DAWAN

Plan de la formation

www.dawan.fr

Formation C++ Intermédiaire : Conception objet avancée + Design patterns

Durée :	3 jours (21 heures)
Tarifs inter-entreprise :	2 175,00 CHF HT (standard) 1 740,00 CHF HT (remisé)
Public :	Développeurs C++
Pré-requis :	Avoir suivi la formation C++ Initiation ou notions équivalentes
Objectifs :	Maîtriser les fondements de la POO - Comprendre la décomposition d'une application d'entreprise en objets (conception/design OO) - Appliquer les principes de regroupement, de structuration et de communication entre les objets d'un système complexe - Concevoir des systèmes OO de manière à favoriser la maintenabilité et faciliter le changement dans un contexte itératif - Appliquer les principes S.O.L.I.D. - Concevoir des applications faiblement couplées et cohésives - Apprendre à implémenter des designs patterns
Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
Sanction :	Attestation de fin de formation mentionnant le résultat des acquis

Référence :	PRO101928-F
Note de satisfaction des participants:	4,85 / 5
Contacts :	commercial@dawan.fr - 09 72 37 73 73
Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
Délais d'accès :	Variable selon le type de financement.
Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

Maîtriser les fondements de la conception objet

Encapsulation : intérêt, bonnes pratiques
Agrégation d'objets
Héritage : cas d'usage, préférence pour la composition
Polymorphisme : ad-hoc, sous-typage, types paramétriques
Objets Valeurs (Value Objects)
Cercle vertueux de l'ignorance

Atelier : construire un schéma de classes cohérent

Gérer l'interaction entre les objets du système

Tell don't ask
Gestion des dépendances
Découpage des règles d'affaires basé sur l'interaction
Conception basée sur les comportements
Loi de Déméter

Atelier : implémentation de patterns de comportements

Concevoir un domaine et découper des objets

Conception par concepts plutôt que par données : concepts, types d'objets
Architecture Héxagonale
Présentation des principes SOLID
Principe de la responsabilité unique (SRP)
Principe de l'ouverture-fermeture (OCP)

Atelier : multiples exemples de mauvaise/bonne implémentation

Introduire une abstraction

Métrique de l'Abstraction-Instabilité (R. C. Martin)
Principe de substitution de Liskov (LSP)
Composition versus héritage
Principe de la ségrégation des interfaces (ISP)

Atelier : analyse d'un code et présentation des métriques - ré-écriture d'exemples concrets

Concevoir une application en couches

Conception modulaire
Principe d’inversion des dépendances (DIP)
Objet de transport (DTO)
Présentation de la clean architecture

Atelier : implémentation d'une applicaion en couches

Comprendre et appliquer les design patterns

Historique et ouvrages de référence
Domaines d'application
Comment appliquer les Design Patterns

Générer des instances

Factory et Abstract Factory pour la création sous condition
Singleton et dérivé : maîtrise des ressources disponibles

Organiser les structures de données

Le Composite, comment simplifier les listes
Proxy et Adapter, les interfaces de l'accès aux méthodes
La Facade : clarifier un composant

Maîtriser le comportement des objets

Strategy : l'usine à méthodes
L'itérateur et ses implémentations existantes
Observer : l'événementiel sans événements

Chaîne de responsabilités et arbres de responsabilité
Visiteur et accès : maîtrise de la collaboration
Aperçu d'autres Design Patterns