HITO 5.2 JUNIT 5

1. Dentro de la carpeta test – java crear un nuevo paquete: com.pruebas

2. Crear un conjunto de Test (clase Java).

3. Crear un método de Test para probar el método count() de la clase SmartPhoneServiceImpl mediante aserciones (assert), en concreto:

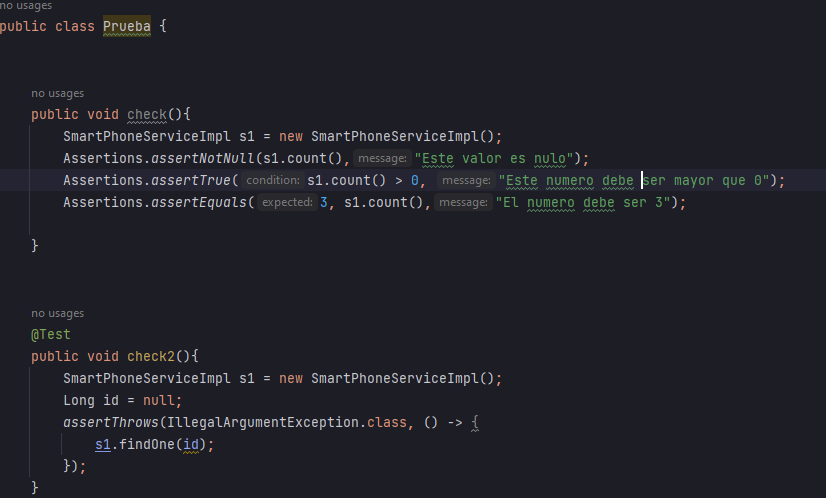
3.1 Que el método count() no devuelve NULL.

3.2 Que el valor que devuelve el método count() es mayor que 0.

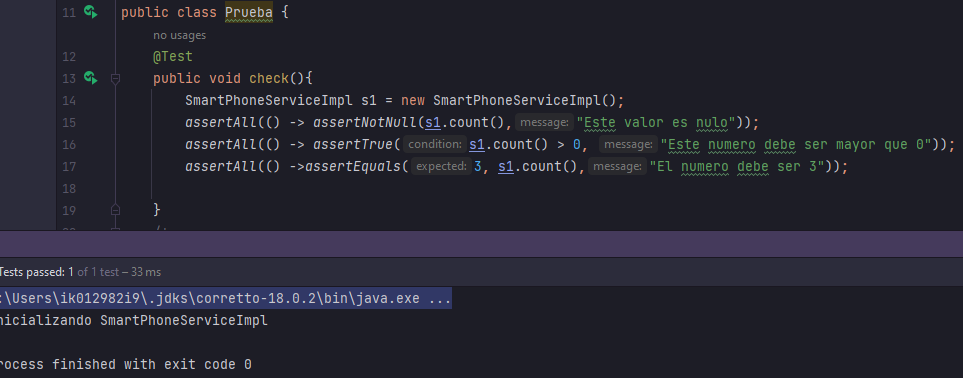
3.3 Que el valor que devuelve el método count() es justo 3.

5. Crear otro método de Test para probar el método findOne(Long id), en concreto:

5.1 Que el método al recibir un ID nulo, lanza una excepción de tipo IllegalArgumentException.

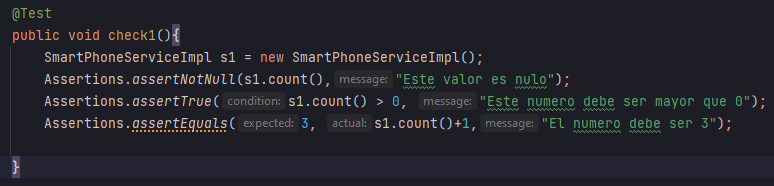


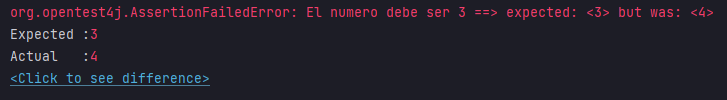
4. Ejecutar estos tests y crear un caso el que alguno de los tests falle.



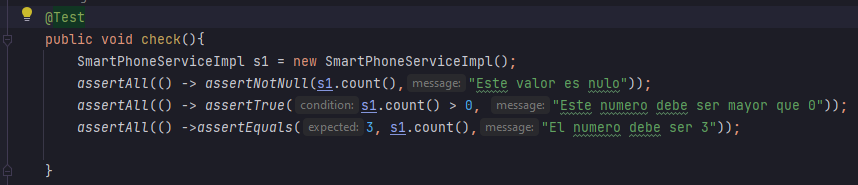
Ejemplo para que falle:

Le sumamos 1 al tercer Assertions





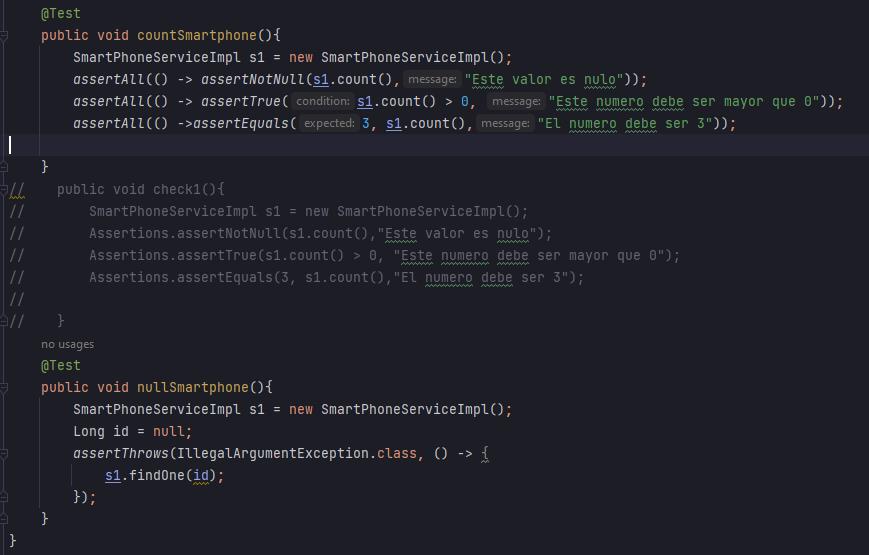
6. Juntar las aseciones 3.1, 3.2 eta 3.3 en una sola (mediante assertAll) y probar cada una ellas se ejecuta de manera independiente.



7. Explicar para que se utilizan las funciones lambdas

En Java, las funciones lambda se utilizan para implementar interfaces funcionales de una manera más concisa y legible. Una interfaz funcional es una interfaz que contiene exactamente un método abstracto, lo que significa que se puede tratar como una función en sí misma.

8. Asignar un nombre a cada método de Test y al conjunto de Tests (único)



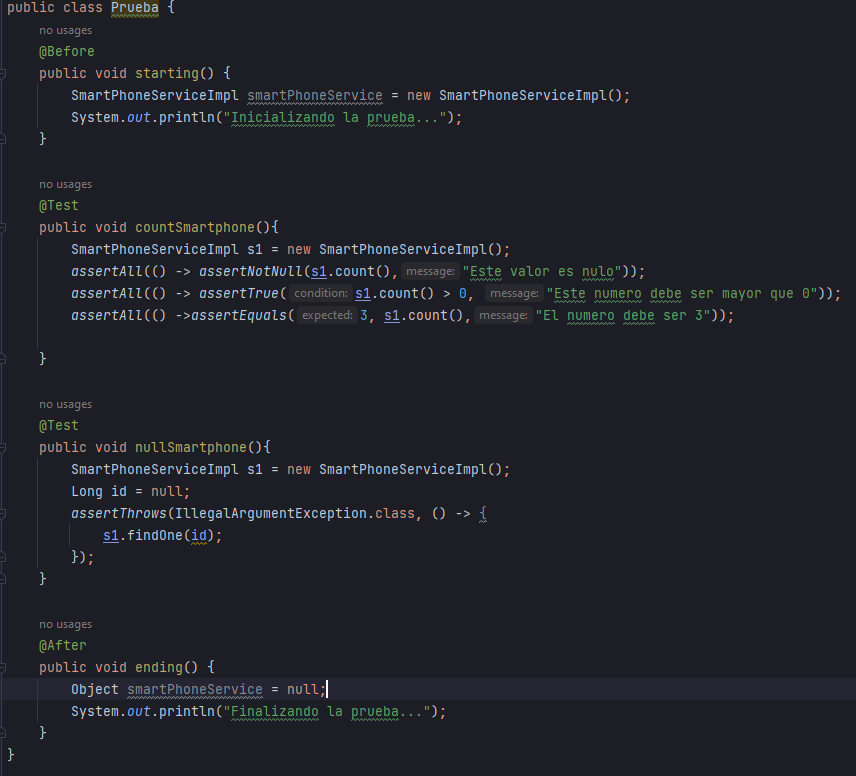
9. Investiga las opciones @BeforeEach y @AfterEach en una clase Test. ¿Para qué sirven? ¿Cuál es su uso? Demuestra que has probado ambas opciones (capturas de pantalla).

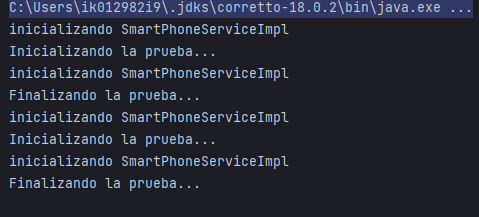
**@BeforeEach** y **@AfterEach** en JUnit son anotaciones de ciclo de vida que se utilizan en las pruebas unitarias para ejecutar ciertos métodos antes y después de cada prueba individual.

**@BeforeEach** indica que un método debe ejecutarse antes de cada prueba en la clase de prueba.

**@AfterEach** indica que un método debe ejecutarse después de cada prueba en la clase de prueba.

Ejemplo usando @BeforeEach & @AfterEach:





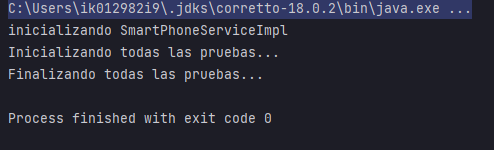
10. Investiga las opciones @BeforeAll y @AfterAll en una clase Test. ¿Para qué sirven? ¿Cuál es su uso? Demuestra que has probado ambas opciones (capturas de pantalla).

**@BeforeAll** y **@AfterAll** son anotaciones utilizadas en JUnit 5 para especificar métodos que deben ser ejecutados antes o después de todas las pruebas de la clase de prueba.

**@BeforeAll** se utiliza para marcar un método que se ejecutará solo una vez antes de que se ejecuten todas las pruebas en una clase de prueba

**@AfterAll** se utiliza para marcar un método que se ejecutará solo una vez después de que se hayan ejecutado todas las pruebas de la clase de prueba





11. Investiga la opción @TestMethodOrder. ¿Para qué sirve? ¿Cuál es su uso? Demuestra que lo has probado (captura de pantalla).

**@TestMethodOrder** se utiliza para definir el orden en el que se ejecutan los métodos de prueba en una clase de prueba. Sirve para asegurar que los métodos de prueba se ejecuten en un orden específico, lo cual es útil en casos donde un método de prueba depende del resultado de otro.

