



CTI FORMATION
Créateur de compétences

Modéliser des données avec Merise et UML

OBJECTIF

Connaître les principes de base de Merise. Apprendre à la fois une méthode d'analyse et une démarche de construction d'un système d'information. Apprendre à mettre en oeuvre UML (Unified Modeling Language) dans tous projets de système d'informations. Acquérir les connaissances nécessaires à son utilisation et à la mise en oeuvre des meilleures pratiques d'analyse et de conception orientée objet.

Public visé :

Développeurs Java devant réaliser des applications mobiles pour téléphone.

Pré-requis :

Avoir déjà utilisé une base de données est un plus.

Modalités de suivi / Appréciation des résultats :

Attestation de fin de stage.

Encadrement :

La formation sera assurée par un formateur WEB expert.

PROGRAMME

APPROCHER UN PROBLÈME

- . Comprendre l'avènement de la méthode Merise
- . Acquérir le vocabulaire commun
- . Les principes et les objectifs de la méthode Merise
- . Le cycle d'abstraction, de vie et de décision

LA MODÉLISATION DES TRAITEMENTS

- . Schéma de circulation des documents
- . Le modèle Conceptuel des Traitements (MCT)
- . Le Modèle Organisationnel de Traitements (MOT)

LE MODÈLE CONCEPTUEL DES DONNÉES (MCD)

- . Le modèle Entités-Associations
- . Les concepts, le formalisme, les contraintes fonctionnelles
- . Les règles de vérification
- . L'élaboration du modèle conceptuel des données

LE MODÈLE LOGIQUE DES DONNÉES (MLD)

- . Les concepts, le formalisme
- . L'élaboration du modèle logique des données brut
- . La valorisation et l'optimisation

LE MODÈLE PHYSIQUE DES DONNÉES (MPD)

- . La normalisation, la décomposition
- . Les dépendances fonctionnelles
- . Les règles de passage d'un MCD vers un MPD

COMPRENDRE LES PRINCIPALES NOTIONS DE L'APPROCHE OBJET

- . Rappel sur les paradigmes de programmation
- . Les objets : identité, état et comportement
- . Relations au monde réel et aux systèmes informatiques
- . Popularité, avantages de l'Orienté Objet
- . Abstraction, encapsulation, classification.
- . Classes et instances et classes abstraites
- . Concept d'héritage
- . Méthodes et envois de messages entre objets polymorphisme
- . Surcharge et redéfinition

POURQUOI MODÉLISER ? APPRÉHENDER LE SPECTRE DE L'ANALYSE ET DE LA CONCEPTION

- . Domaine métier et modélisation d'une solution informatique.
- . Le modèle : un artefact central du processus projet
- . Analyse et conception d'une solution informatique
- . Impacts des langages de programmation
- . Evolution vers l'analyse/conception Objet

PRÉSENTATION GÉNÉRALE D'UML

- . Vues de l'architecte
- . Le coeur de l'UML : les différents types de diagrammes
- . Différences entre diagrammes statiques et dynamiques
- . Présentation de plusieurs démarches de modélisation

DIAGRAMME UML : VUE D'ENSEMBLE DU SYSTÈME, DE CES ÉLÉMENTS ET DE LEURS RELATIONS

- . Diagramme de classe : son rôle et son utilisation
- . Comment identifier les classes utiles ?
- . Une classe en UML : nom, attributs et opérations.
- . Visibilité : publique, privée et protégée
- . Formalisme et notation
- . Relations entre classes (association, généralisation, ...)
- . Multiplicités, rôles, contraintes, ...
- . Classes abstraites, interfaces, packages
- . Diagramme d'objets et instances de classes
- . Diagramme de structure composite

DIAGRAMMES UML DE MODÉLISATION DYNAMIQUE

- . Diagramme de séquence : interactions entre objets au cours du temps
- . Messages synchrone et asynchrone
- . Diagramme global d'interactions : enchaînements possibles entre les scénarios identifiés sous forme de diagrammes de séquences
- . Diagramme de communication : rôle des objets, interactions, concurrence de traitements ...
- . Diagramme d'état transition : états possibles d'un objet et événements déclenchant les transitions
- . Diagramme d'activité : flux des activités pour réaliser une opération, objets en charge de ces activités
- . Diagramme de temps : variations d'une donnée au cours du temps - Notations et exemples