1. JUnit Test Suite

Any class can contain JUnit tests, but by convention the class name ends with Test: PurseTest.java, StackTest.java, etc. The test class is usually in the ***same package*** as the class under test.

2. Boilerplate Code for Class

Your IDE will generate boilerplate code if you create a new "JUnit Test" instead of a plain Java class. Typical code is:

```
import static org.junit.Assert.*;
// Matchers used with assertThat
import static org.hamcrest.CoreMatchers.*;

import org.junit.Before;
import org.junit.Test;

public class PurseTest {
      private static final double TOL = 1.0E-6; // tolerance for comparison
      // a "test fixture" - object to test
   private Purse purse;

      @Before
      public void setUp() throws Exception {
              // any code you want to run before each test
      }

      /** A test method annotated by @Test */
      @Test
      public void testNewPurseIsEmpty() {
        purse = new Purse(100);
        assertEquals( 0, purse.count() );
        // same thing, using assertThat
        assertThat( purse.count(), is(0) );
   }
```

3. Common JUnit Assert methods. These are static methods in org.junit.Assert

assertEquals(*expected, actual*)	assertEquals(0, purse.count())
assertEquals("*message*", *expected*, *actual*)	assertEquals("Should be empty", 0, purse.count())
assertEquals(*expect*, *actual*, *tolerance*)	For compariing floating point values, you should specify a tolerance for two values to be considered "equal". Tolerance may be 0. assertEquals(0.0, purse.getBalance(), 1.0E-6)
assertSame(*expected*, *actual*)	Test if two object variables refer to the same object. This is like: assertTrue(expected == actual)
assertTrue(*boolean_expression*)	
assertFalse(*boolean_expression*)	assertFalse(purse.isFull())
assertNull(*variable*)	assertNull(purse.withdraw(1.0E+100))
assertNotNull(*variable*)	assertNotNull

assertThat(*expected*, *Matcher*)	assertThat(currency, is("Baht"))
	assertThat(money.toString(), contains("Coin"))
	Test results satisfies some condition, specified by a Matcher. See reference for examples.

References

`http://junit.org` JUnit home. Has many examples and how-to.

"*Matchers and assertThat", JUnit Wiki.*

https://github.com/junit-team/junit4/wiki/matchers-and-assertthat

"*Benefit of* assertThat *over other* Assert *Methods",*

https://objectpartners.com/2013/09/18/the-benefits-of-using-assertthat-over-other-assert-methods-in-unit-tests/

Download JUnit from **`junit.org`** to get the JUnit API Javadoc and code samples.