

Clean Code : principes SOLID & refactoring

Référence : **NUMR010**

Durée : **2 jours (14 heures)**

Certification : **Aucune**

CONNAISSANCES PRÉALA

Avoir une bonne maîtrise du langage Java et de ses API fondamentales

Avoir déjà été confronté aux problèmes de gestion de bases de code anciennes, volumineuses et non (ou mal) testées est un avantage

PROFIL DES STAGIAIRES

Développeurs expérimentés

OBJECTIFS

Comprendre la notion de dette technique et les principes SOLID

Comprendre les enjeux de l'écriture de tests automatisés et le coût de la mise en production de code non testé automatiquement

Maîtriser l'écriture de tests automatisés, unitaires et tests d'intégration ainsi que les principes et patterns du « clean code » en programmation objet

Comprendre le principe du travail à partir de katas et les pratiques de l'eXtreme programming : pair programming, coding dojos

Etre capable d'appliquer ces principes et patterns au développement de nouveau code (greenfield development) et au code existant (brownfield development)

CERTIFICATION PRÉPARÉE

Aucune

MÉTHODES PÉDAGOGIQUES

Mise à disposition d'un poste de travail par stagiaire

Remise d'une documentation pédagogique papier ou numérique pendant le stage

La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions

La partie codage porte principalement sur le codage de « Katas », un exercice essentiel pour s'exercer à la pratique du refactoring

Le suivi de cette formation donne lieu à la signature d'une feuille d'emargement

FORMATEUR

Consultant-formateur expert en Clean Code et au codage de Katas

MÉTHODES D'ÉVALUATION DES ACQUIS

Auto-évaluation des acquis par le stagiaire via un questionnaire

Attestation des compétences acquises envoyée au stagiaire

Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

1. Enjeux du clean code : maîtriser la dette technique

Notion de dette technique

Impact de la dette technique sur la maintenance des applications

Impact de la dette technique sur le cycle SCRUM

2. Importance des tests : approches TDD et BDD

Cycle TDD

Écriture des tests JUnit

Écriture des tests Cucumber

3. Introduction des principes du Clean Code

Principes SOLID

Principes de l'eXtreme Programming

4. Travail sur le code legacy (développement brownfield)

Spécificités du travail sur code legacy

Exemple du Kata Rental Movie (code legacy)

Application du pattern Strategy, principe Open Closed

Bilan : détecter les manquements au SRP

5. Application au nouveau code (développement greenfield)

Application au kata Mars Rover

Bilan : application du principe Open Closed

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel:01.71.19.70.30) ou par mail à referent.handicap@edugroupe.com pour vos éventuels besoins d'aménagements, afin de vous offrir la meilleure .