

Objectifs / Durée de la formation

Durée: 2 jours, soit 14 heures

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

Participants / Pré-requis

- Développeurs expérimenté
- La connaissance du développement Objet et de JAVA est particulièrement importante ; des notions d'UML sont souhaitables

Moyens pédagogiques

- Formateur expert dans le domaine
- Mise à disposition d'un ordinateur, support de cours remis à chaque participant, vidéo projecteur, tableau blanc et paperboard
- Feuille de présence signée en demi-journée, évaluation des acquis tout au long de la formation, questionnaire de satisfaction, attestation de stage

Programme

1. Rappel du les stratégies et techniques de conception

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

2. 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 entre les classes

3. 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

4. Utilisation des Design Patterns

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

5. Patterns de construction d'objets

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

6. Patterns de structuration

- Composer des objets pour obtenir de nouvelles fonctionnalités
- Cas pratique

7. Patterns de gestion des comportements

- Identifier des coopérations évolutives et découplées
- Cas pratique

8. Patterns JAVA

- Utiliser les patterns dans une architecture JAVA
- Patterns de création
- Patterns structurels
- Patterns comportementaux
- Cas pratique

9. Synthèse et Conclusion

- Formaliser les solutions techniques
- Réutiliser l'expertise technique
- Comment choisir un Design Pattern ?
- Automatisation des transformations
- Modèles et profils UML