JUnit 4.0

CODIGN SOFTWARE, LLC

What's New

JUnit 4 is a new Java unit testing framework developed by Kent Beck, ErichGamma and David Saff The framework provides an enhanced work-flow that takes advantage of manyJDK 1.5 features.

Annotations

JUnit 4.0 makes extensive use of JDK 1.5 style annotations and is backwards compatible with older versions of JUnit.

@Test replaces 'testMethod'

Previous versions of JUnit require methods under test to be prefixed with the word 'test' in addition to extending the junit.framework.TestCase class.

The latest version of JUnit uses the @Test annotation and extends org.junit.Tests, not the TestCase class.

```
import org.junit.Test

public class CarRentalTest{
   @Test public void GetRental() {
      CarRental c = new CarRental();
        c.getRental(33,5);
        assertEquals(Rental price",10,c.price);}
      }
}
```

*The advantages of this new approach include fostering a naming convention that uses the JDK 1.5 annotation feature, also known as metadata.

*Annotations provide a mechanism to parse and distinguish various components of unit tests.

@Before and @After replaces setUp() and tearDown()

The latest version of JUnit uses the @Before and @After annotations to effectively replace the use of setUp() and tearDown().

A distinct advantage is that developers can now use as many @Before and @After methods as needed.

```
public class CarRentalTest{
HelperClass helper = null;
@Before public void createHelper {
        helper = new HelperClass();
   }
@After void destroyHelper {
        helper.cleanup();
   }
{
```

@BeforeClass and @AfterClass

JUnit 4 introduces @BeforeClass and @AfterClass. Methods annotated with these keywords will execute once before and after the test class. There is a performance advantage in that tests spend less time setting up and tearing down data structures, objects, components, and other expensive operations..

```
public class CarRentalTest{
   DBHandle dbHandle = null;
   @BeforeClass plublic void establishDBHandle() {
          dbHandle = new DBHandle();
          dbHandle.initialize();
   }
   @AfterClass public void cleanupDBHandle() {
          dbHandle.cleanup();
          dbHandle = null;
   }
}
```

@Test(exception)

JUnit 4 lets developers declare an expected thrown exception using the @Test annotation. If an exception is declared but not thrown then the test will fail. Use of the try-catch block is still required to test other aspects of the exception.

```
public class CarRentalTest{
   @Test (expected = NumberFormatException)
       public void illegalNumber() {
          Integer i = new Integer ("xyz");
   }
```

@Ignore

JUnit 4 adds a new annotation called @Ignore, which tells the Runner to ignore the test. The Runner will display output indicating a test was ignored.

```
public class CarRentalTest{
   Ignore public void testForFutureFeature(){
    }
}
```

@Test(timeout=400)

JUnit 4 adds a timeout feature that allows the developer to identify the number of milliseconds a testcase has to execute. If a testcase has not completed before the specified number a milliseconds, the test will fail. This is useful because the runner can drop potentially long-running tests that could hold up the entire test suite.

```
// test is aborted if query takes longer than 1 sec.
Test (timeout = 1000) public void queryDatabase () {
      dbHandle.executeQuery();
}
```

Runners

A JUnit 4 runner for Eclipes will be available in Eclipse 3.2. Until then, developers are encouraged to use the compatibility layer, as described in the JUnit 4.0 FAQ

```
public static junit.framework.Test suite(){
    return new JUnit4Adapter(SimpleTest.class);
```

Developers using other IDEs are encouraged to use the compatibility layer as well.

It is possible to run JUnit tests via the command line as well using:

```
public static void main(String args[]) {
   org.junit.runner.JUnitCore.main("cs.SimpleTest");
```

Assertions

Below is a list of assertions used for JUnit 3.8 and JUnit 4. Two new assertions are introduced with JUnit 4 and are in italics. All assertions are static void.

assertTrue
(boolean) Reports an error if boolean is false
(String, boolean) Adds error String to output
assertFalse
(bolean) Reports an error if boolean is true
(String, boolean) Adds error String to output
assertNull
(Object) Reports an error if object is not null
(String, Object) Adds error String to output
assertNotNull
(Object) Reports an error if object is null
(String, Object) Adds error String to output
assertSame
(Object, Object) Reports error if two objects are not identical
(String, Object, Object) Adds error String to output
assertNotSame
(Object, Object) Reports error if two objects are identical
(Styring, Object, Object) Adds error String to output
assertEquals
(Object, Object) Reports error if two objects are not equal
(String, Object, Object) Adds error String to output
(String, String) Reports delta between two strings if the two strings are not equal
(String, String, String) Adds error String to output
(boolean, boolean) Reports error if the two booleans are not equal
(String, boolean, boolean) Adds error String to output
(byte,byte) Reports error if the two bytes are not equal
(String, byte, byte) Adds error String to output
(char, char)Reports error if two chars are not equal
(String, char, char) Adds error String to output
(short, short) Reports error if two shorts are not equal
(String, short, short) Adds error String to output
(int, int) Reports error if two ints are not equal
(String, int, int) Adds error String to output
(long, long) Reports error if two longs are not equal
(String, long, long) Adds error String to output
(float, float, float) Reports error if the first two floats are not within range specified by third float
(String, float, float, float) Adds error String to output
(double, double, double) Reports error if the first two doubles are not within range specified by third ouble
(String, double, double, double) Adds error String to output
(object[], object[]); Reports error if either the legnth or element of each array are not equal. *New to JUnit 4*
(String, object[], object[]) Adds error String to output. *New to JUnit 4.*

Resources

junit.org
testdriven.com
soureforge.net/projects/junit