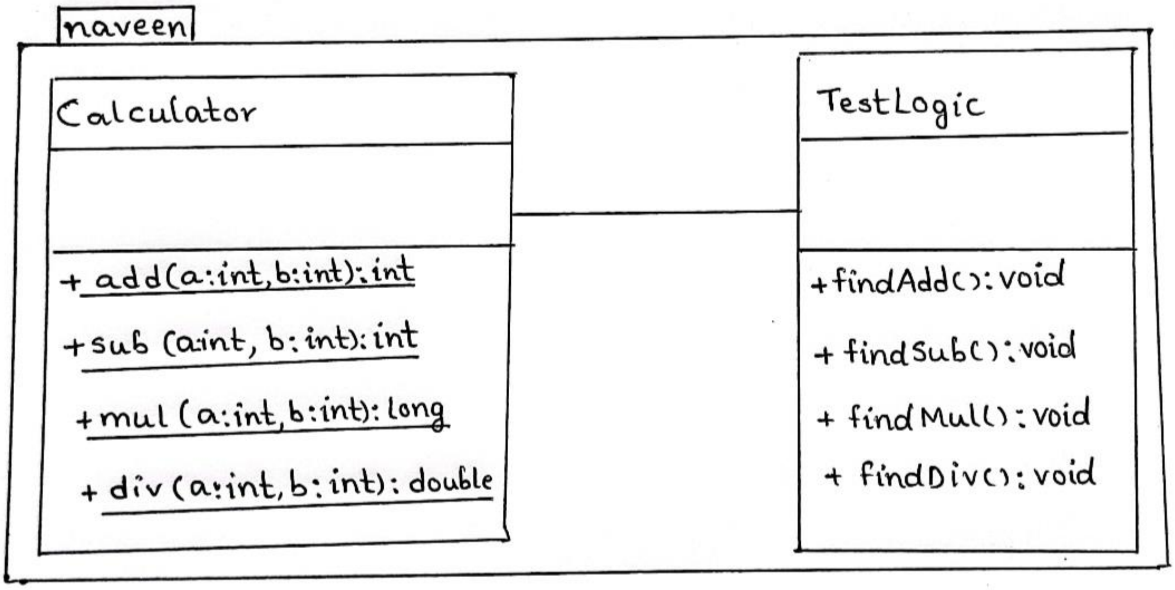
SKILLING EXERCISE-7

Name: Badisa Naveen

Reg.no: 2000031509

Class diagram:



About Junit:

Junit is a “Unit Testing” framework for Java Applications which is already included by default in android studio. It is an automation framework for Unit as well as UI Testing. It contains annotations such as **@Test, @Before**, **@After**, etc.

There are two ways to perform unit testing: 1) manual testing 2) automated testing.

1) **Manual Testing**: If you execute the test cases manually without any tool support, it is known as manual testing. It is time consuming and less reliable
2) **Automated Testing**: If you execute the test cases by tool support, it is known as automated testing. It is fast and more reliable.

Assert:

Assert is a method useful in determining Pass or Fail status of a test case, The assert methods are provided by the class org.junit.Assert which extends java.lang.Object class.

There are various types of assertions like Boolean, Null, Identical etc.

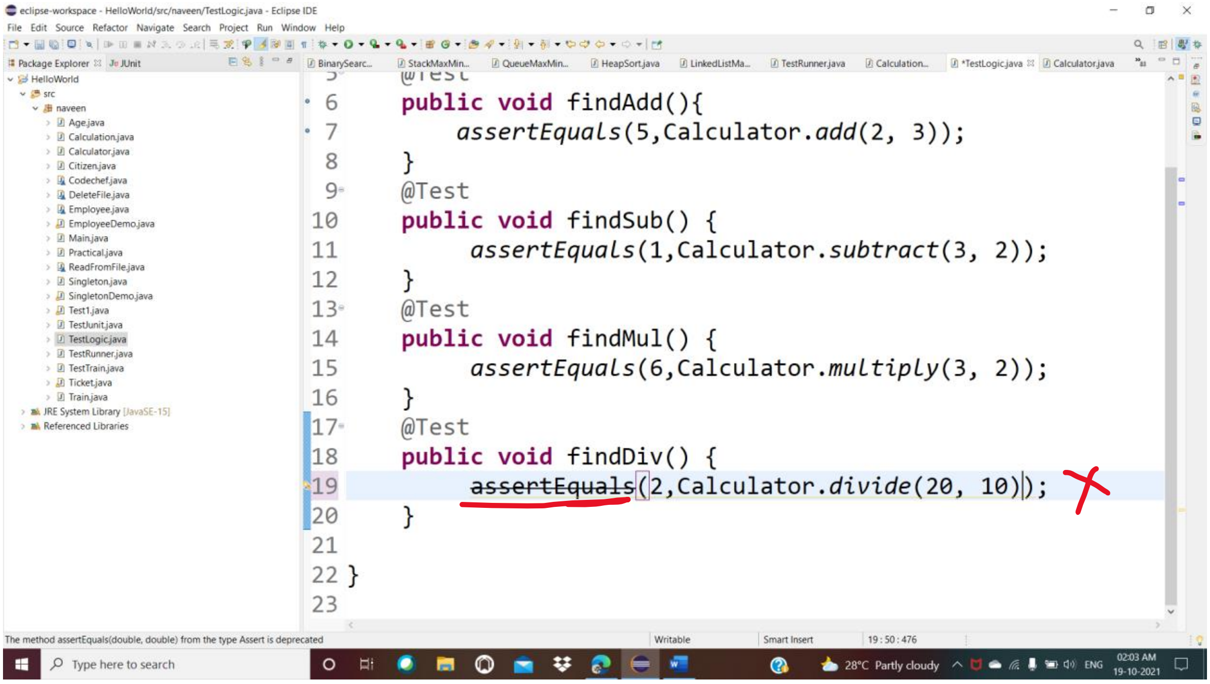
Code:

```
package naveen;
public class Calculator {
public static int add(int a, int b) {
return a + b;
}

public static int subtract(int a, int b) {
return a - b;
}

public static long multiply(int a, int b) {
return a * b;
}

public static double divide(int a, int b) {
double result;
if (b == 0) {
throw new IllegalArgumentException("Divisor cannot divide by zero");
} else {
result = Double.valueOf(a)/Double.valueOf(b);
}
return result;
}
}
```

```
package naveen;
import static org.junit.Assert.*;
import org.junit.Test;
public class TestLogic {
   @Test
   public void findAdd(){
      assertEquals(5,Calculator.add(2, 3));
         }
```

```
    @Test
    public void findSub() {
        assertEquals(1,Calculator.subtract(3, 2));
    }
    @Test
    public void findMul() {
        assertEquals(6,Calculator.multiply(3, 2));
    }
    @Test
    public void findDiv() {
        assertEquals(2,Calculator.divide(20, 10),1);
    }

}
```

Screenshots:



eclipse-workspace - HelloWorld/src/naveen/TestLogic.java - Eclipse IDE

```
    @Test
    public void findAdd(){
        assertEquals(5,Calculator.add(2, 3));
    }
    @Test
    public void findSub() {
        assertEquals(1,Calculator.subtract(3, 2));
    }
    @Test
    public void findMul() {
        assertEquals(6,Calculator.multiply(3, 2));
    }
    @Test
    public void findDiv() {
        assertEquals(2,Calculator.divide(20, 10));
    }

}
```

The method assertEquals(double, double) from the type Assert is deprecated

Note:

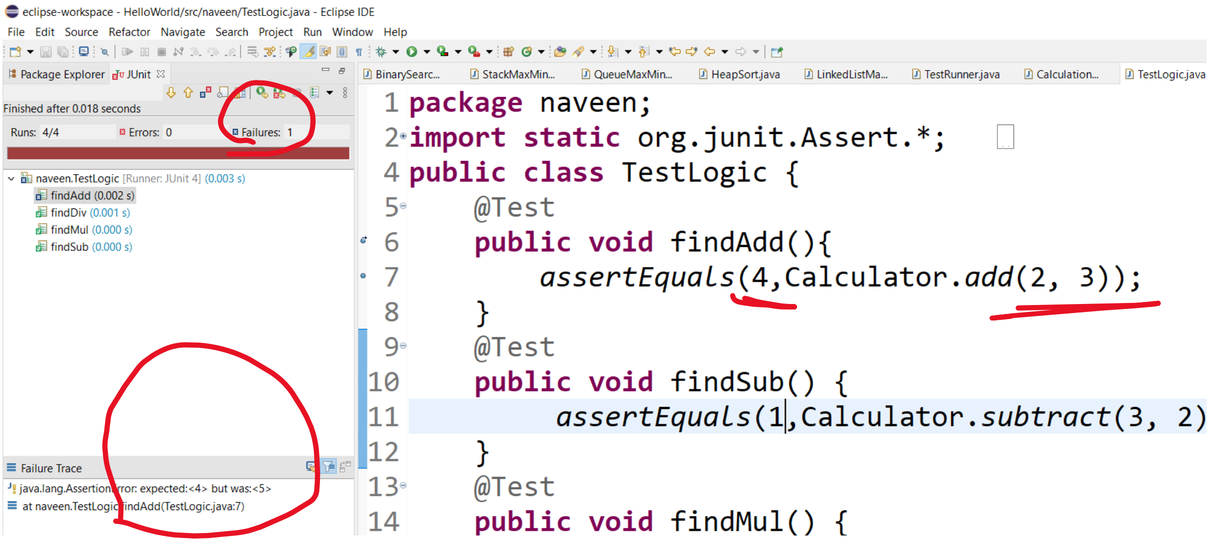
assertEquals(double expected, double actual) Instead of this one , we have to use

Deprecated. *Use* *assertEquals(double expected, double actual, double epsilon)*.

Here epsilon is random value which is smaller than expected and actual value.

Taking one value expected value:

Here findAdd() method is error because by adding 2+3=5 but I given expected value is 4. So it is showing error.



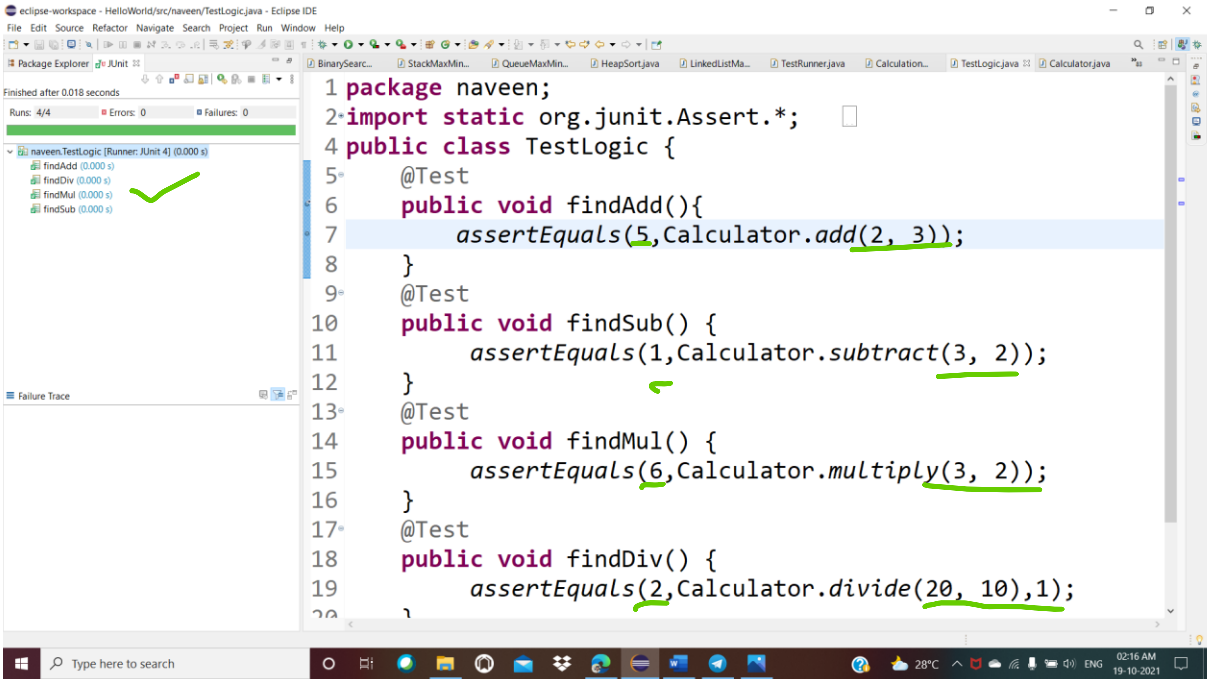
eclipse-workspace - HelloWorld/src/naveen/TestLogic.java - Eclipse IDE

```
Finished after 0.018 seconds
Runs: 4/4    Errors: 0    Failures: 1

naveen.TestLogic [Runner: JUnit 4] (0.003 s)
    findAdd (0.002 s)
    findDiv (0.001 s)
    findMul (0.000 s)
    findSub (0.000 s)
```

```
java.lang.AssertionError: expected:<4> but was:<5>
at naveen.TestLogic.findAdd(TestLogic.java:7)
```

```java
package naveen;
import static org.junit.Assert.*;
public class TestLogic {
    @Test
    public void findAdd(){
        assertEquals(4,Calculator.add(2, 3));
    }
    @Test
    public void findSub() {
        assertEquals(1,Calculator.subtract(3, 2)
    }
    @Test
    public void findMul() {
```

Taking all values correctly:



eclipse-workspace - HelloWorld/src/naveen/TestLogic.java - Eclipse IDE

```
Finished after 0.018 seconds
Runs: 4/4    Errors: 0    Failures: 0

naveen.TestLogic [Runner: JUnit 4] (0.000 s)
    findAdd (0.000 s)
    findDiv (0.000 s)
    findMul (0.000 s)
    findSub (0.000 s)
```

```java
package naveen;
import static org.junit.Assert.*;
public class TestLogic {
    @Test
    public void findAdd(){
        assertEquals(5,Calculator.add(2, 3));
    }
    @Test
    public void findSub() {
        assertEquals(1,Calculator.subtract(3, 2));
    }
    @Test
    public void findMul() {
        assertEquals(6,Calculator.multiply(3, 2));
    }
    @Test
    public void findDiv() {
        assertEquals(2,Calculator.divide(20, 10),1);
    }
```