

JUnit Tests

Annotationen

<code>import org.junit.*</code>	Import, der für die folgenden Annotationen benötigt wird.
<code>@Test</code>	Markiert eine Methode als Test-Methode.
<code>@Before</code>	Markiert eine Methode, die vor jeder einzelnen Test-Methode ausgeführt werden soll.
<code>@After</code>	Markiert eine Methode, die nach jeder einzelnen Test-Methode ausgeführt werden soll.
<code>@BeforeClass</code>	Markiert eine Methode, die einmal vor allen Tests der Test-Klasse ausgeführt werden soll.
<code>@AfterClass</code>	Markiert eine Methode, die einmal nach allen Tests der Test-Klasse ausgeführt werden soll.
<code>@Ignore</code> oder <code>@Ignore("Why disabled")</code>	Markiert, dass ein Test ignoriert werden soll. Ein optionaler Kommentar zur Begründung kann angehängt werden.
<code>@Test(expected = Exception.class)</code>	Der Test schlägt fehl, wenn nicht die angegebene Exception geworfen wird.
<code>@Test(timeout=100)</code>	Der Test schlägt fehl, wenn die Ausführung länger dauert als angegeben (in Millisekunden).

JUnit Tests

Statements

<code>fail([message])</code>	Lässt den Test fehlschlagen.
<code>assertTrue([message,] boolean condition)</code>	Überprüft, ob eine boolean-Bedingung "true" ist.
<code>assertFalse([message,] boolean condition)</code>	Überprüft, ob eine boolean-Bedingung "false" ist.
<code>assertEquals([message,] expected, actual)</code>	Überprüft, ob zwei Werte gleich sind (bei Arrays: Referenz, nicht Elemente!).
<code>assertEquals([message,] expected, actual, tolerance)</code>	Überprüft, ob zwei float- oder double-Werte gleich sind. Es kann eine Toleranz angegeben werden.
<code>assertNull([message,] object)</code>	Überprüft, ob ein Wert "null" ist.
<code>assertNotNull([message,] object)</code>	Überprüft, ob ein Wert nicht "null" ist.
<code>assertSame([message,] expected, actual)</code>	Überprüft, ob zwei Variablen das selbe Objekt referenzieren.
<code>assertNotSame([message,] expected, actual)</code>	Überprüft, ob zwei Variablen nicht das selbe Objekt referenzieren.