TestCase

```
_cleanups : list
_diffThreshold : int
_outcome : NoneType
_subtest : NoneType
_testMethodDoc : str
_testMethodName : str
_type_equality_funcs : dict
assertAlmostEquals
assertEquals
assertNotAlmostEquals
assertNotEquals
assertNotRegexpMatches
assertRaisesRegexp
assertRegexpMatches
assert_
failIf
failIfAlmostEqual
failIfEqual
failUnless
failUnlessAlmostEqual
failUnlessEqual
failUnlessRaises
failureException : AssertionError
longMessage : bool
maxDiff : int
```

```
__call__()
__eq__(other)
__hash__()
__init__(methodName)
__init_subclass__()
__repr__()
__str__()
_addExpectedFailure(result, exc_info)
_addUnexpectedSuccess(result)
_baseAssertEqual(first, second, msg)
_callCleanup()
_callSetUp()
_callTearDown()
_callTestMethod(method)
_deprecate()
_formatMessage(msg, standardMsg)
_getAssertEqualityFunc(first, second)
_truncateMessage(message, diff)
addClassCleanup()
addCleanup()
addTypeEqualityFunc(typeobj, function)
assertAlmostEqual(first, second, places, msg, delta)
assertCountEqual(first, second, msg)
assertDictContainsSubset(subset, dictionary, msg)
assertDictEqual(d1, d2, msg)
assertEqual(first, second, msg)
assertFalse(expr, msg)
assertGreater(a, b, msg)
assertGreaterEqual(a, b, msg)
assertIn(member, container, msg)
assertIs(expr1, expr2, msg)
assertIsInstance(obj, cls, msg)
assertIsNone(obj, msg)
assertIsNot(expr1, expr2, msg)
assertIsNotNone(obj, msg)
assertLess(a, b, msg)
assertLessEqual(a, b, msg)
assertListEqual(list1, list2, msg)
assertLogs(logger, level)
assertMultiLineEqual(first, second, msg)
assertNoLogs(logger, level)
assertNotAlmostEqual(first, second, places, msg, delta)
assertNotEqual(first, second, msg)
assertNotIn(member, container, msg)
assertNotIsInstance(obj, cls, msg)
assertNotRegex(text, unexpected_regex, msg)
assertRaises(expected_exception)
assertRaisesRegex(expected_exception, expected_regex)
assertRegex(text, expected_regex, msg)
assertSequenceEqual(seq1, seq2, msg, seq_type)
assertSetEqual(set1, set2, msg)
assertTrue(expr, msg)
assertTupleEqual(tuple1, tuple2, msg)
assertWarns(expected_warning)
assertWarnsRegex(expected_warning, expected_regex)
countTestCases()
debug()
defaultTestResult()
doClassCleanups()
doCleanups()
enterClassContext(cm)
enterContext(cm)
fail(msg)
id()
run(result)
setUp()
setUpClass()
shortDescription()
skipTest(reason)
subTest(msg)
tearDown()
tearDownClass()
```

TestChat

```
anweisung : str
max_tokens : int
model : str
```

```
setUp()
test_chat()
test_erzeuge_unittest_model()
test_erzeuge_unittest_skript_quelle(skript_quelle)
test_erzeuge_unittest_true_result()
```

TestDateien

```
testdatei : str
```

```
setUp()
tearDown()
test_schreibe_protokol()
```

_Outcome

```
expectedFailure : NoneType
expecting_failure : bool
result : NoneType
result_supports_subtests : bool
success : bool
```

```
__init__(result)
testPartExecutor(test_case, subTest)
```

_SubTest

```
_message
failureException
params
test_case
```

```
__init__(test_case, message, params)
__str__()
_subDescription()
id()
runTest()
shortDescription()
```

_outcome

_subtest   _subtest