

Tester les interactions

Tester en isolation Le test des interactions est un moyen pour tester une classe *en isolation*, cad *indépendamment des classes dont elle dépend*. Exemple : pour une classe `MachineCrous` qui utilise une classe `CarteCrous`, on souhaite tester les fonctionnalités de `MachineCrous` sans du tout utiliser la classe `CarteCrous`. Si on teste `MachineCrous` en utilisant directement `CarteCrous`, c'est plutôt du test d'intégration.

Pourquoi tester en isolation ?

- Si un test de `MachineCrous` échoue, doit-on chercher l'erreur dans `MachineCrous` ou dans `CarteCrous` ?
- Les dépendances ne sont peut-être pas encore développées.
- Les dépendances sont parfois *difficilement contrôlables* :
 - comment simuler une panne de réseau ?
 - que faire si la dépendance a un comportement aléatoire ?
- Les dépendances sont parfois lourdes à mettre en place : base de données.

Tester les interactions Le principe est de :

- remplacer les implantations des dépendances par des *doublures*, ou *mocks*. Les mocks sont des objets du test qui simulent les objets du domaine pour la durée du test.
- vérifier que la classe à tester appelle comme il faut les méthodes des mocks.
- **attention** : on mocke les dépendances, **on ne mocke pas la classe à tester !**

Dépendances explicites Pour pouvoir remplacer les implantations par des doublures, il faut aussi que les dépendances apparaissent explicitement (par exemple en argument du constructeur ou d'une méthode). Ce pattern s'appelle *injection des dépendances* : il permet un *couplage faible* entre les classes et favorise les *architectures testables*.

Dépendances à des abstractions Il faut que la classe à tester dépende d'*interfaces* (abstractions) et non directement des implantations.

Test des interactions / Test d'état On n'a pas systématiquement besoin de l'un ou de l'autre, on a parfois besoin des deux. Le test d'interaction demande obligatoirement un mock.

Description des mocks Les mocks peuvent être codés à la main. Ils peuvent avantageusement être générés automatiquement à partir d'une description de leur comportement (*stubbing*). Les vérifications à faire en interrogeant les mocks peuvent aussi être décrites. cf Mockito.