

# PHPUnit

P. Graffion



**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 débbugger**
- **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!**

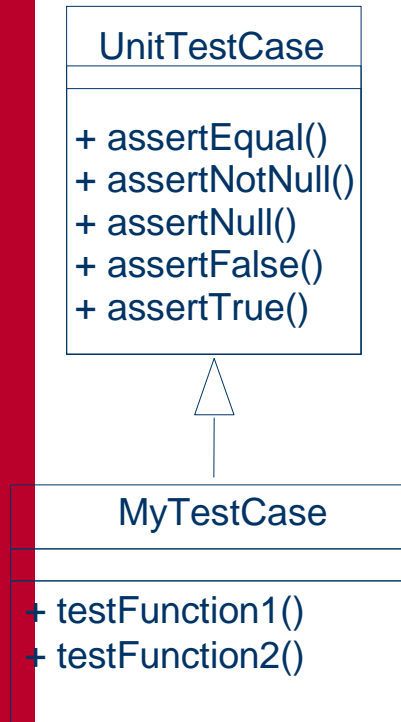
## 2 frameworks principaux disponibles :

- **PHPUnit du repository PEAR:**  
**<https://phpunit.de/>**  
**Le standard désormais**
- **SimpleTest**  
**<http://simpletest.org/>**  
**Plus riche**

# Tests unitaires avec SimpleTest



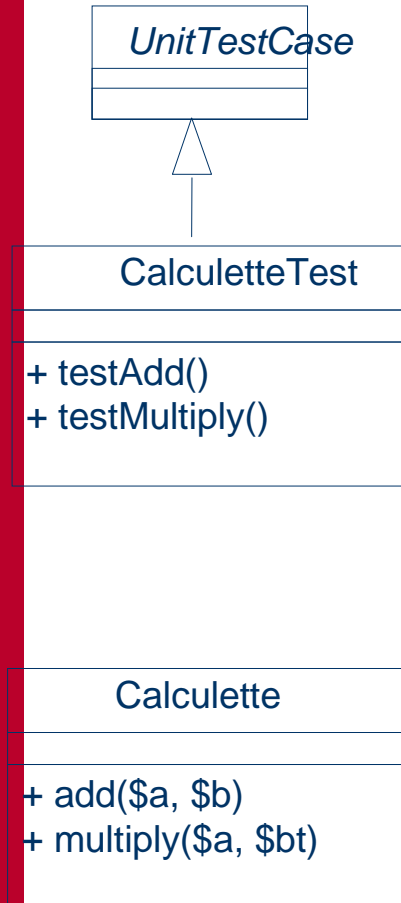
# UnitTestCase



**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);
        $this->assertEqual($expected, $actual);
        $this->assertEqual(1, $calc->add(3, -2));
    }
}

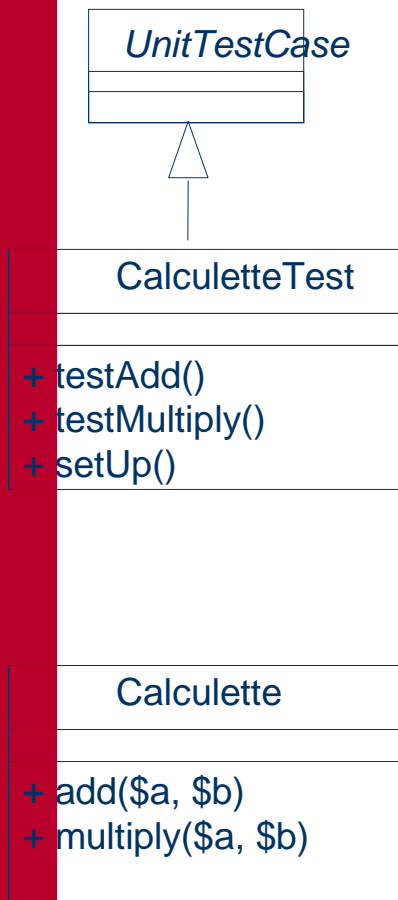
$test = new CalculetteTest();
$test->run(new TextReporter());
?>
  
```



# 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()

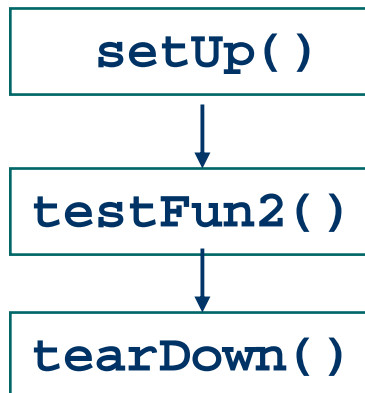
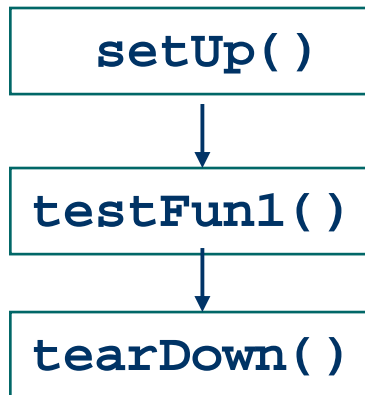
# Exemple avec fixture



```

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  
calculettetest  
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)

# Exécution du test avec un HtmlReporter

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

Problems Console Bookmarks PHP Browser Properties Search CVS Resource History

http://localhost/tpdsi/tests/calc\_test3.php

## calculettetest

Fail: testadd -> Equal expectation fails because [Integer: 6] differs from [Integer: 5] by 1 at [d:\home\web\tpdsi\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)

## Raccourci : autorun.php

```
<?php
require_once('simpletest/autorun.php');
require_once('Calcullette.class.php');
class CalculletteTest extends UnitTestCase {

    function testAdd() {
        $calc = new Calcullette();
        $this->assertEqual(1,$calc->add(3,-2));
    }
}
// $test = new CalculletteTest();
// $test->run(new TextReporter());
?>
```

- le test est déclenché implicitement.
- le reporter utilisé est un TextReporter en ligne de commandes ou un HtmlReporter dans un navigateur.

# 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());
?>
```



## Exemple avec Failures et Exceptions (2)

```
$ php -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

## 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!

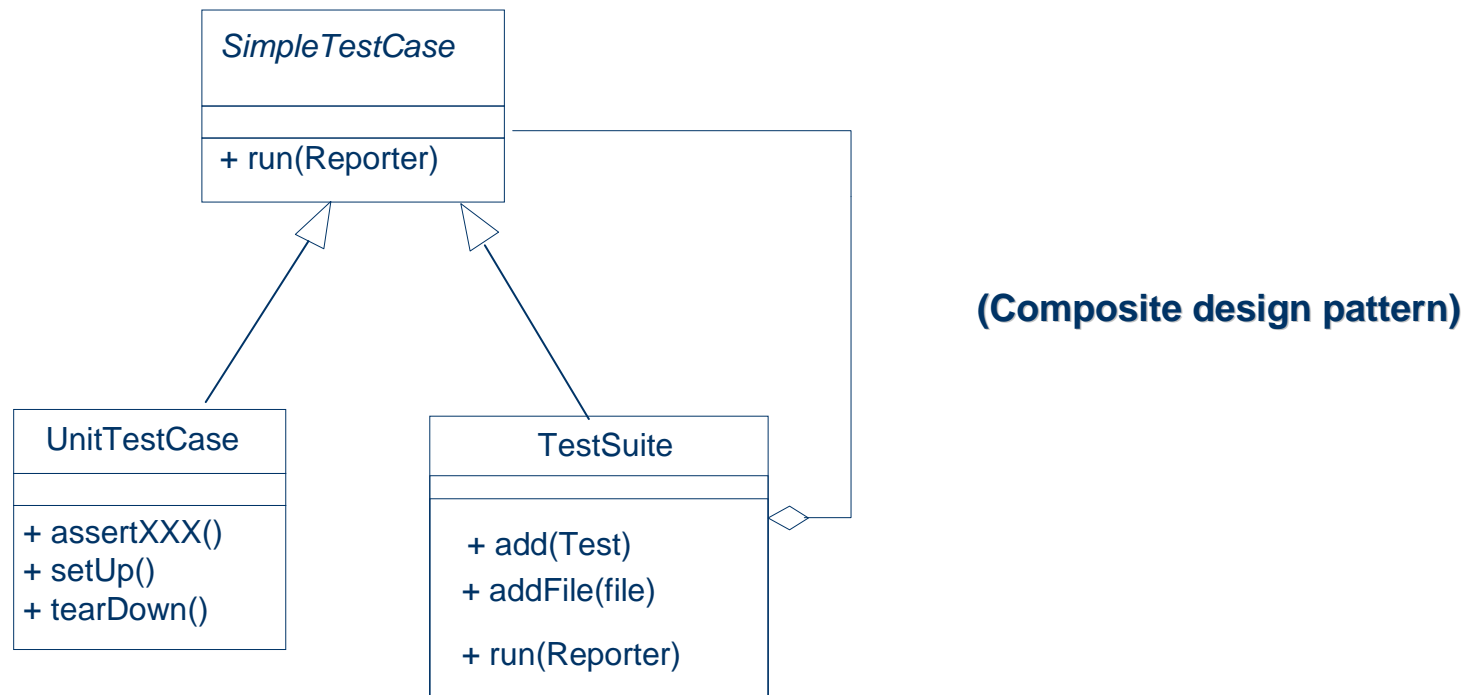
Unexpected PHP error [Division by zero] severity  
[E\_WARNING] in [test3.php] line [13]

(=> Passes: +1, Failures: +0, Exceptions: +1 )

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

# TestSuite

**Une TestSuite est une composition de Test**



**But : exécuter directement plusieurs tests**

# Exécuter un ensemble de tests :addTestCase

- Soient Test1 et Test2 deux classes héritant de UnitTestCase définies dans test1.php et test2.php

```
<?php
require_once('simpletest/autorun.php');
require_once('test1.php');
require_once('test2.php');
class Suite1 extends TestSuite {
    function __construct() {
        parent::__construct();
        $this->add(new Test1());
        $this->add(new Test2());
    }
}
?>
```

- add les ajoutent à la suite

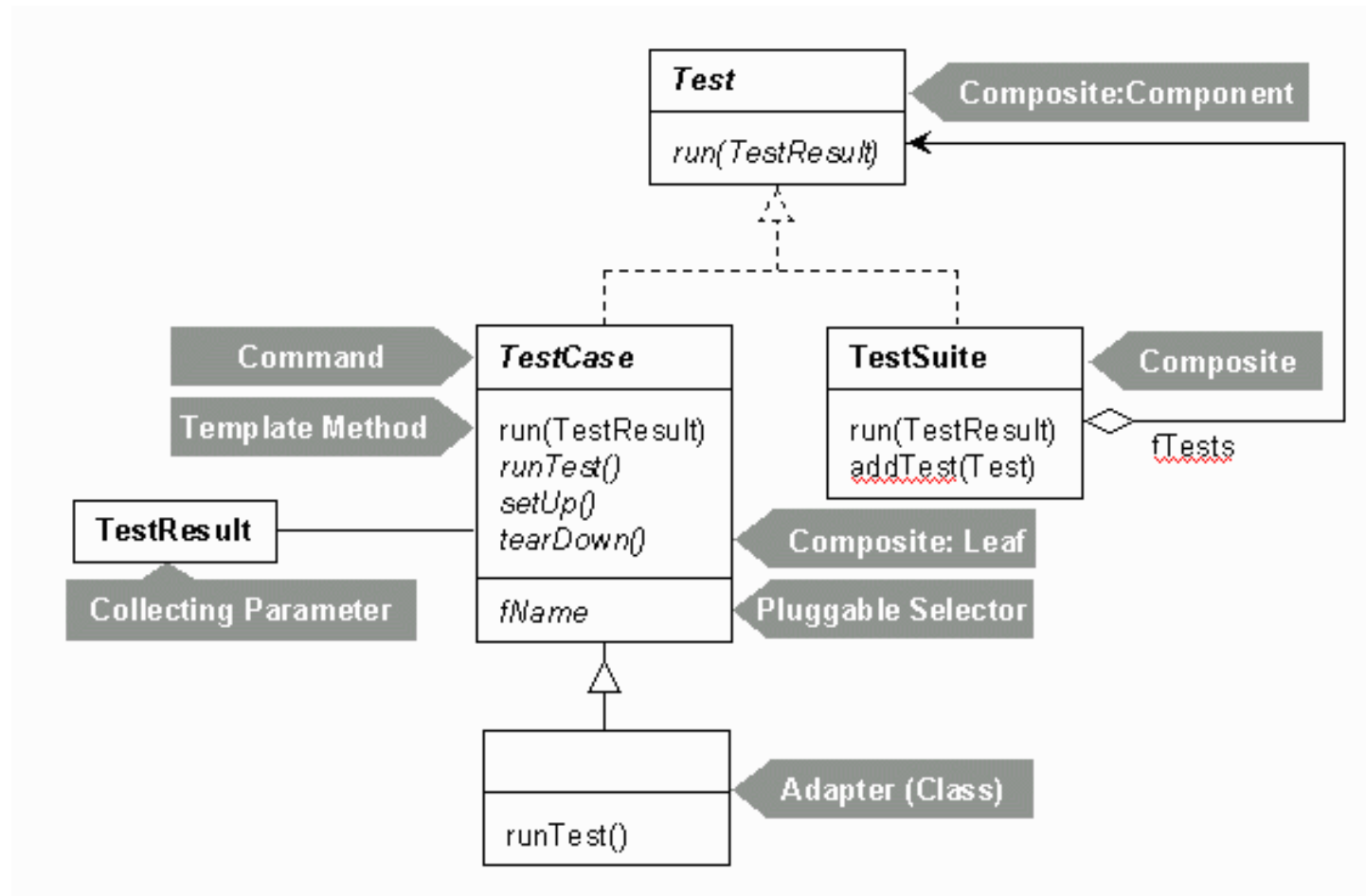
# Exécuter un ensemble de tests : addFile

```
<?php
require_once('simpletest/autorun.php');

class Suite2 extends TestSuite {
    function __construct() {
        parent::__construct();
        $this->addFile('test1.php');
        $this->addFile('test2.php');
    }
}
?>
```

- **Un addFile 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 SimpleTest

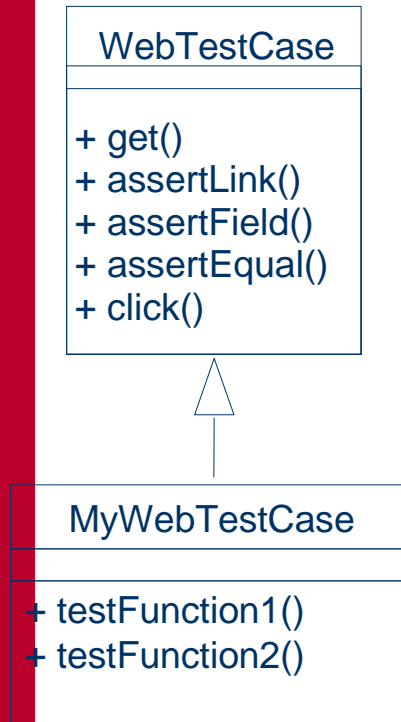


# Testeur web - WebTestCase

**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 ...

# WebTestCase



**Ecrire un test web consiste à :**

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

## WebTestCase – méthodes de navigation

- **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
- ...

# WebTestCase – méthodes de vérification

- **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)

**Veillez décrire votre demande d'intervention:**

Application:	--Autre--	YAHP	<code>&lt;input name=applicationOther</code>	
Priorité:	Urgente		<code>&lt;select name=priority</code>	
Type:	<input checked="" type="radio"/> Anomalie <input type="radio"/> Demande d'évolution			
Résumé:	Les tickets ne sont pas sauvegardés			<code>&lt;input name=oneLiner</code>
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);
    // ...
}
```