About Jd2chm...

The project files of this HTML Help
Were generated with Jd2chm version 0.3
Copyright © 2000-2003 Andre Burgaud
http://www.burgaud.com
### Packages

- `junit.extensions`
- `junit.framework`
Hierarchy For All Packages

Package Hierarchies:
  junit.extensions, junit.framework
Class Hierarchy

- class java.lang.Object
  - class junit.framework.Assert
    - class junit.framework.TestCase (implements junit.framework.Test)
      - class junit.extensions.ExceptionTestCase
    - class junit.extensions.TestDecorator (implements junit.framework.Test)
      - class junit.extensions.RepeatedTest
      - class junit.extensions.TestSetup
  - class junit.framework.TestFailure
  - class junit.framework.TestResult
  - class junit.framework.TestSuite (implements junit.framework.Test)
    - class junit.extensions.ActiveTestSuite
  - class java.lang.Throwable (implements java.io.Serializable)
    - class java.lang.Error
      - class junit.framework.AssertionFailedError
        - class junit.framework.ComparisonFailure
## Interface Hierarchy

- interface junit.framework.**Protectable**
- interface junit.framework.**Test**
- interface junit.framework.**TestListener**

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Class</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV</td>
<td>NEXT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
</table>

PREV NEXT
## Package junit.extensions

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ActiveTestSuite</code></td>
<td>A TestSuite for active Tests.</td>
</tr>
<tr>
<td><code>ExceptionTestCase</code></td>
<td>A TestCase that expects an Exception of class <code>fExpected</code> to be thrown.</td>
</tr>
<tr>
<td><code>RepeatedTest</code></td>
<td>A Decorator that runs a test repeatedly.</td>
</tr>
<tr>
<td><code>TestDecorator</code></td>
<td>A Decorator for Tests.</td>
</tr>
<tr>
<td><code>TestSetup</code></td>
<td>A Decorator to set up and tear down additional fixture state.</td>
</tr>
</tbody>
</table>
Hierarchy For Package junit.extensions

Package Hierarchies:

All Packages
Class Hierarchy

- class java.lang.Object
  - class junit.framework.Assert
    - class junit.framework.TestCase (implements junit.framework.Test)
      - class junit.extensions.ExceptionTestCase
    - class junit.extensions.TestDecorator (implements junit.framework.Test)
      - class junit.extensions.RepeatedTest
      - class junit.extensions.TestSetup
  - class junit.framework.TestSuite (implements junit.framework.Test)
    - class junit.extensions.ActiveTestSuite
JUnit's `ActiveTestSuite` class is a TestSuite designed for active Tests. It runs each test in a separate thread and waits until all threads have terminated. This class extends `TestSuite` and implements the `Test` interface.

### Constructor Summary

**ActiveTestSuite()**

**ActiveTestSuite**(java.lang.Class theClass)

**ActiveTestSuite**(java.lang.Class theClass, java.lang.String name)

**ActiveTestSuite**(java.lang.String name)

### Method Summary

**void run(TestResult result)**

Runs the tests and collects their result in a TestResult.

**void runFinished(Test test)**
```java
void runTest(Test test, TestResult result)
```

<table>
<thead>
<tr>
<th>Methods inherited from class junit.framework.TestSuite</th>
</tr>
</thead>
<tbody>
<tr>
<td>addTest, addTestSuite, countTestCases, createTest, getName, getTestConstructor, setName, testAt, testCount, tests, toString</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

## Constructor Detail

### ActiveTestSuite

```java
public ActiveTestSuite()
```

### ActiveTestSuite

```java
public ActiveTestSuite(java.lang.Class theClass)
```

### ActiveTestSuite

```java
public ActiveTestSuite(java.lang.String name)
```

### ActiveTestSuite

```java
public ActiveTestSuite(java.lang.Class theClass, java.lang.String name)
```

## Method Detail
run

public void run(TestResult result)

Description copied from class: TestSuite
Runs the tests and collects their result in a TestResult.

Specified by: run in interface Test
Overrides: run in class TestSuite

runTest

public void runTest( test, TestResult result)

Overrides: runTest in class TestSuite

runFinished

public void runFinished( test)

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS  SUMMARY: NESTED | FIELD | CONSTR | METHOD  FRAMES  NO FRAMES  DETAIL: FIELD | CONSTR | METHOD
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMEs</td>
<td>NO FRAMEs</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
public class ExceptionTestCase extends TestCase

A TestCase that expects an Exception of class fExpected to be thrown. The other way to check that an expected exception is thrown is:

```java
try {
    shouldThrow();
} catch (SpecialException e) {
    return;
} fail("Expected SpecialException");
```

To use ExceptionTestCase, create a TestCase like:

```java
new ExceptionTestCase("testShouldThrow", SpecialException.class);
```

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExceptionTestCase(java.lang.String name, java.lang.Class exception)</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected void runTest()</td>
<td>Execute the test method expecting that an Exception of class fExpected or one of its subclasses will be thrown</td>
</tr>
</tbody>
</table>

### Methods inherited from class junit.framework.TestCase
- countTestCases
- createResult
- getName
- run
- run
- runBare
- setName
- setUp
- tearDown
- toString

### Methods inherited from class junit.framework.Assert
- assertEquals
- assertEquals
- assertEquals
- assertEquals
- assertEquals
- assertEquals
- assertEquals
- assertEquals
- assertTrue
- assertTrue
- assertTrue
- fail
- fail

### Methods inherited from class java.lang.Object
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

#### ExceptionTestCase

public ExceptionTestCase(java.lang.String name, java.lang.Class exception)

### Method Detail

#### runTest

protected void runTest() throws java.lang.Throwable
Execute the test method expecting that an Exception of class fExpected or one of its subclasses will be thrown

Overrides:

runTest in class TestCase

Throws:

java.lang.Throwable - if any exception is thrown
junit.extensions Class RepeatedTest

java.lang.Object
|   |-- junit.framework.Assert
|     |   |-- junit.extensions.TestDecorator
|     |   |   |-- junit.extensions.RepeatedTest

All Implemented Interfaces:
   Test

public class RepeatedTest extends TestDecorator

A Decorator that runs a test repeatedly.

Field Summary

| Fields inherited from class junit.extensions.TestDecorator |
| fTest |

Constructor Summary

RepeatedTest(Test test, int repeat)

Method Summary

| int | countTestCases() |
|     | Counts the number of test cases that will be run by this test. |

| void | run(TestResult result) |
|      | Runs a test and collects its result in a TestResult |
Methods inherited from class junit.extensions.TestDecorator
basicRun, getTest

Methods inherited from class junit.framework.Assert
assertEquals

Methods inherited from class java.lang.Object
clon, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

RepeatedTest

public RepeatedTest(Test test,
                        int repeat)

Method Detail

countTestCases

public int countTestCases()

Description copied from interface: Test
Counts the number of test cases that will be run by this test.
run

```java
public void run(TestResult result)
```

**Description copied from interface: Test**
Runs a test and collects its result in a TestResult instance.

**Specified by:**
run in interface Test

**Overrides:**
run in class TestDecorator

toString

```java
public java.lang.String toString()
```

**Overrides:**
toString in class TestDecorator
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>

| SUMMARY: NESTED | FIELD | CONSTR | METHOD |
| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |
junit.extensions **Class TestDecorator**

java.lang.Object

    +--- junit.framework.Assert
        +--- junit.extensions.TestDecorator

All Implemented Interfaces:
    Test

Direct Known Subclasses:
    RepeatedTest, TestSetup

---

public class **TestDecorator**
extends Assert
implements Test

A Decorator for Tests. Use TestDecorator as the base class for defining new test decorators. Test decorator subclasses can be introduced to add behaviour before or after a test is run.

---

**Field Summary**

| protected Test fTest |

**Constructor Summary**

| TestDecorator(Test test) |

**Method Summary**

| void basicRun(TestResult result) |

The basic run behaviour.
```java
int countTestCases()

Counts the number of test cases that will be run by this test.

String getTest()

void run(TestResult result)

Runs a test and collects its result in a TestResult instance.

java.lang.String toString()

Methods inherited from class junit.framework.Assert
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
equalities, equals, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,
equalities, equals, equals, equals, equals, equals,

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

### Field Detail

**fTest**

protected `Test` fTest

### Constructor Detail

**TestDecorator**

public `TestDecorator`(`Test` test)
Method Detail

**basicRun**

public void basicRun(TestResult result)

   The basic run behaviour.

---

**countTestCases**

public int countTestCases()

   Description copied from interface: Test
   Counts the number of test cases that will be run by this test.

   Specified by:
       countTestCases in interface Test

---

**run**

public void run(TestResult result)

   Description copied from interface: Test
   Runs a test and collects its result in a TestResult instance.

   Specified by:
       run in interface Test

---

**toString**

public java.lang.String toString()

   Overrides:
       toString in class java.lang.Object
getTest

public Test getTest()
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
public class TestSetup
extends TestDecorator

A Decorator to set up and tear down additional fixture state. Subclass TestSetup and insert it into your tests when you want to set up additional state once before the tests are run.

Field Summary

Fields inherited from class junit.extensions.TestDecorator
fTest

Constructor Summary

TestSetup(Test test)

Method Summary

void run(TestResult result)
   Runs a test and collects its result in a TestResult instance.
protected void setUp()
Sets up the fixture.

protected void tearDown()
Tears down the fixture.

Methods inherited from class junit.extensions.TestDecorator
basicRun, countTestCases, getTest, toString

Methods inherited from class junit.framework.Assert
assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,
equalToFalse, assertEquals, assertEquals, assertEquals,

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait

Constructor Detail

TestSetup

public TestSetup(Test test)

Method Detail

run

public void run(TestResult result)

Description copied from interface: Test
Runs a test and collects its result in a TestResult instance.
Specified by:
run in interface Test

Overrides:
run in class TestDecorator

---

**setUp**

```java
protected void setUp()
    throws java.lang.Exception
```

Sets up the fixture. Override to set up additional fixture state.

```
java.lang.Exception
```

---

**tearDown**

```java
protected void tearDown()
    throws java.lang.Exception
```

Tears down the fixture. Override to tear down the additional fixture state.

```
java.lang.Exception
```
# Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Protectable</code></td>
<td>A <code>Protectable</code> can be run and can throw a Throwable.</td>
</tr>
<tr>
<td><code>Test</code></td>
<td>A <code>Test</code> can be run and collect its results.</td>
</tr>
<tr>
<td><code>TestListener</code></td>
<td>A Listener for test progress</td>
</tr>
</tbody>
</table>

# Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Assert</code></td>
<td>A set of assert methods.</td>
</tr>
<tr>
<td><code>TestCase</code></td>
<td>A test case defines the fixture to run multiple tests.</td>
</tr>
<tr>
<td><code>TestFailure</code></td>
<td>A <code>TestFailure</code> collects a failed test together with the caught exception.</td>
</tr>
<tr>
<td><code>TestResult</code></td>
<td>A <code>TestResult</code> collects the results of executing a test case.</td>
</tr>
<tr>
<td><code>TestSuite</code></td>
<td>A <code>TestSuite</code> is a Composite of Tests.</td>
</tr>
</tbody>
</table>

# Error Summary

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AssertionFailedError</code></td>
<td>Thrown when an assertion failed.</td>
</tr>
<tr>
<td><code>ComparisonFailure</code></td>
<td>Thrown when an assert equals for Strings failed.</td>
</tr>
</tbody>
</table>
Hierarchy For Package junit.framework

Package Hierarchies:

All Packages
Class Hierarchy

- class java.lang.Object
  - class junit.framework.Assert
    - class junit.framework.TestCase (implements junit.framework.Test)
  - class junit.framework.TestFailure
  - class junit.framework.TestResult
  - class junit.framework.TestSuite (implements junit.framework.Test)
- class java.lang Throwable (implements java.io.Serializable)
  - class java.lang.Error
    - class junit.framework.AssertionFailedError
    - class junit.framework.ComparisonFailure
Interface Hierarchy

- interface junit.framework.Protectable
- interface junit.framework.Test
- interface junit.framework.TestListener
public interface `Protectable`

A `Protectable` can be run and can throw a Throwable.

See Also: `TestResult`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>protect()</code></td>
<td><code>public void protect()</code></td>
</tr>
</tbody>
</table>

Run the the following method protected.

### Method Detail

**protect**

```java
public void protect()
    throws java.lang.Throwable
```

Run the the following method protected.

`java.lang.Throwable`
public interface Test

A Test can be run and collect its results.

See Also: TestResult
junit.framework **Interface TestListener**

public interface **TestListener**

A Listener for test progress

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addError</td>
<td>public void addError(Test test, java.lang.Throwable t)</td>
<td>An error occurred.</td>
</tr>
<tr>
<td>addFailure</td>
<td>public void addFailure(Test test, AssertionFailedError t)</td>
<td>A failure occurred.</td>
</tr>
<tr>
<td>endTest</td>
<td>public void endTest(Test test)</td>
<td>A test ended.</td>
</tr>
<tr>
<td>startTest</td>
<td>public void startTest(Test test)</td>
<td>A test started.</td>
</tr>
</tbody>
</table>

### Method Detail

**addError**

public void addError(Test test, java.lang.Throwable t)

An error occurred.

**addFailure**

public void addFailure(Test test, AssertionFailedError t)

A failure occurred.
endTest

public void endTest(Test test)

   A test ended.

startTest

public void startTest(Test test)

   A test started.
**junit.framework** **Class Assert**

java.lang.Object

|-- junit.framework.Assert

**Direct Known Subclasses:**
  - TestCase, TestDecorator

---

public class Assert
extends java.lang.Object

A set of assert methods. Messages are only displayed when an assert fails.

---

### Constructor Summary

<table>
<thead>
<tr>
<th>protected Assert()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protect constructor since it is a static only class</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>static void assertEquals(boolean expected, boolean actual)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asserts that two booleans are equal.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static void assertEquals(byte expected, byte actual)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asserts that two bytes are equal.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static void assertEquals(char expected, char actual)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asserts that two chars are equal.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static void assertEquals(double expected, double actual, double delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asserts that two doubles are equal concerning a delta.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static void assertEquals(float expected, float actual, float delta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asserts that two floats are equal concerning a delta.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static void assertEquals(int expected, int actual)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asserts that two ints are equal.</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
<tr>
<td>static void</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

Constructor Detail

Assert

protected Assert()

Protect constructor since it is a static only class

Method Detail

assertTrue

public static void assertTrue(java.lang.String message,
       boolean condition)

Asserts that a condition is true. If it isn't it throws an
AssertionFailedError with the given message.

assertTrue

public static void assertTrue(boolean condition)

Asserts that a condition is true. If it isn't it throws an
AssertionFailedError.

assertFalse

public static void assertFalse(java.lang.String message,
       boolean condition)
Asserts that a condition is false. If it isn’t it throws an AssertionFailedError with the given message.

**assertFalse**

```java
public static void assertFalse(boolean condition)
```

Asserts that a condition is false. If it isn’t it throws an AssertionFailedError.

**fail**

```java
public static void fail(java.lang.String message)
```

Fails a test with the given message.

**fail**

```java
public static void fail()
```

Fails a test with no message.

**assertEquals**

```java
```

Asserts that two objects are equal. If they are not an AssertionFailedError is thrown with the given message.

**assertEquals**

```java
public static void assertEquals(java.lang.Object expected, java.lang.Object actual)
```
Asserts that two objects are equal. If they are not an AssertionFailedError is thrown.

assertEquals

public static void assertEquals(java.lang.String message,
                                        java.lang.String expected,
                                        java.lang.String actual)

Asserts that two Strings are equal.

assertEquals

public static void assertEquals(java.lang.String expected,
                                        java.lang.String actual)

Asserts that two Strings are equal.

assertEquals

public static void assertEquals(java.lang.String message,
                                        double expected,
                                        double actual,
                                        double delta)

Asserts that two doubles are equal concerning a delta. If they are not an AssertionFailedError is thrown with the given message. If the expected value is infinity then the delta value is ignored.

assertEquals

public static void assertEquals(double expected,
                                        double actual,
                                        double delta)

Asserts that two doubles are equal concerning a delta. If the expected value is infinity then the delta value is ignored.
**assertEquals**

```java
public static void assertEquals(java.lang.String message,
                               float expected,
                               float actual,
                               float delta)
```

Asserts that two floats are equal concerning a delta. If they are not an AssertionFailedError is thrown with the given message. If the expected value is infinity then the delta value is ignored.

---

**assertEquals**

```java
public static void assertEquals(float expected,
                                float actual,
                                float delta)
```

Asserts that two floats are equal concerning a delta. If the expected value is infinity then the delta value is ignored.

---

**assertEquals**

```java
public static void assertEquals(java.lang.String message,
                                long expected,
                                long actual)
```

Asserts that two longs are equal. If they are not an AssertionFailedError is thrown with the given message.

---

**assertEquals**

```java
public static void assertEquals(long expected,
                                long actual)
```

Asserts that two longs are equal.
assertEquals

public static void assertEquals(java.lang.String message, boolean expected, boolean actual)

Asserts that two booleans are equal. If they are not an
AssertionFailedError is thrown with the given message.

assertEquals

public static void assertEquals(boolean expected, boolean actual)

Asserts that two booleans are equal.

assertEquals

public static void assertEquals(java.lang.String message, byte expected, byte actual)

Asserts that two bytes are equal. If they are not an
AssertionFailedError is thrown with the given message.

assertEquals

public static void assertEquals(byte expected, byte actual)

Asserts that two bytes are equal.

assertEquals

public static void assertEquals(java.lang.String message, char expected, char actual)
Asserts that two chars are equal. If they are not an AssertionFailedError is thrown with the given message.

assertEquals

```java
public static void assertEquals(char expected, char actual)

Asserts that two chars are equal.
```

assertEquals

```java
public static void assertEquals(java.lang.String message, short expected, short actual)

Asserts that two shorts are equal. If they are not an AssertionFailedError is thrown with the given message.
```

assertEquals

```java
public static void assertEquals(short expected, short actual)

Asserts that two shorts are equal.
```

assertEquals

```java
public static void assertEquals(java.lang.String message, int expected, int actual)

Asserts that two ints are equal. If they are not an AssertionFailedError is thrown with the given message.
```
**assertEquals**

```java
public static void assertEquals(int expected, int actual)
```

Asserts that two ints are equal.

**assertNotNull**

```java
public static void assertNotNull(java.lang.Object object)
```

Asserts that an object isn't null.

**assertNotNull**

```java
public static void assertNotNull(java.lang.String message, java.lang.Object object)
```

Asserts that an object isn't null. If it is an AssertionError is thrown with the given message.

**assertNull**

```java
public static void assertNull(java.lang.Object object)
```

Asserts that an object is null.

**assertNull**

```java
public static void assertNull(java.lang.String message, java.lang.Object object)
```

Asserts that an object is null. If it is not an AssertionError is thrown with the given message.
assertSame

public static void assertSame(java.lang.String message,
               java.lang.Object expected,
               java.lang.Object actual)

Asserts that two objects refer to the same object. If they are not an AssertionFailedError is thrown with the given message.

assertSame

public static void assertSame(java.lang.Object expected,
               java.lang.Object actual)

Asserts that two objects refer to the same object. If they are not the same an AssertionFailedError is thrown.

assertNotSame

public static void assertNotSame(java.lang.String message,
               java.lang.Object expected,
               java.lang.Object actual)

Asserts that two objects refer to the same object. If they are not an AssertionFailedError is thrown with the given message.

assertNotSame

public static void assertNotSame(java.lang.Object expected,
               java.lang.Object actual)

Asserts that two objects refer to the same object. If they are not the same an AssertionFailedError is thrown.
JUnit Framework Class TestCase

java.lang.Object
    `junit.framework.Assert`
    `junit.framework.TestCase`

All Implemented Interfaces:
    `Test`

Direct Known Subclasses:
    `ExceptionTestCase`

---

class Test

defined fixture to run multiple tests. To define a test case:
1) implement a subclass of TestCase
2) define instance variables that store the state of the fixture
3) initialize the fixture state by overriding `setUp`
4) clean-up after a test by overriding `tearDown`

Each test runs in its own fixture so there can be no side effects among test runs. Here is an example:

```java
public class MathTest extends TestCase {
    protected double fValue1;
    protected double fValue2;

    protected void setUp() {
        fValue1 = 2.0;
        fValue2 = 3.0;
    }
}
```

For each test implement a method which interacts with the fixture. Verify the expected results with assertions specified by calling `assertTrue` with a boolean.
public void testAdd() {
    double result = fValue1 + fValue2;
    assertTrue(result == 5.0);
}

Once the methods are defined you can run them. The framework supports both a static type safe and more dynamic way to run a test. In the static way you override the runTest method and define the method to be invoked. A convenient way to do so is with an anonymous inner class.

    TestCase test = new MathTest("add") {
        public void runTest() {
            testAdd();
        }
    };
    test.run();

The dynamic way uses reflection to implement runTest. It dynamically finds and invokes a method. In this case the name of the test case has to correspond to the test method to be run.

    TestCase = new MathTest("testAdd");
    test.run();

The tests to be run can be collected into a TestSuite. JUnit provides different test runners which can run a test suite and collect the results. A test runner either expects a static method suite as the entry point to get a test to run or it will extract the suite automatically.

    public static Test suite() {
        suite.addTest(new MathTest("testAdd"));
        suite.addTest(new MathTest("testDivideByZero"));
        return suite;
    }

See Also:
    TestResult, TestSuite

---

**Constructor Summary**
**TestCase()**
No-arg constructor to enable serialization.

**TestCase(java.lang.String name)**
Constructs a test case with the given name.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td><strong>countTestCases()</strong> Counts the number of test cases executed by run(TestResult result).</td>
</tr>
<tr>
<td>protected TestResult</td>
<td><strong>createResult()</strong> Creates a default TestResult object</td>
</tr>
<tr>
<td>java.lang.String</td>
<td><strong>getName()</strong> Gets the name of a TestCase</td>
</tr>
<tr>
<td>TestResult</td>
<td><strong>run()</strong> A convenience method to run this test, collecting the results with a default TestResult object.</td>
</tr>
<tr>
<td>void</td>
<td><strong>run(TestResult result)</strong> Runs the test case and collects the results in TestResult.</td>
</tr>
<tr>
<td>void</td>
<td><strong>runBare()</strong> Runs the bare test sequence.</td>
</tr>
<tr>
<td>protected void</td>
<td><strong>runTest()</strong> Override to run the test and assert its state.</td>
</tr>
<tr>
<td>void</td>
<td><strong>setName(java.lang.String name)</strong> Sets the name of a TestCase</td>
</tr>
<tr>
<td>protected void</td>
<td><strong>setUp()</strong> Sets up the fixture, for example, open a network connection.</td>
</tr>
<tr>
<td>protected void</td>
<td><strong>tearDown()</strong> Tears down the fixture, for example, close a network connection.</td>
</tr>
<tr>
<td>java.lang.String</td>
<td><strong>toString()</strong> Returns a string representation of the test case</td>
</tr>
</tbody>
</table>
Methods inherited from class.junit.framework.Assert

assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,
assertFalse, assertFalse, assertNull, assertNull, assertEquals,
assertSame, assertEquals, assertEquals, assertEquals, assertEquals,"}

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait

Constructor Detail

TestCase

public TestCase()

No-arg constructor to enable serialization. This method is not intended to be used by mere mortals without calling setName().

TestCase

public TestCase(java.lang.String name)

Constructs a test case with the given name.

Method Detail

countTestCases

public int countTestCases()

Counts the number of test cases executed by run(TestResult result).
createResult

protected TestResult createResult()

Creates a default TestResult object

See Also:
TestResult

run

public TestResult run()

A convenience method to run this test, collecting the results with a default TestResult object.

See Also:
TestResult

run

public void run(TestResult result)

Runs the test case and collects the results in TestResult.

Specified by:
run in interface Test

runBare

public void runBare()

throws java.lang.Throwable
Runs the bare test sequence.

**Throws:**
java.lang.Throwable - if any exception is thrown

---

**runTest**

protected void runTest()  
throws java.lang.Throwable

Override to run the test and assert its state.

**Throws:**
java.lang.Throwable - if any exception is thrown

---

**setUp**

protected void setUp()  
throws java.lang.Exception

Sets up the fixture, for example, open a network connection. This method is called before a test is executed.

java.lang.Exception

---

**tearDown**

protected void tearDown()  
throws java.lang.Exception

Tears down the fixture, for example, close a network connection. This method is called after a test is executed.

java.lang.Exception

---

**toString**
public java.lang.String **toString**()

Returns a string representation of the test case

**Overrides:**

toString in class java.lang.Object

---

**getName**

public java.lang.String **getName**()

Gets the name of a TestCase

**Returns:**

returns a String

---

**setName**

public void **setName**(java.lang.String name)

Sets the name of a TestCase

**Parameters:**

name - The name to set
junit.framework  **Class TestFailure**

java.lang.Object  

| +-- junit.framework.TestFailure

public class **TestFailure**  
extends java.lang.Object

A **TestFailure** collects a failed test together with the caught exception.

See Also:  
[TestResult](#)

---

### Field Summary

<table>
<thead>
<tr>
<th>Protected</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>fFailedTest</strong></td>
<td>Test</td>
</tr>
<tr>
<td><strong>fThrownException</strong></td>
<td>java.lang.Throwable</td>
</tr>
</tbody>
</table>

---

### Constructor Summary

**TestFailure**(Test **failedTest**, java.lang Throwable **thrownException**)  
Constructs a TestFailure with the given test and exception.

---

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.String</td>
<td>exceptionMessage()</td>
</tr>
</tbody>
</table>
| Test           | **failedTest()**  
| boolean        | **isFailure()**  
|                | **thrownException()**  |
java.lang.Throwable

gets the thrown exception.

java.lang.String

toString()

returns a short description of the failure.

java.lang.String

trace()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

fFailedTest

protected Test fFailedTest

fThrownException

protected java.lang.Throwable fThrownException

Constructor Detail

TestFailure

public TestFailure(Test failedTest,

java.lang.Throwable thrownException)

constructs a TestFailure with the given test and exception.

Method Detail

failedTest
public Test failedTest()

    Gets the failed test.

thrownException

public java.lang.Throwable thrownException()

    Gets the thrown exception.

toString

public java.lang.String toString()

    Returns a short description of the failure.

    Overrides:
        toString in class java.lang.Object

trace

public java.lang.String trace()

exceptionMessage

public java.lang.String exceptionMessage()

isFailure

public boolean isFailure()
public class TestResult
extends java.lang.Object

A TestResult collects the results of executing a test case. It is an instance of the Collecting Parameter pattern. The test framework distinguishes between failures and errors. A failure is anticipated and checked for with assertions. Errors are unanticipated problems like an ArrayIndexOutOfBoundsException.

See Also:
Test

Field Summary

<table>
<thead>
<tr>
<th>protected</th>
<th>fErrors</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.util.Vector</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>fFailures</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.util.Vector</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>fListeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.util.Vector</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>fRunTests</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addError(Test test, java.lang.Throwable t)</td>
<td>Adds an error to the list of errors.</td>
</tr>
<tr>
<td>void addFailure(Test test, AssertionError t)</td>
<td>Adds a failure to the list of failures.</td>
</tr>
<tr>
<td>void addListener(TestListener listener)</td>
<td>Registers a TestListener</td>
</tr>
<tr>
<td>void endTest(Test test)</td>
<td>Informs the result that a test was completed.</td>
</tr>
<tr>
<td>int errorCount()</td>
<td>Gets the number of detected errors.</td>
</tr>
<tr>
<td>java.util.Enumeration errors()</td>
<td>Returns an Enumeration for the errors</td>
</tr>
<tr>
<td>int failureCount()</td>
<td>Gets the number of detected failures.</td>
</tr>
<tr>
<td>java.util.Enumeration failures()</td>
<td>Returns an Enumeration for the failures</td>
</tr>
<tr>
<td>void removeListener(TestListener listener)</td>
<td>Unregisters a TestListener</td>
</tr>
<tr>
<td>protected void run(TestCase test)</td>
<td>Runs a TestCase.</td>
</tr>
<tr>
<td>int runCount()</td>
<td>Gets the number of run tests.</td>
</tr>
<tr>
<td>void runProtected(Test test, Protectable p)</td>
<td>Runs a TestCase.</td>
</tr>
<tr>
<td>boolean shouldStop()</td>
<td>Checks whether the test run should stop</td>
</tr>
<tr>
<td>void startTest(Test test)</td>
<td>Informs the result that a test will be started.</td>
</tr>
<tr>
<td>void stop()</td>
<td>Marks that the test run should stop.</td>
</tr>
<tr>
<td>boolean wasSuccessful()</td>
<td>Returns whether the entire test was successful or not.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

fFailures
protected java.util.Vector fFailures

fErrors
protected java.util.Vector fErrors

fListeners
protected java.util.Vector fListeners

fRunTests
protected int fRunTests

Constructor Detail

TestResult
public TestResult()
public void addError(Test test, java.lang.Throwable t)

    Adds an error to the list of errors. The passed in exception caused the error.

---

**addFailure**

public void addFailure(Test test, AssertionError t)

    Adds a failure to the list of failures. The passed in exception caused the failure.

---

**addListener**

public void addListener(TestListener listener)

    Registers a TestListener

---

**removeListener**

public void removeListener(TestListener listener)

    Unregisters a TestListener

---

**endTest**

public void endTest(Test test)

    Informs the result that a test was completed.

---

**errorCount**

public int errorCount()
Gets the number of detected errors.

```java
errors
public java.util.Enumeration errors()

    Returns an Enumeration for the errors
```

failureCount

```java
failureCount
public int failureCount()

    Gets the number of detected failures.
```

failures

```java
failures
public java.util.Enumeration failures()

    Returns an Enumeration for the failures
```

run

```java
run
protected void run(TestCase test)

    Runs a TestCase.
```

runCount

```java
runCount
public int runCount()

    Gets the number of run tests.
```

runProtected
public void runProtected(Test test, Protectable p)

Runs a TestCase.

shouldStop
public boolean shouldStop()

Checks whether the test run should stop

startTest
public void startTest(Test test)

Informs the result that a test will be started.

stop
public void stop()

Marks that the test run should stop.

wasSuccessful
public boolean wasSuccessful()

Returns whether the entire test was successful or not.
public class TestSuite
extends java.lang.Object
implements Test

A TestSuite is a Composite of Tests. It runs a collection of test cases. Here is an example using the dynamic test definition.

TestSuite suite = new TestSuite();
suite.addTest(new MathTest("testAdd"));
suite.addTest(new MathTest("testDivideByZero"));

Alternatively, a TestSuite can extract the tests to be run automatically. To do so you pass the class of your TestCase class to the TestSuite constructor.

TestSuite suite = new TestSuite(MathTest.class);

This constructor creates a suite with all the methods starting with "test" that take no arguments.

See Also:
Test

Constructor Summary
| **TestSuite()** | Constructs an empty TestSuite. |
| **TestSuite(java.lang.Class theClass)** | Constructs a TestSuite from the given class. |
| **TestSuite(java.lang.Class theClass, java.lang.String name)** | Constructs a TestSuite from the given class with the given name. |
| **TestSuite(java.lang.String name)** | Constructs an empty TestSuite. |

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <strong>addTest(Test test)</strong></td>
<td>Adds a test to the suite.</td>
</tr>
<tr>
<td>void <strong>addTestSuite(java.lang.Class testClass)</strong></td>
<td>Adds the tests from the given class to the suite.</td>
</tr>
<tr>
<td>int <strong>countTestCases()</strong></td>
<td>Counts the number of test cases that will be run by this test.</td>
</tr>
<tr>
<td>static <strong>Test createTest(java.lang.Class theClass, java.lang.String name)</strong></td>
<td>...as the moon sets over the early morning Merlin, Oregon mountains, our intrepid adventurers type...</td>
</tr>
<tr>
<td>java.lang.String <strong>getName()</strong></td>
<td>Returns the name of the suite.</td>
</tr>
<tr>
<td>static java.lang.reflect.Constructor <strong>getTestConstructor(java.lang.Class theClass, java.lang.String name)</strong></td>
<td>Gets a constructor which takes a single String as its argument or a no arg constructor.</td>
</tr>
<tr>
<td>void <strong>run(TestResult result)</strong></td>
<td>Runs the tests and collects their result in a TestResult.</td>
</tr>
<tr>
<td>void <strong>runTest(Test test, TestResult result)</strong></td>
<td></td>
</tr>
<tr>
<td>void <strong>setName(java.lang.String name)</strong></td>
<td>Sets the name of the suite.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>testAt(int index)</td>
<td>Returns the test at the given index</td>
</tr>
<tr>
<td>testCount()</td>
<td>Returns the number of tests in this suite</td>
</tr>
<tr>
<td>tests()</td>
<td>Returns the tests as an enumeration</td>
</tr>
<tr>
<td>toString()</td>
<td></td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

**Constructor Detail**

**TestSuite**

```java
public TestSuite() {
    Constructs an empty TestSuite.
}
```

**TestSuite**

```java
public TestSuite(java.lang.Class theClass, java.lang.String name) {
    Constructs a TestSuite from the given class with the given name.
}
```

**See Also:**

- TestSuite(Class)
Constructs a TestSuite from the given class. Adds all the methods starting with "test" as test cases to the suite. Parts of this method was written at 2337 meters in the Hüffihütte, Kanton Uri

TestSuite

public TestSuite(java.lang.String name)

Constructs an empty TestSuite.

Method Detail

addTest

public void addTest(Test test)

Adds a test to the suite.

addTestSuite

public void addTestSuite(java.lang.Class testClass)

Adds the tests from the given class to the suite

createTest

public static Test createTest(java.lang.Class theClass,
                               java.lang.String name)

...as the moon sets over the early morning Merlin, Oregon mountains, our intrepid adventurers type...

countTestCases

public int countTestCases()
Counts the number of test cases that will be run by this test.

**Specified by:**

[countTestCases](#) in interface [Test](#)

---

**getTestConstructor**

```java
public static java.lang.reflect.Constructor getTestConstructor(java.lang.Class theClass)
throws java.lang.NoSuchMethodException
```

Gets a constructor which takes a single String as its argument or a no arg constructor.

java.lang.NoSuchMethodException

---

**run**

```java
public void run(TestResult result)
```

Runs the tests and collects their result in a TestResult.

**Specified by:**

[run](#) in interface [Test](#)

---

**runTest**

```java
public void runTest(Test test, TestResult result)
```

---

**testAt**

```java
public Test testAt(int index)
```

Returns the test at the given index
testCount

public int testCount()

    Returns the number of tests in this suite

---------------------

tests

public java.util.Enumeration tests()

    Returns the tests as an enumeration

---------------------

toString

public java.lang.String toString()

    Overrides: toString in class java.lang.Object

---------------------

setName

public void setName(java.lang.String name)

    Sets the name of the suite.

    Parameters:
        name - The name to set

---------------------

getName

public java.lang.String getName()

    Returns the name of the suite. Not all test suites have a name and this method can return null.

---------------------
junit.framework  Class AssertionFailedError

java.lang.Object
   |-- java.lang.Throwable
      |-- java.lang.Error
         |-- junit.framework.AssertionFailedError

All Implemented Interfaces:
   java.io.Serializable

Direct Known Subclasses:
   ComparisonFailure

public class AssertionFailedError
   extends java.lang.Error

Thrown when an assertion failed.

See Also:
   Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssertionFailedError()</td>
</tr>
<tr>
<td>AssertionFailedError(java.lang.String message)</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

AssertionFailedError

public AssertionFailedError()

AssertionFailedError

public AssertionFailedError(java.lang.String message)
junit.framework  **Class ComparisonFailure**

java.lang.Object
  |   +--java.lang.Throwable
  |       |   +--java.lang.Error
  |       |       |   +--junit.framework.AssertionFailedError
  |       |       |       |   +--junit.framework.ComparisonFailure

All Implemented Interfaces:
  java.io.Serializable

public class **ComparisonFailure**
extends **AssertionFailedError**

Thrown when an assert equals for Strings failed. Inspired by a patch from Alex Chaffee mailto:alex@purpletech.com

See Also:
  Serialized Form

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComparisonFailure</td>
<td>(java.lang.String message, java.lang.String expected, java.lang.String actual)</td>
</tr>
<tr>
<td>Constructs a comparison failure.</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMessage</td>
<td>Returns &quot;...&quot; in place of common prefix and &quot;...&quot; in place of common suffix between expected and actual.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Throwable
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

ComparisonFailure

public ComparisonFailure(java.lang.String message,
                           java.lang.String expected,
                           java.lang.String actual)

Constructs a comparison failure.

Parameters:
    message - the identifying message or null
    expected - the expected string value
    actual - the actual string value

Method Detail

getMessage

public java.lang.String getMessage()

Returns "..." in place of common prefix and "..." in place of common suffix between expected and actual.

Overrides:
    getMessage in class java.lang.Throwable

See Also:
    Throwable.getMessage()
ActiveTestSuite - class junit.extensions.ActiveTestSuite
   A TestSuite for active Tests.
ActiveTestSuite() - Constructor for class junit.extensions.ActiveTestSuite
ActiveTestSuite(Class) - Constructor for class junit.extensions.ActiveTestSuite
ActiveTestSuite(Class, String) - Constructor for class junit.extensions.ActiveTestSuite
ActiveTestSuite(String) - Constructor for class junit.extensions.ActiveTestSuite

addError(Test, Throwable) - Method in class junit.framework.TestResult
   Adds an error to the list of errors.
addError(Test, Throwable) - Method in interface junit.framework.TestListener
   An error occurred.
addFailure(Test, AssertionFailedError) - Method in class junit.framework.TestResult
   Adds a failure to the list of failures.
addFailure(Test, AssertionFailedError) - Method in interface junit.framework.TestListener
   A failure occurred.
addListener(TestListener) - Method in class junit.framework.TestResult
   Registers a TestListener
addTest(Test) - Method in class junit.framework.TestSuite
   Adds a test to the suite.
addTestSuite(Class) - Method in class junit.framework.TestSuite
   Adds the tests from the given class to the suite
Assert - class junit.framework.Assert
   A set of assert methods.
Assert() - Constructor for class junit.framework.Assert
   Protect constructor since it is a static only class
assertEquals(boolean, boolean) - Static method in class
junit.framework.Assert
  Asserts that two booleans are equal.
assertEquals(byte, byte) - Static method in class junit.framework.Assert
  Asserts that two bytes are equal.
assertEquals(char, char) - Static method in class junit.framework.Assert
  Asserts that two chars are equal.
assertEquals(double, double, double) - Static method in class junit.framework.Assert
  Asserts that two doubles are equal concerning a delta.
assertEquals(float, float, float) - Static method in class junit.framework.Assert
  Asserts that two floats are equal concerning a delta.
assertEquals(int, int) - Static method in class junit.framework.Assert
  Asserts that two ints are equal.
assertEquals(long, long) - Static method in class junit.framework.Assert
  Asserts that two longs are equal.
assertEquals(Object, Object) - Static method in class junit.framework.Assert
  Asserts that two objects are equal.
assertEquals(short, short) - Static method in class junit.framework.Assert
  Asserts that two shorts are equal.
assertEquals(String, boolean, boolean) - Static method in class junit.framework.Assert
  Asserts that two booleans are equal.
assertEquals(String, byte, byte) - Static method in class junit.framework.Assert
  Asserts that two bytes are equal.
assertEquals(String, char, char) - Static method in class junit.framework.Assert
  Asserts that two chars are equal.
assertEquals(String, double, double, double) - Static method in class junit.framework.Assert
  Asserts that two doubles are equal concerning a delta.
assertEquals(String, float, float, float) - Static method in class junit.framework.Assert
  Asserts that two floats are equal concerning a delta.
assertEquals(String, int, int) - Static method in class junit.framework.Assert
Asserts that two ints are equal.
**assertEquals(String, long, long)** - Static method in class junit.framework.Assert

Asserts that two longs are equal.
**assertEquals(String, Object, Object)** - Static method in class junit.framework.Assert

Asserts that two objects are equal.
**assertEquals(String, short, short)** - Static method in class junit.framework.Assert

Asserts that two shorts are equal.
**assertEquals(String, String)** - Static method in class junit.framework.Assert

Asserts that two Strings are equal.
**assertEquals(String, String, String)** - Static method in class junit.framework.Assert

Asserts that two Strings are equal.
**assertFalse(boolean)** - Static method in class junit.framework.Assert

Asserts that a condition is false.
**assertFalse(String, boolean)** - Static method in class junit.framework.Assert

Asserts that a condition is false.
**AssertionFailedError** - error junit.framework.AssertionFailedError

Thrown when an assertion failed.
**AssertionFailedError()** - Constructor for class junit.framework.AssertionFailedError

**AssertionFailedError(String)** - Constructor for class junit.framework.AssertionFailedError

**assertNotNull(Object)** - Static method in class junit.framework.Assert

Asserts that an object isn't null.
**assertNotNull(String, Object)** - Static method in class junit.framework.Assert

Asserts that an object isn't null.
**assertNotSame(Object, Object)** - Static method in class junit.framework.Assert

Asserts that two objects refer to the same object.
**assertNotSame(String, Object, Object)** - Static method in class junit.framework.Assert
Asserts that two objects refer to the same object.

**assertNull(Object)** - Static method in class junit.framework.Assert

Asserts that an object is null.

**assertNull(String, Object)** - Static method in class junit.framework.Assert

Asserts that an object is null.

**assertSame(Object, Object)** - Static method in class junit.framework.Assert

Asserts that two objects refer to the same object.

**assertSame(String, Object, Object)** - Static method in class junit.framework.Assert

Asserts that two objects refer to the same object.

**assertTrue(boolean)** - Static method in class junit.framework.Assert

Asserts that a condition is true.

**assertTrue(String, boolean)** - Static method in class junit.framework.Assert

Asserts that a condition is true.
basicRun(TestResult) - Method in class junit.extensions.TestDecorator
The basic run behaviour.
ComparisonFailure - error junit.framework. ComparisonFailure. Thrown when an assert equals for Strings failed.

ComparisonFailure(String, String, String) - Constructor for class junit.framework. ComparisonFailure
Constructs a comparison failure.

countTestCases() - Method in class junit.extensions. TestDecorator

countTestCases() - Method in class junit.extensions. RepeatedTest

countTestCases() - Method in class junit.framework. TestSuite
Counts the number of test cases that will be run by this test.
countTestCases() - Method in class junit.framework. TestCase
Counts the number of test cases executed by run(TestResult result).
countTestCases() - Method in interface junit.framework. Test
Counts the number of test cases that will be run by this test.
createResult() - Method in class junit.framework. TestCase
Creates a default TestResult object
createTest(Class, String) - Static method in class junit.framework. TestSuite
...as the moon sets over the early morning Merlin, Oregon mountains, our intrepid adventurers type...
**endTest(Test)** - Method in class junit.framework.TestResult
   - Informs the result that a test was completed.

**endTest(Test)** - Method in interface junit.framework.TestListener
   - A test ended.

**errorCount()** - Method in class junit.framework.TestResult
   - Gets the number of detected errors.

**errors()** - Method in class junit.framework.TestResult
   - Returns an Enumeration for the errors

**exceptionMessage()** - Method in class junit.framework.TestFailure

**ExceptionTestCase** - class junit.extensions.ExceptionTestCase
   - A TestCase that expects an Exception of class fExpected to be thrown.

**ExceptionTestCase(String, Class)** - Constructor for class
junit.extensions.ExceptionTestCase

---
fail() - Static method in class junit.framework.Assert
Fails a test with no message.
fail(String) - Static method in class junit.framework.Assert
Fails a test with the given message.
failedTest() - Method in class junit.framework.TestFailure
Gets the failed test.
failureCount() - Method in class junit.framework.TestResult
Gets the number of detected failures.
failures() - Method in class junit.framework.TestResult
Returns an Enumeration for the failures
fErrors - Variable in class junit.framework.TestResult

fFailedTest - Variable in class junit.framework.TestFailure

fFailures - Variable in class junit.framework.TestResult

fListeners - Variable in class junit.framework.TestResult

fRunTests - Variable in class junit.framework.TestResult

fTest - Variable in class junit.extensions.TestDecorator

fThrownException - Variable in class junit.framework.TestFailure
**getMessage()** - Method in class junit.framework.ComparisonFailure
   Returns "..." in place of common prefix and "..." in place of common suffix between expected and actual.

**getName()** - Method in class junit.framework.TestSuite
   Returns the name of the suite.

**getName()** - Method in class junit.framework.TestCase
   Gets the name of a TestCase.

**gtestTest()** - Method in class junit.extensions.TestDecorator
   Gets a constructor which takes a single String as its argument or a no arg constructor.
isFailure() - Method in class junit.framework.TestFailure
JUnit

**junit.extensions** - package junit.extensions

**junit.framework** - package junit.framework
protect() - Method in interface junit.framework.Protectable
Run the following method protected.
A Protectable can be run and can throw a Throwable.
**removeListener(TestListener)** - Method in class junit.framework.TestResult
   Unregisters a TestListener

**RepeatedTest** - class junit.extensions.RepeatedTest.
   A Decorator that runs a test repeatedly.

**RepeatedTest(Test, int)** - Constructor for class junit.extensions.RepeatedTest

**run()** - Method in class junit.framework.TestCase
   A convenience method to run this test, collecting the results with a default TestResult object.

**run(TestCase)** - Method in class junit.framework.TestResult
   Runs a TestCase.

**run(TestResult)** - Method in class junit.extensions.TestSetup

**run(TestResult)** - Method in class junit.extensions.TestDecorator

**run(TestResult)** - Method in class junit.extensionsRepeatedTest

**run(TestResult)** - Method in class junit.extensions.ActiveTestSuite

**run(TestResult)** - Method in class junit.framework.TestSuite
   Runs the tests and collects their result in a TestResult.

**run(TestResult)** - Method in class junit.framework.TestCase
   Runs the test case and collects the results in TestResult.

**run(TestResult)** - Method in class junit.framework.Test
   Runs a test and collects its result in a TestResult instance.

**runBare()** - Method in class junit.framework.TestCase
   Runs the bare test sequence.

**runCount()** - Method in class junit.framework.TestResult
   Gets the number of run tests.

**runFinished(Test)** - Method in class junit.extensions.ActiveTestSuite

**runProtected(Test, Protectable)** - Method in class junit.framework.TestResult
Runs a TestCase.

**runTest()** - Method in class junit.extensions.ExpectedTestCase
Execute the test method expecting that an Exception of class fExpected or one of its subclasses will be thrown

**runTest()** - Method in class junit.framework.TestCase
Override to run the test and assert its state.

**runTest(Test, TestResult)** - Method in class junit.extensions.ActiveTestSuite

**runTest(Test, TestResult)** - Method in class junit.framework.TestSuite
**setName(String)** - Method in class `junit.framework.TestSuite`
Sets the name of the suite.

**setName(String)** - Method in class `junit.framework.TestCase`
Sets the name of a `TestCase`

**setUp()** - Method in class `junit.extensions.TestSetup`
Sets up the fixture.

**setUp()** - Method in class `junit.framework.TestCase`
Sets up the fixture, for example, open a network connection.

**shouldStop()** - Method in class `junit.framework.TestResult`
Checks whether the test run should stop

**startTest(Test)** - Method in class `junit.framework.TestResult`
Informs the result that a test will be started.

**startTest(Test)** - Method in interface `junit.framework.TestListener`
A test started.

**stop()** - Method in class `junit.framework.TestResult`
Marks that the test run should stop.
tearDown() - Method in class junit.extensions.TestSetup
   Tears down the fixture.
tearDown() - Method in class junit.framework.TestCase
   Tears down the fixture, for example, close a network connection.
Test - interface junit.framework.Test.
   A Test can be run and collect its results.
testAt(int) - Method in class junit.framework.TestSuite
   Returns the test at the given index
TestCase - class junit.framework.TestCase.
   A test case defines the fixture to run multiple tests.
TestCase() - Constructor for class junit.framework.TestCase
   No-arg constructor to enable serialization.
TestCase(String) - Constructor for class junit.framework.TestCase
   Constructs a test case with the given name.
testCount() - Method in class junit.framework.TestSuite
   Returns the number of tests in this suite
TestDecorator - class junit.extensions.TestDecorator.
   A Decorator for Tests.
TestDecorator(Test) - Constructor for class junit.extensions.TestDecorator
   Constructs a TestDecorator with the given test.
TestFailure - class junit.framework.TestFailure.
   A TestFailure collects a failed test together with the caught exception.
TestFailure(Test, Throwable) - Constructor for class junit.framework.TestFailure
   Constructs a TestFailure with the given test and exception.
TestListener - interface junit.framework.TestListener.
   A Listener for test progress
TestResult - class junit.framework.TestResult.
   A TestResult collects the results of executing a test case.
TestResult() - Constructor for class junit.framework.TestResult
   Returns the tests as an enumeration

tests() - Method in class junit.framework.TestSuite
   Returns the tests as an enumeration
**TestSetup** - class junit.extensions.**TestSetup**.
   A Decorator to set up and tear down additional fixture state.
**TestSetup(Test)** - Constructor for class junit.extensions.**TestSetup**

**TestSuite** - class junit.framework.**TestSuite**.
   A TestSuite is a Composite of Tests.
**TestSuite()** - Constructor for class junit.framework.**TestSuite**
   Constructs an empty TestSuite.
**TestSuite(Class)** - Constructor for class junit.framework.**TestSuite**
   Constructs a TestSuite from the given class.
**TestSuite(Class, String)** - Constructor for class junit.framework.**TestSuite**
   Constructs a TestSuite from the given class with the given name.
**TestSuite(String)** - Constructor for class junit.framework.**TestSuite**
   Constructs an empty TestSuite.

**thrownException()** - Method in class junit.framework.**TestFailure**
   Gets the thrown exception.
**toString()** - Method in class junit.extensions.**TestDecorator**

**toString()** - Method in class junit.extensions.**RepeatedTest**

**toString()** - Method in class junit.framework.**TestSuite**

**toString()** - Method in class junit.framework.**TestFailure**
   Returns a short description of the failure.
**toString()** - Method in class junit.framework.**TestCase**
   Returns a string representation of the test case
**trace()** - Method in class junit.framework.**TestFailure**
wasSuccessful() - Method in class junit.framework.TestResult

Returns whether the entire test was successful or not.
How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The Overview page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (italic)
- Classes
- Exceptions
- Errors

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description
Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

**Tree (Class Hierarchy)**

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

**Deprecated API**

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

**Index**

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.
Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

This help file applies to API documentation generated using the standard doclet.
All Classes

ActiveTestSuite
Assert
AssertionFailedError
ComparisonFailure
ExceptionTestCase
Protectable
RepeatedTest
Test
TestCase
TestDecorator
TestFailure
TestListener
TestResult
TestSetup
TestSuite
Serialized Form

Package junit.framework

Class
junit.framework.AssertionFailedError
implements Serializable

Class
junit.framework.ComparisonFailure
implements Serializable

Serialized Fields

fExpected
java.lang.String fExpected

fActual
java.lang.String fActual