

Référence	<b>4-JA-DP</b>
Durée	<b>2 jours (14 heures)</b>
Éligible CPF	<b>NON</b>
Mise à jour	<b>27/11/2023</b>

## Développement Java avec les design patterns



### OBJECTIFS PÉDAGOGIQUES

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



### PUBLIC CONCERNÉ

Développeur expérimenté



### PRÉREQUIS

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.



### MODALITÉS D'ÉVALUATION

- Feuille de présence signée en demi-journée,
- Evaluation des acquis tout au long de la formation,
- Questionnaire de satisfaction,
- Attestation de stage à chaque apprenant,
- Positionnement préalable oral ou écrit,
- Evaluation formative tout au long de la formation,
- Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles



### MOYENS TECHNIQUES EN PRÉSENTIEL

Accueil des stagiaires dans une salle dédiée à la formation équipée à minima d'un vidéo projecteur et d'un tableau blanc et/ou paperboard.

Pour les formations nécessitant un ordinateur, un PC est mis à disposition de chaque participant.



### MOYENS TECHNIQUES EN DISTANCIEL

A l'aide d'un logiciel (Teams, Zoom...), d'un micro et éventuellement d'une caméra les apprenants interagissent et communiquent entre eux et avec le formateur.

Sessions organisées en inter comme en intra entreprise.

L'accès à l'environnement d'apprentissage ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.

Pour toute question avant et pendant le parcours, assistance technique à disposition au 04 67 13 45 45.



### ORGANISATION

Délai d'accès : 5 jours ouvrés  
 (délai variable en fonction du financeur)

Les cours ont lieu de 9h à 12h30 et de 13h30 à 17h



### ACCESSIBILITÉ

Les personnes en situation d'handicap sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

Pour tout renseignement, notre référent handicap reste à votre disposition : mteyssedou@ait.fr



### PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention

Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.



### CERTIFICATION POSSIBLE

Aucune

# Développement Java avec les design patterns

## 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 entre 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