

UML 2 - Analyse et conception

Référence : DEUML

Durée : 4 jours

Certification : Aucune

CONNAISSANCES PREALABLES

- Connaître la programmation orientée objet.

PROFIL DES STAGIAIRES

- Analystes. • Architectes logiciels. • Chefs de projets. • développeurs.

OBJECTIFS

- Découvrir les différentes phases de la modélisation objet en UML. • Savoir traduire un besoin fonctionnel en s'appuyant sur les diagrammes UML. • Appréhender UML de façon opérationnelle sur toutes les étapes du projet de développement. • Maîtriser les bonnes pratiques de conception logicielle fondamentale.

CERTIFICATION PREPAREE

Aucune

METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

FORMATEUR

Consultant-Formateur expert Développement

METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

Partie 1 : Présentation, concepts, objectifs

Présentation et concepts

- Les méthodes de modélisation
- Concepts de l'approche Objet : classe, Objet, généralisation, héritage, polymorphisme...
- Niveau de modélisation. UML et la maîtrise d'ouvrage
- UML 2 : présentation des évolutions proposées

Description des processus

- Définition des concepts de base : processus, activité
- Le diagramme des cas d'utilisation. Les cas d'utilisation, au centre de l'approche UML
- Comment décrire les cas ? Différents types de cas : les cas métiers et les cas fonctionnels
- Le diagramme d'activité

- Décrire les activités d'un processus : activité, action, stimuli, flot de données, de contrôle...
- Le diagramme de séquence
- Comment décrire le workflow d'un processus ? Les objets, les messages, l'axe des temps. Illustration
- Typologies de processus utilisés pour la modélisation : métier/support/pilotage, base/variante, générique/spécialisé

Description des informations

- Le diagramme de classe. Décrire les données du système. Classe, association, opération, généricité et spécialisation
- Le diagramme Objet. Comment utiliser un diagramme Objet pour illustrer le diagramme de classe ?
- Le diagramme d'états. Description des différents états des objets. Etat, transition, imbrication d'états

Description des échanges

- Le diagramme de communication
- Comment décrire le contexte du projet ?
- Représentation des échanges entre les objets du système
- Objet, message, ordre condition, paramètres

Description des fonctions

- Définition des fonctions du futur système : les cas d'utilisation fonctionnels
- Collaboration entre les objets logiciels
- Description des opérations systèmes

Processus projet

- Inscrire la modélisation UML dans une démarche existante de conduite de projet
- Méthode de projet "classique"
- Méthode RUP, Rational Unified Process
- Organiser la documentation

Partie 2 : Mise en œuvre d'UML 2

Vie d'un SI et paradigmes de développement

- UML et le processus de développement
- Modèle en V
- Modèle en spirale
- Modèle RAD
- Modèle en Y
- Modèle RUP
- Les diagrammes d'UML

Étude préliminaire

- Élaboration du cahier des charges
- Identifier les acteurs
- Identifier les messages
- Modéliser le contexte
- Représentation du contexte dynamique grâce à un diagramme de collaboration

Capture des besoins fonctionnels : spécification des exigences d'après les cas d'utilisation

- Identification des acteurs
- Identification des cas d'utilisation
- Structuration en packages
- Relations entre cas d'utilisation : inclusion - extension
- Classement des cas d'utilisation
- Planification du projet en itération

Analyse du domaine : les objets métiers

- Identification des concepts du domaine
- Ajout des associations et des attributs
- Généralisation
- Structuration en packages

Introduction de bonnes pratiques de conception

- Principes fondamentaux de manipulation d'abstraction et d'inversion de dépendances
- Introduction de quelques Design Patterns fondamentaux