

Maintenance Applicative



Mathieu Barberot

Tech Lead @ Sully-Group



Marc Bouvier

Développeur Java / Typescript
Senior indépendant

But de ce cours

Vous fournir des *outils* et des *méthodes* pour travailler sur du code écrit par qqn d'autre

Challenge

- garder la connaissance $\Rightarrow$ *doc, code bien nommé, revue de code*
- donner confiance $\Rightarrow$ *tests, ci*
- gagner en efficacité $\Rightarrow$ *qualité du code, modularité*

Toutes ces pratiques sont synergiques

Méthodologie

1. Imiter

2. Assimiler

3. Innover

⇒ Amélioration continue

Fonctionnement

- Quelques slides pour la théorie
- Un peu de live coding pour les premiers pas
- Application sur des exercices
- Mise en place sur votre projet

Le projet



Phénix

Le projet

- Une API REST de gestion des cours de potion

Modalités

- Chaque semaine une nouvelle tâche à faire
 - En appliquant les notions vues en cours le jour même
 - Et en appliquant les notions des semaines précédentes

Livrable

- Le code source
 - Un fichier `nom.prenom.zip`
 - Contient : le dossier `projet`, sans le `.m2/`, ni le `target/`
 - Déposé dans le formulaire de rendu sur Moodle
- Le build et les tests doivent passer

Evaluation

Barème général :

- 5 notions = 5 points, x4 pour avoir une note sur 20
 - Règle Scout : 1 point
 - Tests : 1 point
 - DDD : 1 point
 - SOLID : 1 point
 - Cohésion-Couplage : 1 point

Evaluation

- Appréciation : entre -1 et +1 point
 - Bonus : Respect des consignes, excellent travail
 - Malus : Non-respect des consignes, travail rendu en retard

Merci de votre attention