JUnit promotes the idea of "first testing then coding", which emphasis on setting up the test data for a piece of code which can be tested first and then can be implemented .

Features

JUnit is an open source framework which is used for writing & running tests.

Provides Annotation to identify the test methods.

Provides Assertions for testing expected results.

Provides Test runners for running tests.

JUnit tests allow you to write code faster which increasing quality

JUnit is elegantly simple. It is less complex & takes less time.

JUnit tests can be run automatically and they check their own results and provide immediate feedback. There's no need to manually comb through a report of test results.

JUnit tests can be organized into test suites containing test cases and even other test suites.

Junit shows test progress in a bar that is green if test is going fine and it turns red when a test fails.

Important API's of JUnit

Serial No	Class Name	Functionality
1	Assert	A set of assert methods.
2	TestCase	A test case defines the fixture to run multiple tests.
3	TestResult	A TestResult collects the results of executing a test case.
4	TestSuite	A TestSuite is a Composite of Tests.

Different assert methods:

void **assertEquals**(boolean expected, boolean actual) - Check that two primitives/Objects are equal

void **assertFalse**(boolean condition) - Check that a condition is false

void **assertNotNull**(Object object) - Check that an object isn't null.

void **assertNull**(Object object) - Check that an object is null

void **assertTrue**(boolean condition) - Check that a condition is true.

void **fail**()- Fails a test with no message.

Annotation

Annotations are like meta-tags that you can add to you code and apply them to methods or in class. These annotation in JUnit gives us information about test methods , which methods are going to run before & after test methods, which methods run before & after all the methods, which methods or class will be ignore during execution.

List of annotations and their meaning in JUnit :

Annotation & Description

@Test

The Test annotation tells JUnit that the public void method to which it is attached can be run as a test case.

@Before

Several tests need similar objects created before they can run. Annotating a public void method with @Before causes that method to be run before each Test method.

@After

If you allocate external resources in a Before method you need to release them after the test runs. Annotating a public void method with @After causes that method to be run after the Test method.

@BeforeClass

Annotating a public static void method with @BeforeClass causes it to be run once before any of the test methods in the class.

@AfterClass

This will perform the method after all tests have finished. This can be used to perform clean-up activities.

@Ignore

The Ignore annotation is used to ignore the test and that test will not be executed.