

DÉVELOPPEMENT JAVA AVEC LES DESIGN PATTERNS

Durée

2 jours

Référence Formation

4-JA-DP

Objectifs

Savoir mettre en application les design patterns afin d'optimiser leurs projets de conception logiciel
Cours particulièrement adapté au projet de développement JAVA

Participants

Développeur expérimenté

Pré-requis

La connaissance du développement Objet et de JAVA est particulièrement importante
Des notions d'UML son souhaitables

Moyens pédagogiques

Réflexion de groupe et apports théoriques du formateur
Travail d'échange avec les participants sous forme de réunion-discussion
Utilisation de cas concrets issus de l'expérience professionnelle
Validation des acquis par des questionnaires, des tests d'évaluation, des mises en situation et des jeux pédagogiques.
Remise d'un support de cours.

PROGRAMME

Rappel du les stratégies et techniques de conception

- La conception objet
- Modèle d'analyse
- Formalisme UML

Introduction aux Design Patterns

- Des solutions de conception « sur étagère »
- Définir les besoins techniques : le client
- Définir le 'type' de pattern des classes
- Définir les liens de collaboration entres les classes

Modèle d'analyse et d'architecture

- Définir les règles de conception
- Définir les modèles techniques abstraits
- Systématiser les principes d'architecture
- Automatiser les principes d'architecture

Utilisation des design patterns

- Description formel d'un pattern
- Comment choisir un pattern ?
- Comment utiliser un pattern ?

Patterns de construction d'objets

- Rendre un système indépendant de la façon dont les objets sont créés
- Cas pratique

Patterns de structuration

- Composer des objets pour obtenir de nouvelles fonctionnalités

- Cas pratique

Patterns de gestion des comportements

- Identifier des coopérations évolutives et découplées

- Cas pratique

Patterns JAVA

- Utiliser les patterns dans une architecture JAVA

- Patterns de création

- Patterns structurels

- Patterns comportementaux

- Cas pratique

Synthèse et conclusion

- Formaliser les solutions techniques

- Réutiliser l'expertise technique

- Comment choisir un design pattern ?

- Automatisation des transformations

- Modèles et profils UML