

JUnit

Introduction:

- 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 is used for [Unit Testing](#) of a small chunk of code.

Importance of junit:

- It finds bugs early in the code, which makes our code more reliable.
- JUnit is useful for developers, who work in a test-driven environment.
- Unit testing forces a developer to read code more than writing.
- You develop more readable, reliable and bug-free code which builds confidence during development.

Advantages of junit:

- JUnit 4 can be used with java5 or higher version.
- While using JUnit4, you are not required to [extend **JUnit.framework.TestCase**](#). You can just create a simple java class.
- You need to use annotations in spite of special method name as before.
 - Instead of using **setup** method, you need to use **@before** annotation.
 - Instead of using **teardown** method, put **@after** annotation.
 - Instead of using **testxxxx** before method name, use **@test** annotation.

Steps to install jdk:

- Visit <http://junit.org/junit4/> and click Download and Install

Click junit.jar

In the central repository you are shown all versions of Junit that can be downloaded. Usually, you will select the latest version. Click on jar link to download Junit version 4.12

Visit <https://github.com/junit-team/junit4/wiki/Download-and-Install> again. Click hamcrest-core.jar

Download the Jar

1. Go to java build path window as shown in below figure
2. Now click on "Add External JARs" button to add your downloaded JUnit.jar file with eclipse.

After adding a JUnit.jar file, click on 'OK' button to close java build path window.

JUnit annotation

1. @Test	This annotation is a replacement of org.junit.TestCase which indicates that public void method to which it is attached can be executed as a test Case.
2. @Before	This annotation is used if you want to execute some statement such as preconditions before each test case.
3. @BeforeClass	This annotation is used if you want to execute some statements before all the test cases for e.g. test connection must be executed before all the test cases.
4. @After	This annotation can be used if you want to execute some statements after each Test Case for e.g resetting variables, deleting temporary files ,variables, etc.
5. @AfterClass	This annotation can be used if you want to execute some statements after all test cases for e.g. Releasing resources after executing all test cases.
6. @Ignore	This annotation can be used if you want to ignore some statements during test execution for e.g. disabling some test cases during test execution.
7. @Test(timeout=500)	This annotation can be used if you want to set some timeout during test execution for e.g. if you are working under some SLA (Service

level agreement), and tests need to be completed within some specified time.

Order of execution:

@BeforeClass , executed before all test cases

@Before annotations, executed before each test cases

@Test(timeout),it can be used to enforce timeout in JUnit4 test case

@Test,used to execute the test

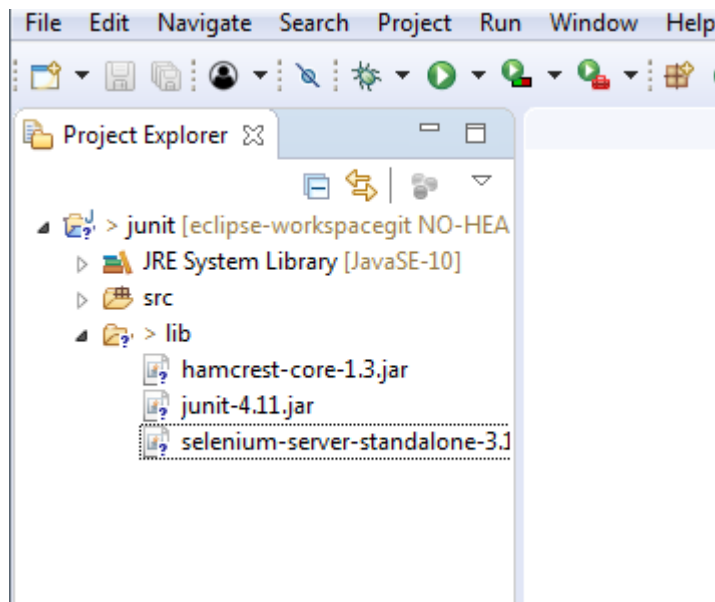
@Test(expected) ,it will check for specified exception during its execution

@After, executed after each test cases

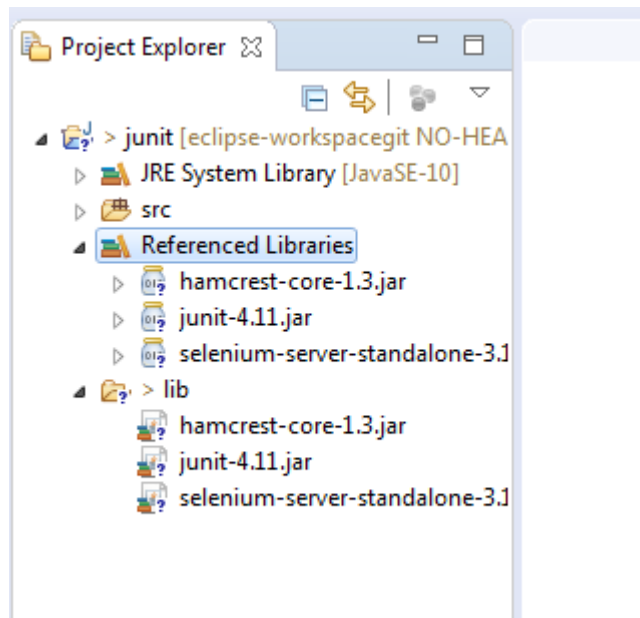
@AfterClass, executed after all test cases

Simple Junit program

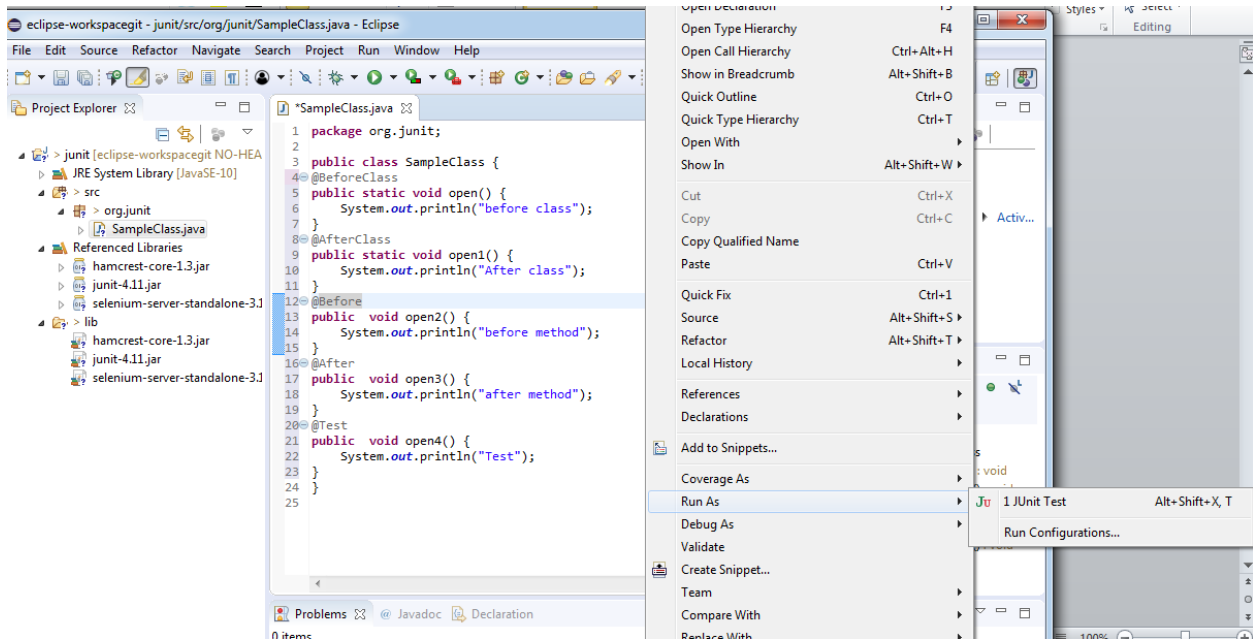
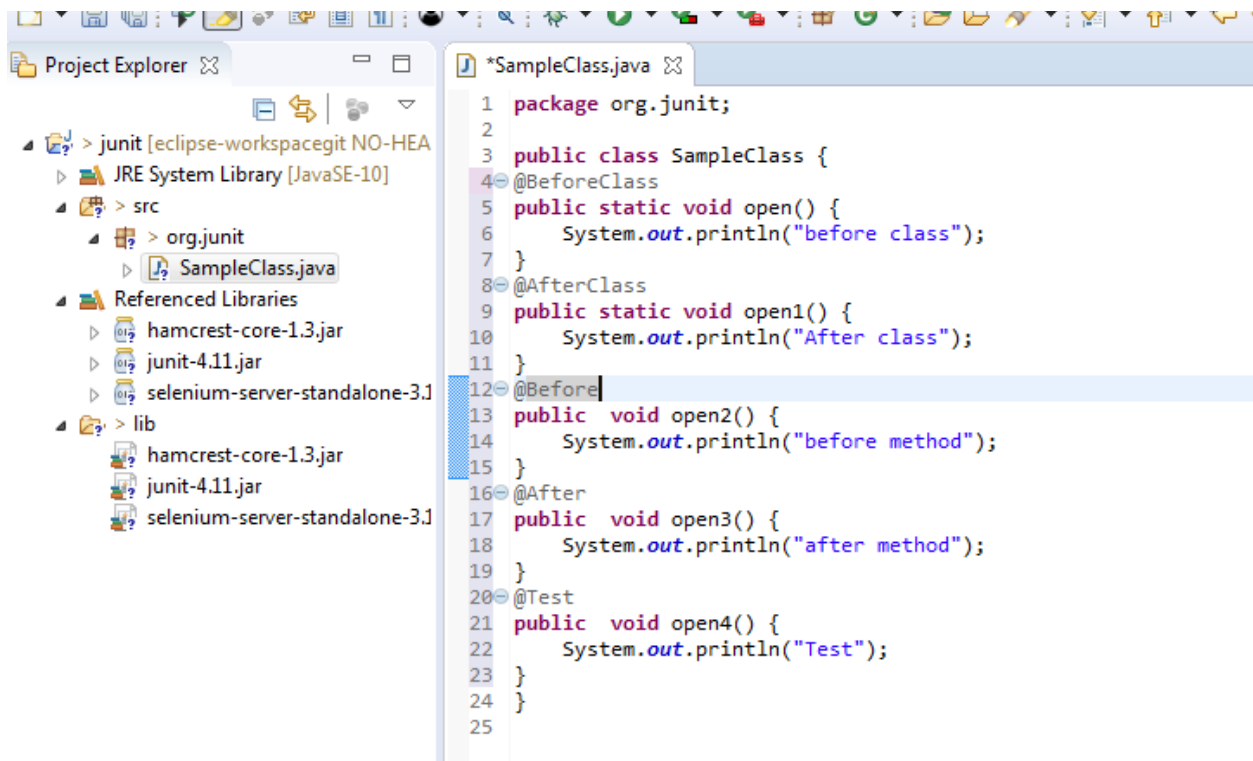
1.Add the jar files

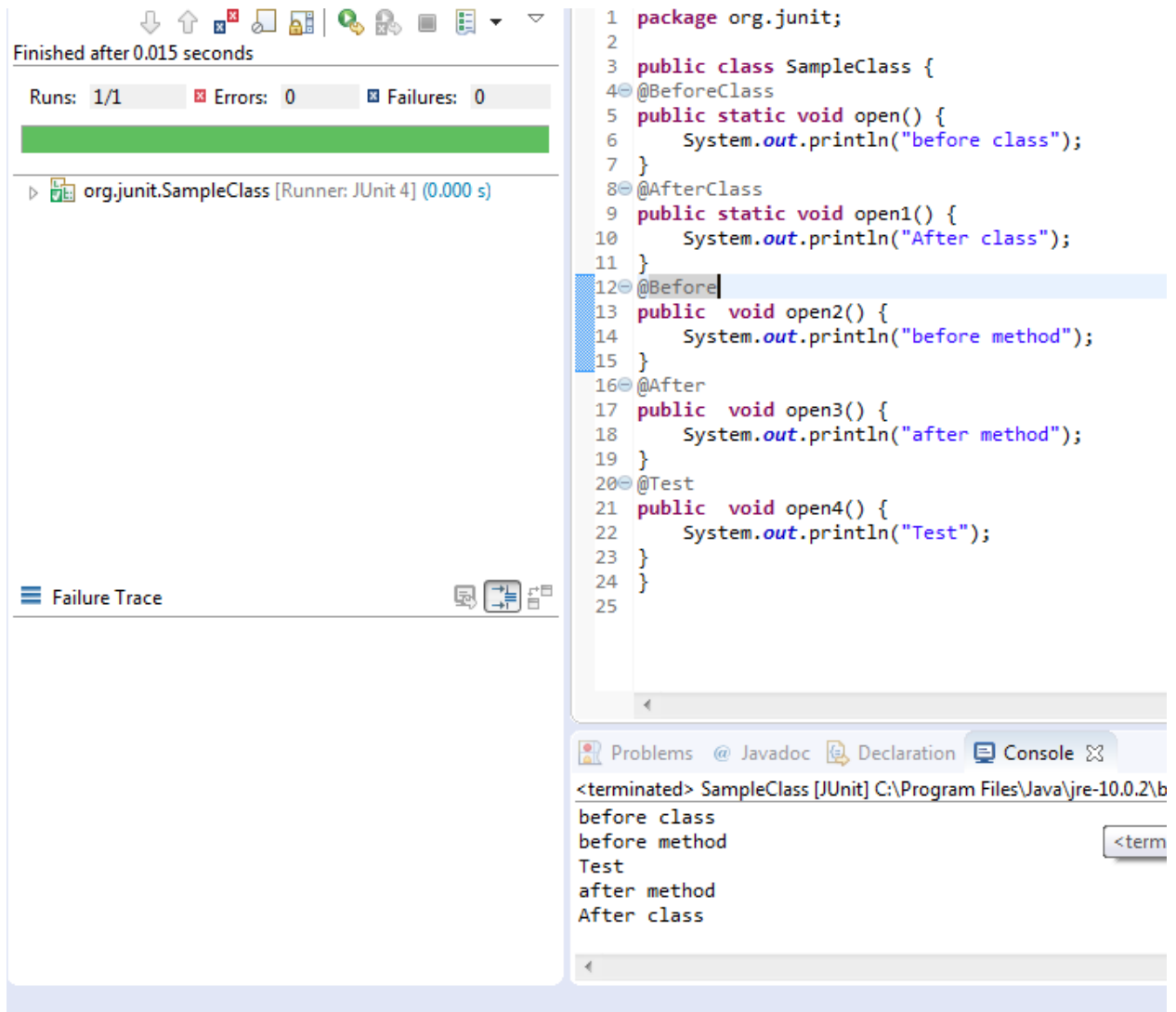


2.configure the jar file add it to build pat

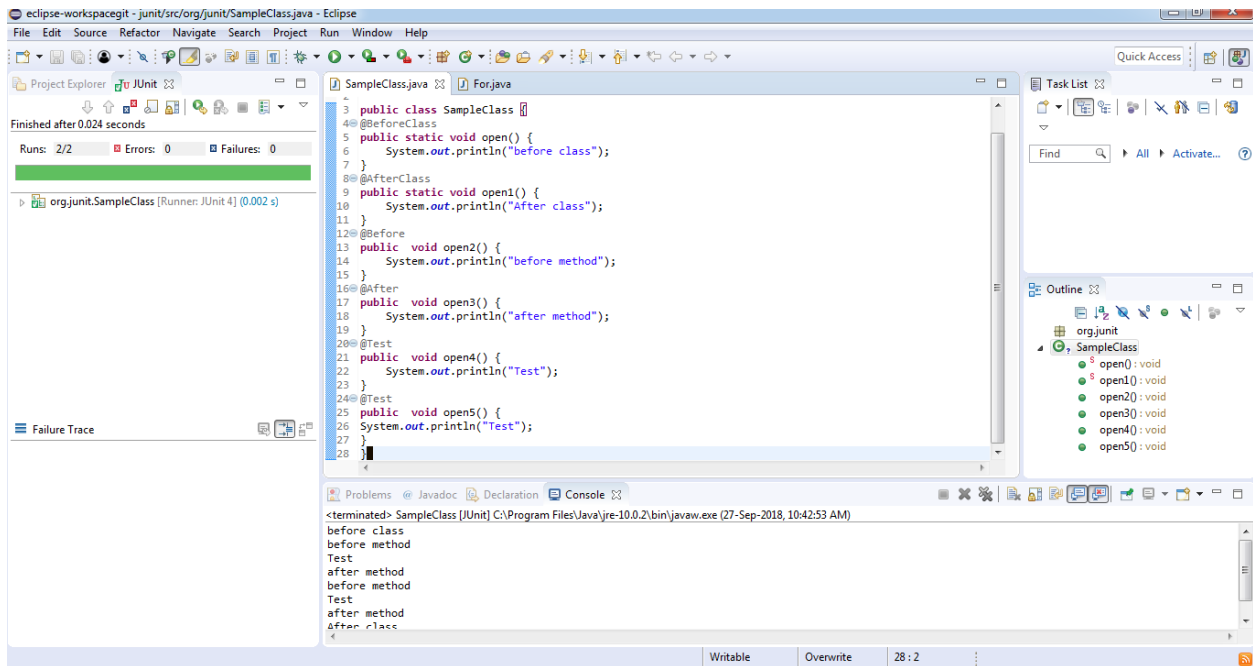


Program for order

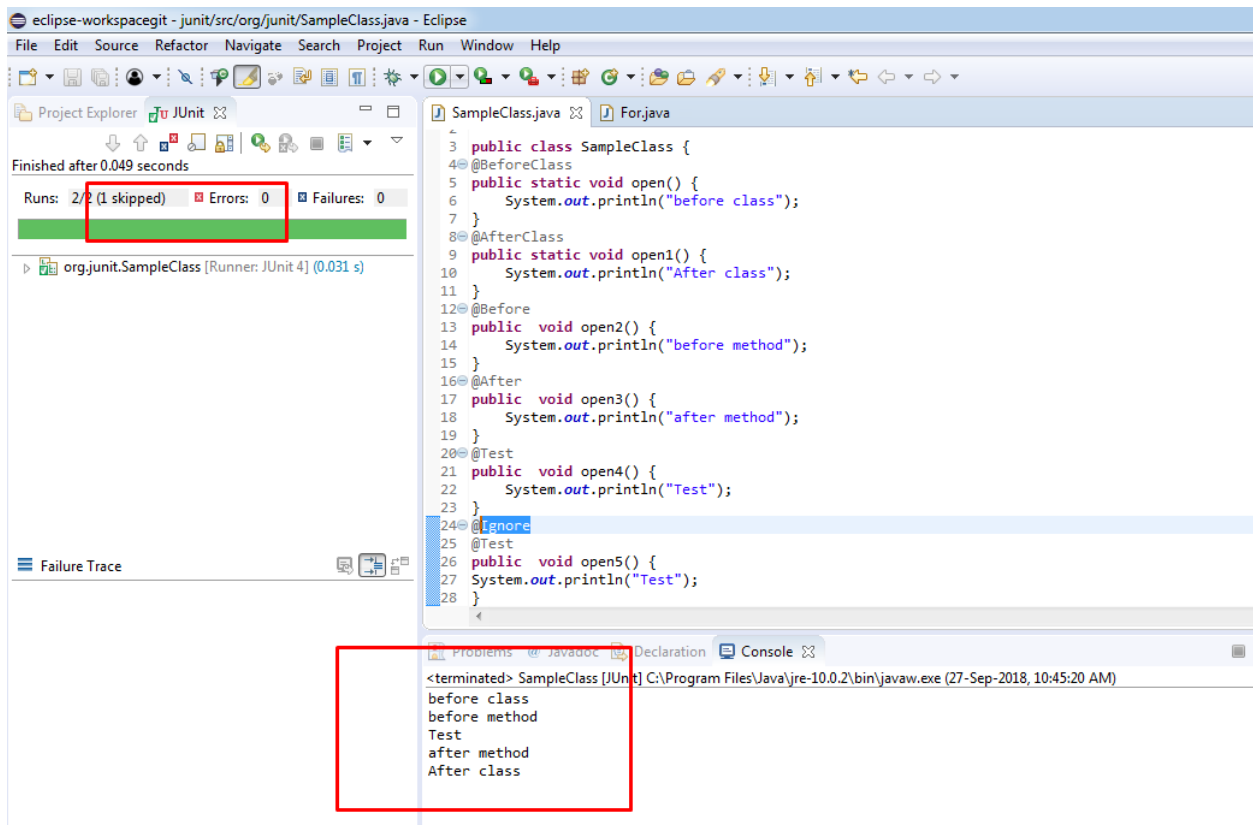




If I am giving more than one testcase the before method and after method runs for each and every test cases.



@ignore will skip the particular test cases



Timeout :

This option will run the test case for the particular time only

If the test case take more than mentioned time it will Fail automatically


Here I run the program without using time out:

```
SampleClass.java For.java
4 import org.openqa.selenium.chrome.ChromeDriver;
5
6 public class SampleClass {
7     @BeforeClass
8     public static void open() {
9         System.out.println("before class");
10    }
11    @AfterClass
12    public static void open1() {
13        System.out.println("After class");
14    }
15    @Before
16    public void open2() {
17        System.out.println("before method");
18    }
19    @After
20    public void open3() {
21        System.out.println("after method");
22    }
23    @Test
24    public void open4() {
25        System.out.println("Test");
26    }
27
28    @Test
29    public void open5() throws InterruptedException {
30        System.out.println("Test");
31        Thread.sleep(5000);
32        System.setProperty("webdriver.chrome.driver", "C:\\Users\\User\\eclipse-workspacegit\\junit\\Drive");
33        WebDriver driver=new ChromeDriver();
34
35        driver.get("https://github.com/");
36    }
37 }
```


Output

Finished after 24.688 seconds

Runs: 2/2 Errors: 0 Failures: 0

▶  org.junit.SampleClass [Runner: JUnit 4] (24.666 s)

Failure Trace

Problems Javadoc Declaration Console

<terminated> SampleClass [JUnit] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (27-Sep-2018, 10:54:28 AM)

before class
before method
Test
after method
before method
Test
Starting ChromeDriver 2.42.591088 (7b2b2dca23cca0862f674758c9a3933e685c27d5) on port 45301
Only local connections are allowed.
Sep 27, 2018 10:54:37 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
after method
After class

Here I use timeout

The screenshot shows the Eclipse IDE with a Java class named `SampleClass.java` open. The code is as follows:

```
4 import org.openqa.selenium.chrome.ChromeDriver;
5
6 public class SampleClass {
7     @BeforeClass
8     public static void open() {
9         System.out.println("before class");
10    }
11    @AfterClass
12    public static void open1() {
13        System.out.println("After class");
14    }
15    @Before
16    public void open2() {
17        System.out.println("before method");
18    }
19    @After
20    public void open3() {
21        System.out.println("after method");
22    }
23    @Test
24    public void open4() {
25        System.out.println("Test");
26    }
27
28    @Test(timeout=900)
29    public void open5() throws InterruptedException {
30        System.out.println("Test");
31        Thread.sleep(5000);
32        System.setProperty("webdriver.chrome.driver", "C:\\Users\\User\\eclipse-workspace\\git\\junit\\Driver\\");
33        WebDriver driver=new ChromeDriver();
34
35        driver.get("https://github.com/");
36    }
```

Below the code editor, the **Runs** tab shows the execution results:

- Runs: 2/2
- Errors: 1
- Failures: 0

The **org.junit.SampleClass [Runner: JUnit 4] (0.914 s)** entry is expanded, showing the following steps:

- open4 (0.001 s)
- open5 (0.913 s)

The **Problems** tab is also visible, showing a single error:

```
<terminated> SampleClass [JUnit] C:\Program Files\Ja
before class
before method
Test
after method
before method
Test
after method
After class
```