

MET101

Conception d'application avec UML

Durée

4 jours.

Pré-requis

Connaissance d'un langage de programmation ou d'une méthode de conception. La connaissance d'un langage orienté objet est un plus.

Orientation

Développeurs, analystes, chefs de projets impliqués dans la conception d'applications à base d'objets.

Dates

19-22 mars 2012 . 2-5 juil 2012
. 5-8 nov 2012

Objectifs

Connaître et savoir construire les principaux diagrammes UML. Concevoir une application avec UML. Se sensibiliser au processus de développement unifié.

Organisation du cours

Théorie : 60%
Pratique : 40%

Configuration matérielle

Une machine par stagiaire.
Environnement de développement Eclipse.

Documentation fournie

Support de cours et exercices corrigés.

Tarif H.T. *

1 personne : 1680 €
2 personnes : 2856 €
3 personnes : 3866 €
* personnes d'une même entreprise inscrites à la même session.

Programme

Présentation

- Historique des méthodes
- Les technologies orientées objet

Concepts fondamentaux

- Abstraction, modularité
- Encapsulation
- Hiérarchie, héritage, généricité
- Composition

Processus de développement itératif

- UML et le processus de développement
- Principes du développement itératif, définition des itérations
- Les méthodes de type UP

La modélisation des exigences

- Modélisation des cas d'utilisation
- Spécification des cas d'utilisation

Le modèle statique

- Notion d'objet, de classe, de relation
- Diagramme de classes
- Diagramme d'objets

Le modèle dynamique

- Notion d'interaction
- Diagramme de séquence
- Diagramme de communication
- Diagramme d'activité

Le métamodèle UML

- Eléments du métamodèle
- Mécanismes communs (stéréotypes, étiquettes, contraintes...)

Les modèles d'architecture et d'implémentation

- Diagramme d'architecture : Notion de package
- Diagramme de composants
- Diagramme de déploiement
- Notion de design patterns

L'implémentation et les tests

- Génération de code
- Définir les tests

Outils de modélisation et de développement

- Construction des diagrammes
- Génération de code
- Reverse engineering, round-trip