

## Software Craftmanship

Software Craftmanship : Clean Code, TDD, BDD et principes SOLID

Un logiciel peut être parfaitement fonctionnel mais poser de nombreux problèmes de fiabilité et de maintenabilité. Le Software Craftmanship ou "l'artisanat du logiciel" propose un ensemble de méthodes et d'approches de haute qualité pour concevoir et construire des bases de code de tous volumes maintenables à coût constant. Ces méthodes s'appuient sur les principes SOLID, les principes du Clean Code et de l'eXtreme Programming.

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.

**La formation a été conçue par José Paumard. Elle sera animée par l'un de nos consultants-formateurs seniors.**

### 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 Craftmanship 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 Devovx (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 Devovx 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-SCS
- Durée : 3 jours ( 21 heures )

Public  
• Développeurs

#### Pré-requis

- Une première confrontation aux problèmes de gestion de bases de code anciennes, volumineuses et non (ou mal) testées est un avantage.

### Objectifs

- Comprendre la notion de développement dirigé par la valeur
- Maîtriser les différentes étapes du développement TDD
- Maîtriser les principes SOLID et les patterns du « clean code » en programmation objet
- Comprendre 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)

### Programme

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 TDD / BDD.

#### Introduction à la pratique du développement dirigé par les tests, le cycle de développement TDD

- Exemple du Kata FizzBuzz
- Développement du Kata
- Bilan : les étapes de la pratique du TDD

#### Introduction des principes du Clean Code et des principes SOLID

#### Pratiques de l'eXtreme Programming

- Pratiques des Katas
- Pratiques des Coding Dojo
- Pratiques du Pair Programming

### Le principe Open / Closed : application au pattern Strategy

- Exemple du Kata RPN Calculator
- Développement du Kata
- Bilan : le pattern Strategy et son implémentation

### Travail sur le code legacy, Single Responsibility Principe

- Spécificités du travail sur code legacy
- Exemple du Kata Rental Movie (code legacy)
- Bilan : détecter les manquements au SRP

### Utilisation de Gherkin / Cucumber pour l'écriture de

### tests

- Écriture de tests en Gherkin : méthodes, organisation, syntaxe
- Écriture de classes Cucumber pour l'exécution de ces tests
- Fonctionnalités avancées : tests paramétrés, tables de données, tags
- Intégration avec Maven, génération de rapports de tests

### Mise en oeuvre sur un kata complexe : Mars Rover

- Ecriture des tests avec Cucumber
- Résolution du kata
- Conclusion

## Modalité

- Stage pratique en présentiel
- Stage pratique en distanciel
- Nombre de stagiaires minimum : 3
- Nombre de stagiaires maximum : 6

## Méthodes pédagogiques

- Exposés
- Cas pratiques
- Echanges d'expérience
- Mises en situation

## Profils des intervenants

- Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.

## Modalités d'évaluation

- Evaluation des acquis de la formation par le biais de cas pratiques et/ou mises en situation.
- Attestation de formation remise à chaque participant.

## Démarche qualité

- Questionnaire d'évaluation de satisfaction à chaud complété par chaque participant à l'issue de la formation.

## Moyens pédagogiques

- Salle équipée de PC (1 poste par stagiaire), vidéo-projecteur.
- Espace de pause.

Dernière mise à jour le 09/06/2021