

# JUnit - TestCase

Br. Rafael Rodríguez Guzmán

```
public abstract class TestCase extends Assert implements  
Test
```

Un caso de pruebas define el “*fixture*” para correr múltiples pruebas. Para definir un caso de pruebas es necesario:

- 1) Implementar una subclase de TestCase.
- 2) Definir variables de instancia que almacenen el estado del *fixture*.
- 3) Inicializar el estado del *fixture* sobreescribiendo *tearDown*.
- 4) Limpiar después del test sobreescribiendo *tearDown*.

La implementación de TestCase es la forma en que se definen pruebas en JUnit 3, sin embargo, en JUnit 4 esta forma de indicar qué métodos serán métodos o casos de prueba se sustituyó por la anotación @Test

# Casos de prueba - estilo JUnit 3

```
import junit.framework.TestCase;

public class DummyTestA extends TestCase {

    public void testSum() {
        int a = 5;
        int b = 10;
        int result = a + b;
        assertEquals(15, result);
    }
}
```

La clase de prueba debe extender de TestCase.

El método debe ser nombrado con el prefijo “test” para que se reconozca como una prueba.

Todos los métodos con el prefijo “test” se ejecutarán en el TestCase.

# Casos de prueba - estilo JUnit 4

```
import org.junit.*;  
import static org.junit.Assert.*;
```

```
public class DummyTestB {
```

```
    @Test
```

```
    public void Sum() {
```

```
        int a = 5;
```

```
        int b = 10;
```

```
        int result = a + b;
```

```
        assertEquals(15, result);
```

```
    }
```

```
}
```

Las clases se declaran como clases normales.

A cada caso de prueba se asigna a un método con la anotación `@Test`.

Si queremos una clase que ejecute todos los casos de prueba (parecido a `TestCase`) podemos usar un `TestSuite`.

```
TestSuite suite = new TestSuite("All tests");
```

```
suite.addTestSuite(DummyTestA.class);  
suite.addTestSuite(DummyTestAbis.class);
```

# TestCase methods

- `int countTestCases()`: Cuenta el número de casos de pruebas.
- `TestResult createResult()`: Crea un objeto `TestResult` por defecto.
- `String getName()`: Obtiene el nombre del `TestCase`.
- `TestResult run()`: Método para correr una prueba, recolectando los resultados en un objeto `TestResult`.
- `void run(TestResult result)`: Corre el caso de prueba y recolecta los resultados en un objeto `TestResult`.
- `void setName(String name)`: Establece el nombre del caso de prueba.
- `void setUp()`: Establece el *fixture*. Por ejemplo, abre la conexión con la base de datos.
- `void tearDown()`: Termina el *fixture*. Por ejemplo, cierra la conexión con una base de datos.
- `String toString()`: Regresa un *String* con la representación del caso de prueba.



```
import junit.framework.TestCase;  
import org.junit.Before;  
import org.junit.Test;
```

```
public class TestClassJU3 extends TestCase {  
    protected double fValue1;  
    protected double fValue2;
```

```
    @Before
```

```
    public void setUp() {  
        System.out.println("setup");  
        fValue1 = 2.0;  
        fValue2 = 3.0;  
    }
```

@Test

```
public void testAdd() {
```

```
    //count the number of test cases
```

```
    System.out.println("No of Test Case = "+ this.countTestCases());
```

```
    //test getName
```

```
    String name= this.getName();
```

```
    System.out.println("Test Case Name = "+ name);
```

```
    //test setName
```

```
    this.setName("testNewAdd");
```

```
    String newName= this.getName();
```

```
    System.out.println("Updated Test Case Name = "+ newName);
```

```
    assertEquals(5.0, fValue1+fValue2);
```

```
}
```

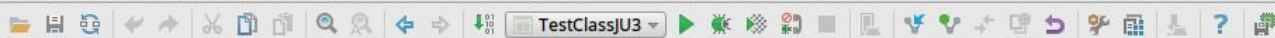
*//tearDown used to close the connection or clean up activities*

```
public void tearDown( ) {  
    System.out.println("clean up.");  
}  
}
```

Resultado en consola:

```
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=gasp  
setup  
No of Test Case = 1  
Test Case Name = testAdd  
Updated Test Case Name = testNewAdd  
clean up.
```

Process finished with exit code 0



JUnit5 src main java TestClassJU3

Project

JUnit5 ~/Documents/JUnit5

- .idea
- src
  - main
    - java
      - AppTest1
      - AppTest2
      - AssertionsDemo
      - AssumptionsDemo
      - DisplayNameDemo
      - FirstJUnit5Test.java
      - FirstJUnit5Tests
      - ParameterizedTestDemo
      - Person
      - TestClassJU3**
      - TestingAStackDemo
- resources

TestClassJU3.java

```
5 public class TestClassJU3 extends TestCase {
6     protected double fValue1;
7     protected double fValue2;
8
9     @Before
10    public void setUp() {
11        System.out.println("setup");
12        fValue1= 2.0;
13        fValue2= 3.0;
14    }
15
16    @Test
17    public void testAdd() {
18        //count the number of test cases
19        System.out.println("No of Test Case = "+ this.countTestCases());
20
21        //test getName
22        String name= this.getName();
23        System.out.println("Test Case Name = "+ name);
24
25        //test setName
```

Run TestClassJU3

0 of 1 test

TestClassJU3

- testAdd

```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=gasp
setup
No of Test Case = 1
Test Case Name = testAdd
Updated Test Case Name = testNewAdd
clean up.

Process finished with exit code 0
```

Run TODO Version Control Terminal

1 Event Log

Failed to start: 0 passed, 1 not started (a minute ago)

10:1 LF UTF-8 Git: master

# Referencias

Java JUnit tutorial - JUnit TestCase. Recuperado de:  
[http://www.java2s.com/Tutorials/Java/JUnit/0060\\_\\_JUnit\\_TestCase.htm](http://www.java2s.com/Tutorials/Java/JUnit/0060__JUnit_TestCase.htm)