

Sommaire :

1. Méthode de développement du logiciel avec UML

- 1.1 Introduction au génie logiciel
- 1.2 Objectifs de la modélisation orientée objet
- 1.3 UML comme méthode de développement
- 1.4 Processus unifié et approche itérative
Introduction aux processus (cas du Unified Process / RUP)
Étapes principales : analyser – concevoir – implémenter – vérifier
- 1.5 Gestion de projet logiciel avec UML

2. L'essence d'une modélisation UML

- 2.1 Concepts clés d'UML
- 2.2 Notions d'objets, classes et relations
- 2.3 Diagrammes structuraux (classes, composants, packages)
- 2.4 Diagrammes comportementaux (cas d'utilisation, séquences, activités)
- 2.5 Diagrammes de déploiement et d'artefacts

3. Le langage OCL : de la précision à l'ingénierie des modèles

- 3.1 Introduction à OCL (Object Constraint Language)
- 3.2 Syntaxe et sémantique d'OCL
- 3.3 Navigation et expressions dans OCL
- 3.4 Contraintes d'intégrité, invariants, pré- et post-conditions
- 3.5 Applications d'OCL à la vérification des modèles
- 3.6 Cas pratiques d'OCL

4. Modélisation métier et des besoins

- 4.1 Analyse des besoins
- 4.2 Diagrammes de cas d'utilisation
- 4.3 Description des scénarios et règles métier
- 4.4 Modélisation du domaine (objets du monde réel)
- 4.5 Glossaire et modèles métier

5. Analyse : une vue logique du système

- 5.1 Diagrammes d'objets pour capturer des situations
- 5.2 Diagrammes de classes analytiques
- 5.3 États-Transitions pour comportements complexes
- 5.4 Diagrammes d'activités pour processus métier et flux
- 5.5 Exemple d'étude de cas d'analyse avec solutions

6. Vérification de modèles

- 6.1 Rôle de la vérification dans l'ingénierie des modèles
- 6.2 Cohérence des diagrammes

- 6.3 Utilisation de contraintes OCL pour vérification
- 6.4 Validation des besoins et des modèles analytiques
- 6.5 Exercices corrigés de vérification

7. Conception : de la transformation de modèles

- 7.1 De l'analyse à la conception
- 7.2 Diagrammes de conception orientés objets
- 7.3 Architecture logicielle avec modèles UML
- 7.4 Diagrammes de composants et de déploiement
- 7.5 Modèles dépendants de la plateforme

8. Codage, test et qualité

- 8.1 Génération de code à partir de modèles
- 8.2 Bonnes pratiques de codage orienté objet
- 8.3 Stratégies de tests basées sur les modèles
- 8.4 Assurance qualité logiciel et métriques
- 8.5 Exercices de test et de vérification