

Design Patterns

Connaître et utiliser les Design Patterns

Maîtriser les bonnes pratiques de conception, c'est dans un premier temps l'assurance de savoir mieux appréhender un environnement technique qui les utilise.

C'est aussi pouvoir échanger plus facilement avec des experts techniques, sur les bases d'un langage et de métaphores communs.

C'est enfin la capacité à mettre en oeuvre une pratique éprouvée face à une problématique type que l'on reconnaît, sans avoir à chercher.

Les Design Patterns constituent la boîte à outil au quotidien de tout développeur expert objet.

Détails

- Code : MM-DP
- Durée : 3 jours (21 heures)

Public

- Architectes
- Chefs de projets
- Développeurs
- Ingénieurs
- Maîtrises d'oeuvres
- Maîtrises d'ouvrages

Pré-requis

- Connaissance de Java

Objectifs

- Comprendre les Design Patterns
- Savoir trouver le (ou les) Design Pattern(s) répondant à une problématique donnée
- Savoir les appliquer pour répondre à un besoin spécifique.

Programme

Les fondamentaux de l'orienté objet et UML

- Notion d'objet, de classe, d'encapsulation
- Les relations d'héritage, d'implémentation d'interface et les associations
- Les diagrammes UML

L'origine des Design Patterns

- La genèse
- Rôle et intérêts
- Le catalogue et son organisation

L'utilisation des Design Patterns

- Identifier le problème
- Trouver le pattern répondant au problème
- Créer sa solution en appliquant le pattern

Les Patterns créateurs

- Singleton (Singleton)
- Fabrique (Factory Method)
- Fabrique Abstraite (Abstract Factory)
- Monteur (Builder)
- Prototype (Prototype)

Les Patterns structuraux

- Composite (Composite)
- Adaptateur (Adapter)
- Décorateur (Decorator)
- Façade (Facade)

- Proxy (Proxy)
- Poids-mouche (Flyweight)
- Pont (Bridge)

Les Patterns comportementaux

- Itérateur (Iterator)
- Stratégie (Strategy)
- Patron de méthode (Template Method)
- État (State)
- Observateur (Observer)
- Commande (Command)
- Visiteur (Visitor)
- Chaîne de responsabilité (Chain of responsibility)
- Interpréteur (Interpreter)
- Médiateur (Mediator)
- Memento (Memento)

Les GRASP patterns (General Responsibility Assignment Software Patterns)

- Les pratiques pour affecter les responsabilités aux classes
- Expert
- Faible couplage
- Forte cohésion
- Polymorphisme
- Pure Fabrication
- Créateur
- Contrôleur
- Loi de Demeter

Modalités

- **Type d'action** : Acquisition des connaissances
- **Moyens de la formation** : Formation présentielle - 1 poste par stagiaire - 1 vidéo projecteur - Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** : Exposés - Cas pratiques - Synthèse
- **Validation** : Exercices de validation - Attestation de stages
- **Profils des intervenants** : Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.