***DAY 23 HOME TASKS***

### ***Task 01: Methods of Assertion Class in JUnit***

The org.junit.Assert class provides static methods like:

| **Method** | **Description** |
| --- | --- |
| assertEquals(expected, actual) | Checks if two values are equal |
| assertNotEquals(expected, actual) | Checks if two values are not equal |
| assertTrue(condition) | Checks if a condition is true |
| assertFalse(condition) | Checks if a condition is false |
| assertNull(object) | Verifies object is null |
| assertNotNull(object) | Verifies object is **not** null |
| assertSame(expected, actual) | Checks if both refer to **same object** (==) |
| assertNotSame(expected, actual) | Checks if both refer to **different objects** |
| assertArrayEquals(expected, actual) | Compares arrays for equality |
| fail() | Marks test as failed immediately |

### ***Task 02: Implement assertNotSame, assertSame, and assertArrayEquals()***

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionDemo {

@Test

public void testAssertSame() {

String a = "hello";

String b = "hello";

assertSame("Both should point to same object", a, b);

}

@Test

public void testAssertNotSame() {

String a = new String("hello");

String b = new String("hello");

assertNotSame("Should be different objects", a, b);

}

@Test

public void testAssertArrayEquals() {

int[] expected = {1, 2, 3};

int[] actual = {1, 2, 3};

assertArrayEquals("Arrays should be equal", expected, actual);

}

}

### ***Task 03: @Ignore Test***

import org.junit.Ignore;

import org.junit.Test;

public class IgnoreDemo {

@Ignore("This test is ignored for now")

@Test

public void testToIgnore() {

System.out.println("This test should be skipped.");

}

@Test

public void testNormal() {

System.out.println("This test will run.");

}

}

**Note:** The ignored test will be skipped during execution. It will show as "ignored" in your test report/logs.

### ***Task 04: Timeout Test***

import org.junit.Test;

public class TimeoutDemo {

@Test(timeout = 1000)

public void testTimeout() throws InterruptedException {

// Simulate long-running task

Thread.sleep(2000); // Exceeds timeout -> test will fail

}

}

If test execution > 1000ms (1 sec), JUnit will throw org.junit.runners.model.TestTimedOutException.

### ***Task 05: Exception Test***

import org.junit.Test;

public class ExceptionDemo {

@Test(expected = ArithmeticException.class)

public void testArithmeticException() {

int x = 5 / 0; // Boom! Throws ArithmeticException

}

}

We can combine them into one test file like this:

import org.junit.\*;

import static org.junit.Assert.\*;

public class AllTasksTest {

@Test

public void testSameAndNotSame() {

String str1 = "abc";

String str2 = "abc";

assertSame(str1, str2);

String str3 = new String("abc");

assertNotSame(str1, str3);

}

@Test

public void testArrayEquals() {

int[] arr1 = {1, 2, 3};

int[] arr2 = {1, 2, 3};

assertArrayEquals(arr1, arr2);

}

@Ignore("Skipping this for now")

@Test

public void testIgnored() {

fail("This test should be ignored");

}

@Test(timeout = 1000)

public void testTimeoutFail() throws InterruptedException {

Thread.sleep(1500); // will cause timeout

}

@Test(expected = ArithmeticException.class)

public void testExpectedException() {

int result = 10 / 0;

}

}

**How to Run All This:**

* Add JUnit 4 to your pom.xml if using Maven:

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>