



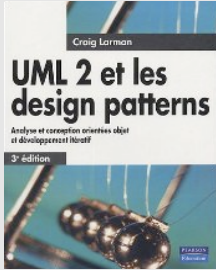
zenika
ARCHITECTURE INFORMATIQUE

www.zenika.com

UML & Design Patterns

Modélisation et conception avec UML et les Design Patterns

Repartez avec le livre



Durée
3 jours

Répartition
60% théorie
40% pratique

Pré-requis
Connaissance de Java

Public
Architecte
Développeur
Chef de projet

Tarif (déjeuners inclus)
1490 € (HT)
Tarif (1 mois avant)
1290 € (HT)

Lieu
Paris 9ème

Sessions 2009
7 au 9 janvier
2 au 4 mars
4 au 6 mai
1 au 3 juillet
7 au 9 septembre
3 au 5 novembre

Intra-entreprise sur
demande

Inscription et
renseignements
+33(0)1.45.26.19.15
training@zenika.com
www.zenika.com

Objectifs

- Apprendre le langage UML et sa notation
- Comprendre l'utilisation d'UML au sein d'un processus de développement
- Maîtriser les principes fondamentaux du Design Objet
- Apprendre à mettre en oeuvre les principaux Design Patterns au sein d'un modèle Objet

Contenu

Introduction à la modélisation Objet

- Qu'est ce qu'un modèle ? Pourquoi modéliser ?
- Rappel des fondamentaux de l'approche Objet
- Les enjeux de la programmation orientée Objet

Le langage UML

- Historique (l'unification des méthodes, le rôle de l'OMG)
- UML et les processus de développement agiles
- Un processus itératif incrémental
- Planification du processus
- Introduction au concept MDA (Model Driven Architecture)

Les cas d'utilisation

- Les différents axes de modélisation UML
- Définition de la notion de scénario
- Scénario et cas d'utilisation
- Diagrammes des cas d'utilisation
- La notion d'Acteur
- Les relations entre cas d'utilisation

Les diagrammes de classes

- Les différents objectifs du diagramme de classes
 - Le niveau conceptuel
 - Le niveau spécification
- Les associations (navigabilité et multiplicité)
- Relations d'agrégation et de composition
- La relation d'héritage de classe
- Classe abstraites versus classes concrètes
- Le Design Pattern Template Method
- Visibilité des attributs et des méthodes
- Importance de la notion d'API (comment optimiser la signature des méthodes)
- Design Pattern de la Métaclasse

Les diagrammes d'interaction

- Objectifs et enjeux des diagrammes de séquence et de collaboration
- Notation diagramme de séquence
- Notation diagramme de collaboration

Le diagramme de packages

- La notion de dépendances (principes de gestion des dépendances)
- Comment maîtriser ses dépendances ? (principe DIP) ?
- Le couplage faible par Interface (Design Pattern Factory, l'injection des dépendances)
- Le Design des packages (packages d'abstraction et packages d'implémentation, les métriques)

Les principes fondamentaux du Design Objet

- Présentation des enjeux de la conception agile
- Introduction aux Patterns d'analyse GRASP (expert en information, créateur, forte cohésion, faible couplage, contrôleur, pattern polymorphe)
- Principe de responsabilité unique (SRP)
- Principe d'Ouverture/Fermeture (OCP)
- Principe de substitution de Liskov
- Composition versus Héritage

Le diagramme d'états-transitions

- L'état d'un objet
- La notion de transition
- La notion de Garde sur une transition
- Mise en oeuvre du Design Pattern Etat

Le diagramme d'Activité

- La notion d'activité
- La notion de travées

Les mécanismes d'extension d'UML

- Les 4 types d'extensions (les notes, les stéréotypes, les tagged values, les contraintes)