

## PHPUnit

Fayçal Hamdi

Maître de conférences – CNAM – Paris

*à partir de transparents de Pascal Graffion*

## Objectifs

Framework de tests écrit pour faciliter :

- l'écriture de tests
  - tests unitaires
  - tests de recette
- l'exécution de ces tests
- l'exploitation de ces tests

## Origine

- Développement piloté par les tests (Test Driven Development)
- Framework Junit écrit en Java par :
  - Kent Beck (Extreme programming)
  - Erich Gamma (Design patterns)
  - ([www.junit.org](http://www.junit.org))
- Désormais décliné dans tous les langages!

## Pourquoi tester?

- Pour gagner du temps!
- C'est plus efficace que de débbuger
- Permet de valider le logiciel
- Permet de valider les remaniements de code
- Permet de détecter les régressions

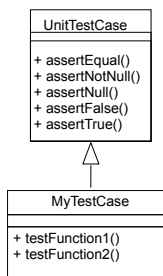
Test Driven Development (TDD) :  
on écrit les tests avant de coder l'application!

2 frameworks disponibles :

- PHPUnit du repository PEAR:  
<http://pear.php.net/package/PHPUnit/>
- SimpleTest  
[http://www.lastcraft.com/simple\\_test.php](http://www.lastcraft.com/simple_test.php)  
<http://simpletest.sourceforge.net/fr/>  
Plus riche

## TESTS UNITAIRES AVEC SIMPLETEST

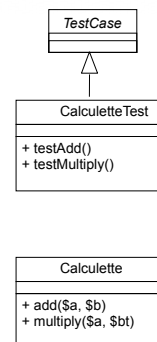
## UnitTestCase



Ecrire une classe de test consiste à :

- hériter de la classe UnitTestCase
- implémenter plusieurs méthodes nommées test<f>()
- écrire des assertions :
  - assertTrue(1 > 0)
  - assertEquals(7, 3+4)

## Premier exemple



```

<?php
require_once('simpletest/unit_tester.php');
require_once('simpletest/reporter.php');
require_once('Calculette.class.php');
class CalculetteTest extends UnitTestCase {

    function testAdd() {
        $calc = new Calculette();
        $expected = 7;
        $actual = $calc->add(3, 4);
        $this->assertEquals($expected, $actual);
        $this->assertEquals(1, $calc->add(3, -2));
    }

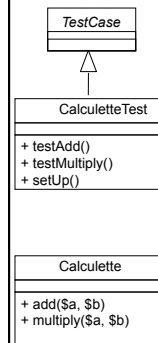
}

$test = new CalculetteTest();
$test->run(new TextReporter());
?>
    
```

## Fixture

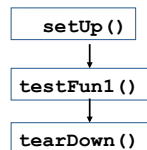
- Ensemble des objets utilisés dans plusieurs fonctions d'une même classe de test
- Chaque objet utilisé est alors
  - déclaré en tant que variable d'instance de la classe de test
  - initialisé dans la méthode setUp()
  - éventuellement libéré dans la méthode tearDown()

## Exemple avec fixture

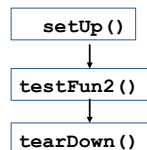


```
class CalculetteTest extends UnitTestCase {
    private /* Calculette */ $calc; // fixture
    function setUp() {
        $this->calc = new Calculette();
    }
    function tearDown() {
        $this->calc = null;
    }
    function testAdd() {
        $expected = 7;
        $actual = $this->calc->add(3, 4);
        $this->assertEqual($expected, $actual);
        $this->assertEqual(6, $this->calc->add(3, 2));
    }
    function testMultiply() {
        $actual = $this->calc->multiply(3, 4);
        // ...
    }
}
```

## setUp et tearDown



- setUp est exécutée avant chaque méthode testXXX



- tearDown est exécutée après chaque méthode testXXX

## Exécuter un test isolément

- On instancie la classe de test

```
$test = new CalculetteTest();
```

- On exécute le test en lui passant un rapporteur

```
$test->run(new TextReporter());
```

- Il existe différents rapporteurs :
  - TextReporter,
  - HtmlReporter, ...
- Toutes les méthodes dont le nom commence par test sont exécutées

## Exécution du test avec un TextReporter

```
$test = new CalculetteTest();  
$test->run(new TextReporter());
```

```
$ php calc_test2.php  
calcullettetest  
1) Equal expectation fails because [Integer: 6] differs from  
[Integer: 5] by 1 at [calc_test2.php line 18]  
in testadd  
FAILURES!!!  
Test cases run: 1/1, Passes: 2, Failures: 1, Exceptions: 0
```

Nombre d'assertions  
vérifiées

Nombre d'échecs

Nombre d'erreurs  
ou d'exceptions (PHP5)

le cnam

13

## Exécution du test avec un HtmlReporter

```
$test = new CalculetteTest();  
$test->run(new HtmlReporter());
```

calcullettetest

Fail: testadd -> Equal expectation fails because [Integer: 6] differs from [Integer: 5] by 1 at [d:\home\web\tpds\tests\calc\_test3.php line 18]

1/1 test cases complete: 2 passes, 1 fails and 0 exceptions

Nombre d'assertions vérifiées

Nombre d'échecs

Nombre d'erreurs ou d'exceptions (PHP5)

le cnam

14

## Exemple avec Failures et Exceptions (1)

```
<?php  
require_once('simpletest/unit_tester.php');  
require_once('simpletest/reporter.php');  
class test3 extends UnitTestCase {  
    function testPass(){  
        $this->assertEqual(3,2+1, "This should pass");  
    }  
    function testFail(){  
        $this->assertEqual(4,2+1,"This should fail");  
        $this->assertEqual(3,2+1, "This should pass");  
    }  
    function testMissingArgumentException(){  
        strlen();  
        $this->assertEqual(3,2+1, "This should pass");  
    }  
    function testDivideByZeroException() {  
        $i = 3/0;  
        $this->assertEqual(3,2+1, "This should pass");  
    }  
}  
$test = new test3();  
$test->run(new TextReporter());  
?>
```

le cnam

15

## Exemple avec Failures et Exceptions (2)

```
$ php -n -d include_path=".;c:\Applications\php\includes"  
test3.php 2>err  
test3  
1) This should fail at line [9] in testfail  
Exception 1!  
Unexpected PHP error [Wrong parameter count for strlen()]  
severity [E_WARNING] in [test3.php] line [13]  
Exception 2!  
Unexpected PHP error [Division by zero] severity [E_WARNING] in  
[test3.php] line [17]  
FAILURES!!!  
Test cases run: 1/1, Passes: 4, Failures: 1, Exceptions: 2
```

Nombre d'assertions  
vérifiées

Nombre d'échecs

Nombre d'erreurs

le cnam

16

### Exemple avec Failures et Exceptions (3)

```
function testFail(){
    $this->assertEqual(4,2+1,"This should fail");
    $this->assertEqual(3,2+1, "This should pass");
}
```



```
1) This should fail at [test3.php line 10]
   in testFail
(=> Passes: +1, Failures: +1, Exceptions: +0 )
```

Note : le 2-ième assertEquals est exécuté même si le premier a échoué!

### Exemple avec Failures et Exceptions (4)

```
function testMissingArgumentException() {
    strlen();
    $this->assertEqual(3,2+1, "This should pass");
}
```



**Exception 1!**  
Unexpected PHP error [Wrong parameter count for strlen()] severity [E\_WARNING] in [test3.php] line [17]  
(=> Passes: +1, Failures: +0, Exceptions: +1 )

Note : assertEquals est exécutée bien qu'une erreur PHP ait été détectée (comportement != de Junit)!

### Exemple avec Failures et Exceptions (5)

```
function testDivideByZeroException() {
    $i = 3/0;
    $this->assertEqual(3,2+1, "This should pass");
    //exécuté!
}
```

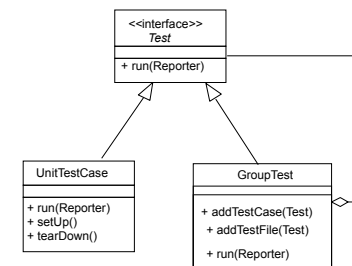


**Exception 2!**  
Unexpected PHP error [Division by zero] severity [E\_WARNING] in [test3.php] line [13]  
(=> Passes: +1, Failures: +0, Exceptions: +1 )

Note : assertEquals est exécutée bien qu'une erreur PHP ait été détectée (comportement != de Junit)!

### GroupTest

#### Un GroupTest est une composition de Test



(Composite design pattern)

But : exécuter directement plusieurs tests

## Exécuter un ensemble de tests : addTestCase

- Soient test1 et test2 2 classes héritant de UnitTestCase définies dans test1.php et test2.php

```
<?php
require_once('simpletest/unit_tester.php');
require_once('simpletest/reporter.php');
require_once('test1.php');
require_once('test2.php');
$test = new GroupTest('All tests');
$test->addTestCase(new test1());
$test->addTestCase(new test2());
$test->run(new TextReporter());
?>
```

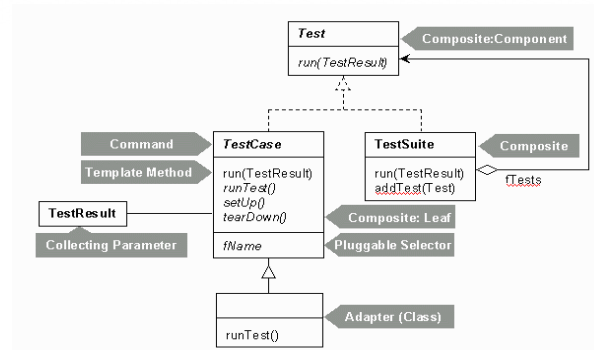
- addTestCase les ajoutent à \$test

## Exécuter un ensemble de tests : addTestFile

```
<?php
require_once('simpletest/unit_tester.php');
require_once('simpletest/reporter.php');
$test = new GroupTest('All tests');
$test->addTestFile('test1.php');
$test->addTestFile('test2.php');
$test->run(new TextReporter());
?>
```

- Un addTestFile pour chaque fichier (qu'on n'a pas à déclarer auparavant)

## JUnit et Design patterns

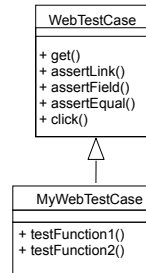


(<http://junit.sourceforge.net/junit3.8.1/doc/cookstour/cookstour.htm>)

## TESTS WEB AVEC WEBTESTCASE

Permet de récupérer une page web et de tester les actions sur cette page :

- vérifier son contenu,
- naviguer (suivre les liens hypertexte),
- gérer les formulaires ...



Ecrire un test web consiste à :

- hériter de la classe WebTestCase
- obtenir la page web
  - get(\$url)
- implémenter plusieurs méthodes nommées test<f>()
  - assertText(\$textToFind)
  - assertField('contents', \$contents);
- écrire des assertions :

- get(\$url, \$parameters = false)  
envoie une requête GET avec ces paramètres
- clickSubmit(\$label)  
clique sur le premier bouton avec cette
- clickSubmitById(\$id)  
clique sur le bouton avec cet attribut d'identification
- clickLink(\$label)  
clique sur une ancre avec ce texte d'étiquette visible
- clickLinkById(\$id)  
clique sur une ancre avec cet attribut d'identification
- ...

- assertText(\$text)  
passe si le texte est trouvé dans la partie visible de la page
- assertPattern(\$pattern)  
passe si l'expression rationnelle Perl \$pattern matche une partie du contenu de la page (recherche dans tout le code HTML)
- assertTitle(\$title)  
passe si le titre de la page correspond exactement
- assertLink(\$label)  
passe si un lien avec ce texte est présent
- assertField(\$name, \$value)  
passe si une balise input avec ce nom contient cette valeur
- ...

## Exemple de test web (1)

```
<?php
require_once('simpletest/unit_tester.php');
require_once('simpletest/reporter.php');
require_once('simpletest/web_tester.php');

class CreateTicketTests extends WebTestCase {

    function setUp() {
        $createTicketURL = $this->getProjectRootUrl()
            . 'tp2/saisieTicket.html';
        $this->get($createTicketURL);
    }
    // ...
}
```

## Exemple de test web (2)

Veuillez décrire votre demande d'intervention:

Application:  YAHP

Priorité:

Type: ☒ Anomalie ☐ Demande d'évolution

Résumé:

Description détaillée:

Pièce-jointe:  Parcourir...

Envoyer

## Exemple de test web (3)

```
// fonction utilitaire
function fillCreateForm($applicationOther, $priority,
    $oneLiner, $contents, $application = -1) {
    $this->setField('applicationOther',
        $applicationOther);
    // test si le champ existe bien
    $this->assertField('applicationOther',
        $applicationOther);
    $this->setField('oneLiner', $oneLiner);
    // test si le champ existe bien
    $this->assertField('oneLiner', $oneLiner);
    // ...
}
```

## Exemple de test web (4)

```
// un test web
function testCreateUrgentTicketOnYahd() {
    $applicationOther = 'YAHD';
    $oneLiner = 'oneLiner contents';
    $priorityCode = 4; $priorityText = 'Urgente';
    $contents = 'contents field content';
    $this->fillCreateForm($applicationOther,
        $priorityCode, $oneLiner, $contents);
    // submit the form
    $this->click('Envoyer');
    // check the contents of the new page
    $this->assertText($applicationOther);
    $this->assertText($priorityText);
    // ...
}
```