



Formation Bonnes pratiques et anti-patterns de développement logiciel

■ Durée :	5 jours (35 heures)
■ Tarifs inter-entreprise :	3 475,00 CHF HT (standard) 2 780,00 CHF HT (remisé)
■ Public :	Développeurs
■ Pré-requis :	Maîtrise de la programmation orientée objets (Java, C#, PHP)
■ Objectifs :	Acquérir les bonnes pratiques de conception objet - Apprendre les anti-patterns de développement logiciel
■ Modalités pédagogiques, techniques et d'encadrement :	<ul style="list-style-type: none">• 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 :	<ul style="list-style-type: none">• 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 :	PRO101594-F
■ Note de satisfaction des participants:	4,93 / 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 principes SOLID

Présentation des principes SOLID

Principe de la responsabilité unique (SRP)

Principe de l'ouverture-fermeture (OCP)

Principe de substitution de Liskov (LSP)

Principe de la ségrégation des interfaces (ISP)

Principe d'inversion des dépendances (DIP)

Appliquer les bonnes pratiques

Style de code et mesure

Profiling et performance

Calcul de métriques : longueur de classe, de méthode, nombre de paramètres, de champs, complexité (cyclomatique, NPATH)

Bonnes pratiques de codage :

- Limiter autant que possible les retours par fonction
- Éviter les instructions en cascade
- Éviter les instructions break et continue
- Éviter les conditionnelles négatives
- Remplacer les nombres et les chaînes par des constantes

Règles de conception :

- Patterns à appliquer
- Substituer les instructions switch par du polymorphisme
- Éviter les champs protégés dans les classes final
- Couplage : typer par des interfaces

Documentation du code : langue, génération et maintien

Découvrir des anti-patterns de développement

Anti-pattern : définition, types (architectural, organisationnel, développement)

Panorama d'anti-patterns :

- The Blob (God Class)
- Singleton
- Poltergeist
- Hard coding
- Continuous Obsolescence
- Lava Flow
- Ambiguous Viewpoint
- Functional Decomposition
- Dead End
- Input Kludge
- Walking through a Minefield
- Mushroom Management
- Boat Anchor
- Object cesspool
- Golden hammer
- Premature optimization
- Spaghetti code
- Magic numbers
- Copy-and-Paste Programming
- Yo-yo problem
- Accidental complexity