- Conversion entre diagramme de séquences et diagramme de communication

Expression des aspects temps réel

- Messages synchrones et asynchrones
- Représentation des contraintes temporelles et des délais de transmission
- Mise en œuvre des diagrammes de temps

Représenter des machines d'état

Diagrammes d'état : modélisation du comportement sous forme d'états

- États, transitions et événements
- Sous-états concurrents
- États historiques et états synchronisés

Diagrammes d'activités : spécification de la logique comportementale

- Modélisation des workflows
- Partitionnement des activités en utilisant les lignes de visibilité
- Concurrence et synchronisation des activités parallèles

Modélisation architecturale

Paquetages et interfaces

- Distinction entre classes et interfaces
- Publication des interfaces des classes et des paquetages
- Utilisation des interfaces

Diagrammes de composants et de déploiement

- Description des dépendances
- Déploiement des composants sur des fils d'exécution, des processus et des processeurs
- Utilisation de diagrammes de structure composite pour décrire la structure interne

Design patterns

- Patterns, mécanismes et frameworks
- Représentation des design patterns
- Référencement de design patterns

Utilisation d'UML

Architecture MDA (Model-Driven Architecture)

- MOF (Meta-Object Facility)
- CWM (Common Warehouse Meta-model)

Phases du cycle de vie

- Utilisation d'UML dans le Processus Unifié

- Modélisation des processus métier
- Récupération des spécifications
- Analyse système
- Conception de logiciels