Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

Clean Code, principes SOLID et refactoring

La pérennité d'une base de code repose sur la qualité de son écriture et sur la maîtrise du niveau de sa dette technique. La bonne connaissance des principes SOLID permet de percevoir les non-qualités dans les bases de code existantes et de comprendre les règles du Clean Code. Cette compréhension permet de sécuriser et de garantir le succès de l'étape de refactoring, objet de cette formation.
La formation s'adresse à des développeurs Java expérimentés ayant une bonne connaissance du langage et de ses API fondamentales. Une première confrontation aux problèmes de gestion de bases de code anciennes, volumineuses et non (ou mal) testées est un avantage.

A propos de José Paumard :
José est universitaire et indépendant depuis une vingtaine d'années, Java Champion, Java Rockstar et auteur pour Pluralsight . Coach en Software Craftsmanship depuis 4 ans, entre autres dans une grande banque Parisienne, il publie un catalogue raisonné de katas craft sur GitHub.io. Il enseigne les technologies Java / Java EE à l'université Paris 13 en école d'ingénieur et intervient auprès de sociétés en formation, architecture et expertise. Il est speaker invité à JavaOne (San Francisco), et intervient régulièrement à Devoxx (Belgique, France, Angleterre) et dans de nombreuses autres conférences européennes (JPrime, JFokus, JTres, etc…) . Il publie des articles pour Java Magazine, Oracle Technology Network et anime le blog technique « Java le soir ». Il est cofondateur de Devoxx France qu'il a coorganisé les 3 premières années. Il est actuellement trésorier de l'association BJPC, organisatrice des soirées du Paris JUG. Il est enfin membre du groupe d'experts pour CDI 2.0 (Context dans Dependency Injection, JSR 365).

Détails

- Code : MP-CCD
- Durée : 2 jours (14 heures)

Public

- Consultants
- Consultants informatiques
- Developpeurs
- Développeurs expérimentés
- Professionnels de l'IT

Pré-requis

- 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

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

Programme

Remarque

- Le programme se compose de 30% présentation sur slides, 70% de codage
- La partie codage porte principalement sur le codage de « Katas », un exercice essentiel pour s'exercer à la pratique du refactoring

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

Importance des tests : approches TDD et BDD

- Cycle TDD
- Écriture des tests JUnit
- Écriture des tests Cucumber

Introduction des principes du Clean Code

- Principes SOLID
- Principes de l'eXtreme Programming

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

Application au nouveau code (développement greenfield)

- Application au kata Mars Rover
- Bilan : application du principe Open Closed

Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

Modalités

- Type d'action :Acquisition des connaissances
- Moyens de la formation :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- Modalités pédagogiques :Exposés – Cas pratiques – Synthèse
- Validation :Exercices de validation – Attestation de stages