**What is Junit?**

JUnit is an open source Unit Testing Framework for JAVA. It is useful for java Developers to write and run repeatable tests. Erich Gamma and Kent Beck initially develop it. It is an instance of xunit architecture. As the name implies, it is used for unit testing of a small chunk of code.

Developers who are following test-driven methodology must write and execute unit test first before any code. Once you are done with code, you should execute all tests, and it should pass. Every time any code is added, you need to re-execute all test cases and makes sure nothing is broken.

JUnit does not require server for testing web application, which makes the testing process fast. JUnit framework also allows quick and easy generation of test cases and test data. The **org.Junit** package consist of many interfaces and classes for JUnit Testing such as Test, Assert, After, Before, etc.

**Implementation of Junit:-**

Create one maven project-

Add the following dependency of either Junit 4 or Junit 5 as per latest version available:

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

<dependency>

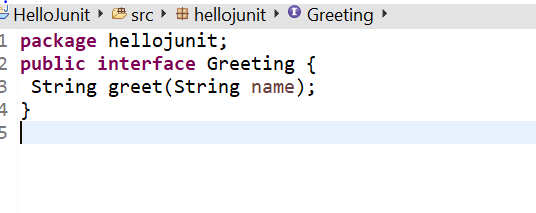
<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

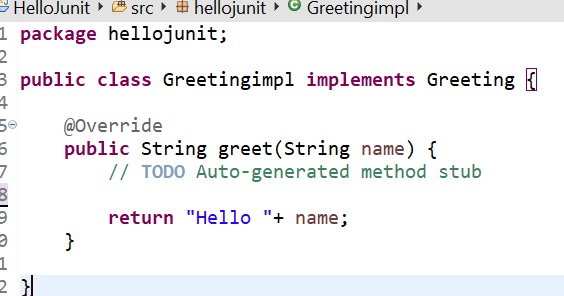
<version>5.9.1</version>

</dependency>

Let’s Create one interface i.e. Greeting as following:



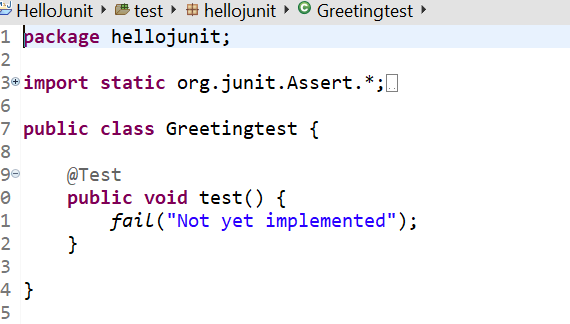
Create one class which implements this interface as below:



After then in test folder create one junit test case:

Here @Test – specifies method is test method and method should always be public void, Otherwise it will throw error.

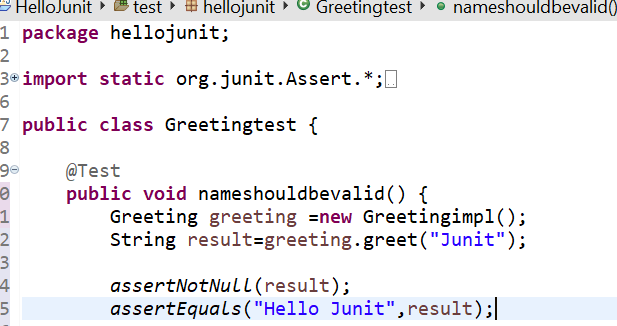
Fail Message - We can fail a test explicitly when it is incomplete or not yet implemented



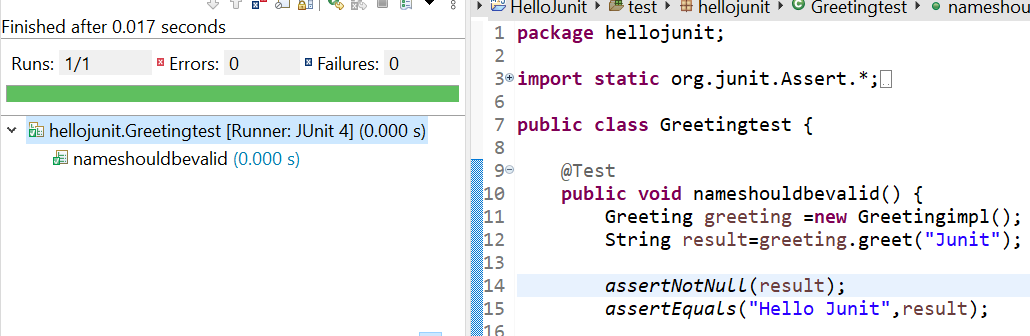
Create one test method :

Assert Not Null:- Checks value should not be null

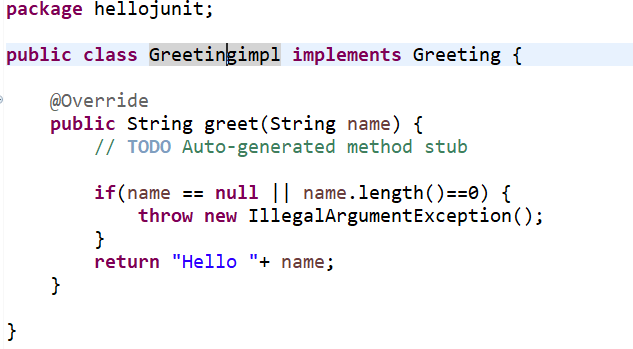
Assert Equals(expected value, actual value):- Checks the two objects are equal



Run the testcase as Junit:

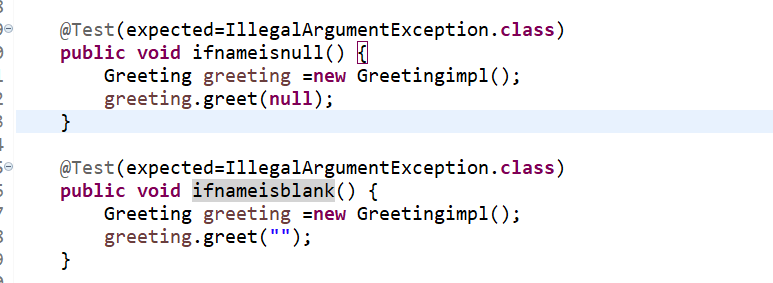


Add some logic for exception in code:

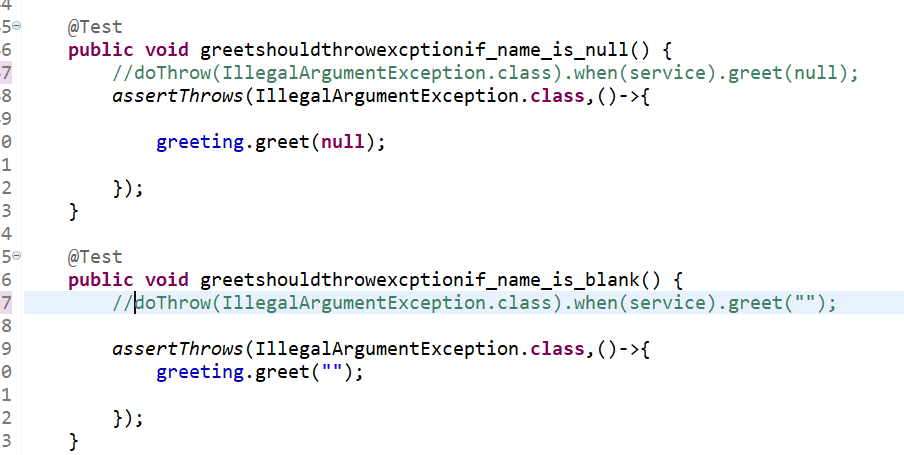


Create method to test for exception :

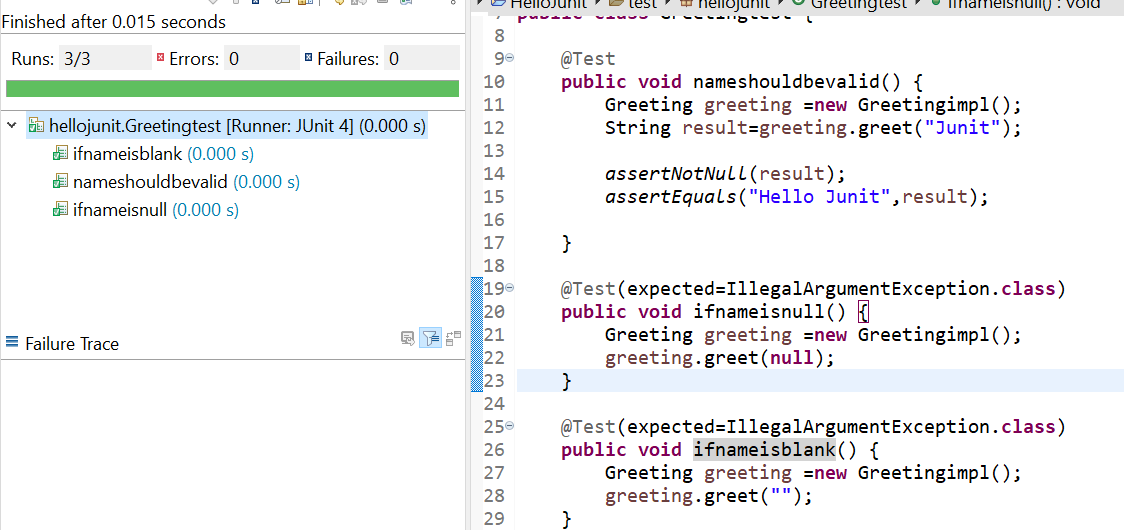
Here expected can be pass as argument with @Test in junit4



Junit 5 exception testcase:

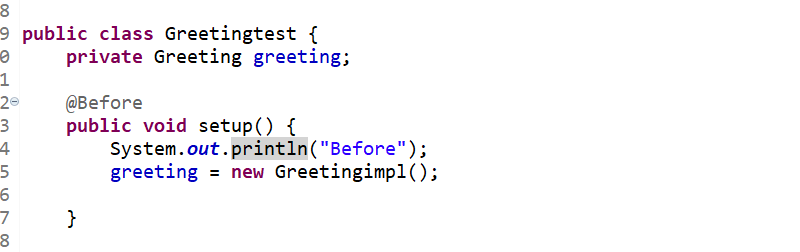


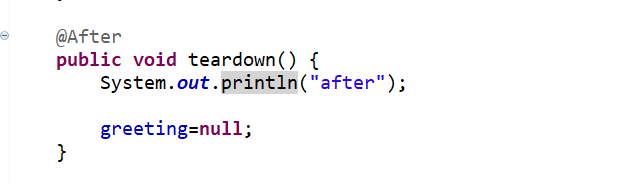
Run test as Junit test:



@Before – specifies the method will be invoked before each test

@After-- specifies that method will be invoked after each test.

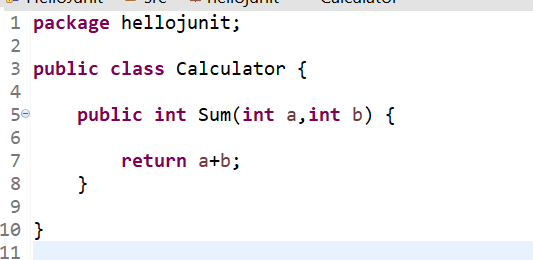




Parameterized Tests:-

A parameterized test is a test case that is invoked by using a predefined input and expected output. These tests are typically used to test classes that have no external dependencies and that provide methods which have no side effects. Parameterized test is to execute the same test over and over again using different values. It helps developer to save time in executing same test which differs only in their inputs and expected results.

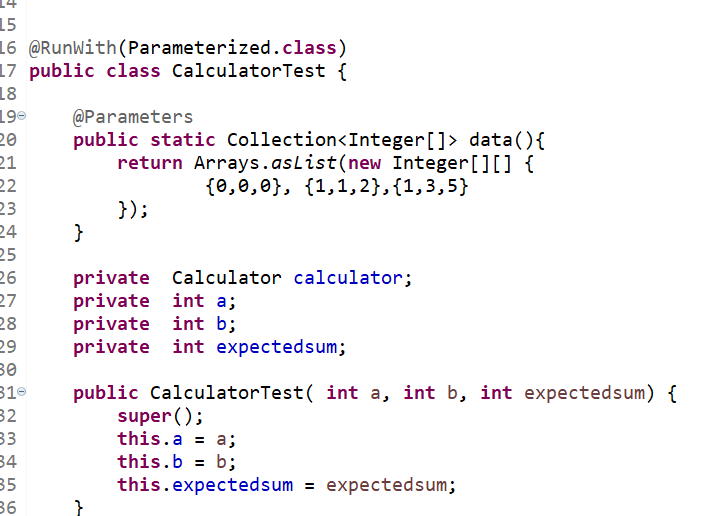
Create one class e.g. Calculator:



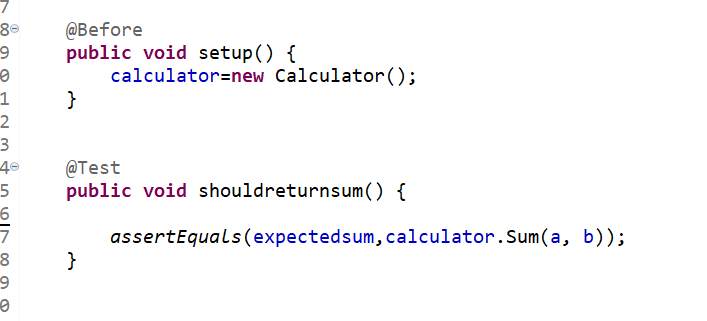
Create testcase of above class:

- Use **parameterized class** argument along **@Runwith**

- Mark static input fields with **@Parameters**



Create test method :



Run the test as Junit test:

