TEST AUTOMATION FUNDAMENTALS

KavinSchool

TESTNG OVERVIEW

KavinSchool

Learning Objectives

- ❑ TestNG Concepts
- ❑ TestNG Asserts
- ❑ Exception Handling
- ❑ Suites of Suite
- ❑ Running TestNG using Maven

1

TESTNG CONCEPTS

Learning Objectives

- ❑ History of TestNG
- ❑ TestNG Annotations
- ❑ Order of Execution
- ❑ Annotation Attribute
- ❑ testng.xml – structure
- ❑ Using Groups with Tests

HISTORY OF TESTNG

- TestNG is a Testing Framework
- Developed by Cédric Beust
- TestNG was developed out of the frustration of JUnit 3.x limited features set
- Uses annotations to define the test methods
- Flexible test configuration using external testng.xml file
- Support for parameters and data-driven testing using @DataProvider
- Allows distribution of tests on slave machines.
- Features to execute tests based on groups

TestNG

TESTNG CONFIG ANNOTATIONS

Annotations	Description
@BeforeSuite	Runs before all tests in the current suite have run
@AfterSuite	Runs after all tests in the current suite have run
@BeforeTest	Runs before any test method belonging to the classes inside the <test> tag have run
@AfterTest	Runs after all the test methods belonging to the classes inside the <test> tag have run
@BeforeGroups	Runs shortly before the first test method that belongs to any of the groups get invoked
@AfterGroups	Runs shortly after the last test method that belongs to any of the groups get invoked

TESTNG CONFIG ANNOTATIONS

Annotations	Description
@BeforeClass	Runs before the first test method in the current class gets invoked
@AfterClass	Runs after all the test methods in the current class have been run
@BeforeMethod	Runs before each test method
@AfterMethod	Runs after each test method

ORDER OF EXECUTION

- Order of Execution for a SquareOrderTest with groups run with “area” is given below

```
SquareOrderTest @BeforeSuite beforeSuite
SquareOrderTest @BeforeTest beforeTest
SquareTest @BeforeGroups setUpGroups functional
Side: 2.0 expResultArea: 4.0 expResultPerimeter: 8.0
SquareOrderTest @BeforeClass beforeClass
SquareOrderTest @BeforeMethod beforeMethod
SquareOrderTest @Test testArea Begins
Side: 2.0 expResultArea: 4.0 ActualResult: 4.0
SquareOrderTest @Test testArea Ends
SquareOrderTest @AfterMethod afterMethod
SquareOrderTest @AfterClass afterClass
SquareOrderTest @AfterTest afterTest
SquareTest @AfterGroups tearDownGroups functional
SquareOrderTest @AfterSuite afterSuite
```

testng_square_order_test.xml

Pay attention to the alwaysRun=true setup at different annotation level

TESTNG.XML

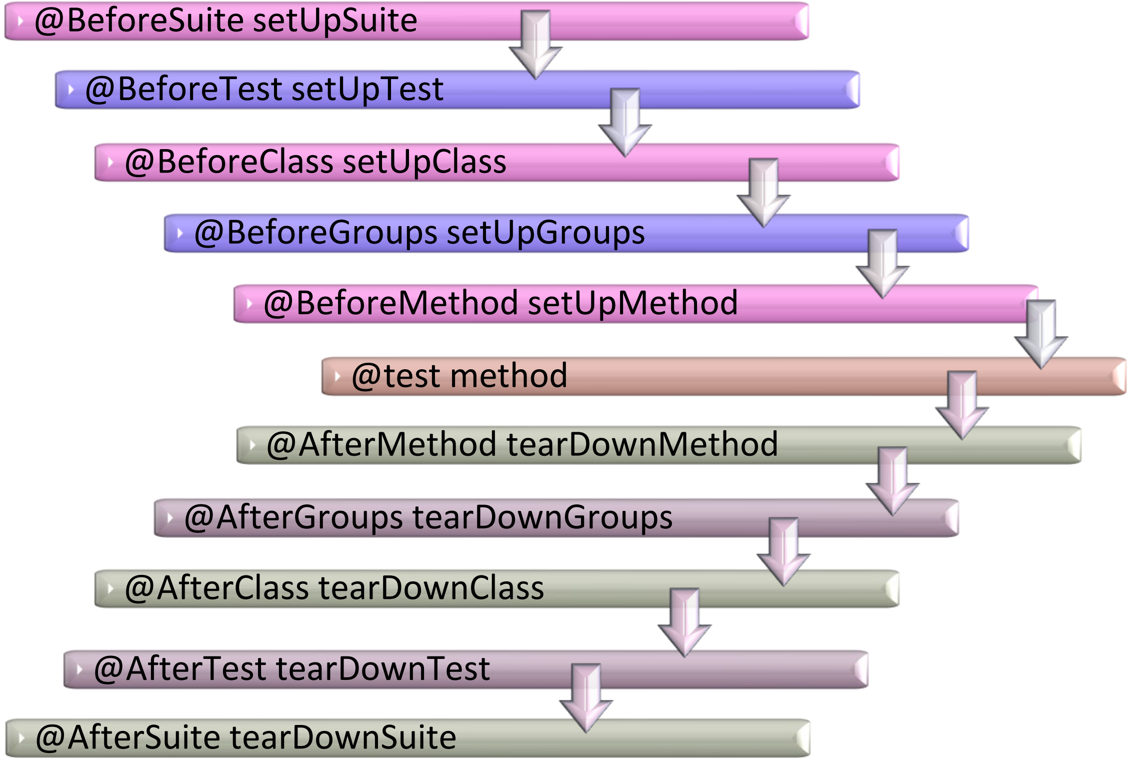
- Executing Groups with “Area”

Code

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="5" verbose="1" name="ShapeSuite" annotations="JDK">
   <test name="Shape Test">
      <groups>
         <run>
            <include name="area"/>
         </run>
      </groups>
      <classes>
         <class name="Com.KavinSchool.Shape.SquareTest"/>
      </classes>
   </test>
</suite>
```

group “area” only executed, where other @BeforeGroups are commented out

ORDER OF EXECUTION



@BeforeSuite setUpSuite

@BeforeTest setUpTest

@BeforeClass setUpClass

@BeforeGroups setUpGroups

@BeforeMethod setUpMethod

@test method

@AfterMethod tearDownMethod

@AfterGroups tearDownGroups

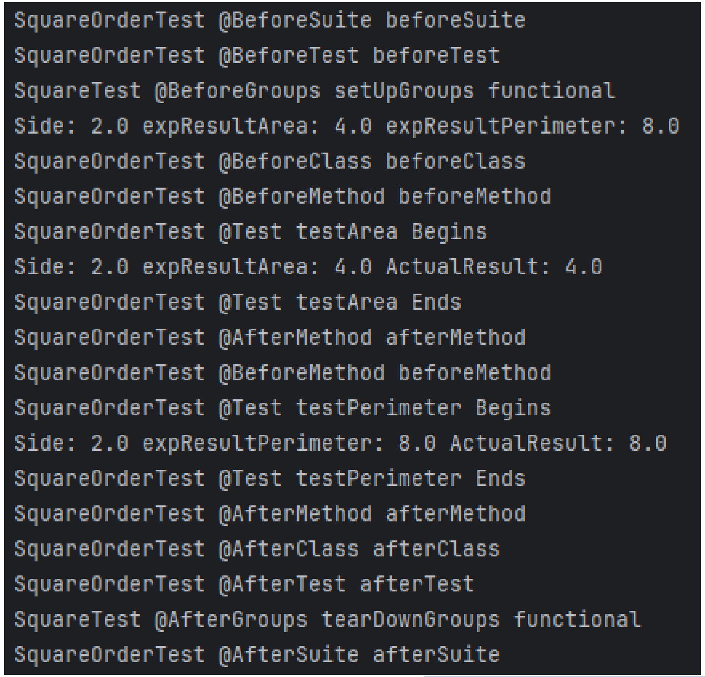
@AfterClass tearDownClass

@AfterTest tearDownTest

@AfterSuite tearDownSuite

ORDER OF EXECUTION

- Order of Execution for a SquareOrderTest run with group “functional” where “area” and “perimeter” both are functional methods.



```
SquareOrderTest @BeforeSuite beforeSuite
SquareOrderTest @BeforeTest beforeTest
SquareTest @BeforeGroups setUpGroups functional
Side: 2.0 expResultArea: 4.0 expResultPerimeter: 8.0
SquareOrderTest @BeforeClass beforeClass
SquareOrderTest @BeforeMethod beforeMethod
SquareOrderTest @Test testArea Begins
Side: 2.0 expResultArea: 4.0 ActualResult: 4.0
SquareOrderTest @Test testArea Ends
SquareOrderTest @AfterMethod afterMethod
SquareOrderTest @BeforeMethod beforeMethod
SquareOrderTest @Test testPerimeter Begins
Side: 2.0 expResultPerimeter: 8.0 ActualResult: 8.0
SquareOrderTest @Test testPerimeter Ends
SquareOrderTest @AfterMethod afterMethod
SquareOrderTest @AfterClass afterClass
SquareOrderTest @AfterTest afterTest
SquareTest @AfterGroups tearDownGroups functional
SquareOrderTest @AfterSuite afterSuite
```

TESTNG.XML

- Run the groups which are tagged as “functional”

Code

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="5" verbose="1" name="ShapeSuite" annotations="JDK">
   <test name="Shape Test">
      <groups>
         <run>
            <include name="“functional"/>
         </run>
      </groups>
      <classes>
         <class name="Com.KavinSchool.Shape.SquareTest"/>
      </classes>
   </test>
</suite>
```

group “functional” is includes both “area” and “perimeter”

CONFIG ANNOTATIONS ATTRIBUTES

Attributes	Description
alwaysRun	For @BeforeXXXX annotation methods (not for BeforeGroups): If set to true, this configuration method will be run regardless of what groups it belongs to. For @AfterXXXX methods (not for AfterGroups): If set to true, this configuration method will be run even if one or more methods invoked previously failed or was skipped

CONFIG ANNOTATIONS ATTRIBUTES

Attributes	Description
dependsOnGroups	The list of groups this method depends on
dependsOnMethods	The list of methods this method depends on
enabled	Whether methods on this class/method are enabled (@ignore of Junit)
groups	The list of groups this class/method belongs to
inheritGroups	If set to true, this method will belong to groups specified in the @Test annotation at the class level.

TESTNG.XML

- testng.xml is an XML file that describes the runtime definition of a TestNG test suite.
- testng.xml file allows you run various groups of test without re-compiling the code
- To achieve this, within the test code, you should define test groups and within testng.xml decide which groups to include/exclude.
- testng.xml is an XML file that contains the configuration of a test suite
- You can create multiple xml files (testng1.xml, testng2.xml,…) and run them at the same time

TESTNG.XML STRUCTURE

- The root tag of testng.xml is <suite>
- For a single package
  - A <suite> tag may contain zero or more <test> tags
  - A <test> tag may contain zero or one <classes> and or <groups> tags
  - A <classes> tag may contain one or more <class> tags
  - A <class> tag may contain zero or more <methods> tags
- For multiple packages
  - A <suite> tag may contain zero or more <packages> tags
  - A <packages> tag may contain zero or more <package> tags

TESTNG.XML STRUCTURE

- The following tags allows to include and/or exclude of a things:
  - <define> groups (using existing groups)
  - <package>
  - <methods>
- To include list of testng.xml files use the below tags
  - A <suite> tag may contain zero or more <suite-files> tag
  - A <suite-files> tag may contain <suite-file> tags

BEFORE/AFTER SUITE ALWAYSRUN

- Configuration annotation methods

Code

```
@BeforeSuite(alwaysRun = true)
    public static void setUpSuite() throws Exception {
        System.out.println("SquareTest @BeforeSuite setUpSuite");
    }

    @AfterSuite(alwaysRun = true)
    public static void tearDownSuite() throws Exception {
        System.out.println("SquareTest @AfterSuite tearDownSuite");
    }

    @BeforeClass(alwaysRun = true)
    public static void setUpClass() throws Exception {
        System.out.println("SquareTest @BeforeClass setUpClass");
    }
```

The @BeforeSuite method gets first executed

If alwaysRun is set true, then this method guaranteed to run

BEFORE/AFTER GROUPS

- @BeforeGroups and @AfterGroups

Code

```
@BeforeGroups(groups = {"functional","area","perimeter"})
   public void setUpGroupsFunctional() {
      System.out.println("SquareTest @BeforeGroups setUpGroups functional");
      side = 2;   instance = new Square(side);
      expResultArea = 4.0;  expResultPerimeter = 8.0;
   }

   @AfterGroups(groups = {"functional","area","perimeter"})
   public void tearDownGroupsFunctional() {
      System.out.println("SquareTest @AfterGroups tearDownGroups functional");
      instance = null;
   }
```

This method is guaranteed to run before of any functional, area or parameter test methods

release any group-based object creation

TEST

- @Test

Code

```
@Test(groups = {"functional", "perimeter"})
   public void testPerimeter() {
      System.out.println("SquareTest perimeter");
      double result = instance.perimeter();
      assertEquals(expResultPerimeter, result, 0.0);
      System.out.println("Side: " + side + " expResultPerimeter: " + expResultPerimeter + " ActualResult: " + result);
   }
```

testPerimeter is part of the group ‘functional’ as well as ‘perimeter’

This sout is for information purpose only

ENABLED ATTRIBUTE

- @Test(enabled = false)

Code

```
//@Ignore
  @Test(enabled = false)
  public void testDraw() {
    System.out.println("SquareTest draw");
    Square instance = null;
    instance.draw();
    fail("The test case is a prototype.");
  }
```

testDraw method testing is disabled, so it will never get executed

This is equivalent of @Ignore of JUnit annotation

GROUPS WITH RUN

- Create a groups with run and include with name “perimeter”

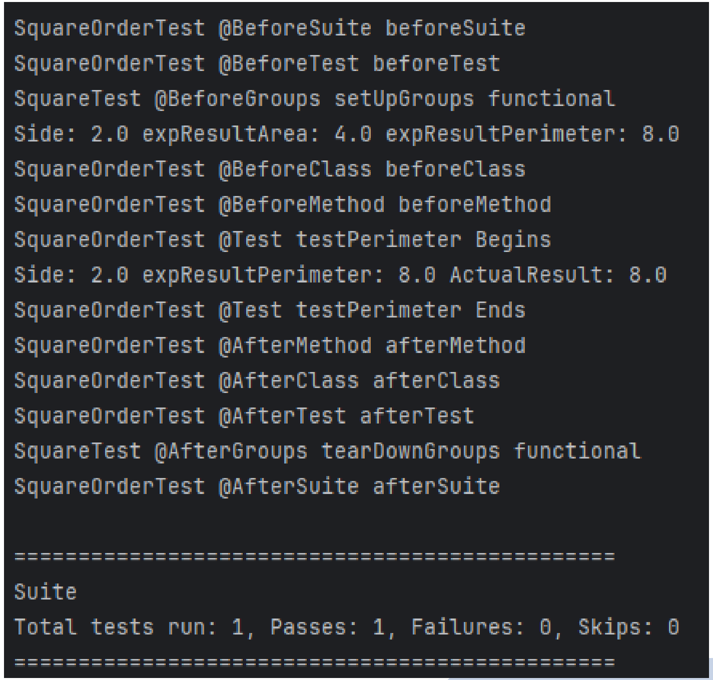
Code

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="5" verbose="1" name="ShapeSuite" annotations="JDK">
  <test name="Shape Test">
    <groups>
      <run>
        <include name=“perimeter"/>
      </run>
    </groups>
    <classes>
      <class name="Com.KavinSchool.Shape.SquareTest"/>
    </classes>
  </test>
</suite>
```

The methods which are under the group “perimeter” are only executed

PERIMETER GROUP RESULTS

- Only Perimeter related groups are executed



```
SquareOrderTest @BeforeSuite beforeSuite
SquareOrderTest @BeforeTest beforeTest
SquareTest @BeforeGroups setUpGroups functional
Side: 2.0 expResultArea: 4.0 expResultPerimeter: 8.0
SquareOrderTest @BeforeClass beforeClass
SquareOrderTest @BeforeMethod beforeMethod
SquareOrderTest @Test testPerimeter Begins
Side: 2.0 expResultPerimeter: 8.0 ActualResult: 8.0
SquareOrderTest @Test testPerimeter Ends
SquareOrderTest @AfterMethod afterMethod
SquareOrderTest @AfterClass afterClass
SquareOrderTest @AfterTest afterTest
SquareTest @AfterGroups tearDownGroups functional
SquareOrderTest @AfterSuite afterSuite


===============================================
Suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```

2

TESTNG ASSERTS

Learning Objectives

- ❑ Assert Methods
- ❑ Soft Assertions

ASSERT

Method	Description
assertEqualsNoOrder	Asserts that two arrays contain the same elements in no order
assertEquals	Asserts that two values are equal. For float and double when comparing allows for a delta difference, includes arrays comparison
assertTrue	Asserts that a condition is true. If it isn't it throws an AssertionError with the given message
assertFalse	Asserts that a condition is false (allows AssertionError message)

ASSERT

- The assertion methods are with parameter order of actualValue, expectedValue [, message]

Method	Description
assertSame	Asserts that two objects refer to the same object (allows for an AssertionError message)
fail	Fails a test with or without a message
assertNull	Asserts that an object is null (allows AssertionError message)
assertNotNull	Asserts that an object isn't null (allows AssertionError message)

ASSERTFILE

- Assertion tool for File centric assertions. The order is actualValue, expectedValue [, message]

Method	Description
assertDirectory	Assert that given value is a directory.
assertFile	Asserts that given value is a file
assertLength	Asserts that a given file of exactly expected characters or a directory of exactly expected entries
assertMinLength	Asserts that a given file of at least expected characters or a directory of at least expected entries.

ASSERTFILE

Method	Description
assertMaxLength	Asserts that a given file of at least expected characters or a directory of at most expected entries
assertReadable	Asserts that a given file is readable
assertWriteable	Asserts that a given file is writeable
assertReadWrite	Asserts that a given file is readable and writeable

SOFT ASSERTIONS

Hard Assert:

- When an assertion fails, immediately fail a test and stop execution of the rest of the test case

Soft Assert:

- When an assertion fails, the test case will not fail immediately but proceed to the rest of the test case.
- Helpful when you want to verify something like form field labels but want to proceed with ought failing the entire test case.

SOFT ASSERTIONS

Soft Assert:

- SoftAssert softAssert = new SoftAssert()
- softAssert.assertTrue(5>200);
- softAssert.assertEquals(5, 200);
- softAssert.assertAll();
- When an assertion fails, don't throw an exception but record the failure, calling assertAll() will cause an exception to be thrown if at least one assertion is failed.

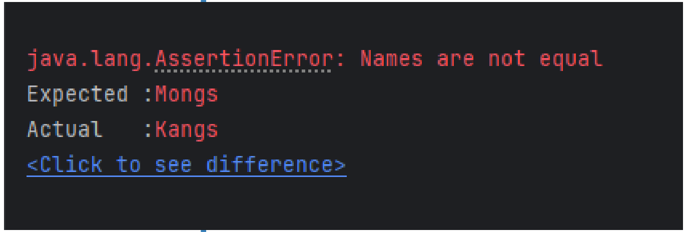
USAGE OF HARD ASSERT

- Assertion with hard example

Code

```
@Test
public void testHardAsserts() {
    Assertion hardAssert = new Assertion();
    hardAssert.assertEquals("Kangs", "Mongs", "Names are not equal");
    hardAssert.assertEquals(2, 5, "Numbers are not equal");
}
```

Generic Assertion Class



```
java.lang.AssertionError: Names are not equal
Expected :Mongs
Actual   :Kangs
<Click to see difference>
```

The second assertEquls never get executed the first one fails

USAGE OF SOFT ASSERT

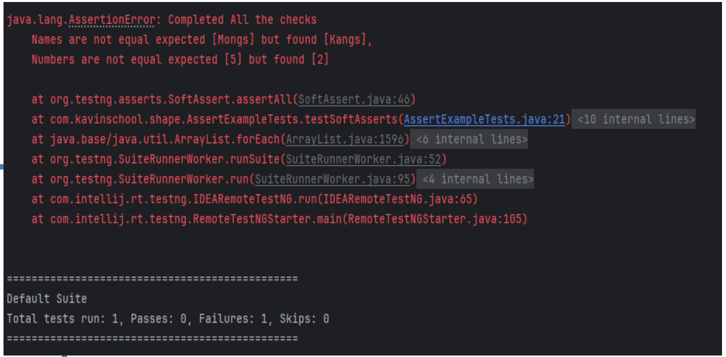
- Soft Assert example

SoftAssertion Class

Code

```
@Test
public void testSoftAsserts() {
    SoftAssert softAssert = new SoftAssert();
    softAssert.assertEquals("Kangs", "Mongs", "Names are not equal");
    softAssert.assertEquals(2, 5, "Numbers are not equal");
    softAssert.assertAll("Completed All the checks");
}
```

The second assertEquls will get executed even the first one fails



```
java.lang.AssertionError: Completed All the checks
    Names are not equal expected [Mongs] but found [Kangs],
    Numbers are not equal expected [5] but found [2]

    at org.testng.asserts.SoftAssert.assertAll(SoftAssert.java:46)
    at com.kavinschool.shape.AssertExampleTests.testSoftAsserts(AssertExampleTests.java:21) <10 internal lines>
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1596) <6 internal lines>
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95) <4 internal lines>
    at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:65)
    at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:105)


===============================================
Default Suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
===============================================
```

3

TESTNG EXCEPTION HANDLING

Learning Objectives

❑ expectedExpections

❑ Using Groups to run exceptions

EXPECTEDEXPECTIONS

- Usage of **expectedExpections** is shown below:

Code

```
@Test(groups = {"exceptions"}, expectedExceptions = ColorRangeException.class )
public void testSetColorName() throws ColorRangeException {
   System.out.println("CircleTest setColorName");
   int expResult = -CircleColor;
   instance.setColorName(-CircleColor);
   int result = instance.getColorName();
   assertEquals(expResult, result);
}
```

expectedExpections to handle ColorRangeException

TESTNG-CIRCLE.XML

- Include “Exceptions” group to run

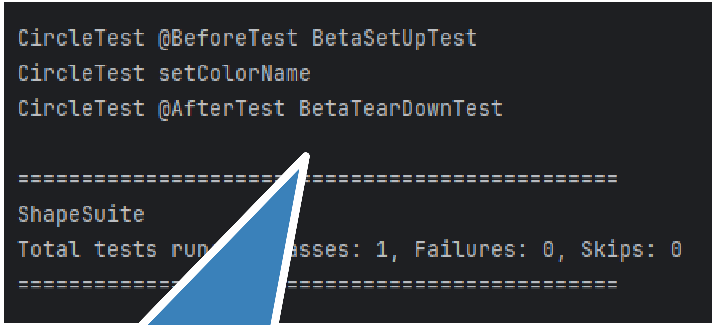
Code

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite thread-count="5" verbose="1" name="ShapeSuite" annotations="JDK">
   <test name="Shape Test">
      <groups>
         <run>
            <include name="exceptions"/>
            <include name="functional"/>
            <exclude name="non-functional"/>
         </run>
      </groups>
      <classes>
         <class name="Com.KavinSchool.Shape.CircleTest"/>
      </classes>
   </test></suite>
```

Include all the functional groups and exceptions

EXCEPTIONS HANDLING

- CircleTest handles the Exceptions for testSetColorName

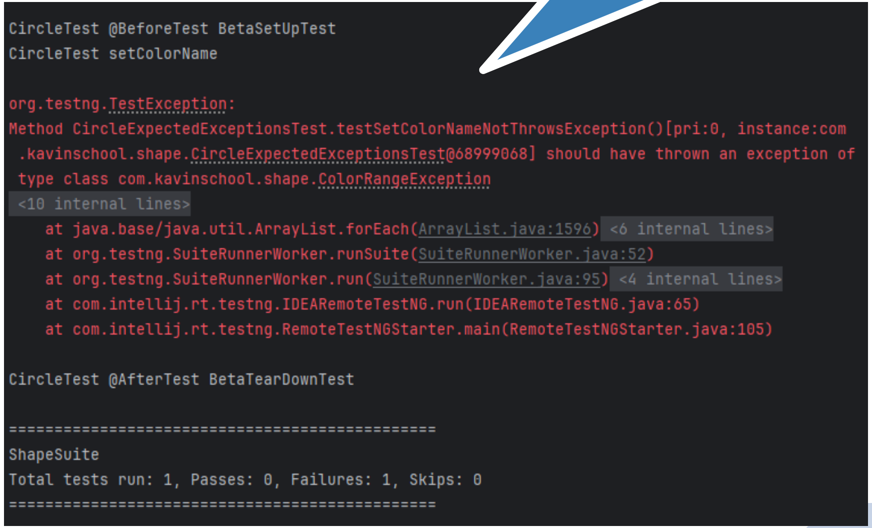


```
CircleTest @BeforeTest BetaSetUpTest
CircleTest setColorName
CircleTest @AfterTest BetaTearDownTest

====================      ==========================
ShapeSuite
Total tests run        asses: 1, Failures: 0, Skips: 0
============          ===========================
```

When expected exceptions are thrown

When expected exceptions are not thrown



```
CircleTest @BeforeTest BetaSetUpTest
CircleTest setColorName

org.testng.TestException:
Method CircleExpectedExceptionsTest.testSetColorNameNotThrowsException()[pri:0, instance:com
 .kavinschool.shape.CircleExpectedExceptionsTest@68999068] should have thrown an exception of
 type class com.kavinschool.shape.ColorRangeException
 <10 internal lines>
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1596) <6 internal lines>
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95) <4 internal lines>
    at com.intellij.rt.testng.IDEARemoteTestNG.run(IDEARemoteTestNG.java:65)
    at com.intellij.rt.testng.RemoteTestNGStarter.main(RemoteTestNGStarter.java:105)

CircleTest @AfterTest BetaTearDownTest

===============================================
ShapeSuite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
===============================================
```

4

TESTNG
SUITES OF SUITE

DEFINING SUITES OF SUITE

- Call many testng.xml files in one suite

Code

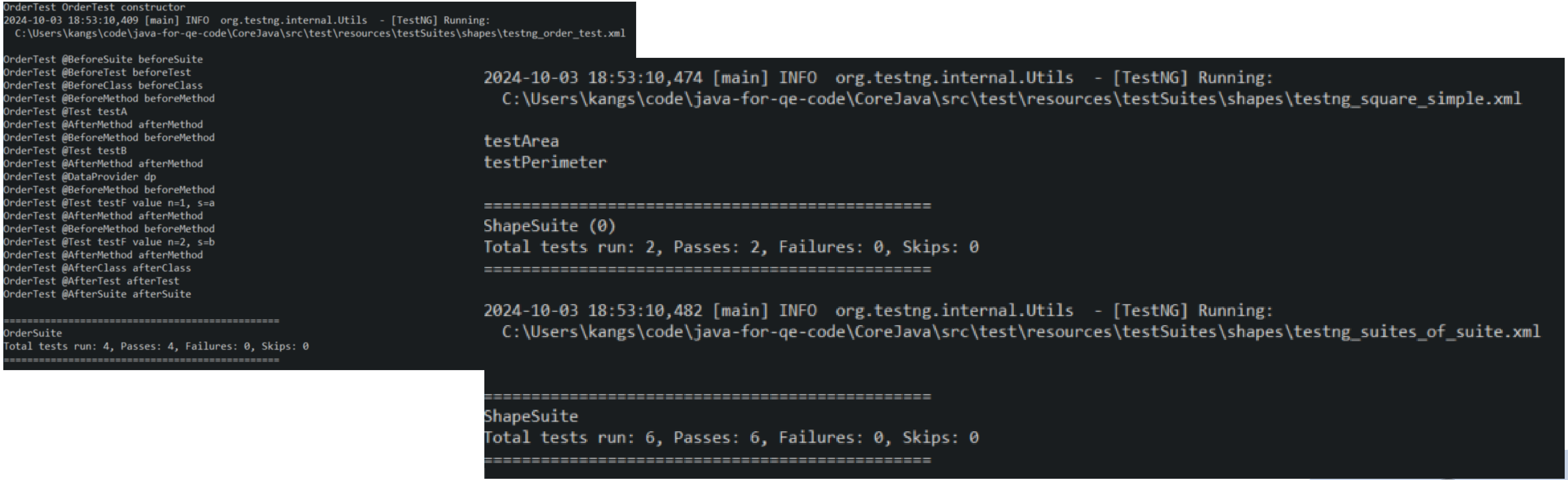
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.1.dtd" >
<suite thread-count="2" verbose="1" name="ShapeSuite" annotations="JDK"
parallel="classes">
<suite-files>
<suite-file path="./testng_order_test.xml" />
<suite-file path="./testng_square_simple.xml" />
</suite-files>
</suite>
```

The <suite-files> tag contains each testng file

Include failed.xml from test-output to re-run failed tests

DEFINE SUITE-FILES

- Each suite executed and the results are displayed



```
OrderTest OrderTest constructor
2024-10-03 18:53:10,409 [main] INFO  org.testng.internal.Utils  - [TestNG] Running:
  C:\Users\kangs\code\java-for-qe-code\CoreJava\src\test\resources\testSuites\shapes\testng_order_test.xml

OrderTest @BeforeSuite beforeSuite
OrderTest @BeforeTest beforeTest
OrderTest @BeforeClass beforeClass
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testA
OrderTest @AfterMethod afterMethod
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testB
OrderTest @AfterMethod afterMethod
OrderTest @DataProvider dp
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testF value n=1, s=a
OrderTest @AfterMethod afterMethod
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testF value n=2, s=b
OrderTest @AfterMethod afterMethod
OrderTest @AfterClass afterClass
OrderTest @AfterTest afterTest
OrderTest @AfterSuite afterSuite

===============================================
OrderSuite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
===============================================
```

```
2024-10-03 18:53:10,474 [main] INFO  org.testng.internal.Utils  - [TestNG] Running:
  C:\Users\kangs\code\java-for-qe-code\CoreJava\src\test\resources\testSuites\shapes\testng_square_simple.xml

testArea
testPerimeter

===============================================
ShapeSuite (0)
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================

2024-10-03 18:53:10,482 [main] INFO  org.testng.internal.Utils  - [TestNG] Running:
  C:\Users\kangs\code\java-for-qe-code\CoreJava\src\test\resources\testSuites\shapes\testng_suites_of_suite.xml


===============================================
ShapeSuite
Total tests run: 6, Passes: 6, Failures: 0, Skips: 0
===============================================
```

5

MAVEN WITH TESTING

MAVEN

- In Your POM file add Maven dependency

Code

```
<dependencies>
    <dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>7.10.2</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Dependencies will allow Maven to download the testng.jar file

MAVEN

- In Your POM file add Maven dependency

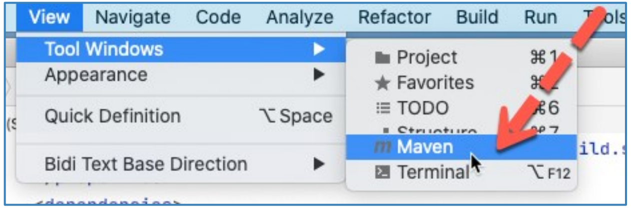
Code

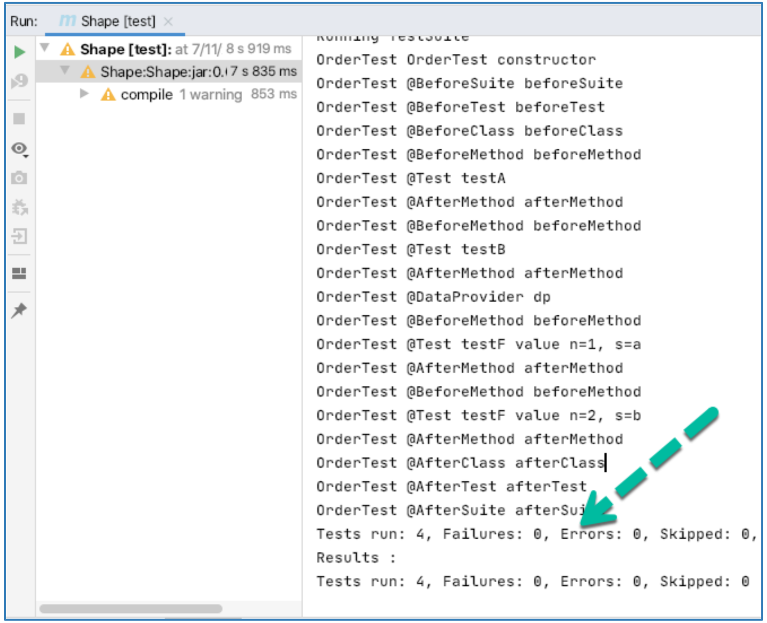
```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>${maven-surefire-plugin.version}</version>
            <configuration>
                <suiteXmlFiles>
                    <suiteXmlFile>control\testng_order_test.xml</suiteXmlFile>
                </suiteXmlFiles>
            </configuration>
        </plugin>
    </plugins>
</build>
```

List suite XML files with the relative path location

RUNNING MAVEN TEST IN INTELLIJ IDE

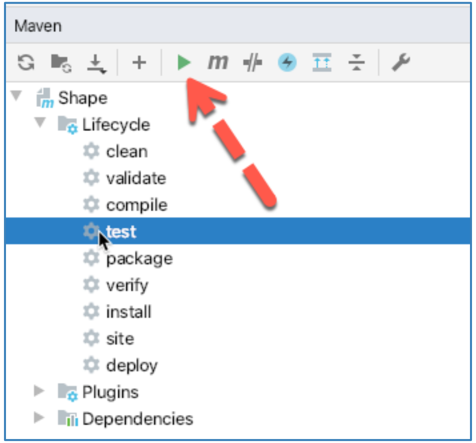
- In IntelliJ → View → Maven
- In the Maven → Select test → Click Run Button



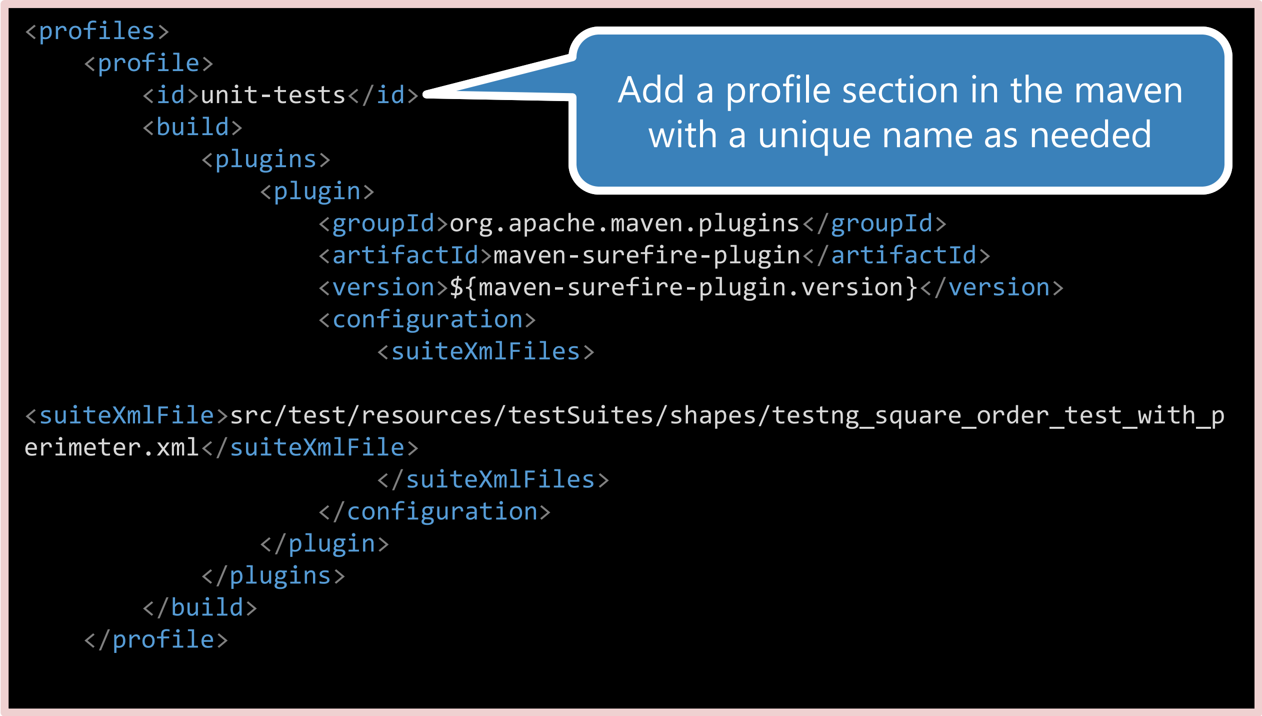

Run:
Shape [test]
Shape [test]: at 7/11/ 8 s 919 ms
Shape:Shape:jar:0.1 7 s 835 ms
compile 1 warning 853 ms

```
Running TestSuite
OrderTest OrderTest constructor
OrderTest @BeforeSuite beforeSuite
OrderTest @BeforeTest beforeTest
OrderTest @BeforeClass beforeClass
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testA
OrderTest @AfterMethod afterMethod
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testB
OrderTest @AfterMethod afterMethod
OrderTest @DataProvider dp
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testF value n=1, s=a
OrderTest @AfterMethod afterMethod
OrderTest @BeforeMethod beforeMethod
OrderTest @Test testF value n=2, s=b
OrderTest @AfterMethod afterMethod
OrderTest @AfterClass afterClass
OrderTest @AfterTest afterTest
OrderTest @AfterSuite afterSuite
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0,
Results :
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
```

Run As → Maven test



MAVEN – ADD A PROFILE



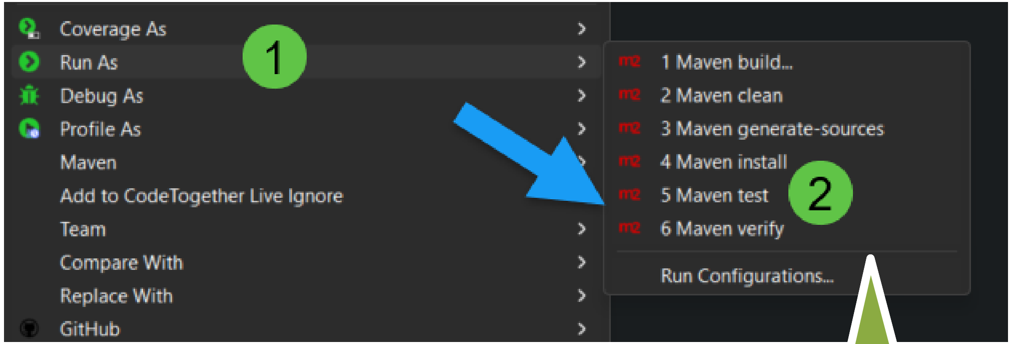
```
<profiles>
    <profile>
        <id>unit-tests</id>
        <build>
            <plugins>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-surefire-plugin</artifactId>
                    <version>${maven-surefire-plugin.version}</version>
                    <configuration>
                        <suiteXmlFiles>

<suiteXmlFile>src/test/resources/testSuites/shapes/testng_square_order_test_with_p
erimeter.xml</suiteXmlFile>
                        </suiteXmlFiles>
                    </configuration>
                </plugin>
            </plugins>
        </build>
    </profile>
```

Add a profile section in the maven with a unique name as needed

RUNNING MAVEN TEST IN ECLIPSE IDE

- Right click on pom.xml → Run As → Maven Test

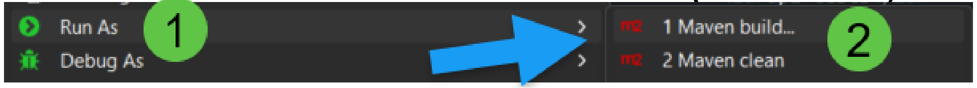


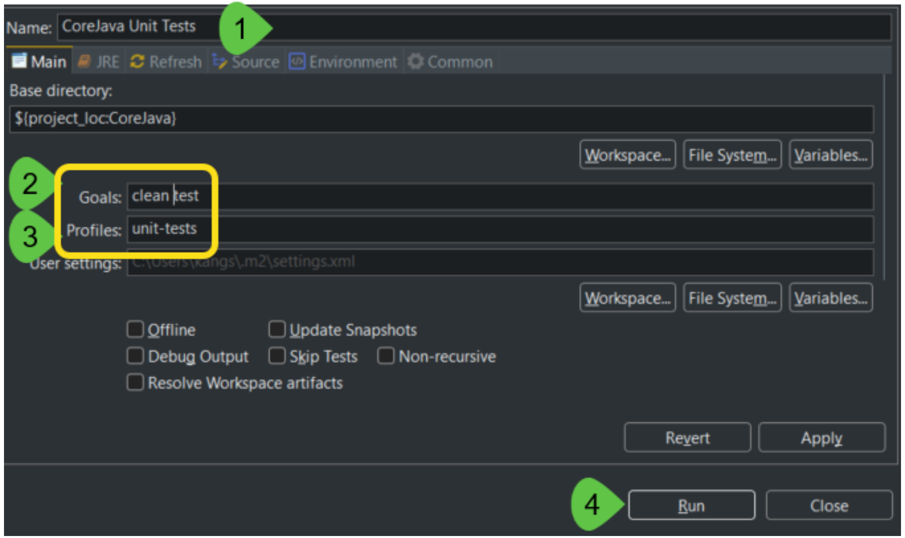
Run As → Maven test

RUNNING MAVEN TEST ACTIVATING PROFILE

In the console, you will see the output and outcome of the build

- Right-click on pom.xml → Run As → Maven build…
- Provide Goal and Profile name (unit-tests, Javadoc, etc.), then click the Run button





Name: CoreJava Unit Tests

Base directory: ${project_loc:CoreJava}

Goals: clean test

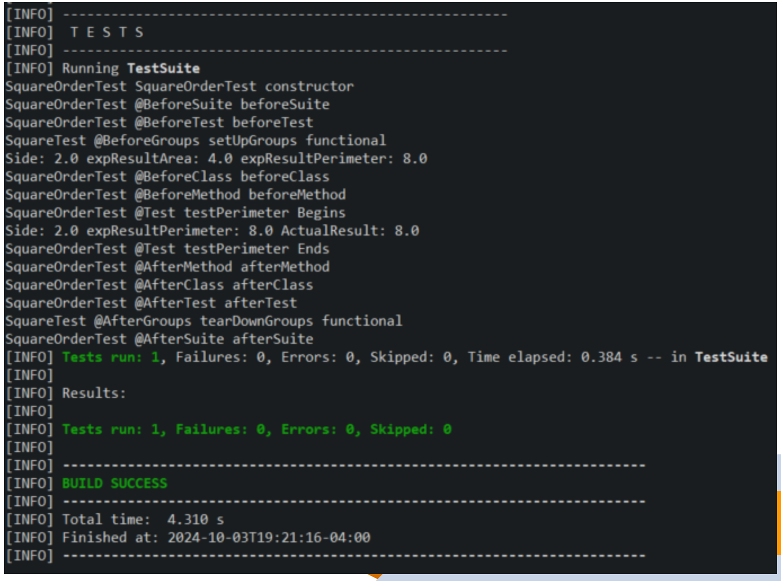
Profiles: unit-tests

User settings: C:\Users\rangs\.m2\settings.xml

☐ Offline ☐ Update Snapshots

☐ Debug Output ☐ Skip Tests ☐ Non-recursive

☐ Resolve Workspace artifacts



```
[INFO] ------------------------------------------------------
[INFO]  T E S T S
[INFO] ------------------------------------------------------
[INFO] Running TestSuite
SquareOrderTest SquareOrderTest constructor
SquareOrderTest @BeforeSuite beforeSuite
SquareOrderTest @BeforeTest beforeTest
SquareTest @BeforeGroups setUpGroups functional
Side: 2.0 expResultArea: 4.0 expResultPerimeter: 8.0
SquareOrderTest @BeforeClass beforeClass
SquareOrderTest @BeforeMethod beforeMethod
SquareOrderTest @Test testPerimeter Begins
Side: 2.0 expResultPerimeter: 8.0 ActualResult: 8.0
SquareOrderTest @Test testPerimeter Ends
SquareOrderTest @AfterMethod afterMethod
SquareOrderTest @AfterClass afterClass
SquareOrderTest @AfterTest afterTest
SquareTest @AfterGroups tearDownGroups functional
SquareOrderTest @AfterSuite afterSuite
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.384 s -- in TestSuite
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.310 s
[INFO] Finished at: 2024-10-03T19:21:16-04:00
[INFO] ------------------------------------------------------------------------
```

QUIZ

QUIZ#1

What is the correct way to disable a test in TestNG?

a. Using @Test(exclude = true) annotation

b. Using @Test(enabled = false) annotation

c. Using @Test(include = false) annotation

d. None of the above

QUIZ#1

What is the correct way to disable a test in TestNG?

a. Using @Test(exclude = true) annotation

b. Using @Test(enabled = false) annotation

c. Using @Test(include = false) annotation

d. None of the above

The correct answer is: b

Using @Test(enabled = false) annotation

Thank You

THANKS!

Your feedback is welcome
support@kavinschool.com