

Conception UML avec les Design Patterns



UML est un langage graphique de modélisation des données et des traitements. C'est une formalisation très aboutie et non-proprétaire de la modélisation objet utilisée en génie logiciel. Les Design Patterns fonctionnent sur le principe suivant : à un problème donné, on apporte une solution précise et exposant à la fois le principe, la mise en oeuvre, les avantages, les inconvénients et des exemples d'application

Objectifs

- Comprendre les concepts sous-tendant les Design Patterns
- Savoir comment trouver le (ou les) Design Pattern(s) répondant à une problématique donnée, les utiliser et les modifier pour répondre au besoin spécifique
- Disposer d'un bagage formel pour faciliter la recherche de solutions stéréotypiques
- Comprendre comment enrichir le catalogue de Design Patterns métiers

Public concerné

- Développeurs Java,
- Concepteurs
- Architectes

Pré requis

- Expérience de la programmation en Java et avoir suivi le stage "Analyse et conception avec UML 2.0" (OB210)

Une formation de 3 jours

Caractéristiques
Tarif : 1575 € HT par personne
Numéro de formateur : 11753687675
Nombre d'heures : 21
Référence : OB213
Contact : Patrick LE GOFF
Telephone : 01.76.60.66.10
Email : contact@kaptive.com

Description des modules

num	Module
1	Rappels / approfondissements sur UML
Détails	<ul style="list-style-type: none"> - Syntaxes et utilités des diagrammes UML fondamentaux (Uses Cases ...) - Détails sur le diagramme de classes (notations et concepts) - Classes abstraites, interfaces ... - Packages, composants ...
2	Présentation des Designs Patterns
Détails	<ul style="list-style-type: none"> - Patterns, anti-patterns - Notion de Framework
3	Essentiels des patterns "GOF"
Détails	<ul style="list-style-type: none"> - Liste des patterns "GOF" - Singleton - Stratégie - Méthode de fabrication - Façade - Décorateur (enveloppe)
4	I.O.C. / Injection de dépendance
Détails	<ul style="list-style-type: none"> - Inversion de contrôle, méthodes d'injections - Conteneur léger (illustration avec Spring)
5	Patterns pour JEE et le N-tiers
Détails	<ul style="list-style-type: none"> - DAO (Data Access Object) - VO (Value Object) / DTO (Data Transfert Object) - Business Delegate (proxy métier / fonctionnel) - MVC (Modèle / Vue / Contrôleur)
6	Grands principes objets
Détails	<ul style="list-style-type: none"> - Gestion des évolutions et dépendances (OCP, LSP ...) - Organisation des applications en modules - Gestion de la stabilité d'une application
7	Autres Patterns "GOF"
Détails	<ul style="list-style-type: none"> - Observateur - Médiateur - Composite - Adaptateur ...
8	Patterns GRASP (en annexes)
Détails	<ul style="list-style-type: none"> - Affectations des responsabilités - Expert, faible couplage, forte cohésion - Création - Contrôleur, polymorphisme, fabrication pure - Indirection, protection des variations