



Informatique

IUT Nord Franche-Comté

Tests unitaires en Java

Bases de syntaxe JUnit 5

Fabrice AMBERT



Informatique

IUT Nord Franche-Comté

DÉPARTEMENT INFO | IUT BELFORT-MONTBÉLIARD
UNIVERSITÉ DE FRANCHE-COMTÉ

UNIVERSITÉ de
FRANCHE-COMTÉ

Prérequis et Utilité

- JUnit 5 nécessite Java 1.8 minimum
- JUnit 5 supporte les λ -expressions de Java contrairement à JUnit 4
- JUnit 5 peut supporter les tests écrits pour JUnit 4

Anatomie d'un test unitaire

```
@Test    Annotation désignant la méthode comme un test
public void testXXX() {    public n'est pas obligatoire – private est interdit
    //Define                Instructions de mise en contexte
    //When                  Instruction sous test
    //Then                  Observation et vérification de l'oracle
}
```

Vérification d'un oracle

Utiliser la classe **Assertions** de JUnit

`Assertions.assertEquals(expected, actual)`

vérifier que le résultat calculé, **actual**, est égal au résultat attendu, **expected**.

`Assertions.assertTrue(actual)`

vérifier que le résultat booléen calculé, **actual**, est égal à vrai.

`Assertions.assertFalse(actual)`

vérifier que le résultat booléen calculé, **actual**, est égal à faux.

Tester l'apparition d'une exception

Utiliser la classe **Assertions** de JUnit

```
@Test  
public void testXXX() {  
  
    Assertions.assertThrows(ExceptionAttendue.class, Executable);  
}
```

Quelques annotations

@Disabled

```
@Disabled ("motif de la désactivation")
```

```
@Test
```

```
public void testXXX() {
```

```
    // Define
```

```
    // When
```

```
    // Then
```

```
}
```

Lors de l'exécution de la suite de tests, le test est ignoré mais est mentionné dans le rapport d'exécution

Quelques annotations - 2

@BeforeEach @AfterEach

@BeforeEach

```
public void setUp() {
```

```
}
```

Cette méthode est exécuté avant chaque test de la suite

@AfterEach

```
public void tearDown() {
```

```
}
```

Celle ci après chaque test de la suite

Quelques annotations - 3

@BeforeAll

@AfterAll

@BeforeAll

```
public void setUpSuite() {
```

```
}
```

Cette méthode est exécuté une fois avant tous les test de la suite

@AfterAll

```
public void tearDownSuite() {
```

```
}
```

Celle ci une fois après tous les tests de la suite

Pour aller plus loin

Junit 5 met à disposition d'autres annotations permettant des gestions fines des tests (plusieurs formes de Disabled), de disposer de tests paramétrés (qui vont pouvoir s'utiliser sur plusieurs jeux de données) ...

<https://junit.org/junit5/docs/current/user-guide/>