SCTR’s Pune Institute of Computer Technology
Dhankawadi, Pune

A MINI PROJECT REPORT ON
Testing on Calculator Application written in Java

Under the guidance of
Prof. Laxmi Pawar

DEPARTMENT OF COMPUTER ENGINEERING
ACADEMIC YEAR 2022-23

Mehul Oswal - 41444
Amit Purohit - 41450

Problem Statement:

Create a small application by selecting relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Report for the same

Objectives:

- Learn how to prepare test cases inclusive of test procedures for identified test scenarios
- Perform selective Black-box and White-box testing.
- Prepare Test Reports based on Test Pass/Fail Criteria

Outcomes: Students will be able to understand unit and integration testing with tool and test report

Hardware and Software Requirements: 8 GB Ram, x64 bit System, Windows 10, Eclipse IDE, JDK, TestNG

Theory:

Unit Testing: Unit Testing of software applications is done during the development (coding) of an application. The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure The goal of Unit Testing is to isolate each part of the program and show that the individual parts are correct. Unit Testing is usually performed by the developer.

Tools: There are several tools. Some are listed below:

1. Jtest: Parasoft Jtest is an IDE plugin that leverages open-source frameworks (Junit, Mockito, PowerMock, and Spring) with guided and easy one-click actions for creating, scaling, and maintaining unit tests. By automating these time-consuming aspects of unit testing, it frees the developer to focus on business logic and create more meaningful test suites.
2. Junit: Junit is a free to use testing tool used for Java programming language. It provides assertions to identify test method. This tool test data first and then inserted in the piece of code.
3. NUnit: NUnit is widely used unit-testing framework use for all .net languages. It is open-source tool which allows writing scripts manually. It supports data-driven tests which can run in parallel.
4. JMockit: JMockit is open-source Unit testing tool. It is code coverage tool with line and path metrics. It allows mocking API with recording and verification syntax. This tool offers Line coverage, Path Coverage, and Data Coverage.
5. EMMA: EMMA is an open-source toolkit for analysing and reporting code written in Java language. Emma support coverage types like method, line, basic block. It is Java-based so it is without external library dependencies and can access to the source code.

Bug Taxonomy:

Bug taxonomies help in providing fast and effective feedback so that they can easily identify possible reasons for failure of the software. Using bug taxonomy, a large number of potential bugs can be grouped into few categories. Whenever a new bug is reported, using bug taxonomy, a tester can easily analyse and put that bug into any of these categories. At the end of testing, Testers can understand the type of categories of bugs that frequently occurred and thereby in successive rounds of testing he can focus on writing more test cases that would help to detect such bugs. In addition, test leaders can guide their testers to focus on such frequently occurring bugs.

The summary of the Bug Taxonomy is given below:

- Requirements, Features, and Functionality Bugs
- Structural Bugs
- Data Bugs
- Coding Bugs
- Interface, Integration, and System Bugs
- Test and Test Design Bugs
- Testing and Design Style

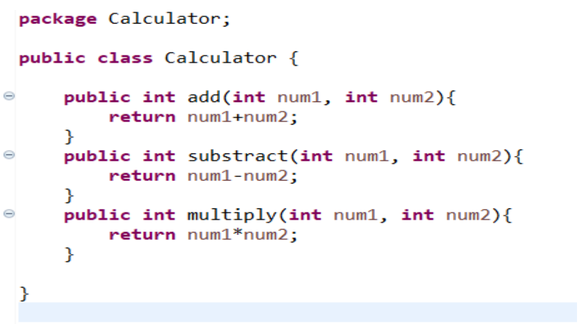
Integration Testing: In integration Testing, individual software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing Integration Test Case: Integration Test Case differs from other test cases in the sense it focuses mainly on the interfaces & flow of data/information between the modules. Here priority is to be given for the integrating links rather than the unit functions which are already tested.

Desktop Application Testing by using Junit Tool: JUnit is a framework for implementing testing in Java. It provides a simple way to explicitly test specific areas of a Java program, it is extensible and can be employed to test a hierarchy of program code either singularly or as multiple units. Using a testing framework is beneficial because it forces you to explicitly declare the expected results of specific program execution routes. When debugging it is possible to write a test which expresses the result you are trying to achieve and then debug until the test comes out positive. By having a set of tests that test all the core components of the project it is possible to modify specific areas of the project and immediately see the effect the modifications have on the other areas by the results of the test, hence, side-effects can be quickly realized. JUnit promotes the idea of first testing then coding, in that it is possible to setup test data for a unit which defines what the expected output is and then code until the tests pass. It is believed by some that this practice of "test a little, code a little, test a little, code a little..." increases programmer productivity and stability of program code whilst reducing programmer stress and the time spent debugging. JUnit is a simple open-source Java testing framework used to write and run repeatable automated tests. It is an instance of the xUnit architecture for unit testing framework. Eclipse supports creating test cases and running test suites, so it is easy to use for your Java applications.

JUnit features include:

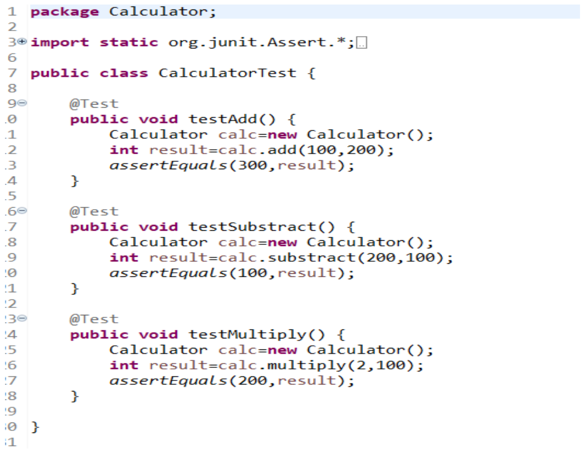
- Assertions for testing expected results
- Test fixtures for sharing common test data
- Test suites for easily organizing and running tests
- Graphical and textual test runners

Implementation:



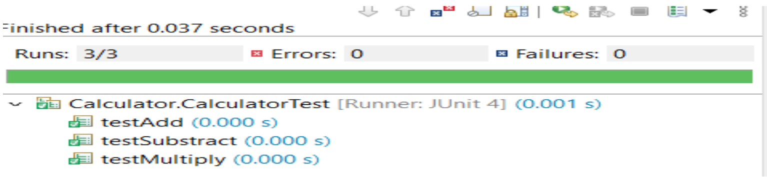
```java
package Calculator;

public class Calculator {

    public int add(int num1, int num2){
        return num1+num2;
    }
    public int substract(int num1, int num2){
        return num1-num2;
    }
    public int multiply(int num1, int num2){
        return num1*num2;
    }

}
```

Calculator.java



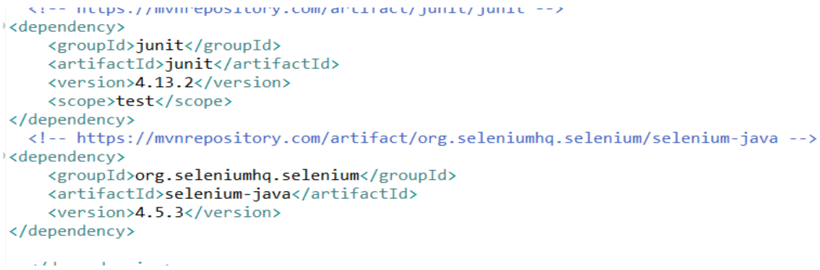
```java
package Calculator;

import static org.junit.Assert.*;

public class CalculatorTest {

    @Test
    public void testAdd() {
        Calculator calc=new Calculator();
        int result=calc.add(100,200);
        assertEquals(300,result);
    }

    @Test
    public void testSubstract() {
        Calculator calc=new Calculator();
        int result=calc.substract(200,100);
        assertEquals(100,result);
    }

    @Test
    public void testMultiply() {
        Calculator calc=new Calculator();
        int result=calc.multiply(2,100);
        assertEquals(200,result);
    }
}
```

CalculatorTest.java



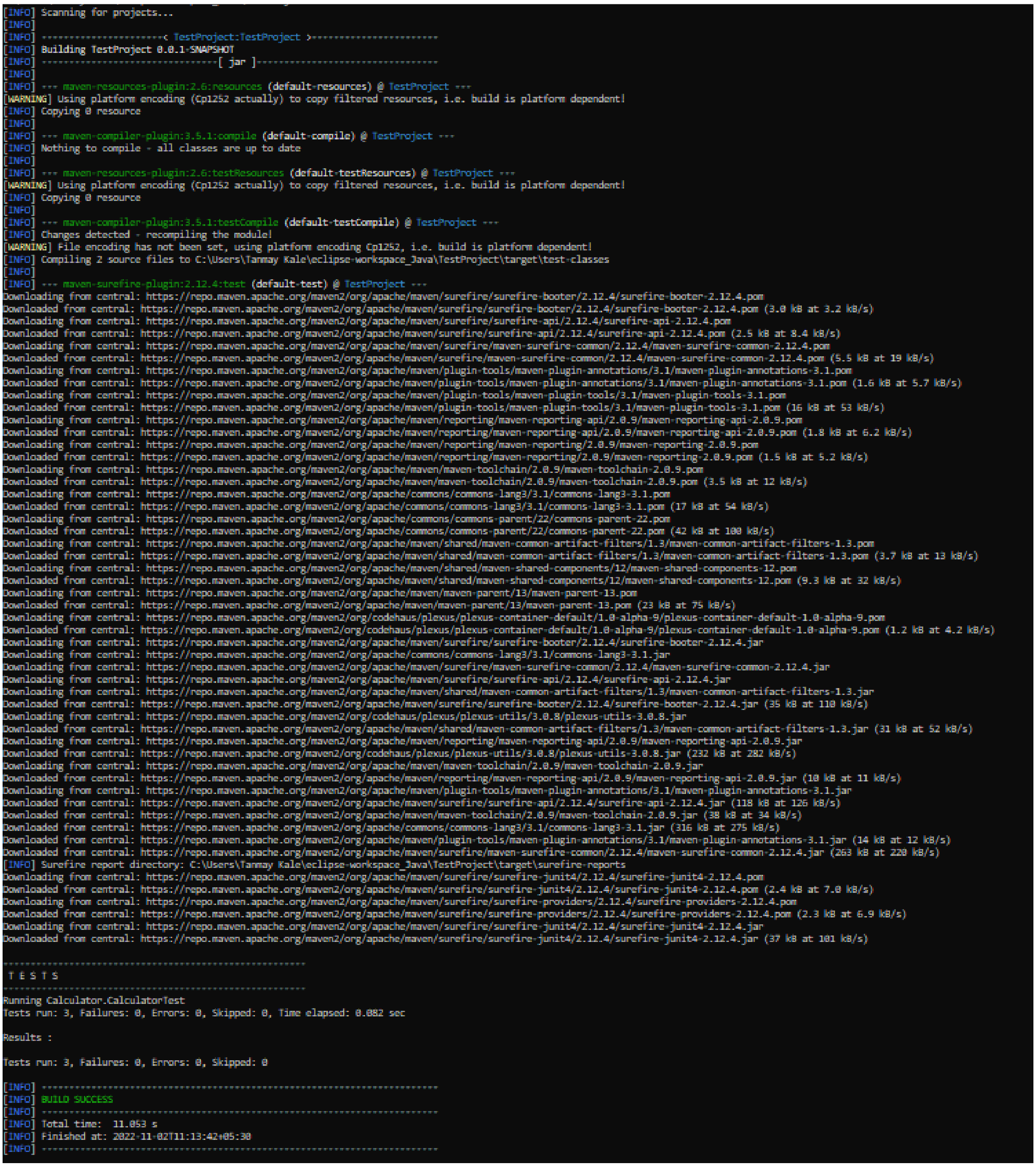
```
Finished after 0.037 seconds
Runs: 3/3        Errors: 0        Failures: 0

Calculator.CalculatorTest [Runner: JUnit 4] (0.001 s)
    testAdd (0.000 s)
    testSubstract (0.000 s)
    testMultiply (0.000 s)
```

Output



```xml
    <!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13.2</version>
    <scope>test</scope>
</dependency>
    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.5.3</version>
</dependency>
```

Including Dependency for Junit and Selenium in MavenProject



```
[INFO] Scanning for projects...
[INFO]
[INFO] -----------------------< TestProject:TestProject >-----------------------
[INFO] Building TestProject 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ TestProject ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ TestProject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ TestProject ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:testCompile (default-testCompile) @ TestProject ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 2 source files to C:\Users\Tanmay Kale\eclipse-workspace_Java\TestProject\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ TestProject ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-booter/2.12.4/surefire-booter-2.12.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-booter/2.12.4/surefire-booter-2.12.4.pom (3.0 kB at 3.2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-api/2.12.4/surefire-api-2.12.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-api/2.12.4/surefire-api-2.12.4.pom (2.5 kB at 8.4 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/maven-surefire-common/2.12.4/maven-surefire-common-2.12.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/maven-surefire-common/2.12.4/maven-surefire-common-2.12.4.pom (5.5 kB at 19 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugin-tools/maven-plugin-annotations/3.1/maven-plugin-annotations-3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugin-tools/maven-plugin-annotations/3.1/maven-plugin-annotations-3.1.pom (1.6 kB at 5.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugin-tools/maven-plugin-tools/3.1/maven-plugin-tools-3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugin-tools/maven-plugin-tools/3.1/maven-plugin-tools-3.1.pom (16 kB at 53 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/2.0.9/maven-reporting-api-2.0.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/2.0.9/maven-reporting-api-2.0.9.pom (1.8 kB at 6.2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting/2.0.9/maven-reporting-2.0.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting/2.0.9/maven-reporting-2.0.9.pom (1.5 kB at 5.2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-toolchain/2.0.9/maven-toolchain-2.0.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-toolchain/2.0.9/maven-toolchain-2.0.9.pom (3.5 kB at 12 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.1/commons-lang3-3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.1/commons-lang3-3.1.pom (17 kB at 54 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/22/commons-parent-22.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/22/commons-parent-22.pom (42 kB at 100 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-common-artifact-filters/1.3/maven-common-artifact-filters-1.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-common-artifact-filters/1.3/maven-common-artifact-filters-1.3.pom (3.7 kB at 13 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-components/12/maven-shared-components-12.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-components/12/maven-shared-components-12.pom (9.3 kB at 32 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/13/maven-parent-13.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/13/maven-parent-13.pom (23 kB at 75 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.pom (1.2 kB at 4.2 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-booter/2.12.4/surefire-booter-2.12.4.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.1/commons-lang3-3.1.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/maven-surefire-common/2.12.4/maven-surefire-common-2.12.4.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-api/2.12.4/surefire-api-2.12.4.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-common-artifact-filters/1.3/maven-common-artifact-filters-1.3.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-booter/2.12.4/surefire-booter-2.12.4.jar (35 kB at 110 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.8/plexus-utils-3.0.8.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-common-artifact-filters/1.3/maven-common-artifact-filters-1.3.jar (31 kB at 52 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/2.0.9/maven-reporting-api-2.0.9.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.8/plexus-utils-3.0.8.jar (232 kB at 282 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-toolchain/2.0.9/maven-toolchain-2.0.9.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/2.0.9/maven-reporting-api-2.0.9.jar (10 kB at 11 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugin-tools/maven-plugin-annotations/3.1/maven-plugin-annotations-3.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-api/2.12.4/surefire-api-2.12.4.jar (118 kB at 126 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-toolchain/2.0.9/maven-toolchain-2.0.9.jar (38 kB at 34 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.1/commons-lang3-3.1.jar (316 kB at 275 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugin-tools/maven-plugin-annotations/3.1/maven-plugin-annotations-3.1.jar (14 kB at 12 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/maven-surefire-common/2.12.4/maven-surefire-common-2.12.4.jar (263 kB at 220 kB/s)
[INFO] Surefire report directory: C:\Users\Tanmay Kale\eclipse-workspace_Java\TestProject\target\surefire-reports
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.pom (2.4 kB at 7.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-providers/2.12.4/surefire-providers-2.12.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-providers/2.12.4/surefire-providers-2.12.4.pom (2.3 kB at 6.9 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.12.4/surefire-junit4-2.12.4.jar (37 kB at 101 kB/s)

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running Calculator.CalculatorTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.082 sec

Results :

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  11.053 s
[INFO] Finished at: 2022-11-02T11:13:42+05:30
[INFO] ------------------------------------------------------------------------
```

Maven Automation Test report

Conclusion: Hence we have successfully implemented Unit Testing and generated Test Report of same using JUnit and Maven Automation tool for Calculator Application