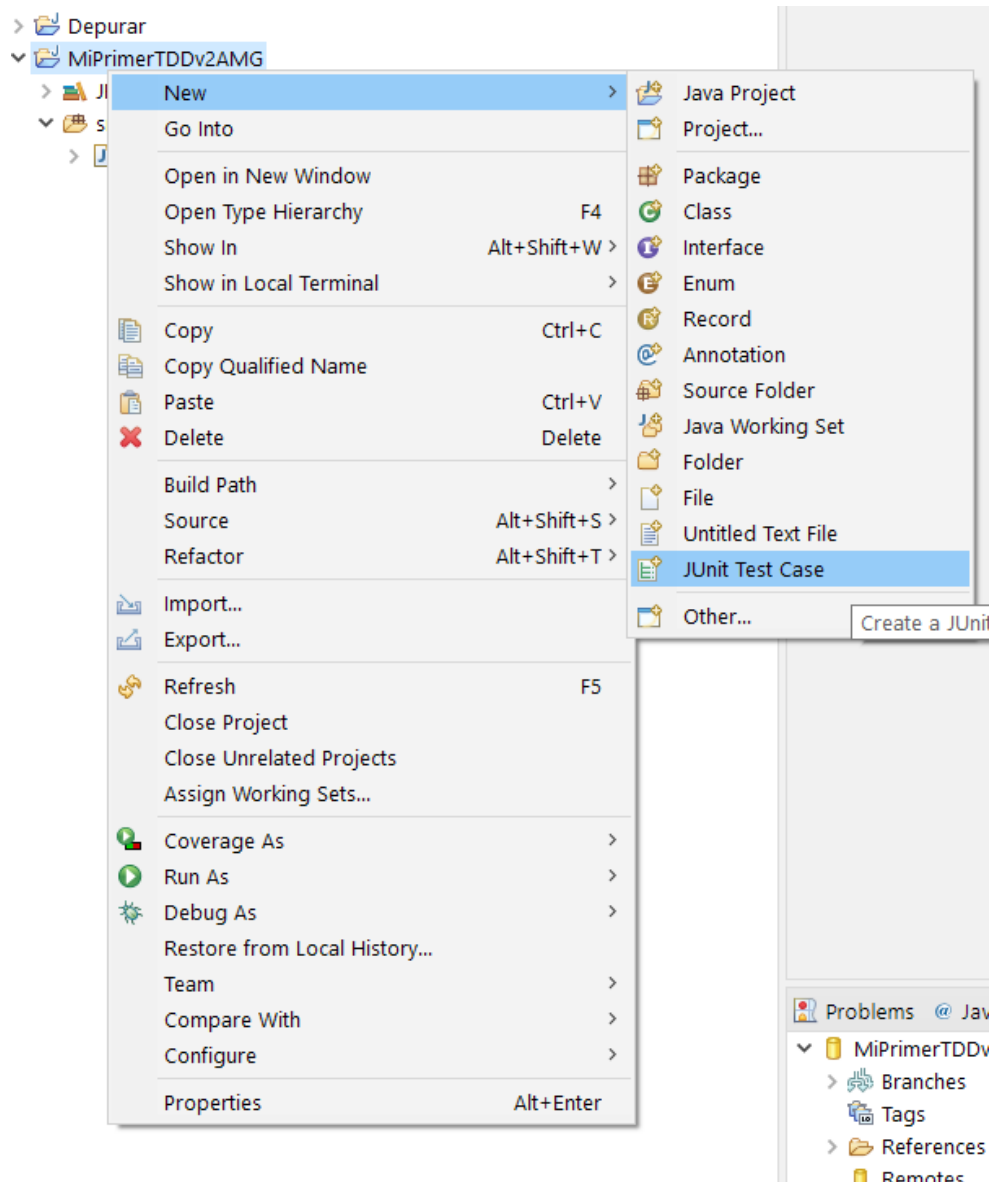
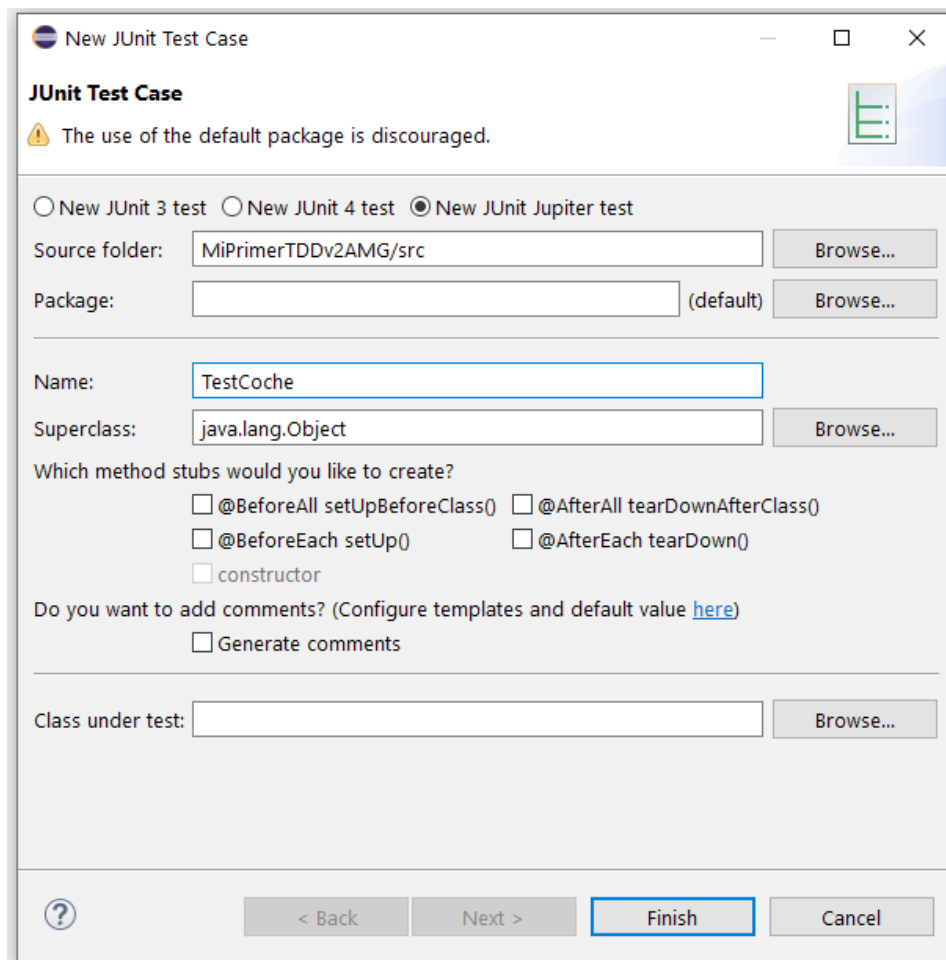


MI PRIMER TDDv2 – ÁNGEL MACIÁ GARCÍA

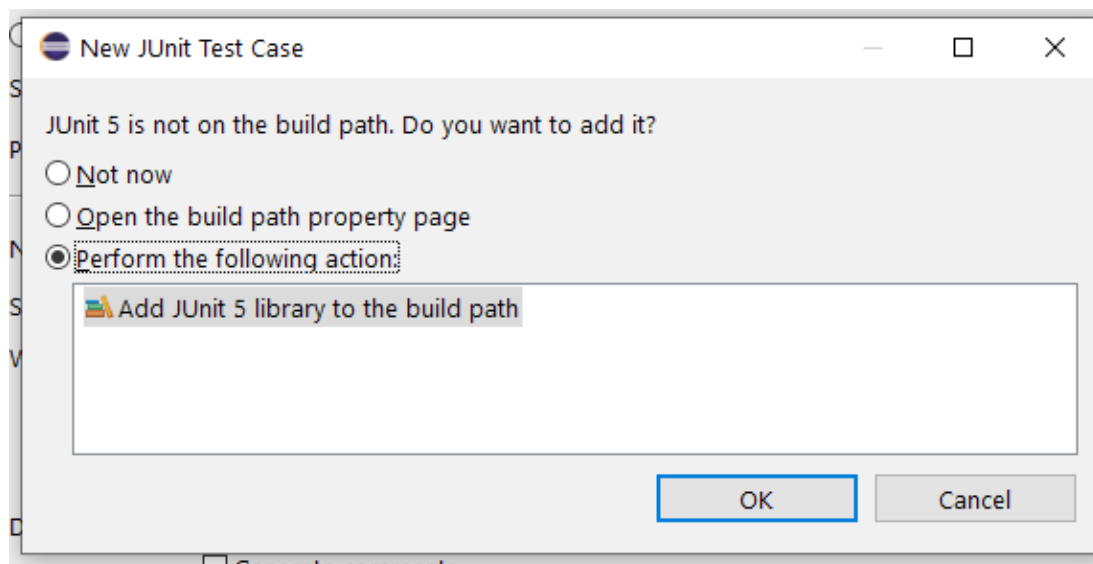
Paso 1: Para empezar, creamos un nuevo proyecto de Java en Eclipse. Lo siguiente es crear una clase para los test que vayamos a realizar, debemos pulsar clic derecho en el nombre del proyecto e ir a “New > JUnit Test Case”.



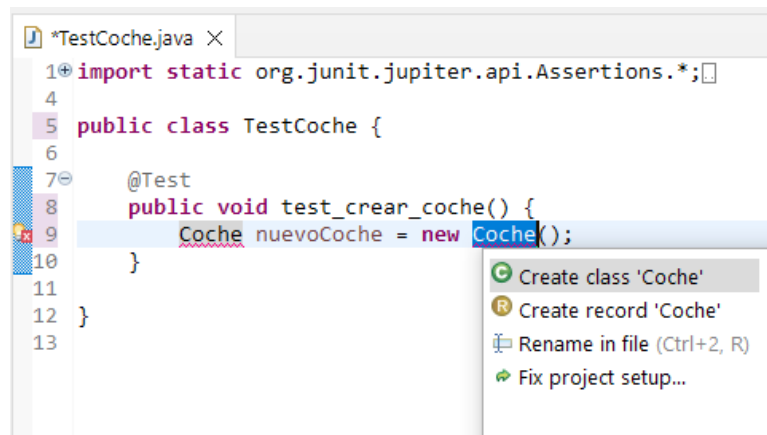
Se abre una ventana para la creación de la clase Junit, debemos seleccionar "New JUnit Jupiter test", de nombre de la clase ponemos "TestCoche" y las demás configuraciones las dejamos por defecto. Al acabar pulsamos "Finish".



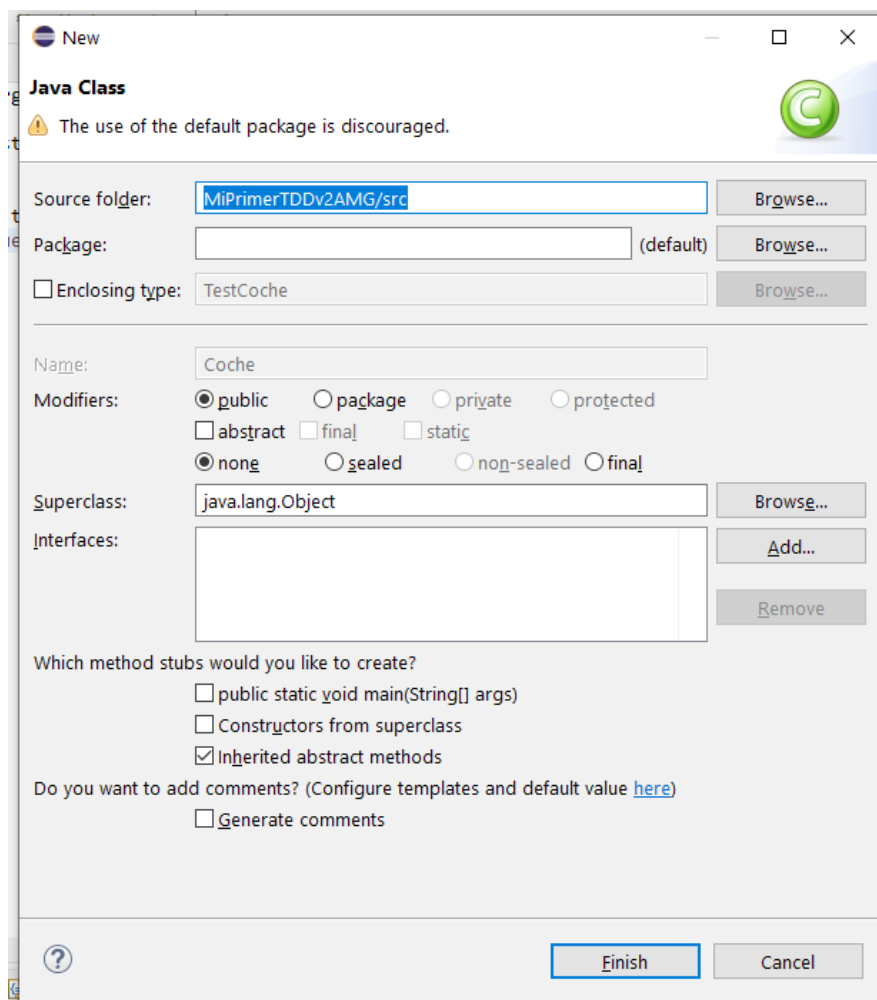
Se abre otra ventana preguntando si queremos añadir Junit 5, seleccionamos la última opción y pulsamos "OK".



Paso 2: Hemos creado la clase para Test, creamos el primer test llamado "test_crear_coche" y dentro creamos un objeto "nuevoCoche" de la clase Coche. Aparece un error, puesto que la clase Coche no existe, así que pulsamos en la bombilla de la izquierda de la línea y seleccionamos "Create class Coche".



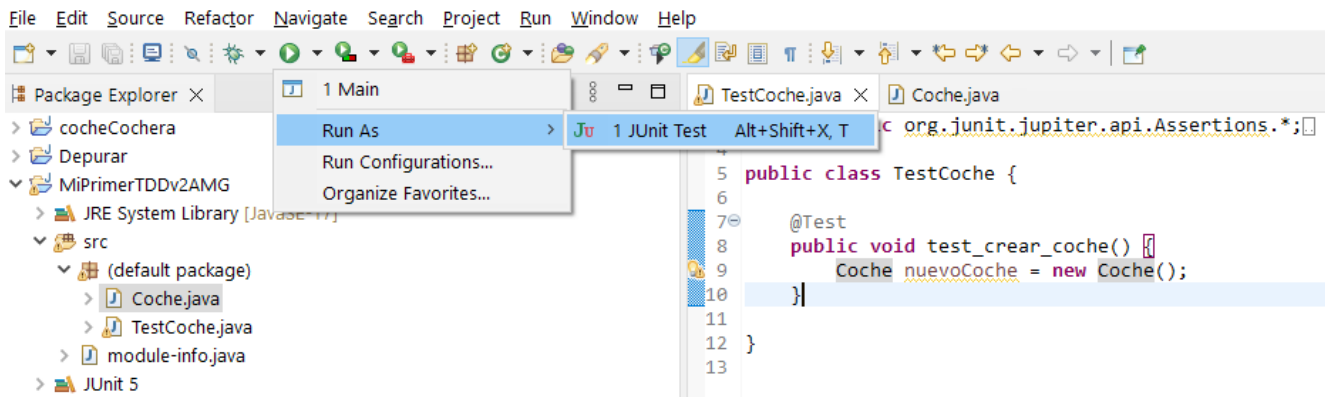
Se abre una ventana sobre la creación de la nueva clase, no cambiamos nada y pulsamos "Finish".



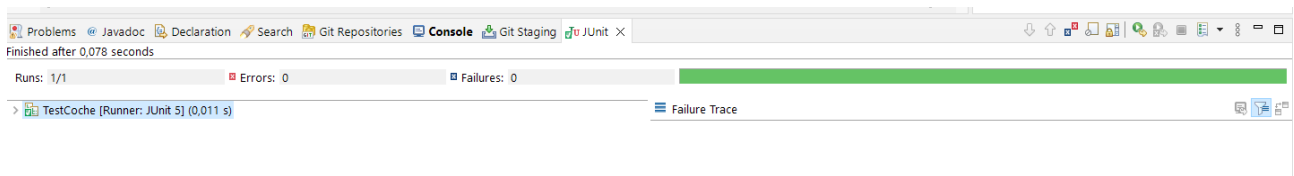
Comprobamos que la clase se ha creado correctamente.

```
*TestCoche.java  Coche.java X
1
2 public class Coche {
3
4 }
5
```

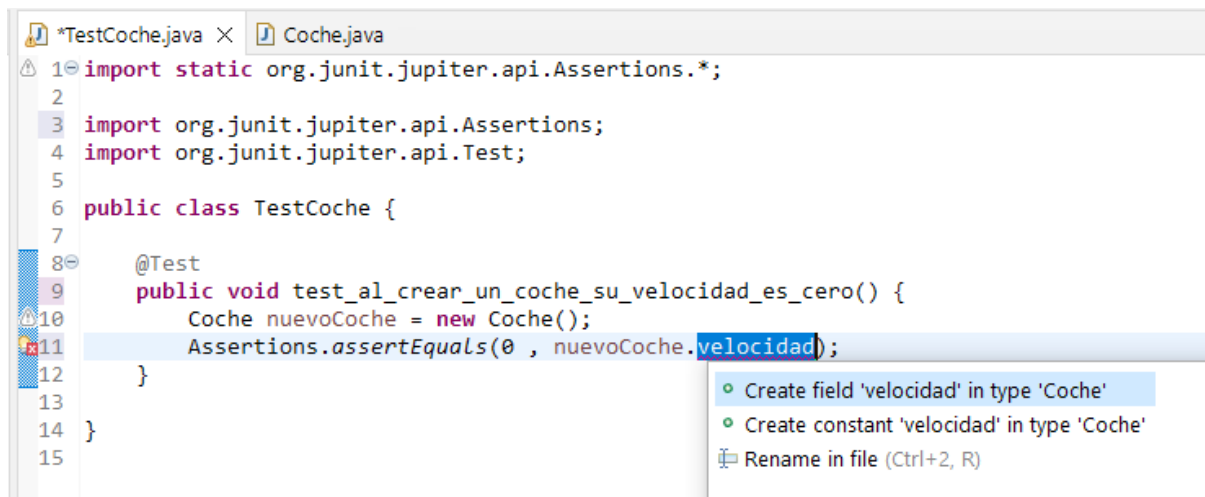
Vamos a probar ahora si el primer test funciona, para ello debemos ejecutar el código pulsando el botón de "Play" y seleccionamos "Run as > Junit Test".



El test ha funcionado correctamente.



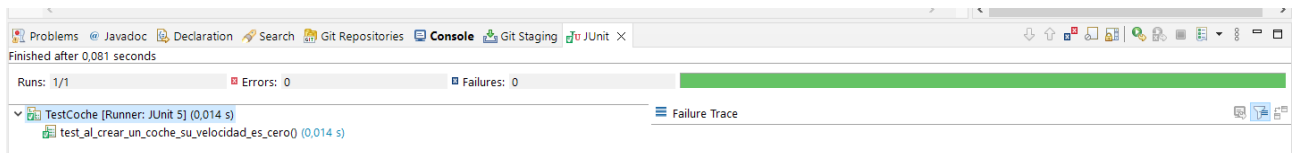
Paso 3: Modificamos el test que teníamos cambiando el nombre a "test_al_crear_un_coche_su_velocidad_es_cero", dentro del test además añadimos un assertEquals en el que comprobamos que el valor de velocidad de "nuevoCoche" sea 0. Como velocidad no existe y debemos crearlo, pulsamos en la bombilla de la izquierda y seleccionamos "Create field velocidad in type Coche".



Volvemos a la clase Coche para ver si ha creado el atributo velocidad. Vemos que lo ha creado de tipo "Integer", lo cambiamos a "int".

```
*TestCoche.java  *Coche.java X
1
2 public class Coche {
3
4     public int velocidad;
5
6 }
7
```

Ejecutamos de nuevo el Test y vemos que no hay ningún problema.



Paso 4: Creamos otro Test, cuyo nombre será "test_al_acelerar_un_coche_su_velocidad_aumenta". Para empezar, este test contendrá lo mismo que el anterior, pero le añadiremos que "nuevoCoche" llama al método acelerar con un parámetro de 30. En la aserción pondremos que el valor esperado es 30. Al hacer todo esto, en acelerar hay un error puesto que no existe, hacemos clic en la bombilla y lo creamos.

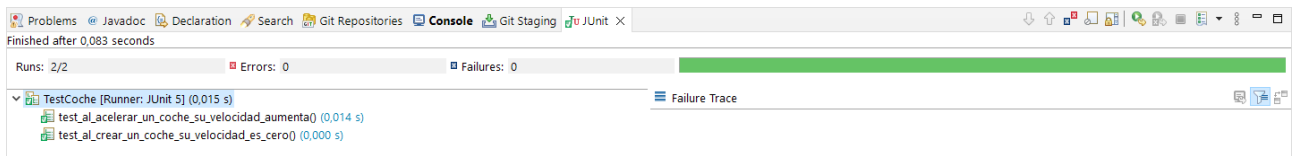
```
12     }
13     @Test
14     public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
15         Coche nuevoCoche = new Coche();
16         nuevoCoche.acelerar(30);
17         Assertions.assertEquals(30, nuevoCoche.velocidad);
18     }
19 }
20 }
21
```

Create method 'acelerar(int)' in type 'Coche'
Add cast to 'nuevoCoche'
Rename in file (Ctrl+2, R)

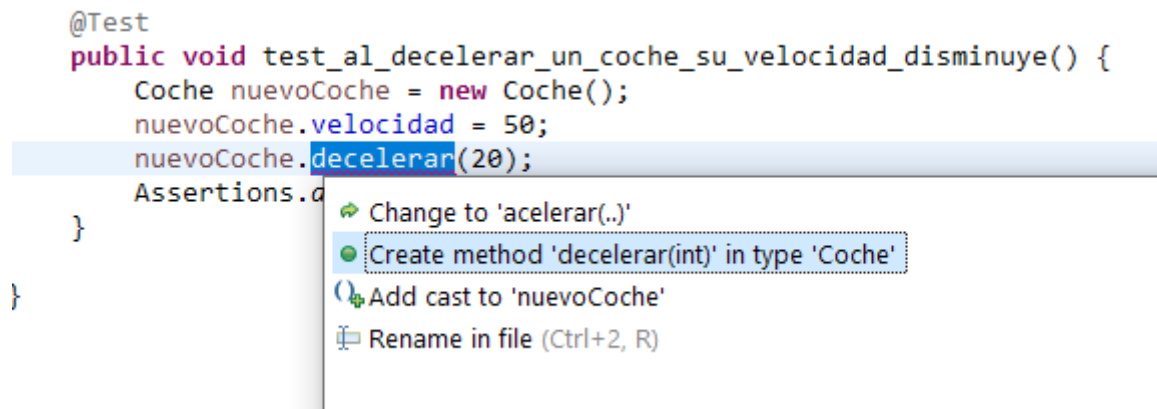
Vamos a la clase Coche y podemos ver que acelerar se ha creado, cambiamos el parámetro "int i" por "int aceleracion". Este método suma la aceleración pasada por parámetro a la velocidad del objeto coche.

```
*TestCoche.java  *Coche.java X
1
2 public class Coche {
3
4     public int velocidad;
5
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9
10 }
11
```

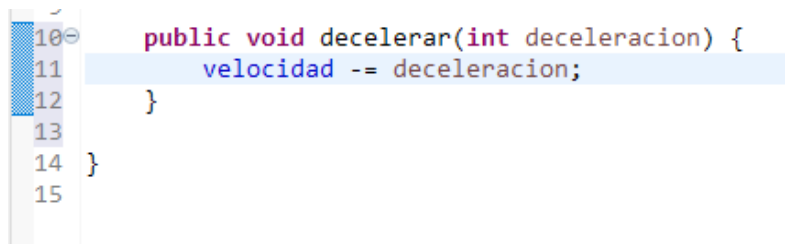
Ejecutamos los Test y comprobamos que ambos funcionan.



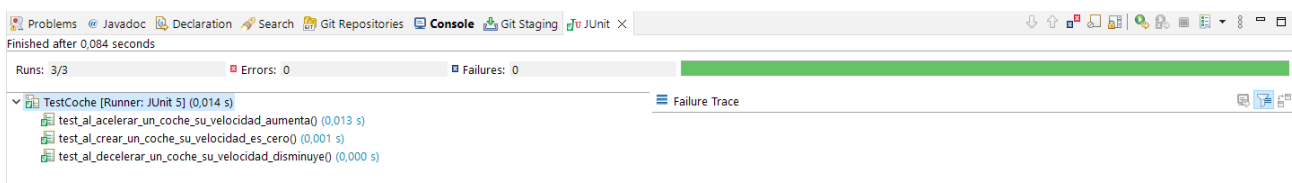
Paso 5: Creamos otro test muy similar al anterior, sólo cambia que el valor de velocidad que tendrá "nuevoCoche" será 50. Además "nuevoCoche" llama a un nuevo método "decelerar", en el cual por parámetro se le pasa un 20. Igual que las veces anteriores, debemos crear el nuevo método "decelerar" pulsando en la bombilla y seleccionando la opción "Create method decelerar in type Coche".



Como anteriormente, comprobamos en "Coche" que el método se ha creado. También cambiamos el parámetro "int i" por "int deceleracion". El método le restará el valor de deceleración al valor de la velocidad.



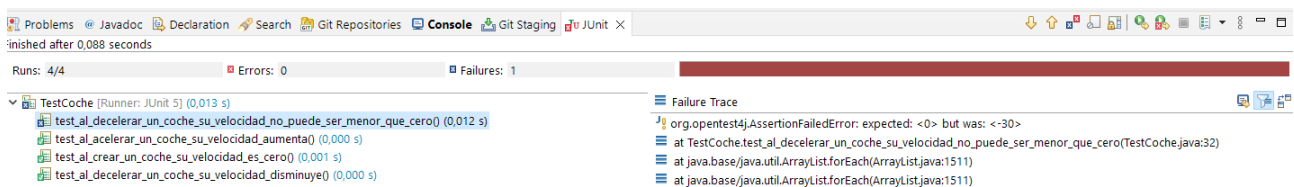
De nuevo ejecutamos los Test y comprobamos que todos funcionan correctamente.



Paso 6: Creamos otro Test copiando lo del anterior. Cambiamos el valor que se pasa por parámetro en "decelerar" a 80, y también cambiamos en "assertEquals" que el valor esperado de velocidad del "nuevoCoche" sea 0.

```
@Test
public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar(80);
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
```

Si ejecutamos ahora los Test, podemos ver que hay un fallo, y es debido a que en este último Test creado, el valor de velocidad es -30, y el valor esperado era 0.



Para solucionar este problema lo que haremos será ir al método "decelerar" y añadimos una condición "if". Este "if" que añadimos comparará si velocidad es menor que 0, y si se da el caso, entonces el valor de velocidad se igualará a 0.

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

Por último, ejecutamos de nuevo todos los Test para ver si queda algún fallo que resolver. Vemos que no hay ningún problema.

