

Junit Testing in Java

COMP2511, CSE, UNSW



UNSW
SYDNEY

Software Testing

❖ Different types of testing:

- Object Oriented Design document describes responsibilities of classes and methods (APIs) → **Unit Testing**
- System Design Document → Integration Testing
- Requirements Analysis Document → System Testing
- Client Expectation → Acceptance Testing

❖ **Unit Testing** is also useful for **refactoring** tasks.

❖ In this course, we will focus on Unit testing.

JUnit

- ❖ **JUnit** is a popular unit testing (open source) framework for testing Java programs.
- ❖ Most popular IDEs facilitate easy integration of Junit.
- ❖ Basic Junit Terminology:
 - **Test Case** – Java class containing test methods
 - **Test Method** – a method that executes the test code, annotated with `@Test`, in a Test Case
 - **Asserts** - asserts or assert statements check an expected result versus the actual result
 - **Test Suites** – collection of several Test Cases

JUnit Example: assertEquals, assertTrue

```
14  /**
15   * Tests for Pineapple on Piazza
16   * @author Nick Patrikeos
17   */
18  public class PiazzaTest {
19
20      @Test
21      public void testExampleUsage() {
22          // Create a forum and make some posts!
23          PiazzaForum forum = new PiazzaForum("COMP2511");
24          assertEquals("COMP2511", forum.getName());
25
26          Thread funThread = forum.publish("The Real Question - Pineapple on Piazza", "Who likes pineapple on piazza?");
27
28          funThread.setTags(new String[] { "pizza", "coding", "social", "hobbies" });
29          assertTrue(
30              Arrays.equals(new String[] { "coding", "hobbies", "pizza", "social" }, funThread.getTags().toArray()));
31
32          funThread.publishPost("Yuck!");
33          funThread.publishPost("Yes, pineapple on pizza is the absolute best");
34          funThread.publishPost("I think you misspelled pizza btw");
35          funThread.publishPost("I'll just fix that lol");
36
37          assertEquals(5, funThread.getPosts().size());
38      }
39
40      @Test
41      public void testSearchByTag() {
42          PiazzaForum forum = new PiazzaForum("COMP2511");
43
44          Thread labThread = forum.publish("Lab 01", "How do I do the piazza exercise?");
45          Thread assignmentThread = forum.publish("Assignment", "Are we back in blackout?");
46          labThread.setTags(new String[] { "Java" });
47          assignmentThread.setTags(new String[] { "Java" });
48
49          List<Thread> searchResults = forum.searchByTag("Java");
50          assertEquals("Lab 01", searchResults.get(0).getTitle());
51          assertEquals("Assignment", searchResults.get(1).getTitle());
52      }
53  }
```

```
15 public class ArchaicFsTest {
16     @Test
17     public void testCdInvalidDirectory() {
18         ArchaicFileSystem fs = new ArchaicFileSystem();
19
20         // Try to change directory to an invalid one
21         assertThrows(UNSWNoSuchFileException.class, () -> {
22             fs.cd("/usr/bin/cool-stuff");
23         });
24     }
25
26     @Test
27     public void testCdValidDirectory() {
28         ArchaicFileSystem fs = new ArchaicFileSystem();
29
30         assertDoesNotThrow(() -> {
31             fs.mkdir("/usr/bin/cool-stuff", true, false);
32             fs.cd("/usr/bin/cool-stuff");
33         });
34     }
35
36     @Test
37     public void testCdAroundPaths() {
38         ArchaicFileSystem fs = new ArchaicFileSystem();
39
40         assertDoesNotThrow(() -> {
41             fs.mkdir("/usr/bin/cool-stuff", true, false);
42             fs.cd("/usr/bin/cool-stuff");
43             assertEquals("/usr/bin/cool-stuff", fs.cwd());
44             fs.cd("..");
45             assertEquals("/usr/bin", fs.cwd());
46             fs.cd("../bin/..");
47             assertEquals("/usr", fs.cwd());
48         });
49     }
50
51     @Test
52     public void testCreateFile() {
53         ArchaicFileSystem fs = new ArchaicFileSystem();
54
55         assertDoesNotThrow(() -> {
56             fs.writeToFile("test.txt", "My Content", EnumSet.of(FileWriteOptions.CREATE, FileWriteOptions.TRUNCATE));
57             assertEquals("My Content", fs.readFromFile("test.txt"));
58             fs.writeToFile("test.txt", "New Content", EnumSet.of(FileWriteOptions.TRUNCATE));
59             assertEquals("New Content", fs.readFromFile("test.txt"));
60         });
61     }
62 }
```

Junit: Dynamic and parameterized tests

For information on Dynamic and parameterized tests,

- see the tutorial at <https://www.vogella.com/tutorials/JUnit/article.html>

❖ For more information on JUnit, read the **user guide** at:

- <https://junit.org/junit5/docs/current/user-guide/>

End