

# ACTIVIDAD DE EJEMPLO

## Primera prueba con JUnit

### Contenido

Preparación de la clase a probar.....	2
Realizando las pruebas con JUnit.....	3
Ejecución de las pruebas.....	5

**JUnit** es un conjunto de clases que permite probar las clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado:

- Si la clase cumple la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba.
- En caso de que el valor esperado sea diferente al retornado por el método durante la ejecución, **JUnit** devolverá un fallo en el método correspondiente.

## Preparación de la clase a probar

A continuación vamos a ver un ejemplo de utilización de **JUnit**.

La clase sobre la que vamos a realizar las pruebas se llama **ArrayWrapper**, y consiste en una simple implementación de un wrapper sobre **int array[]**.

Contiene los siguientes métodos:

- **Constructor** al que se le pasa un array de enteros y lo guarda en el atributo privado.
- **size** devuelve el tamaño
- **setElementAt** agrega un nuevo elemento indicando su posición
- **getArray** devuelve el array

El código es el siguiente:

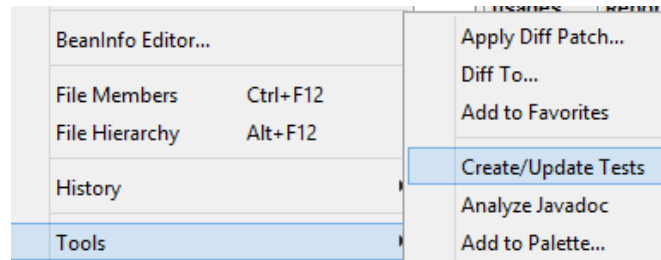
```
package tema8;

public class ArrayWrapper {
    int arr[]; //contenedor de elementos

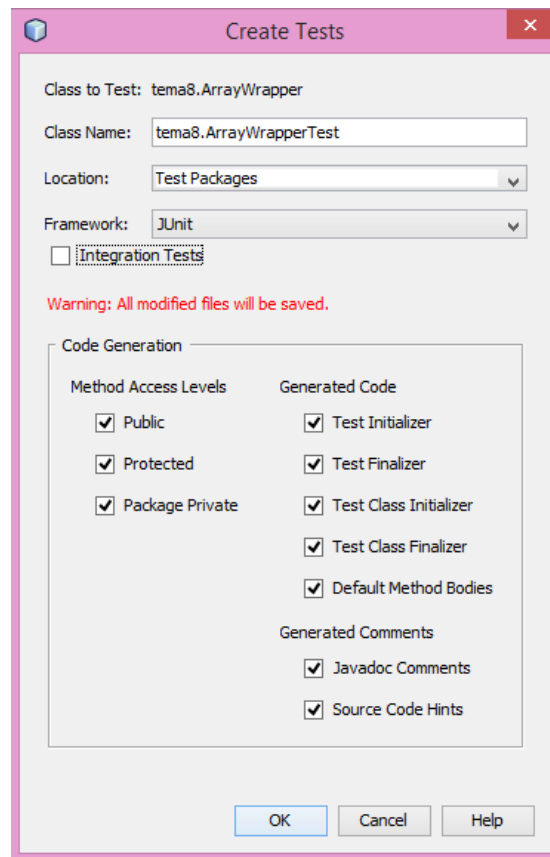
    //Constructor
    public ArrayWrapper(int par[]){
        this.arr = par;
    }
    //tamaño
    public int size(){
        return this.arr.length;
    }
    //devuelve cadena con contenido del array
    @Override
    public String toString(){
        StringBuilder str = new StringBuilder();
        int i = 0;
        str.append("[");
        for (i = 0; i < this.arr.length - 1; i++) {
            str.append(arr[i] + ",");
        }
        str.append(arr[i] + "]");
        return str.toString();
    }
    //agrega un elemento en la posición indicada
    public void setElementAt(int elemento, int indice){
        arr[indice] = elemento;
    }
    //obtiene el array
    public int[] getArray(){
        return this.arr;
    }
}
```

## Realizando las pruebas con JUnit

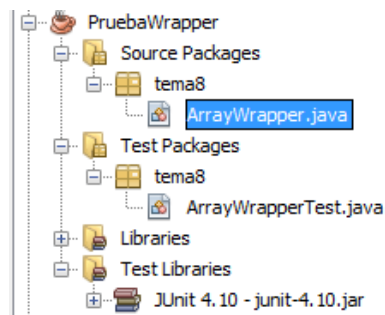
JUnit ya viene incorporado en Netbeans. Con el botón derecho sobre la clase a probar, seleccionaremos **"Tools> Create Tests"**:



En la pantalla que se abre, configuraremos el nombre, localización y tipo de pruebas a realizar:



En este momento, nuestro proyecto debería tener los siguientes componentes:



Vemos que nos ha añadido automáticamente las librerías correspondientes (en este caso **junit 4.10**).

Si echamos un vistazo a la nueva clase **ArrayWrapperTest.java**, vemos que nos ha creado los siguientes métodos:

- **Métodos setUp y tearDown**, usados para inicializar recursos, variables, etc. para que sea igual para todos los métodos.

En nuestro caso inicializaremos el array con el contenido 1, 2, 3, 4, 5.

- o El método **setUp** que se ejecuta al comienzo de cada uno de los métodos test.
- o El método **tearDown** se ejecuta al final de cada test.

```
public class ArrayWrapperTest {
    private int arr[];
    private ArrayWrapper instance;
    private int TAMANO = 5;

    @Before
    public void setUp() {
        arr = new int[TAMANO];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = i;
        }
        instance = new ArrayWrapper(arr);
    }
    @After
    public void tearDown() {
        instance = null;
    }
}
```

- Un método para probar cada uno de los métodos que queramos probar de nuestro **ArrayWrapper**. Será necesario borrar las dos últimas líneas de cada método y codificar a mano la funcionalidad que queramos probar.

Los casos de prueba se consideran exitosos o fallidos dependiendo de una sentencia que se incluye en los casos de prueba y que es de tipo assert. Los asertos que se pueden usar los siguientes (el parámetro **message** es opcional en todos los casos):

- o assertTrue(String **message**, boolean **test**)
- o assertFalse(String **message**, boolean **test**)
- o assertEquals(String **message**, **expected**, **actual**)
- o assertSame(String **message**, Object **expected**, Object **actual**)
- o assertNotSame(String **message**, Object **expected**, Object **actual**)
- o assertNull(String **message**, Object **object**)
- o assertNotNull(String **message**, Object **object**)
- o fail(String **message**)

Por ejemplo, para probar el método **testSize**, usaremos **assertEquals** para comprobar que el tamaño que me devuelva el método size es el mismo que habíamos fijado en el **setUp**:

```
/**
 * Test of size method, of class ArrayWrapper.
 */
@Test
public void testSize() {
    System.out.println("size");
    int expResult = TAMANO;
    int result = instance.size();
    assertEquals(expResult, result);
}
```

De igual forma, para probar el método **toString**, usamos el método **assertEquals** para comprobar que tras instanciar el array, y llamar al método **toString**, el resultado es el esperado([0,1,2,3,4]). Codificaremos lo siguiente:

```
/**
 * Test of toString method, of class ArrayWrapper.
 */
@Test
public void testToString() {
    System.out.println("toString");
    String expectedResult = "[0,1,2,3,4]";
    String result = instance.toString();
    assertEquals(expectedResult, result);
}
```

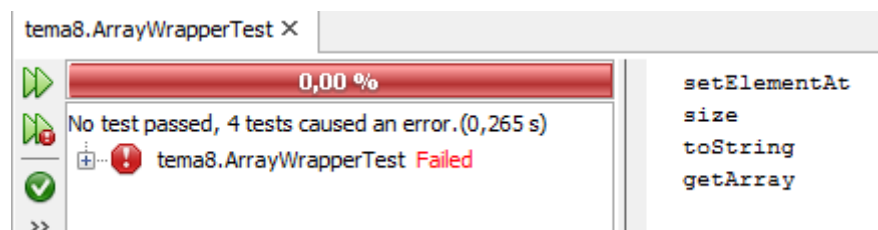
Por último, probaremos el método **setElementAt**, agregando el elemento 99 en la posición 2. Aprovechamos para probar el método **getArray**, y comprobamos que el elemento que ocupa la posición 2 efectivamente es el 99.

```
/**
 * Test of setElementAt method, of class ArrayWrapper.
 */
@Test
public void testSetElementAt() {
    System.out.println("setElementAt");
    int elemento = 99;
    int indice = 2;
    instance.setElementAt(elemento, indice);
    int ret[] = instance.getArray();
    assertEquals(elemento, ret[2]);
}
```

## Ejecución de las pruebas

Para ejecutar las pruebas seleccionaremos la opción **TestFile** que nos ofrece el menú contextual de la clase de pruebas. En la parte inferior (Output) se nos ofrecerá el resultado de la ejecución.

Si tiene fallos:



Si ha sido exitosa:

