

Junit Tutorial Notes

https://github.com/newtonhobbs/JunitTutorial28Minutes

Date: @February 20, 2023

Topic: Introduction

Recall

- In src/main/java all the java packages/class are there
- In src/test/java all the packages/class to write the JUnit test case.
- @Test annotation is use to tells JUnit that the public void method to which it is attached can be run as a test case.



Notes

method after the @Test annotation should be public and void

assertEquals(expected, actual);
 eg: assertEquals("ABC", "ABCD"); - return failed unit test

assertFalse("Message to print after if method return true", method());

eg: assertFalse("The condition failed",

are First And Last Two Characters The Same ("ABCD")); - return failed unit test with message "The condition failed"

assertTrue("Message to print after if method return false", method());

eg: assertTrue("The condition failed",

areFirstAndLastTwoCharactersTheSame("ABCD")); - return failed unit test with message "The condition failed"

• Example:

```
Package Exp... X Julit - StringHelperTest.java StringHelper.java X
                                            package com.in28minutes.junit.helper;
                        E 🕏 :

✓ № JunitCourse

                                                public class StringHelper {
   //AACD => CD, ACD => CD, CDEF => CDEF
public String truncateAInFirst2Positions(String str) {
  if (str.length() <= 2)
    return str.replaceAll("A", "");</pre>

→ 

⊕ com.in28minutes.junit.helper

        >  StringHelper.java
     src/main/resources
   v 🌐 com.in28minutes.junit.helper
                                                         String first2Chars = str.substring(0, 2);
String stringMinusFirst2Chars = str.substring(2);
         > 🚺 StringHelperTest.java
     src/test/resources
                                                         return first2Chars.replaceAll("A", "")
   > 🔼 JRE System Library [J2SE-1.5]
                                                                   + stringMinusFirst2Chars;
   > M JUnit 4
                                                    }
   🗸 🇁 src
       a main
                                           179
                                                    public boolean areFirstAndLastTwoCharactersTheSame(String str) {
       test 🗁
                                                         if (str.length() <= 1)
    return false;
if (str.length() == 2)</pre>
                                           19
     target
                                           20
21
     m pom.xml
                                           22
                                                              return true;
                                           23
                                           24
                                                         String first2Chars = str.substring(0, 2);
                                           25
26
                                                         String last2Chars = str.substring(str.length() - 2):
                                           27
28
                                                         return first2Chars.equals(last2Chars);
                                           29
30
                                           31 }
```

Packages and method for which we have to write the unit test case

```
‡ Package Exp... × ♂v JUnit □ □ □ StringHelperTest.java × □ StringHelper.java
                                              1 package com.in28minutes.junit.helper;
                          E 🕏 🕺
3⊕ import static org.junit.Assert.*;[

→ 

⊕ com.in28minutes.junit.helper

                                             7 public class StringHelperTest {
        > 1 StringHelper.java
     src/main/resources
                                                         public void test() {
    StringHelper helper = new StringHelper();
    assertEquals("CD", helper.truncateAInFirst2Positions("AACD"));
    assertEquals("CD", helper.truncateAInFirst2Positions("ACD"));
    assertEquals("CDEF", helper.truncateAInFirst2Positions("CDEF"));
   🗸 🌐 com.in28minutes.junit.helper
         > I StringHelperTest.java
     # src/test/resources
   > A JRE System Library [J2SE-1.5]
                                             16
17 }
   > 🛋 JUnit 4
   🗸 🗁 src
                                               18
        (2) main
        test
     target
      m pom.xml
```

Junit test for above method: public String truncateAInFirst2Positions(String str) (Version:1)

```
# Package Explorer × ♂ Unit
                                                  🖹 💲 🖁 🗖 🗓 StringHelperTest.java 🗴 📝 StringHelper.java
✓ 🔛 JunitCourse
✓ 🕮 src/main/java
                                                                                     package com.in28minutes.junit.helper;
                                                                                   .
3⊕ import static org.junit.Assert.*;[]
      v 🌐 com.in28minutes.junit.helper
          > J StringHelper.java
                                                                                     public class StringHelperTest {
      src/main/resources

✓ 

src/test/java

✓ 

com.in28minutes.junit.helper
                                                                                           StringHelper helper = new StringHelper();
                                                                               10
11
129
13
14
15
16
179
20
21
229
23
24
25
26
278
28
                                                                                           //AACD => CD, ACD => CD, CDEF => CDEF, CDAA => CDAA
     public void testTruncateAInFirst2Positions_AinFirst2Positions() {
    assertEquals("CD", helper.truncateAInFirst2Positions("AACD"));
     ■ JRE System Library [J2SE-1.5]
     JUnit 4
                                                                                          }
   ∨ ြ src
         public void testTruncateAInFirst2Positions_AinFirstPosition()
    assertEquals("CD", helper.truncateAInFirst2Positions("ACD")
      target
      pom.xml
                                                                                            public void testTruncateAInFirst2Positions_AnotinFirst2Positions() {
    assertEquals("CDAA", helper.truncateAInFirst2Positions("CDAA"));
                                                                                            public void testTruncateAInFirst2Positions_AnotPrestent() {
    assertEquals("CDEF", helper.truncateAInFirst2Positions("CDEF"));
```

Junit test for above method : public String <u>truncateAlnFirst2Positions</u>(String str) (Version:2) || improvement in naming

```
E 🕏 ᠄
 public class StringHelperTest {
          > # com.in28minutes.junit.helper
                                                                                                                                                          StringHelper helper = new StringHelper();
                           > I StringHelper.java
                                                                                                                                                              //AACD => CD, ACD => CD, CDEF => CDEF, CDAA => CDAA
                                                                                                                                                      @Test
public void testTruncateAInFirst2Positions_AinFirst2Positions() {
    assertEquals("CD", helper.truncateAInFirst2Positions("AACD"));
               #Test
cominal/resources
#Final/resources
          public void testTruncateAInFirst2Positions_AnotinFirst2Positions() {
    assertEquals("CDAA", helper.truncateAInFirst2Positions("CDAA"));
                 pom.xml
                                                                                                                                                             public void testTruncateAInFirst2Positions_AnotPrestent() {
    assertEquals("CDEF", helper.truncateAInFirst2Positions("CDEF"));
                                                                                                                                                               //ABCD => false, ABAB => true, AB => true, A => false
                                                                                                                                                             wlest
public void testAreFirstAndLastTwoCharactersTheSame_BasicNegativeScenario() {
    assertFalse("The condition failed", helper.areFirstAndLastTwoCharactersTheSame("ABCD"));
```

Junit test for above method : **public boolean** areFirstAndLastTwoCharactersTheSame(String str) (Version:1) || using of assertFalse

```
| Package Ex... X | Ju Unit | D | StringHelperfestjava X | StringHelperjava | StringHelpe
                                                                          E 🕏 🛚
  public void testTruncateAInFirst2Positions_AnotinFirst2Positions() {
    assertEquals("CDAA", helper.truncateAInFirst2Positions("CDAA"));

→ 

⊕ com.in28minutes.junit.helper

                         >   StringHelper.java
                # src/main/resources
                                                                                                                                       @Test
public void testTruncateAInFirst2Positions AnotPrestent() {
    assertEquals("CDEF", helper.truncateAInFirst2Positions("CDEF"));

✓ 

src/test/java

✓ 

com.in28minutes.junit.helper
                        >  StringHelperTest.java
             //ABCD => false, ABAB => true, AB => true, A => false
          > ml JUnit 4
                                                                                                                                        public void testAreFirstAndLastTwoCharactersTheSame_BasicNegativeScenario() {
    assertFalse("The condition failed", helper.areFirstAndLastTwoCharactersTheSame("ABCD"));
          ✓ 🧀 src

🧀 main

    test

               target pom.xml
                                                                                                                                        @Test
public void testAreFirstAndLastTwoCharactersTheSame_BasicPositiveScenario() {
    assertTrue("The condition failed", helper.areFirstAndLastTwoCharactersTheSame("ABAB"));
    .
                                                                                                                                         @Test
public void testAreFirstAndLastTwoCharactersTheSame_TwoChar() {
    assertTrue("The condition failed", helper.areFirstAndLastTwoCharactersTheSame("AB"));
    .
                                                                                                                                          public void testAreFirstAndLastTwoCharactersTheSame_SingleCharOnly() {| assertFalse("The condition failed", helper.areFirstAndLastTwoCharactersTheSame("A"));
```

Junit test for above method : **public boolean** areFirstAndLastTwoCharactersTheSame(String str) (Version:2) || using of assertTrue

Code

```
if (str.length() <= 1)
    return false;
if (str.length() == 2)
    return true;

String first2Chars = str.substring(0, 2);

String last2Chars = str.substring(str.length() - 2);

return first2Chars.equals(last2Chars);
}</pre>
```

```
package com.in28minutes.junit.helper;
import static org.junit.Assert.*;
import org.junit.Test;
public class StringHelperTest {
  StringHelper helper = new StringHelper();
  //AACD \Rightarrow CD, ACD \Rightarrow CD, CDEF \Rightarrow CDEF, CDAA \Rightarrow CDAA
  @Test
  public void testTruncateAInFirst2Positions_AinFirst2Positions() {
    assert \verb|Equals("CD", helper.truncateAInFirst2Positions("AACD"));\\
  @Test
  \verb"public void testTruncateAInFirst2Positions_AinFirstPosition"() \ \{
   assertEquals("CD", helper.truncateAInFirst2Positions("ACD"));
  @Test
  public \ void \ testTruncateAInFirst2Positions\_AnotinFirst2Positions() \ \{
   assertEquals("CDAA", helper.truncateAInFirst2Positions("CDAA"));
  @Test
  public void testTruncateAInFirst2Positions_AnotPrestent() {
   assertEquals("CDEF", helper.truncateAInFirst2Positions("CDEF"));
  //ABCD \Rightarrow false, ABAB \Rightarrow true, AB \Rightarrow true, A \Rightarrow false
  \verb"public void testAreFirstAndLastTwoCharactersTheSame\_BasicNegativeScenario() \ \{ \ \ \ \}
   assert False ("The \ condition \ failed", \ helper.are First And Last Two Characters The Same ("ABCD"));
  }
  \verb"public void testAreFirstAndLastTwoCharactersTheSame\_BasicPositiveScenario() \ \{
   assert \verb|True| ("The condition failed", helper.are \verb|FirstAndLastTwoCharactersTheSame("ABAB")); \\
  }
  public void testAreFirstAndLastTwoCharactersTheSame_TwoChar() {
   assert \verb|True| ("The condition failed", helper.are \verb|FirstAndLastTwoCharactersTheSame("AB")); \\
  @Test
  public\ void\ testAreFirstAndLastTwoCharactersTheSame\_SingleCharOnly()\ \{
    assertFalse ("The \ condition \ failed", \ helper.are First And Last Two Characters The Same ("A"));
}
```

SUMMARY:

Date: @February 20, 2023

Topic: Junit @Before and @After

Recall

Notes

- @Before Methods annotated with the @Before annotation are run before each test.
- @After Methods annotated with the @After annotation are run after each test.



Method under @Before and @After should be public and void

- @BeforeClass (Once-only setup) It runs only once for the entire test class before any
 of the tests are executed, and prior to any @Before method(s).
- @AfterClass (Once-only tear down) It runs only once after all test case methods and @After annotations have been executed.



Method under @BeforeClass and @AfterClass should be **public**, **void and static**.



SUMMARY:

Date: @February 20, 2023

Topic: Junit @AssertArrayEquals And Handling Exception

Recall

Notes

@AssertArrayEquals(expectedArray, actualArray);
 Check expectedArray is equal to actual or not if it equal than test pass otherwise fail Example:

```
public void testArraySort_RandomArray() {
    int[] numbers = {12, 3, 4, 1};
    int[] expected = {1, 3, 4, 12};
    Arrays.sort(numbers);
    assertArrayEquals(expected, numbers);
}
```

```
📮 Package Expl... 🗶 🚮 ʊ JUnit 📅 🗖 🚺 StringHelperTest.java 🔝 StringHelper.java 🕡 ArrayCompareTest.java 🗴
                          □ 🕏 🕄
                                        4 package com.in28minutes.junit.helper;
Ge import static org.hamcrest.CoreMatchers.nullValue;
7 import static org.junit.Assert.*;
8

→ 

⊕ com.in28minutes.junit.helper

       > I StringHelper.java
                                        9 import java.util.Arrays;
    src/main/resources
  11 import org.junit.Test;
    > 🔃 ArrayCompareTest.java
                                       13@ /**
                                      14 * @author NHI437
15 *
       > I StringHelperTest.java
    src/test/resources
                                       16
  > 🔼 JRE System Library [J2SE-1.5]
                                       17 public class ArrayCompareTest {
  > 🛋 JUnit 4
                                      18
  🗸 🐎 src
                                               public void testArraySort_RandomArray() {
   int[] numbers = {12, 3, 4, 1};
   int[] expected = {1, 3, 4, 12};
   Arrays.sort(numbers);
      🗁 main
      海 test
    🗁 target
    m pom.xml
                                                   assertArrayEquals(expected, numbers);
```

Handling Exception

1. Using Try-Catch

```
//Method:1 for Exception
@Test
public void testArraySort_NullArray1() {
    int[] numbers = null;
    try {
        Arrays.sort(numbers);
    } catch (NullPointerException e) {
        // TODO: handle exception
    }
}
```

2. Using @Test(expected = ———)

```
//Method:2 for Exception (Provides by JUnit)
@Test(expected = NullPointerException.class)
public void testArraySort_NullArray2() {
   int[] numbers = null;
   Arrays.sort(numbers);
}
```



if we use @Test(expected = Exception1) means we expected Excepting will occur and if there is no or other exception will occur than this test will give failed Test.

Example:

```
//Here mention that we expect "NullPointerException" if,
//There is no "NullPointerException" than this test will
//show test failed.
@Test(expected = NullPointerException.class)
public void testArraySort_NullArray3() {
   int[] numbers = {};
   //Here we pass empty array but expected NullPointerException
   //i.e we got Test failed.
   Arrays.sort(numbers);
}
```

Example:

```
🔋 Package Expl... 🗙 😈 JUnit 📅 🗖 🚺 StringHelperTest.java 🚺 StringHelper.java 🕡 ArrayCompareTest.java 🗴
                                               public class ArrayCompareTest {

✓ (#) src/main/java

✓ (#) com.in28minutes.junit.helper
                                                                                        public void testArraySort_RandomArray() {
                                                                                                int[] numbers = {12, 3, 4, 1};
int[] expected = {1, 3, 4, 12};
Arrays.sort(numbers);
              >   StringHelper.java
                                                                    23
24
25
26
27
28
30
31
32
33
33
36
37
36
37
40
41
42
43
44
45
46
47
48
49
50
51
52
                                                                                                assertArrayEquals(expected, numbers);

✓ 

## src/test/iava

✓ 

    Com.in28minutes.junit.helper

             > 🔝 ArrayCompareTest.java
                                                                                        //Method:1 for Exception
               > I StringHelperTest.java
                                                                                       @Test
public void testArraySort_NullArray1() {
   int[] numbers = null;
   try {
        Arrays.sort(numbers);
   } catch (NullPointerException e) {
        // TODO: handle exception
        src/test/resources
       → JRE System Library [J2SE-1.5]
     > 🛋 JUnit 4
    }
             🐎 test
        //Method:2 for Exception (Provides by 3Unit)
@Test(expected = NullPointerException.class)
public void testArraySort_NullArray2() {
   int[] numbers = null;
   Arrays.sort(numbers);
}
         m pom.xml
                                                                                       ...
./Here mention that we expect "NullPointerException" if,
//There is no "NullPointerException" than this test will
//show test failed.
@Test(expected = NullPointerException.class)
public void testArraySort_NullArray3() {\( \frac{1}{2} \) int[] numbers = {\( \frac{1}{2} \) };
//Here we pass empty array but expected NullPointerException
//i.e. we got Test failed.
Arrays.sort(numbers);
}
```

Code:

```
package com.in28minutes.junit.helper;
import static org.hamcrest.CoreMatchers.nullValue;
import static org.junit.Assert.*;
import java.util.Arrays;
import org.junit.Test;
public class ArrayCompareTest {
 public void testArraySort_RandomArray() {
   int[] numbers = {12, 3, 4, 1};
   int[] expected = {1, 3, 4, 12};
   Arrays.sort(numbers);
   assertArrayEquals(expected, numbers);
 }
 //Method:1 for Exception
 public void testArraySort_NullArray1() {
   int[] numbers = null;
   try {
     Arrays.sort(numbers);
   } catch (NullPointerException e) {
     // TODO: handle exception
   }
 //Method:2 for Exception (Provides by JUnit)
 @Test(expected = NullPointerException.class)
 public void testArraySort_NullArray2() {
   int[] numbers = null;
   Arrays.sort(numbers);
 //Here mention that we expect "NullPointerException" if,
 //There is no "NullPointerException" than this test will
  //show test failed.
 @Test(expected = NullPointerException.class)
 public void testArraySort_NullArray3() {
   int[] numbers = {};
    //Here we pass empty array but expected NullPointerException
    //i.e we got Test failed.
```

```
Arrays.sort(numbers);
}
```



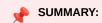
Date: @February 20, 2023

Topic: Performance Test (@Test(timeout = —— milisec)

Recall Notes

Code

```
//Performance Test
@Test(timeout = 100)
public void testSort_performance() {
  int array[] = {12, 23, 4};
  for(int i=1; i<=1000000; i++) {
    array[0] = i;
    Arrays.sort(array);
  }
}</pre>
```



Date: @Today

Topic: Parametrized Test (@RunWith(Parameterized.class)) and Junit Test Suite

Recall Notes

- We can't have two parameterized test in same class we have to make two different class for it. Because we have have two constructor in same class.
 example:
 - 1. StringHelperParameterizedTest class for ${{\text{"AACD", "CD"}, {\text{"ACD", "CD"}}, {\text{"CDEF"}, {\text{"CDAA"}}}}$
 - 2. StringHelperParameterizedTest2 class for -
- · Parameterized class should be under the annotation -

@RunWith(Parameterized.class)

Parameter method should be under the annotation - @Parameters



Method under the @Parameters annotation should be public and static.

- Example Code
 - 1. AACD => CD, ACD => CD, CDEF => CDEF, CDAA => CDAA

```
🔋 Package Explorer 🗙 💇 JUnit 👚 🗖 🔝 StringHelperParameterizedTest.java 🗙 🔝 StringHelperParameterizedTest2.java
                                                                                   | package com.in28minutes.junit.nelper;
| package com.in28minutes.junit.nelper;
| selection | package com.in28minutes.juni
```

```
package com.in28minutes.junit.helper;
import\ static\ org. hamcrest. Core {\tt Matchers.nullValue};
import static org.junit.Assert.*;
import iava.util.Arravs:
import java.util.Collection;
import java.util.List;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;
@RunWith(Parameterized.class)
public class StringHelperParameterizedTest {
      //AACD => CD, ACD => CD, CDEF => CDEF, CDAA => CDAA
      StringHelper helper = new StringHelper();
      private String input;
      private String expextedOutput;
      public StringHelperParameterizedTest(String input, String expextedOutput) {
           this.input = input;
           this.expextedOutput = expextedOutput;
     @Parameters
      public static Collection<String[]> testConditions() {
           String \ expectedOutputs[][] = \{ \{ \text{"AACD", "CD"} \}, \ \{ \text{"ACD", "CD"} \}, \ \{ \text{"CDEF", "CDEF"} \}, \ \{ \text{"CDAA", CD", CD"} \}, \ \{ \text{"CDEF", CDEF", CDEF"} \}, \ \{ \text{"CDAA", CD", CD", CD"} \}, \ \{ \text{"CDEF", CDEF", CDEF"
           Collection<String[]> ExpectedList = Arrays.asList(expectedOutputs);
           return ExpectedList;
     }
     @Test
      public void testTruncateAInFirst2Positions() {
           assertEquals(expextedOutput, helper.truncateAInFirst2Positions(input));
}
```

StringHelperParameterizedTest2.java

2. ABCD => false, ABAB => true, AB => true, A => false

```
package com.in28minutes.junit.helper;
import static org.hamcrest.CoreMatchers.nullValue;
import static org.junit.Assert.*;
import java.util.Arrays;
import java.util.Collection;
import java.util.List;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import\ org.junit.runners.Parameterized.Parameters;\\
@RunWith(Parameterized.class)
public class StringHelperParameterizedTest2 {
  //ABCD => false, ABAB => true, AB => true, A => false
  StringHelper helper = new StringHelper();
  String input:
  \verb"public StringHelperParameterizedTest2(String input) \{
   this.input = input;
 }
  @Parameters
  public static List<String> testConditions() {
   String[] expectedOutputs = {"ABCD", "ABAB", "AB", "A"};
    return Arrays.asList(expectedOutputs);
 public void testAreFirstAndLastTwoCharactersTheSame() {
   assertFalse("The condition failed", helper.areFirstAndLastTwoCharactersTheSame(input));
}
```

• **Junit Test Suite** - To Combine the unit test as one and run this file will run all the class that are in that test suite.

How to create Junit test suite?

Click New Project \rightarrow Search Junit suite \rightarrow select the test classes that you want to include in test suite \rightarrow Name it and click on create. example:

@SuiteClasses({ StringHelperParameterizedTest.class, StringHelperParameterizedTest2.class,add all the class that you want to combine....})

