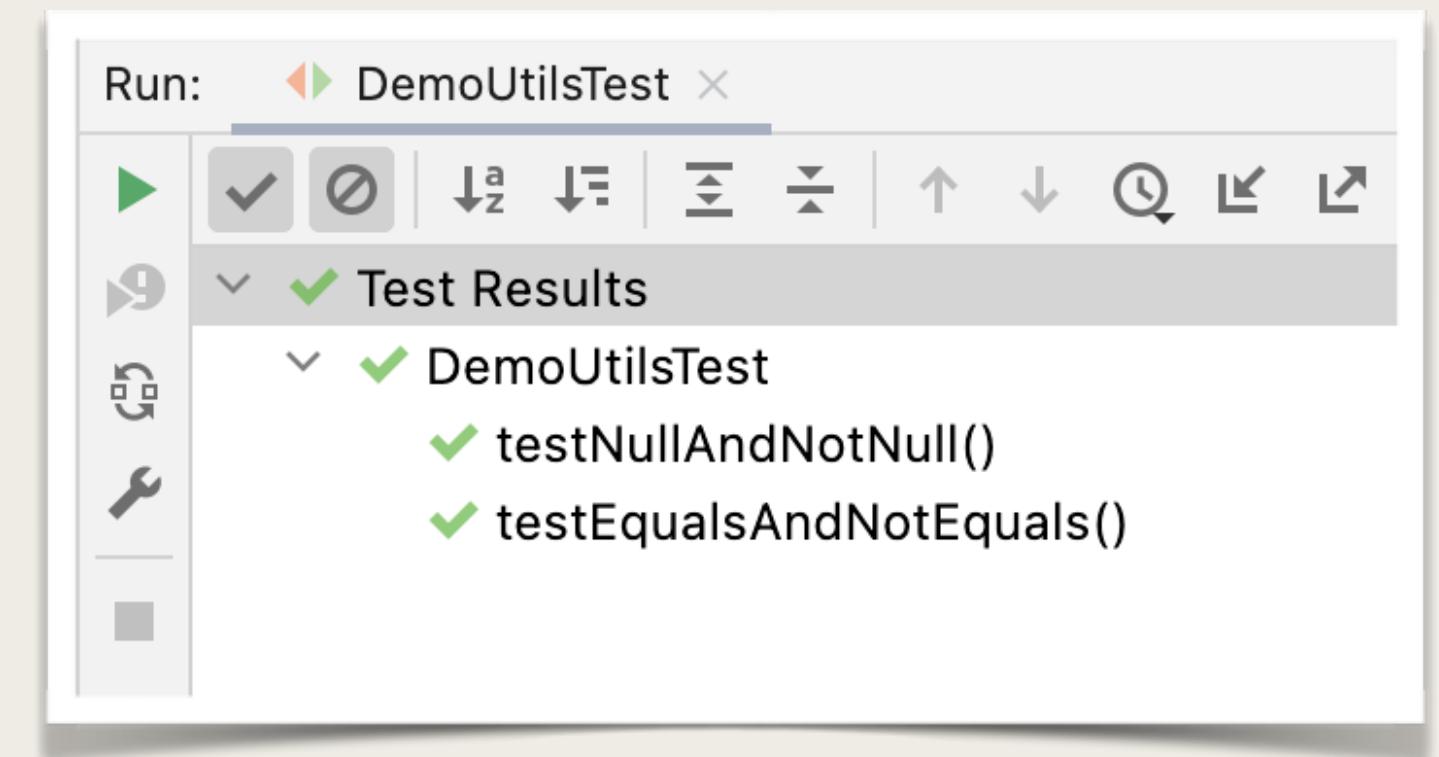


# JUnit Custom Display Names



# Custom Display Names

- Currently the method names are listed in test results
- We'd like to give custom display names
  - Provide a more descriptive name for the test
  - Include spaces, special characters: *Test for Equality to support JIRA #123*
  - Useful for sharing test reports with project management and non-techies



# @DisplayName Annotation

Annotation	Description
@DisplayName	<p>Custom display name with spaces, special characters and emojis.</p> <p><i>Useful for test reports in IDE or external test runner</i></p>

## DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayName;
import static org.junit.jupiter.api.Assertions.*;

class DemoUtilsTest {

    DemoUtils demoUtils;

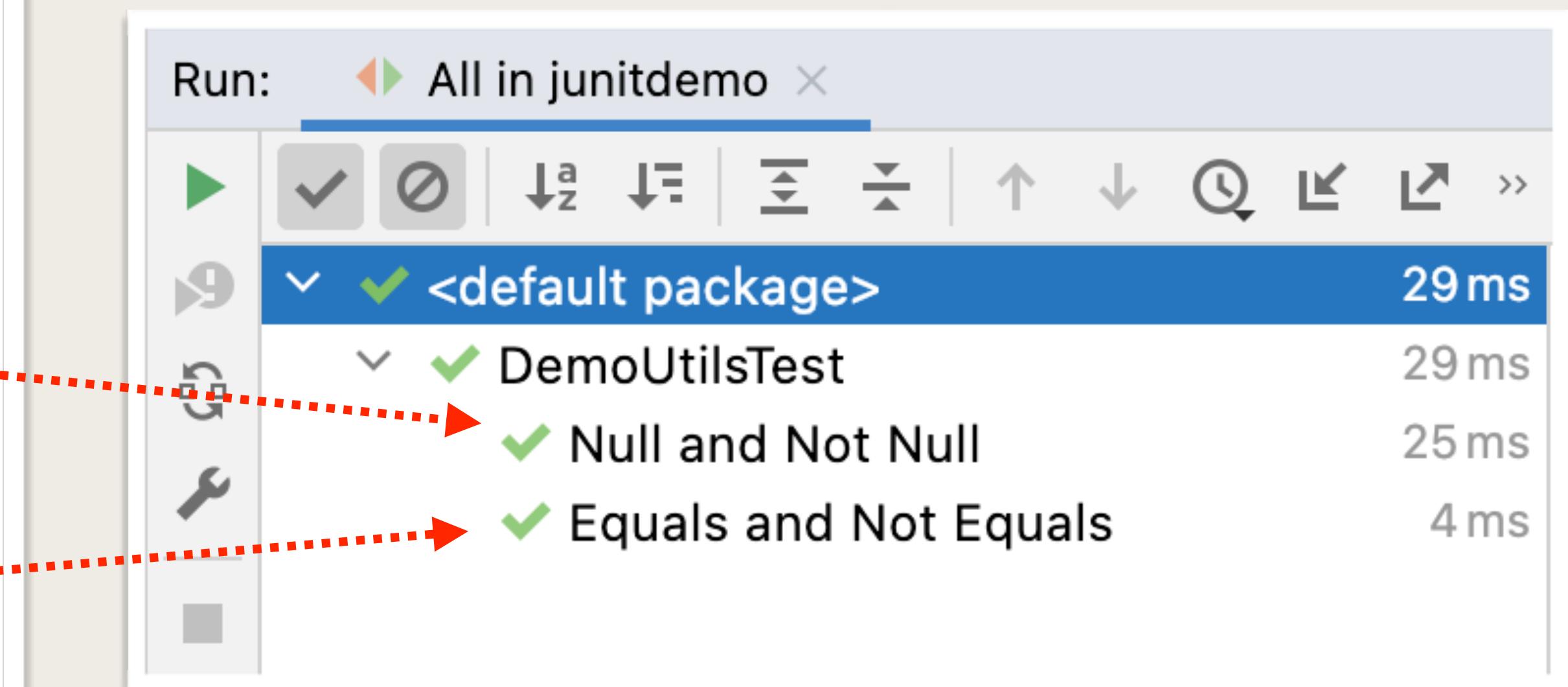
    @BeforeEach
    void setupBeforeEach() {
        // set up
        demoUtils = new DemoUtils();
    }

    @Test
    @DisplayName("Null and Not Null")
    void testNullAndNotNull() {
        String str1 = null;
        String str2 = "luv2code";

        assertNull(demoUtils.checkNotNull(str1), "Object should be null");
        assertNotNull(demoUtils.checkNotNull(str2), "Object should not be null");
    }

    @Test
    @DisplayName("Equals and Not Equals")
    void testEqualsAndNotEquals() {

        // execute and assert
        assertEquals(6, demoUtils.add(2, 4), "2+4 must be 6");
        assertNotEquals(8, demoUtils.add(1, 9), "1+9 must not be 8");
    }
}
```



# Display Name Generators

- JUnit can generate display names for you

Name	Description
Simple	Removes trailing parentheses from test method name
ReplaceUnderscores	Replaces underscores in test method name with spaces
IndicativeSentences	Generate sentence based on test class name and test method name

# Simple Generator

## DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayNameGeneration;
import org.junit.jupiter.api.DisplayNameGenerator;
import static org.junit.jupiter.api.Assertions.*;

@DisplayNameGeneration(DisplayNameGenerator.Simple.class)
class DemoUtilsTest {

    DemoUtils demoUtils;

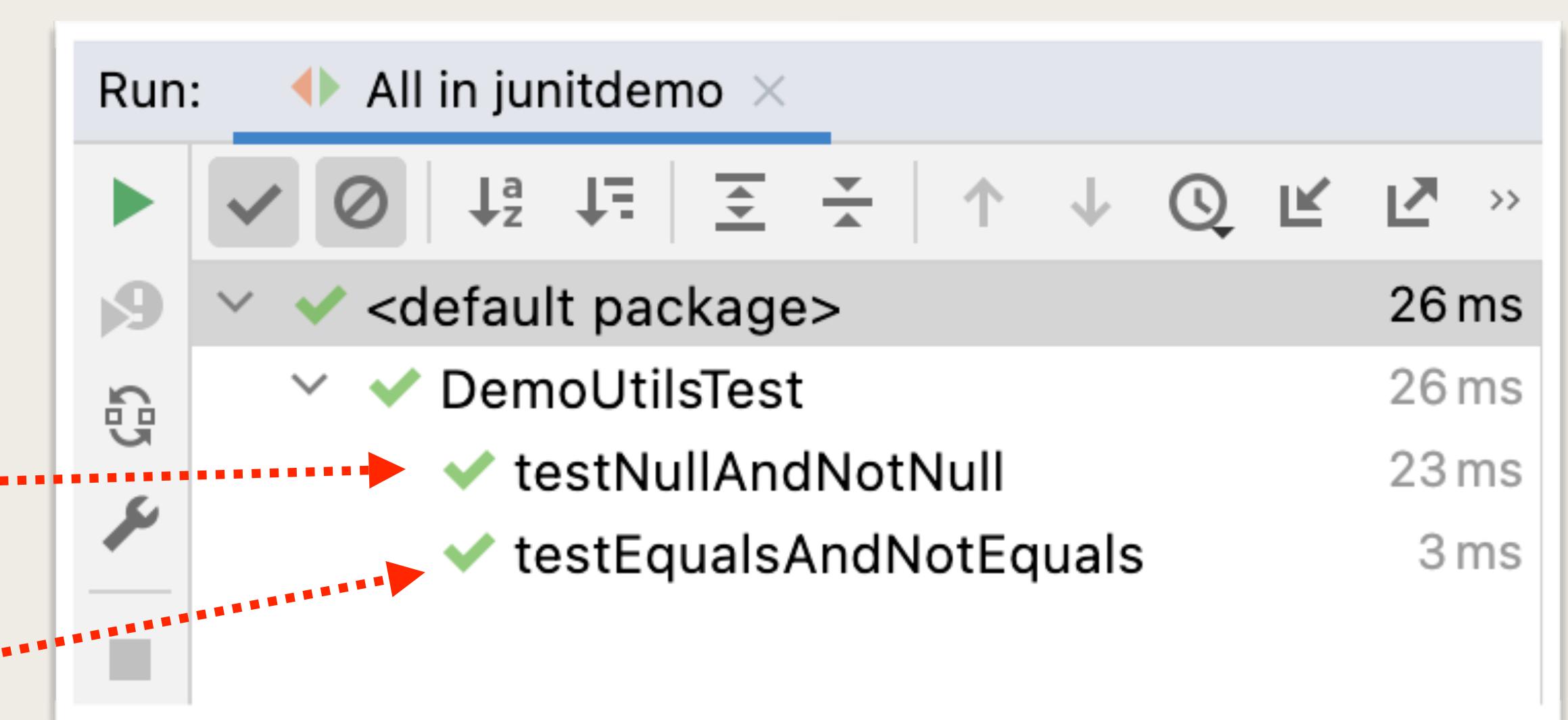
    @BeforeEach
    void setupBeforeEach() {
        // set up
        demoUtils = new DemoUtils();
    }

    @Test
    void testNullAndNotNull() {
        String str1 = null;
        String str2 = "luv2code";

        assertNull(demoUtils.checkNull(str1), "Object should be null");
        assertNotNull(demoUtils.checkNull(str2), "Object should not be null");
    }

    @Test
    void testEqualsAndNotEquals() {
        // execute and assert
        assertEquals(6, demoUtils.add(2, 4), "2+4 must be 6");
        assertNotEquals(8, demoUtils.add(1, 9), "1+9 must not be 8");
    }
}
```

Removes trailing parentheses from test method name



# Simple Generator

## DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayNameGeneration;
import org.junit.jupiter.api.DisplayNameGenerator;
import static org.junit.jupiter.api.Assertions.*;

@DisplayNameGeneration(DisplayNameGenerator.Simple.class)
class DemoUtilsTest {

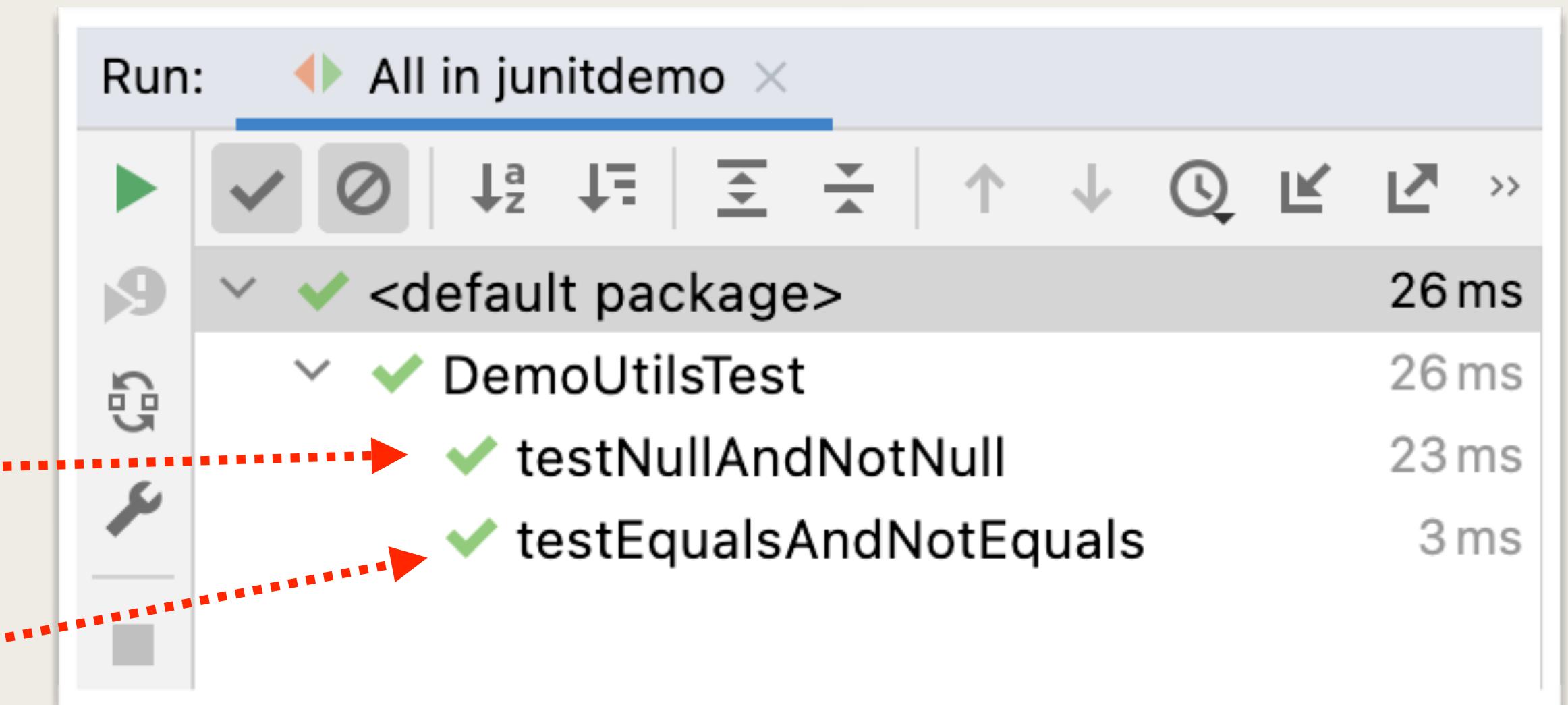
    DemoUtils demoUtils;

    @BeforeEach
    void setupBeforeEach() {
        // set up
        demoUtils = new DemoUtils();
    }

    @Test
    void testNullAndNotNull() {
        String str1 = null;
        String str2 = "luv2code";

        assertNull(demoUtils.checkNull(str1), "Object should be null");
        assertNotNull(demoUtils.checkNull(str2), "Object should not be null");
    }

    @Test
    void testEqualsAndHashCode() {
        // execute and assert
        assertEquals(6, demoUtils.add(2, 4), "2+4 must be 6");
        assertNotEquals(8, demoUtils.add(1, 9), "1+9 must not be 8");
    }
}
```



# Replace Underscores Generator

## DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayNameGeneration;
import org.junit.jupiter.api.DisplayNameGenerator;
import static org.junit.jupiter.api.Assertions.*;

@DisplayNameGeneration(DisplayNameGenerator.ReplaceUnderscores.class)
class DemoUtilsTest {

    DemoUtils demoUtils;

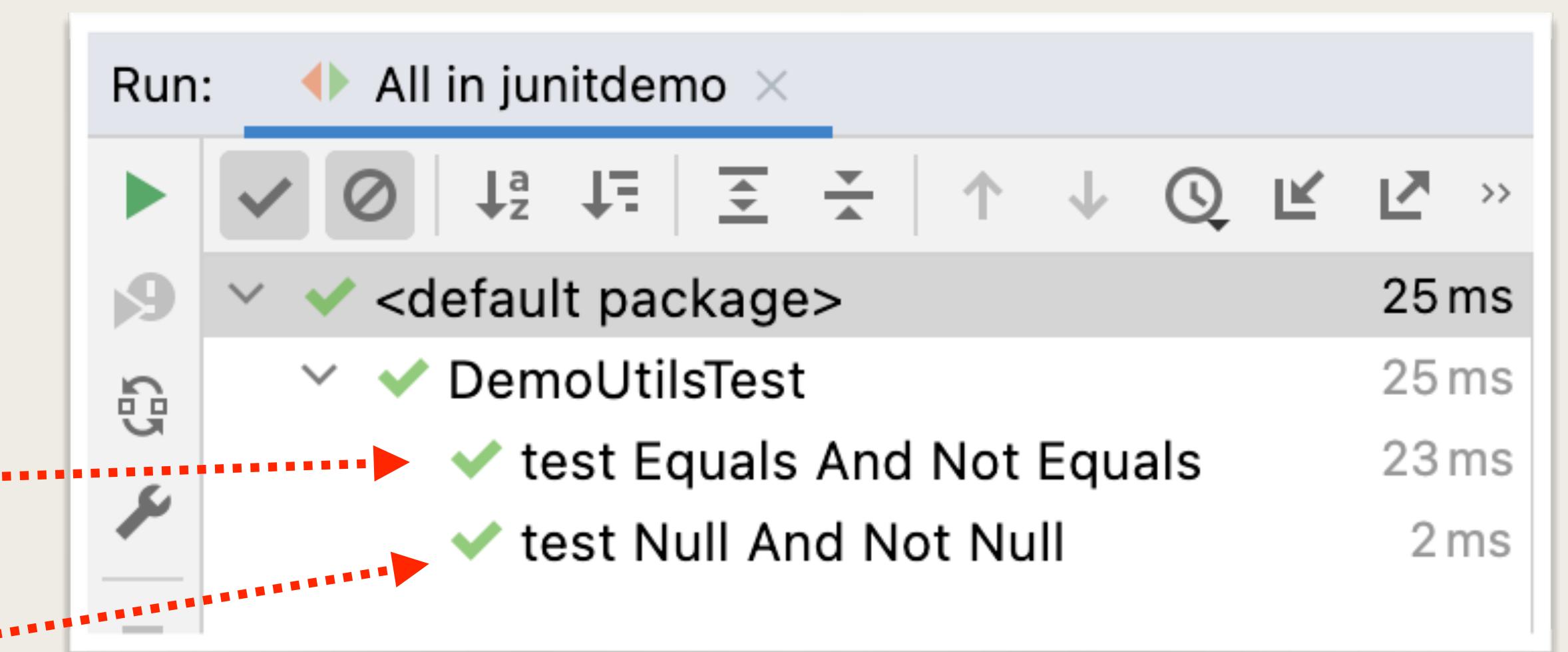
    @BeforeEach
    void setupBeforeEach() {
        // set up
        demoUtils = new DemoUtils();
    }

    @Test
    void test_Equals_And_Not_Equals() {
        // execute and assert
        assertEquals(6, demoUtils.add(2, 4), "2+4 must be 6");
        assertNotEquals(8, demoUtils.add(1, 9), "1+9 must not be 8");
    }

    @Test
    void test_Null_And_Not_Null() {
        String str1 = null;
        String str2 = "luv2code";

        assertNull(demoUtils.checkNull(str1), "Object should be null");
        assertNotNull(demoUtils.checkNull(str2), "Object should not be null");
    }
}
```

Replaces underscores with spaces



# Indicative Sentences Generator

## DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayNameGeneration;
import org.junit.jupiter.api.DisplayNameGenerator;
import static org.junit.jupiter.api.Assertions.*;

@DisplayNameGeneration(DisplayNameGenerator.IndicativeSentences.class)
class DemoUtilsTest {

    DemoUtils demoUtils;

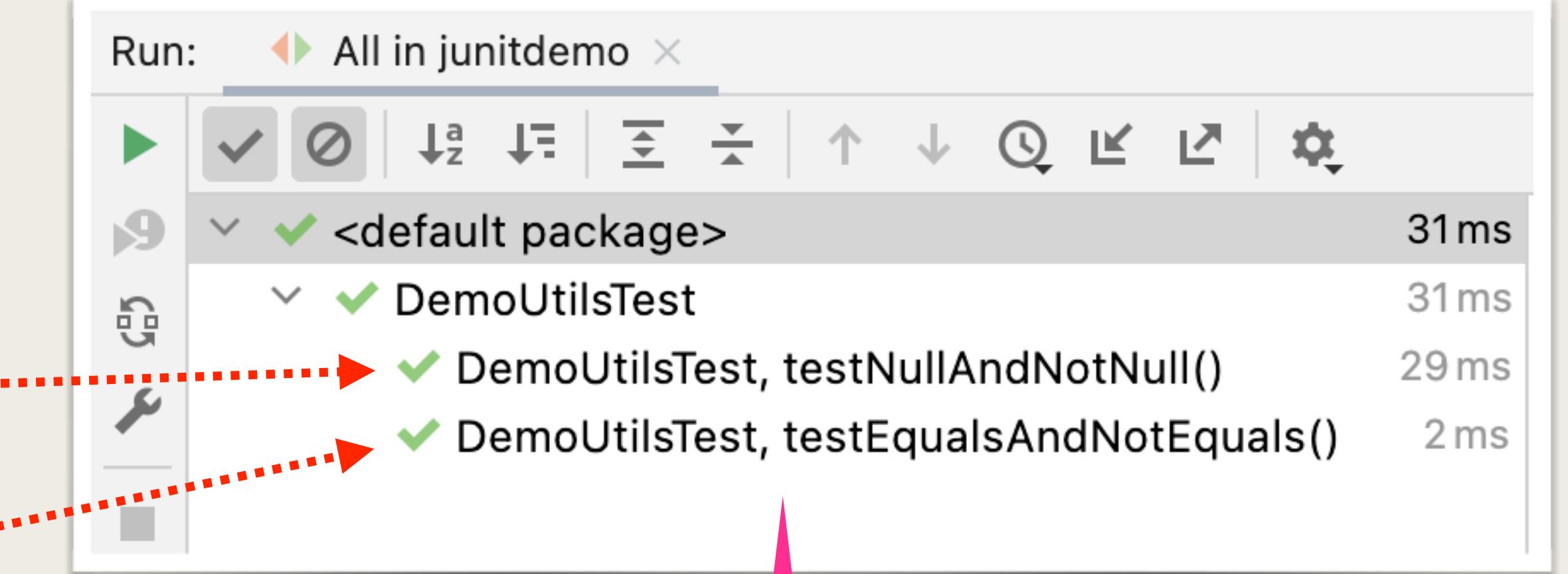
    @BeforeEach
    void setupBeforeEach() {
        // set up
        demoUtils = new DemoUtils();
    }

    @Test
    void testNullAndNotNull() {
        String str1 = null;
        String str2 = "luv2code";

        assertNull(demoUtils.checkNull(str1), "Object should be null");
        assertNotNull(demoUtils.checkNull(str2), "Object should not be null");
    }

    @Test
    void testEqualsAndNotEquals() {
        // execute and assert
        assertEquals(6, demoUtils.add(2, 4), "2+4 must be 6");
        assertNotEquals(8, demoUtils.add(1, 9), "1+9 must not be 8");
    }
}
```

Generate sentence based on  
test class name and test method name



<testClassName>, <methodName>