Assert library comparisons

Category	Junit4	Junit5	Truth	Asserti
Basic	assertEquals("Frodo", Frodo.getName()	assertEquals("Frodo", Frodo.getName()	assertThat(frodo.getName()).isEqualTo("Frodo")	assertThat(frodo.getName()).isEqualTo("Frodo")
busic	assertTrue(condition)	assertTrue(condition)	* ****	
	assertFalse(condition)	assertFalse(condition)		
	assertNull(object)	assertNull(object)		
	assertNotNull(object)	assertNotNull(object)	assertThat(frodo).isNotEqualTo(sauron)	assertThat(frodo).isNotEqualTo(sauron)
	aborenounary (object)	abbertioenarr (object)	about time (110do) . 10mo Englatio (0ddion)	abbeternat (110do) 110not29da110 (0dd20n)
Extended	N/A	N/A	assertThat(frodo.getName()).startsWith("Fro")	assertThat(frodo.getName()).startsWith("Fro")
data			assertThat(frodo.getName()).endsWith("do")	assertThat(frodo.getName()).endsWith("do")
methods				<pre>assertThat(frodo.getName()).isEqualToIgnoringCase("frodo")</pre>
			Not supported: isEqualToIgnoringCase	11000)
Array	assertArrayEquals(expectedArray,	assertArrayEquals(expectedArray,		
,	actualArray)	actualArray)		
Custom	assertEquals("Custom message", expected,	assertEquals(expected, actual, "Custom		
message	actual)	message")		
		assertEquals(expected, actual, () -> "Custom		
		message")		
Grounod	N/A	assertAll("Grouped assertion",	assertAll("Grouped assertion",	assertAll("Grouped assertion",
Grouped assertions	14/11	<pre>assertAll("Grouped assertion", () -> Assertions.assertEquals(expected),</pre>	() -> Assertions.assertEquals(expected1, actual1),	() -> Assertions.assertEquals(expected),
assertions		actual1),	() -> Assertions.assertEquals(expected2, actual2)	actual1),
		() -> Assertions.assertEquals(expected2,)	() -> Assertions.assertEquals(expected2,
		actual2)		actual2)
))
	fail("This test should fail")	fail("This test should fail")		fail("This test should fail")
Explicit	fall("This test should fall")	fall("This test should fall")	Not supported	fall("This test should fall")
failing test				
Exception	@Test(expected =	assertThrows(ExpectedException.class, ()	Not natively supported	assertThatThrownBy(() -> {
Testing	IllegalArgumentException.class)	-> {		// Code that is expected to throw the exception
.com.g	<pre>public void testExceptionWithExpected()</pre>	<pre>// code that should throw the exception }</pre>		throw new IllegalArgumentException("This is an
	{)		error message") })
	throw new IllegalArgumentException("This			.isInstanceOf(IllegalArgumentException.class)
	is an error message"); }			.hasMessageContaining("This is an error message")
Timeout	@Test(timeout = 2000)	assertTimeout(Duration.ofSeconds(2), () -> {	Not natively supported	Not natively supported
assertions	public void testTimeout() throws	// Code that should complete within 2		
assertions	InterruptedException {	seconds		
	Thread.sleep(1500); // a delay	Thread.sleep(1000);		
	}	});		
Florest	N/A	N/A	Limited fluency:	assertThat(frodo.getName()).startsWith("Fro")
Fluent assertions	IV/ FA	N/A	primited indepto:	<pre>.endsWith("do")</pre>
3351 (10113			assertThat(fellowshipOfTheRing).hasSize(2);	.isEqualToIgnoringCase("frodo")
			assertThat(fellowshipOfTheRing).contains(frodo);	
			assertThat(fellowshipOfTheRing).contains(sam);	
			assertThat(fellowshipOfTheRing).doesNotContain(Sauron)	
Massasina	assertEquals("check " + frodo.qetName()	assertEquals(frodo.getAge(), 33, "check " +	assertThat(frodo.getAge()).as("check %s's age",	assertThat(frodo.getAge()).as("check %s's age",
Messaging	<pre>assertEquals("cneck " + frodo.getName() + "'s age", frodo.getAge(), 33);</pre>	<pre>frodo.getName() + "'s age");</pre>	<pre>frodo.getName()).isEqualTo(33)</pre>	assertThat([rodo.getAge()).as("check %s's age", frodo.getName()).isEqualTo(33)
	. 3 ago , 110ao.getage(), 33//	11000.gctName() 5 age),	11000.gccname()).10Equatio(00)	110do.gcchame(//.15Equatio(33/
Soft Asserts	N/A	N/A	SoftAssertions softly = new SoftAssertions();	SoftAssertions softly = new SoftAssertions();
			<pre>softly.assertThat(frodo.getAge()).isNegative();</pre>	softly.assertThat(frodo.getAge()).isNegative();
			<pre>softly.assertThat(frodo.getAge()).isEqualTo(32);</pre>	softly.assertThat(frodo.getAge()).isEqualTo(32);
			softly.assertAll();	softly.assertAll();