

Mis à jour le 20/01/2025

S'inscrire

# Formation Clean Code : Produisez un code de qualité

3 jours (21 heures)

## Présentation

Un [code propre](#) facilement compréhensible et modifiable pour développer votre structure ou votre application rapidement. C'est la promesse du Clean Code.

Il ne s'agit pas d'un ensemble de règles rigides à respecter, mais une association de principes renforçant la qualité de vos développements. La finalité du Clean Code est de permettre à n'importe quel programmeur de pouvoir éditer et comprendre simplement le code de vos projets.

La mise en place du Clean Code au sein de votre organisation a donc de nombreux bénéfices comme la réduction des erreurs, une meilleure délivrabilité ainsi qu'une plus grande collaboration.

Notre formation Clean Code vous présentera ses différents principes comme le cadre SOLID, vous découvrirez les bonnes pratiques de [nommage](#), de structure et de formatage, vous appréhendez le Test Driven Development (TDD) et le Test Driven Development (BDD). Enfin, nous vous donnerons les bonnes pratiques de productivité et de collaboration.

À l'issue de cette formation, vos collaborateurs seront aptes à comprendre et à mettre en place une démarche de Clean Code au sein de votre organisation.

## Objectifs

- Comprendre le Clean Code, ses principes et son intérêt
- Bien structurer et formater ses programmes
- Comprendre le BDD et son utilité
- Comprendre le TDD et son utilité
- Connaître les bonnes pratiques de productivité et de communication en programmation

## Public visé

- Développeurs Web et applicatif
- Architectes
- Administrateur

## Pré-requis

Expérience en programmation.

## Programme de notre formation Clean Code

### Introduction au Clean Code

- Qu'est-ce qu'un code propre ?
- Le bon et le mauvais codeur
- Pourquoi utiliser le Clean Code ? Mesurer le coût d'un code défaillant
- Les points clés du Clean Code

### Les principes SOLID

- Historique
- Le principe de substitution de Liskov
- L'inversion de dépendance
- La responsabilité unique
- La séparation des interfaces
- Le principe ouvert fermé
- Don't Repeat Yourself (DRY)

### Les conventions de nommage

- Des noms simples et clairs
- Éviter l'ambiguïté
- Des noms pouvant être recherchés et prononcés
- Les pièges à éviter
- Le nommage des classes
- Le nommage des méthodes

### Une bonne structure du code

- L'importance d'une bonne structure
- La structure des données

- La structure des objets
- Le formatage vertical
- Le formatage horizontal

## Les fonctions

- La structure
- Éviter les effets secondaires
- Un niveau d'abstraction par fonction
- Utiliser les bons arguments
- Utiliser les bonnes déclarations
- Séparer les commandes

## Les classes et les commentaires

- Organiser ses classes
- L'importance de commenter
- Bon commentaire vs mauvais commentaire
- Ne pas utiliser le commentaire pour améliorer son code

## Test Driven Development (TDD)

- Pourquoi utiliser le TDD ?
- Les principes fondamentaux
- Réaliser des tests clean
- L'acronyme FIRST
- Présentation des outils
  - junit
  - Mochajs
  - Mockito
  - Jest
  - Pytest

## Behaviour Driven Development (BDD)

- Présentation du BDD
- Pourquoi automatiser ses tests ?
- BDD vs TDD
- Définir son projet et les prés-requis
- Les outils de BDD
  - Behat
  - Zephyr SScale
  - Easy B
  - JDave
  - JBehave

## Gestion du temps

- Les bénéfices et les risques des réunions
- Quand se réunir et comment ?
- Garder son focus et éviter le multitâche
- Time boxing
- Automatiser les tâches répétitives
- Utiliser les IDE

## Collaboration

- Utiliser les outils de communication
- Assigner des rôles spécifiques
- Les bonnes pratiques de communication
- Partager ses connaissances

## Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

## Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

## Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

## Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

## Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

## Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.

