**Paso 1.- Crear un proyecto con la clase que queremos probar.**

Abrimos Eclipse, creamos un proyecto JUnitLab y a continuación la clase Java que queremos probar.

Ejemplo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteUna instancia de esta clase representa una suscripción a una publicación. La variable precio contiene el precio de la suscripción. El precio recoge el total en euros y céntimos. El periodo de suscripción se representa en meses y se recoge en la variable periodo.

**Paso 2.- Crear una clase de prueba.**

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamentePara crear una clase de prueba, nos posicionamos sobre la clase que queremos probar, pulsamos el botón derecho y seleccionamos la opción “New”>>”JUnit Test Case” .

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEn el cuadro que aparece pulsamos el botón Next, y en la última pantalla aparecerán los métodos de la clase sobre la que queremos crear un caso de prueba, seleccionamos el método a probar y pulsamos finish.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteLa herramienta crea una clase de prueba, donde incluye las librerías de Junit necesarias para su ejecución, así como un método de prueba vacío donde deberemos escribir nuestro caso de prueba.

**Paso 3.- Crear los casos de prueba.**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteComo ejemplo, podemos rellenar la clase anterios con este código:

Todo caso de prueba se compone básicamente de 3 partes:

1. En una variable se indica cuál es el valor esperado por el método que queremos comprobar.
2. Se ejecuta el método que queremos probar con los datos de entrada adecuados y se guarda el resultado en otra variable.
3. Se comprueba la relación entre el valor esperado y el resultado del método, a través de una aserción. En este caso la aserción assertEquals indica que el caso de prueba es correcto sólo si el valor de las variables esperado y resultado son iguales.

**Paso 4.- Ejecutar paso de prueba.**

Se ejecutan las clases de prueba, las cuales son una serie de tests en cada clase del código.

Si hay fallas, estas salen a flote, indican dónde y por qué fallaron.

Los errores que las clases de prueba no se limitan a un solo tipo, Pueden ser como tal un error en el programa, un error de definición, o un error de lógica.

Las clases de pruebas de clases permiten modificar el código para en un siguiente test, confirmar si ese era el origen del problema o no.

Por ejemplo:

Es decir, que esperaba 1.0 y ha recibido como respuesta 100.0. En este punto podemos observar que el valor esperado no estaba correctamente definido, y que en vez de un error en el programa, es un error de la definición del caso de prueba.

**Paso 5.- Medición de la cobertura del código.**

La efectividad de los tests se mide en su cobertura.

La cobertura se refiere a cuántas líneas leen los test respecto al total. Si el test pasa por el 100% no garantiza que no hayan errores, pero sigue siendo un testeo riguroso.

El software que ayuda a este tipo de problemas puede medir la cobertura de estos tests. Esta medición de cobertura ocurre cuando se hacen las pruebas de caja blanca (que usa el código fuente). Este tipo de software es muy útil, ya que permite correr las clases de prueba varias veces, y poco a poco se cubre más código el cual se considerará dentro de los próximos testeos. La idea es llegar al 100% de cobertura.

Bibliografía:

OpenCourseWare. (2018, octubre). Pruebas en eclipse con JUnit y EclEmma. https://ocw.ehu.eus/. Recuperado 12 de octubre de 2022, de <https://ocw.ehu.eus/pluginfile.php/51924/mod_resource/content/3/lab1-JUnit.pdf>