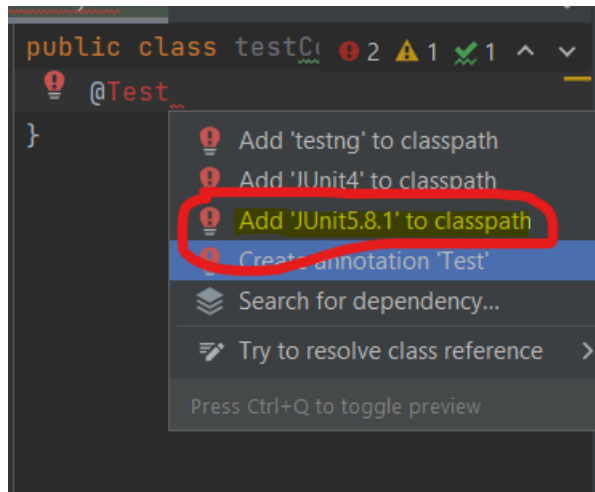


1. Creamos nuestro proyecto con el nombre “TDD Practica Coche”. Utilizamos el entorno de IntelliJ, con el SDK 19 y el JUNIT5.
2. Dentro del proyecto creamos un nuevo directorio “Tests” y la marcamos como raíz. Para esto hacemos lo siguiente:
 - Click derecho en la carpeta “Tests” → *mark directory as* → *test sources root*
3. En ella, creamos una *java class* llamada “TestCoche” y lo indicamos con *@Test*. Si nos aparece como que no reconoce la librería de *Junit5/test*, la añadimos nosotros presionando *Alt+Enter* sobre el *@test*

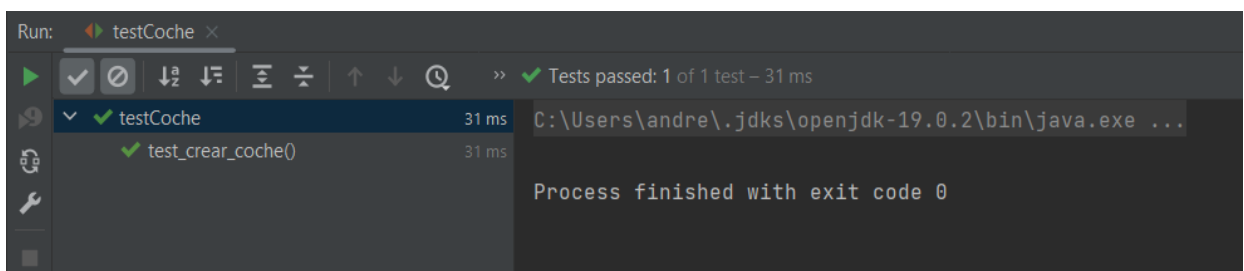


Después de esto, creamos nuestro primer test llamado “Test_crear_coche” y el objeto nuevo de la clase “Coche”.

```
Coche nuevo = new Coche(){};
```

Como hemos creado primero el test y hemos referenciado la clase Coche sin estar creada, el propio *IntelliJ* nos da la opción de crear dicha clase con *ALT + Enter* y la creamos en la carpeta *Src*.

4. Ahora solo falta probar el test. Ejecutamos y comprobamos.

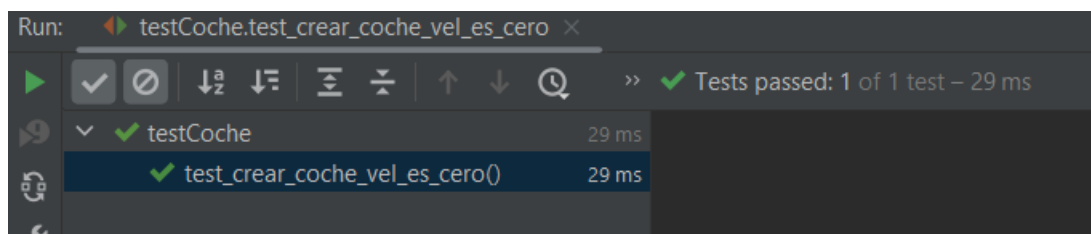


Como podemos observar, el test de la prueba de crear un nuevo objeto, se ha realizado correctamente.

5. Vamos a hacer un test un poco más elaborado. Vamos a comprobar que al crear un coche, su velocidad sea 0.
6. Creamos el método y para comprobar que esté correcto, el dato esperado y el que le damos, tiene que ser el mismo. Para ello, usaremos la declaración *assertEquals*, que compara los dos datos y devuelve *true* o *false* si es igual o no.

```
@Test
public void test_crear_coche_vel_es_cero(){
    Coche nuevoCoche = new Coche();
    Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
}
```

Ejecutamos el test y comprobamos.



Podemos ver que este test también está correctamente.

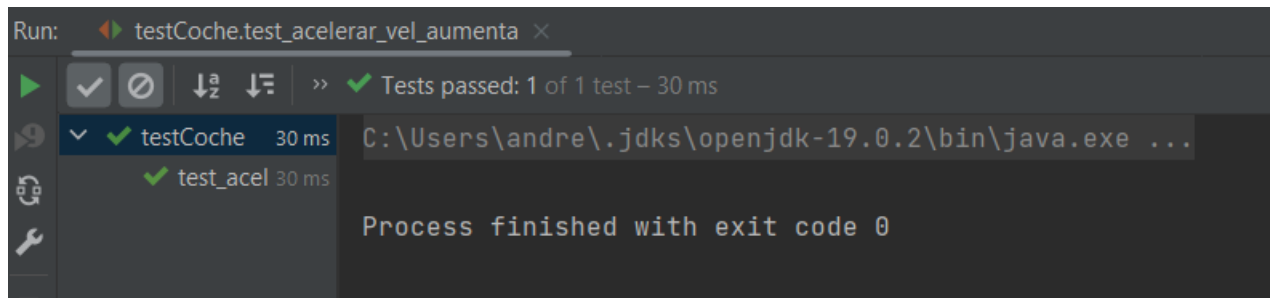
7. Ahora, implementaremos el método *acelerar* para variar la velocidad del coche.

```
@Test
public void test_acelerar_vel_aumenta(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar(30);
    Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
}
```

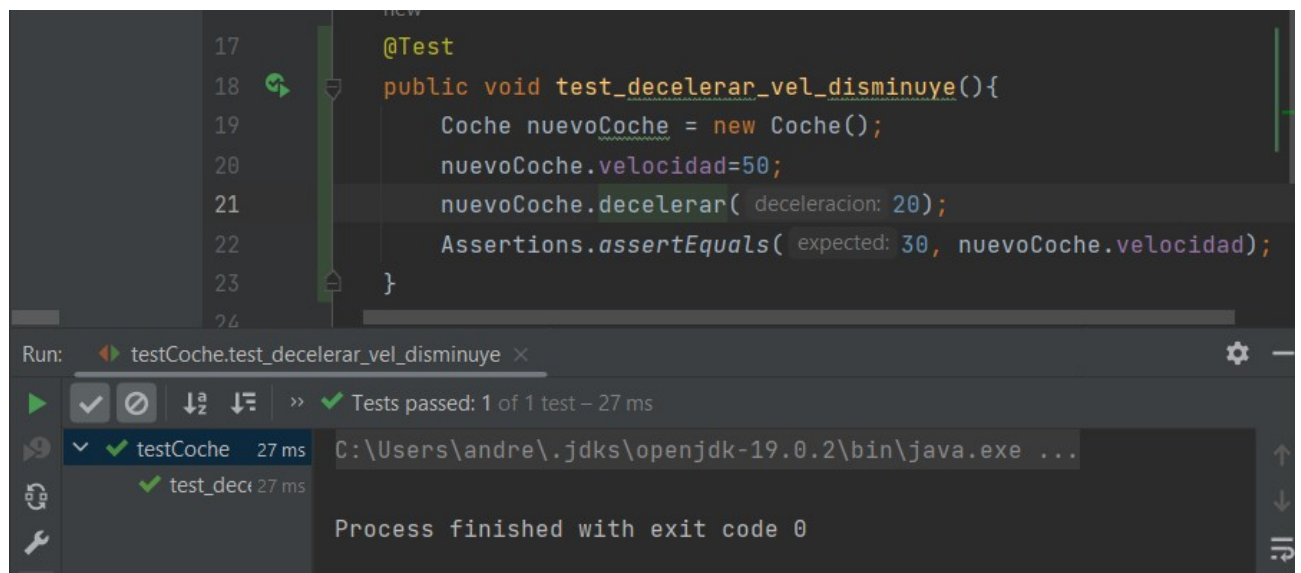
Como *acelerar* no existe todavía, hacemos lo mismo de antes: nos situamos en él y presionamos ALT + Enter y creamos el método *acelerar* y le decimos que la *velocidad* aumente en *aceleración*.

```
1 usage new *
public void acelerar(int aceleracion) {
    velocidad+=aceleracion;
}
```

Ejecutamos y vemos que hemos pasado el test correctamente.



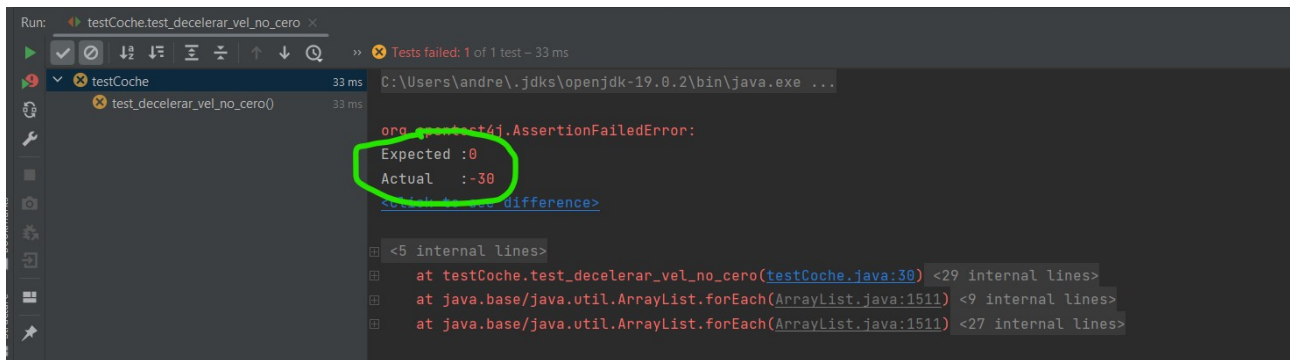
- Ahora, siguiendo los mismos pasos, vamos a crear el método *decelerar*. Como al crear *Coche*, la velocidad inicial es 0, no puede decelerar, por lo que le tenemos que pasar la velocidad que queramos. Por ejemplo, 50 y que el coche decelere 20. Entonces, el parámetro de resultado esperado que le tenemos que pasar a *Assertions.assertEquals* es de 30.



- Por último, vamos a crear un test que si el coche decelera, la velocidad no puede ser menor a cero.



En este ejemplo, llevamos una velocidad de 50 y una deceleración de 80, por lo que el coche debería de frenar. Pero cuando comprobamos el test nos da error.



Como podemos ver, el error que nos da es que el resultado esperado es de 0 pero el resultado real es de -30 y nuestro test lo hemos hecho con la idea de que no puede ser 0, por lo que hay que arreglarlo. Para ello, nos vamos a nuestra clase *Coche* y modificamos el método *decelerar* con una condición que si la velocidad es menor de 0 la velocidad será 0.

```
2 usages new *
public void decelerar(int deceleracion) {
    velocidad -= deceleracion;
    if (velocidad < 0) {
        velocidad = 0;
    }
}
```

Comprobamos el test y vemos que ahora sí funciona.

