

# Formation Bonnes pratiques et anti-patterns de développement logiciel

<b>Durée :</b>	5 jours
<b>Public :</b>	Développeurs
<b>Pré-requis :</b>	Maîtrise de la programmation orientée objets (Java, C#, PHP)
<b>Objectifs :</b>	Acquérir les bonnes pratiques de conception objet - Apprendre les anti-patterns de développement logiciel
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	PRO101594-F
<b>Note de satisfaction des participants:</b>	4,93 / 5

## 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