

# JUnit 4

# Exemple très simple!

```
ExempleSimple.java
class ExempleSimple

1 public class ExempleSimple{
2     public static int calculer(int a, int b) {
3         int res = a + b;
4         if (a == 0){
5             res = b * 2;
6         }
7         if (b == 0) {
8             res = a * a;
9         }
10        return res;
11    }
```

```
ExempleSimpleTest.java
ExempleSimpleTest.java > No Selection

1 import junit.framework.*;
2
3 public class ExempleSimpleTest extends TestCase{
4
5     public void testCalculer() throws Exception {
6
7         assertEquals(2, MaClasse.calculer(1,1));
8
9     }
10 }
```

Java ExempleSimple.java

```
java -cp junit.jar;. junit.textui.TestRunner ExempleSimple
/java/testjunit>java -cp junit.jar;. junit.textui.TestRunner
ExempleSimpleTest
```

```
.
Time: 0,01
OK (1 test)
```

# Trouvez l'intrus

```
ExempleSimpleTest.java
1 import junit.framework.*;
2
3 public class ExempleSimpleTest extends TestCase{
4
5     public void testCalculer() throws Exception {
6
7         assertEquals(2, MaClasse.calculer(1,1));
8
9     }
10 }
```

(A)

```
ExempleSimpleTest.java
1 import junit.framework.*;
2
3 public class ExempleSimpleTest extends TestCase{
4
5     public void TestCalculer() throws Exception
6
7         assertEquals(2, MaClasse.calculer(1,1));
8
9     }
10 }
```

(B)



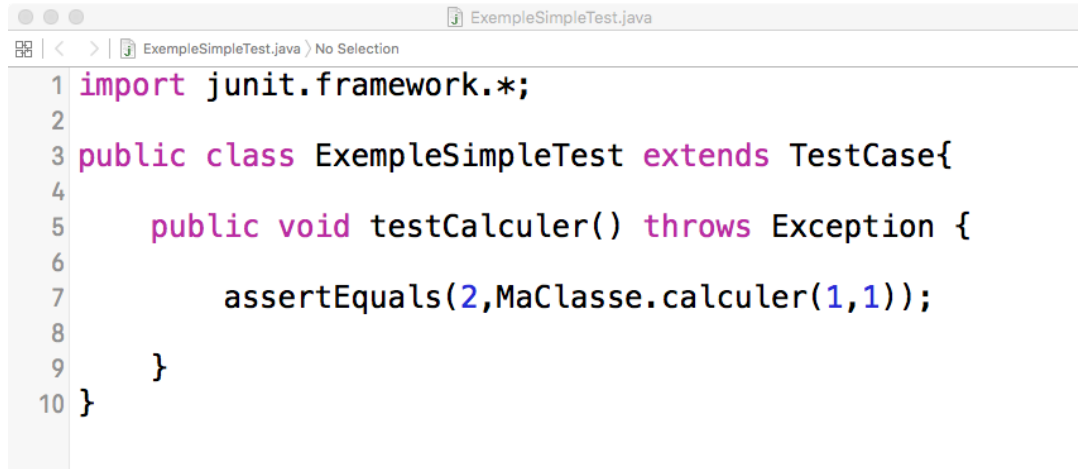
```
/java/testjunit>java -cp junit.jar;. junit.textui.TestRunner
ExempleSimpleTest
.F
Time: 0,01
There was 1 failure:
1) warning(junit.framework.TestSuite$1)
junit.framework.AssertionFailedError: No tests found in MaClasseTest
FAILURES!!!
Tests run: 1, Failures: 1, Errors: 0
```

# Classe de Test

Dans cette classe, il faut écrire une méthode dont le nom commence par "**test**" en minuscule suivi du nom du cas de test (généralement le nom de la méthode à tester).

Chacune de ces méthodes doit avoir les caractéristiques suivantes :

- elle doit être déclarée public
- elle ne doit renvoyer aucune valeur
- elle ne doit pas posséder de paramètres.



```
1 import junit.framework.*;
2
3 public class ExempleSimpleTest extends TestCase{
4
5     public void testCalculer() throws Exception {
6
7         assertEquals(2, MaClasse.calculer(1,1));
8
9     }
10 }
```

# L'écriture des cas de tests

Chaque méthode de test contient généralement des traitements en trois étapes :

- Instanciation des objets requis
- Invocation des traitements sur les objets
- Vérification des résultats des traitements

JUnit ne garantit pas l'ordre d'exécution des cas de tests  
puisque ceux-ci sont obtenus par introspection

# Les assertions

## `assertEquals()`

Vérifier l'égalité de deux valeurs de type primitif ou objet (en utilisant la méthode `equals()`).

Il existe de nombreuses surcharges de cette méthode :

- pour chaque type primitif,
- pour un objet de type `Object`
- pour un objet de type `String`

# Les assertions

**assertFalse()** : Vérifier que la valeur fournie en paramètre est fausse

**assertTrue()** : Vérifier que la valeur fournie en paramètre est vraie

**assertNull()** : Vérifier que l'objet fourni en paramètre soit null

**assertNotNull()** : Vérifier que l'objet fourni en paramètre ne soit pas null

# Les assertions

## assertSame() :

Vérifier que les deux objets fournis en paramètre font référence à la même entité

## Exemples identiques :

```
assertSame("Les deux objets sont identiques", obj1,  
obj2) ;
```

```
assertTrue("Les deux objets sont identiques ", obj1  
== obj2) ;
```

## assertNotSame() :

Vérifier que les deux objets fournis en paramètre ne font pas référence à la même entité