

Les tests unitaires

ACHRAF BAKKARI

KLARA NOSAN

ELMEHDI YAKOUT

introduction



Plan

- ▶ C'est quoi un test
- ▶ L'importance des tests
- ▶ Comment faire un bon test
- ▶ Comment écrire un test
- ▶ Les types des asserts
- ▶ Comment exécuter un test
- ▶ Résultats des tests

C'est quoi un test

Le test unitaire ou test de composants est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

L'importance des tests

- ▶ Informer le développeur si les changements qu'il a fait ont cassé le système de l'application
- ▶ Eviter les bugs dans une application

Comment faire un bon test

- ▶ Un test doit être répété autant fois qu'on veut et donne le même résultat
- ▶ Un test doit marcher tout seul sans aucune intervention
- ▶ On doit écrire un test pour chaque fonctionnalité
- ▶ Un test doit être compréhensible par tout le monde

Comment écrire un test

- ▶ La classe de test doit être une sous classe de TestCase :
 - ▶ **TestCase subclass: #MyClassTest**
- ▶ La méthode de test doit se commencer par le mot test (pour être inclus dans la suite des tests)

Les types des asserts

- ▶ `assert: aBoolean`
- ▶ `deny: anExpression == assert: anExpression not`
- ▶ `assert: expression equals: aValue == assert: expression = aValue`
- ▶ `should: expression raise: erreur`
- ▶ `shouldnt: expression raise: erreur`

Comment exécuter un test

- ▶ System Browser
- ▶ Test Runner
- ▶ Playground

Résultats des tests

- ▶ Passed (vert)
- ▶ Failed (jaune)
- ▶ Erreur (rouge)



Merci pour votre attention