



UCE Techniques de Tests

TP 2-B

UE Génie Logiciel Avancé
M1 ILSEN



- Objectif
 - Ecrire des tests unitaires en mode « Extreme Programming »
 - Utiliser les possibilités de **fixtures** offertes par SimpleTest
 - Utiliser les objets fantaisies "mock objects" de SimpleTest
 - A la fois en tant qu'acteur et critique
 - Regrouper les cas de test en une suite de tests
- La classe à implémenter RevolverA5Coups
 - On veut étudier son comportement lorsque l'on ne charge qu'une cartouche, dans ce cas les méthodes de RevolverA5Coups sont
 - viderBarillet() chargerUneCartouche() tournerAleatoirementBarillet()
 - rangerSousClef () appuyerSurDetente() qui retourne BANG ou CLIC

TP 2-B



- La classe Hasard
 - Cette classe est utilisée par `tournerAleatoirementBarillet()`
 - Méthode `tirageEntre2Bornes($a,$b)`
 - Cette méthode renvoie un nombre entier compris entre les deux bornes a et b incluses
- Vous devez dans cette partie du TP
 - Concevoir vos différents cas de tests
 - Ecrire vos différents cas de test (un cas par fichier)
 - Les regrouper dans une suite de tests (une seule instruction)
 - Rendre plus lisible le rapport de la suite de test
 - Respecter les règles de prudence pour la manipulation d'une arme

TP 2-B



- Précisions sur RevolverA5Coups
 - tournerAleatoirementBarillet(Hasard \$hasard)
 - Prend un objet de la classe Hasard comme argument
 - RangerSousClef () est une méthode static de RevolverA5Coups
- Un cas de test par fichier, pourquoi ?
 - Pas de justification technique
 - Pour utiliser les possibilités de création d'une suite de test
 - Définissez les fixitures dans une classe mère
 - Créer la classe MockHasard, ranger le revolver après chaque test !
 - Dérivez les cas de tests de cette classe mère (un test par classe fille)
 - « Collectez » les cas de tests dans une suite de tests

TP 2-B



- Exemple de test pour viderBarillet

```
<?php
require_once('../simpletest/autorun.php');
require_once(RevolverA5Coups.php');
class TP2BTestViderBarillet extends TP2BUnitTestCase { // si le barillet est bien vide pas de BANG !
function testViderBarillet() {
    $revolver = new RevolverA5coups();
    for($i = 0;$i < 5;$i++) {
        $this->assertEqual($revolver->appuyerSurDetente(),'CLIC');
    }
}
}
?>
```

TP 2-B



- Exemple d'objet mock de la classe Hasard

```
<?php
require_once('../simpletest/autorun.php');
require_once('Hasard.php');
Mock::generate('Hasard');
class MyTestCase2 extends UnitTestCase {
function testSomething() {
    $hasard = new MockHasard(); // création objet mock
    $hasard->returns('tirageEntre2Bornes',1); // défini mock acteur
    $hasard->expect('tirageEntre2Bornes',array(1,5)); // défini mock critique
    $hasard->tirageEntre2Bornes(1,5); // renvoi toujours 1 et vérifie que les bornes sont bien 1 et 5
}
}
?>
```