

Frameworks para Pruebas de Software

Semana 14

Agenda

- ¿Qué es JUnit?
- Beneficios de JUnit 5
- Configuración de un proyecto en Netbeans
- Escribiendo pruebas
- Anotaciones
- Tipos de Asserts en JUnit 5
- Ejecución de pruebas

¿Qué es JUnit?

- JUnit es un marco de prueba
- Java
- Permite escribir y ejecutar pruebas de manera eficiente.
- JUnit 4 y JUnit 5



Beneficios de JUnit 5

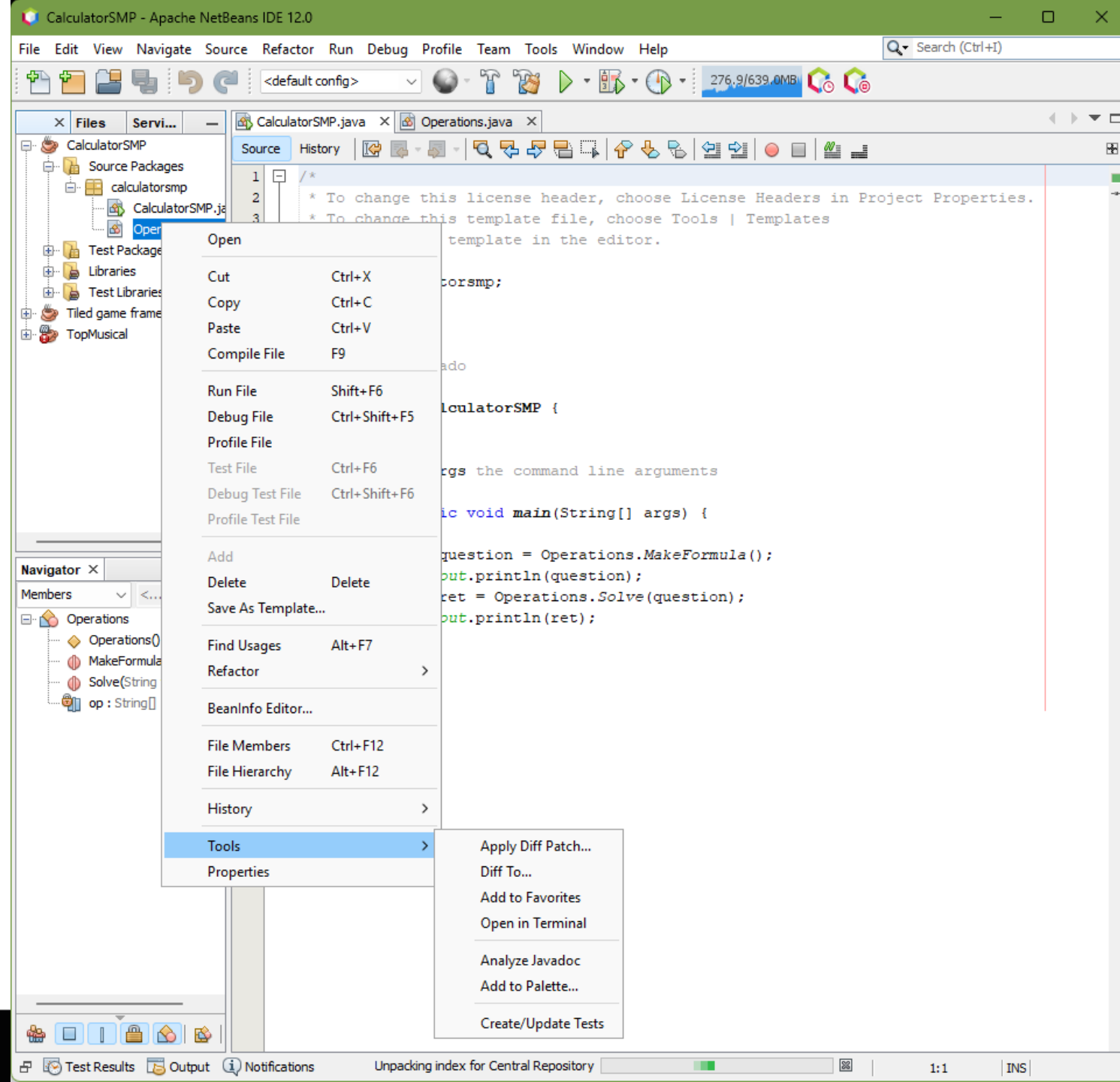
- Anotaciones Mejoradas.
- Arquitectura Moderna. Arquitectura modular y extensible.
- Soporte para Java 8 y Superior.
- JUnit 5 admite la inyección de dependencias directamente en las pruebas, permitiendo una mejor modularidad y facilitando la configuración y reutilización de recursos.
- Ejecución Paralela, especialmente en entornos de construcción continua.

Configuración de un proyecto en Netbeans

- La forma más sencilla es generar una clase de pruebas.
- El IDE agrega automáticamente las dependencias necesarias.
- Solo se debe escoger la versión del framework

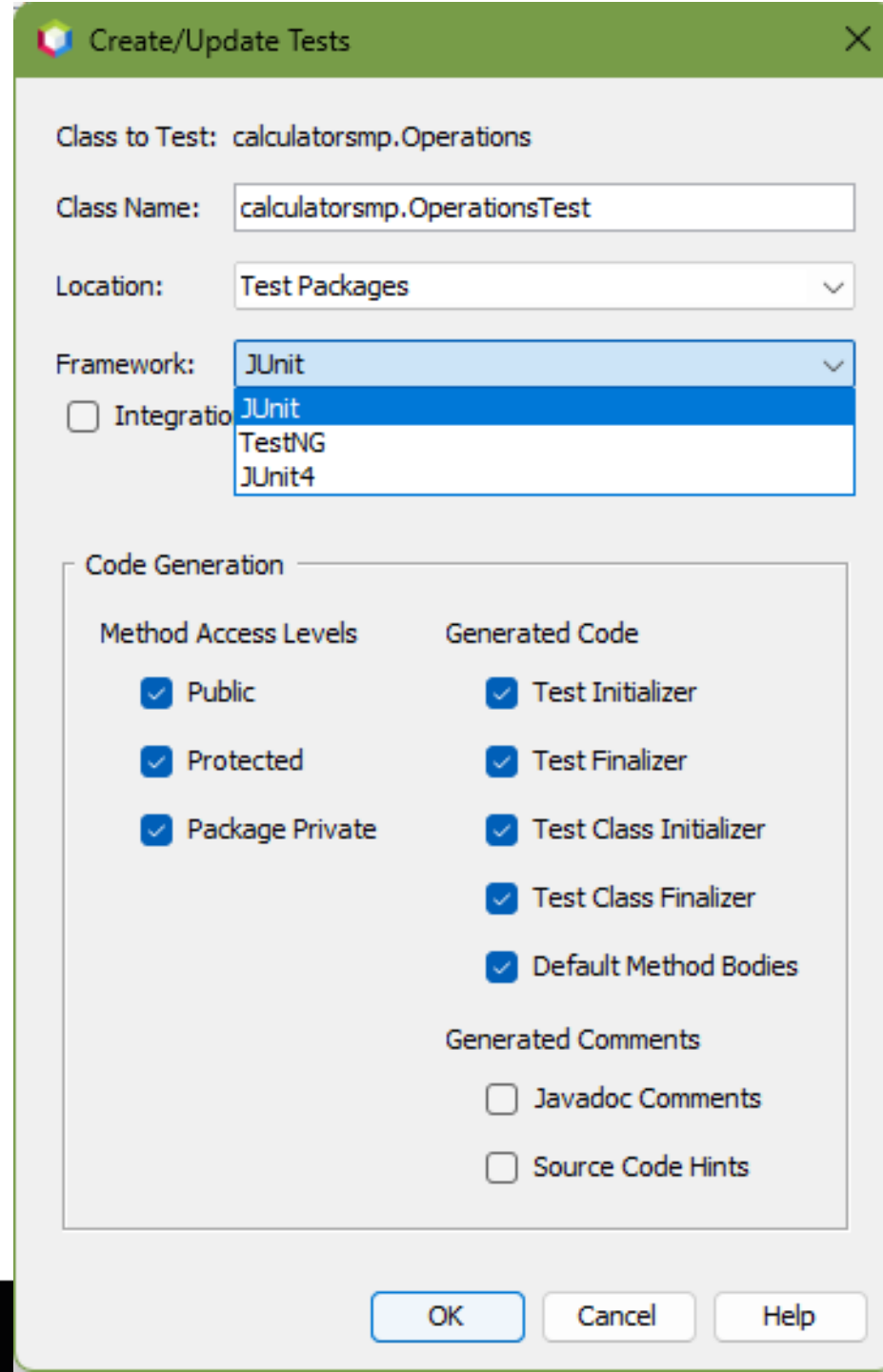
Generando una clase de pruebas

1. Abrir el proyecto Java en Netbeans.
2. Ir a la carpeta del código fuente.
3. Dar clic derecho sobre una clase java.
4. Ir a Tools → Create/Update Tests.



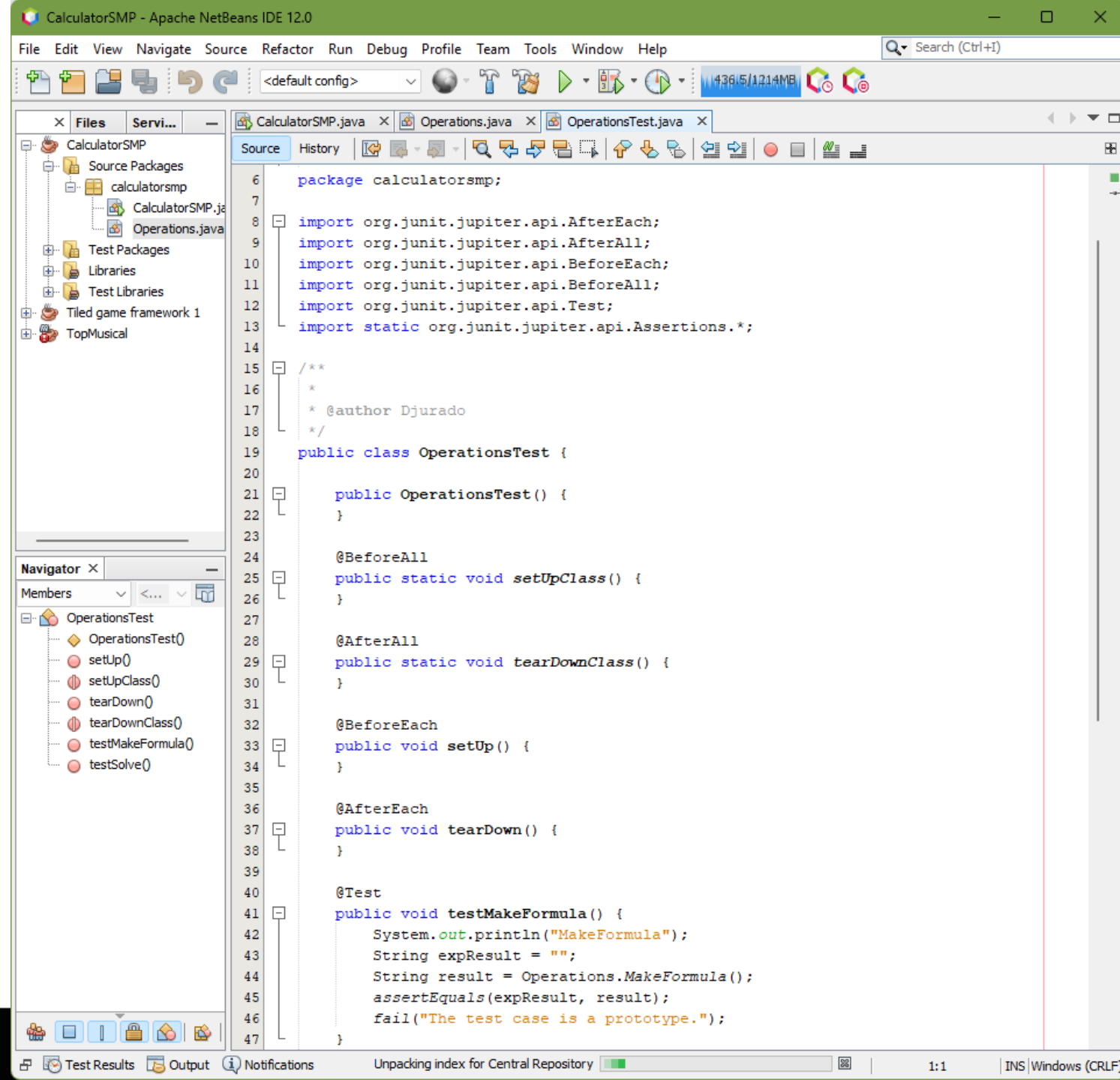
Elegir la versión del framework

1. JUnit se refiere a la versión 5, pero también se puede escoger JUnit 4.
2. Elegir lo que se desea que agregue de forma automática.
3. Aceptar.



Clase de pruebas generada

- Agrega dependencias
- Agrega métodos de configuración previa y posterior a las pruebas.
- Agrega un test por cada método de la clase a probar. (Esto puede cambiarse)
- Agrega las anotaciones correspondientes.



Escribiendo pruebas

- Tener claro el escenario que se desea probar; es decir, identificar cada **caso de prueba**.
- Agregar las **anotaciones** a los nuevos tests (métodos) creados.
- Agregar la comprobación (**asserts**) adecuada al caso de prueba que se desea implementar.
- Asegurarse de que **cada línea de código** haya sido ejecutada por alguno de los tests implementados.

Anotaciones

Principales

- **@Test**: Indica que el método es una prueba.
- **@BeforeEach** y **@AfterEach**: Se ejecutan antes y después de cada prueba, respectivamente.
- **@DisplayName**: Asigne nombres descriptivos a sus pruebas para mayor claridad.

Otras

- **@Timeout(long timeout)**
- **@RepeatedTest(int value)**
- **@Disabled**
- **@BeforeAll** y **@AfterAll**

Tipos de Asserts en JUnit 5

1. **assertEquals(expected, actual):** Verifica que los valores esperados y reales sean iguales.
2. **assertTrue(boolean condition):** Verifica que la condición dada sea verdadera.
3. **assertFalse(boolean condition):** Verifica que la condición dada sea falsa.
4. **assertNull(Object obj):** Verifica que el objeto dado sea nulo.
5. **assertThrows(ExceptionType.class, executable):** Verifica que se lance una excepción específica durante la ejecución.

Tipos de Asserts en JUnit 5

1. **assertArrayEquals(expectedArray, actualArray):** Verifica que los arreglos esperados y reales sean iguales.
2. **assertNotEquals(unexpected, actual):** Verifica que los valores no sean iguales.
3. **assertNotNull(Object obj):** Verifica que el objeto dado no sea nulo.
4. **assertSame(expected, actual):** Verifica que los objetos referencien la misma instancia en la memoria.

Ejecución de pruebas

- Ejecutar Prueba Individual: Hacer clic derecho en un método de prueba y seleccionar "Ejecutar".
- Ejecutar Todas las Pruebas: Hacer clic derecho en el proyecto y seleccionar "Test File" o "Test Project".

Lectura adicional

- <https://howtodoinjava.com/junit-5-tutorial/>