

# Conception avec les Design Patterns

Référence : DEDP001

Durée : 5 jours

Certification : Aucune

## CONNAISSANCES PREALABLES

- Posséder une bonne connaissance de la programmation objet et de la programmation d'application Web.

## PROFIL DES STAGIAIRES

- Analystes. • Architectes. • Chefs de projets. • Développeurs.

## OBJECTIFS

- Comprendre la philosophie des Design Patterns. • Connaître les principaux patterns. • Appréhender la démarche d'utilisation des modèles pour appliquer le bon patron. • Appliquer les Design Patterns dans une architecture technique de conception.

## CERTIFICATION PREPAREE

Aucune

## METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

## FORMATEUR

Consultant-Formateur expert Développement

## METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

## CONTENU DU COURS

### Le diagramme fonctionnel

- Diagramme de cas d'utilisation
- Comment élaborer un modèle des besoins ?
- Définition des acteurs
- Définition des cas d'utilisation
- Délimitation du système
- Description des cas d'utilisation
- La structuration du diagramme des cas d'utilisation
- Les meilleures pratiques pour élaborer un modèle des besoins

### Les diagrammes structurels

- Diagramme de classes
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement
- Diagrammes de packages (UML V2.0)

- Diagrammes de structures composites (UML V2.0)

### Briques de base

- Introduction au concept de Design Pattern
- Factory: l'usine à code
- Dependency injection: choisir son algo à la volée
- Mise en pratique de dependency injection
- Template method: enchainement concret d'actions abstraites
- Mise en pratique de template method
- Chaining : faciliter l'enchainement de transformations
- Mise en pratique du chaining

### Principes fondamentaux de conception

- Les Patterns Grasp d'affectation des responsabilités
- Principe d'ouverture/fermeture : OCP
- Inversion des dépendances : DIP

- Substitution de Liskov : LSP
- Séparation des interfaces : ISP

### **Les Design Patterns du GoF et autres Patterns**

- Les Design Patterns de comportement : itérateur, stratégie, Template Method, état, observateur, médiateur, visiteur
- Les Design Patterns de création : singleton, fabrique abstraite, builder
- Les Design Patterns de structure : composite, pont, adaptateur, décorateur, façade, Proxy, Extension Object, Value Object

### **Patterns et architecture**

- Couches logicielles
- Modèle Vue Contrôleur : MVC
- Frameworks
- Approche par composants

### **Organiser un modèle et en contrôler la qualité**

- Règles d'organisation en packages
- Métriques de packages