

JUnit:

`assertEquals("ABC", "ABC(n"));`

↳ compare two strings.

expected

actual value

↓
result of
operation

`public void test()`

{

`assertEquals("ABC", "ABCD");`

}

right test

↓
refactor

↓
inline

↓
reference will be
changed.

Any test method should be public and void.

no other will accept.

`assertTrue` and

`assertFalse`.

ABC → true

AB → true

AB → true
A → false

@Before

```
public void setUp() {  
    sop("Before test");  
}
```

@Test

```
public void test1() {  
    sop("test1");  
}
```

@Test

```
public void test2() {  
    sop("test2");  
}
```

@After

```
public void tearDown() {  
    sop("After test");  
}
```

@BeforeClass → executes only once

```
public static void beforeClass()
```

```
{
```

```
sop("Before class");
```

```
}
```

Before class
always static.

@AfterClass

```
public static void afterClass() {  
    Sop("inAfterClass");  
}
```

}

Comparing array in JUnit.

@Test

```
public void test() {
```

```
    int numbers[] = {12, 3, 4, 1};
```

```
    int[] expected = {1, 3, 4, 12};
```

```
    Arrays.sort(numbers);
```

```
    assertEquals(expected, numbers);
```

Testing exceptions

@Test(expected = NullPointerException.class)

```
public void testArraySort() {
```

```
    int[] numba = null;
```

```
    Arrays.sort(numba);
```

```
}
```

@Test (timeOut = 1000)

```
public void testSortPerformance() {
```

```
    int array[] = {12, 23, 4}
```

```
    for (int i = 1; i < 1000; i++)
```

```
    {
```

```
        array[0] = i;
```

```
        Arrays.sort(array)
```

```
    }
```

```
}
```

Parameterized tests:

@RunWith (parameterized.class)

```
public class StringHelperParameterizedTest {
```

```
    StringHelper helper = new StringHelper();
```

@Parameters

```
    public static Collection<String> testCondition() {
```

```
        { "AAcd", "cd" }
```

```
        String expected[][] = { { "AAcd", "cd" },  
                                { "Acd", "cd" } };
```

```
        return Arrays.asList(expected);
```

↓

converting array to collection

```
}
```

private String input;
private String expected;

Mockito: Used to mock

TodoService is a interface
→ Dependency

public interface TodoService {

public List<String> retrieveTodo(String user);

public class TodoBusinessImpl {

private TodoService todoService;

}

↳ System under test

TodoServiceStub

Stub is a class which return dummy data.

1. import static org.junit.Assert;