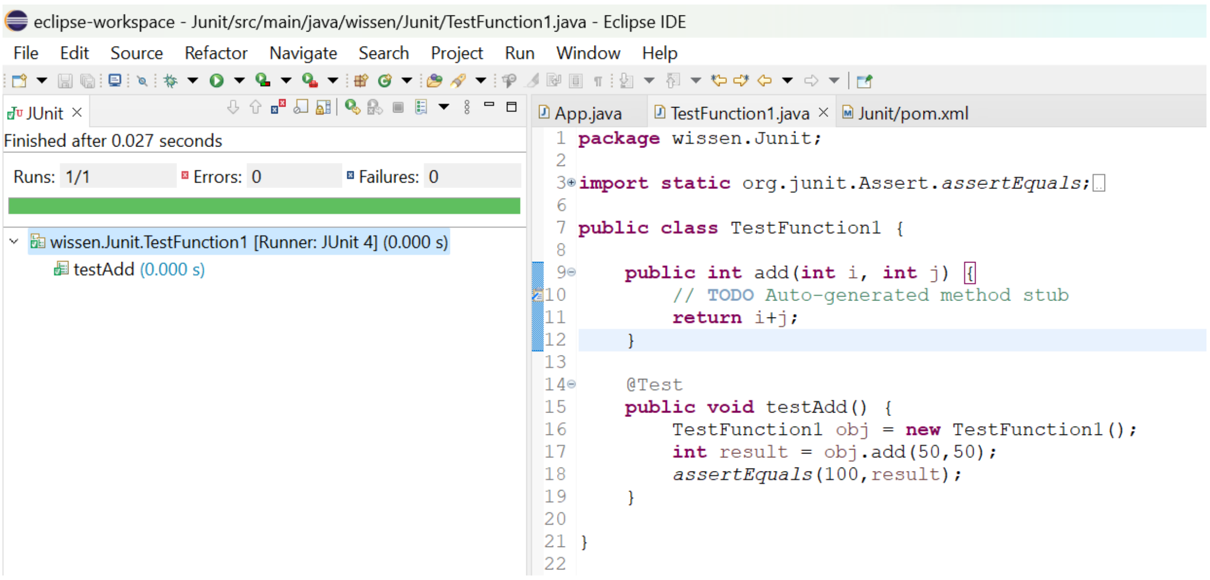
Junit

Writing a function with JUnit.



eclipse-workspace - Junit/src/main/java/wissen/Junit/TestFunction1.java - Eclipse IDE

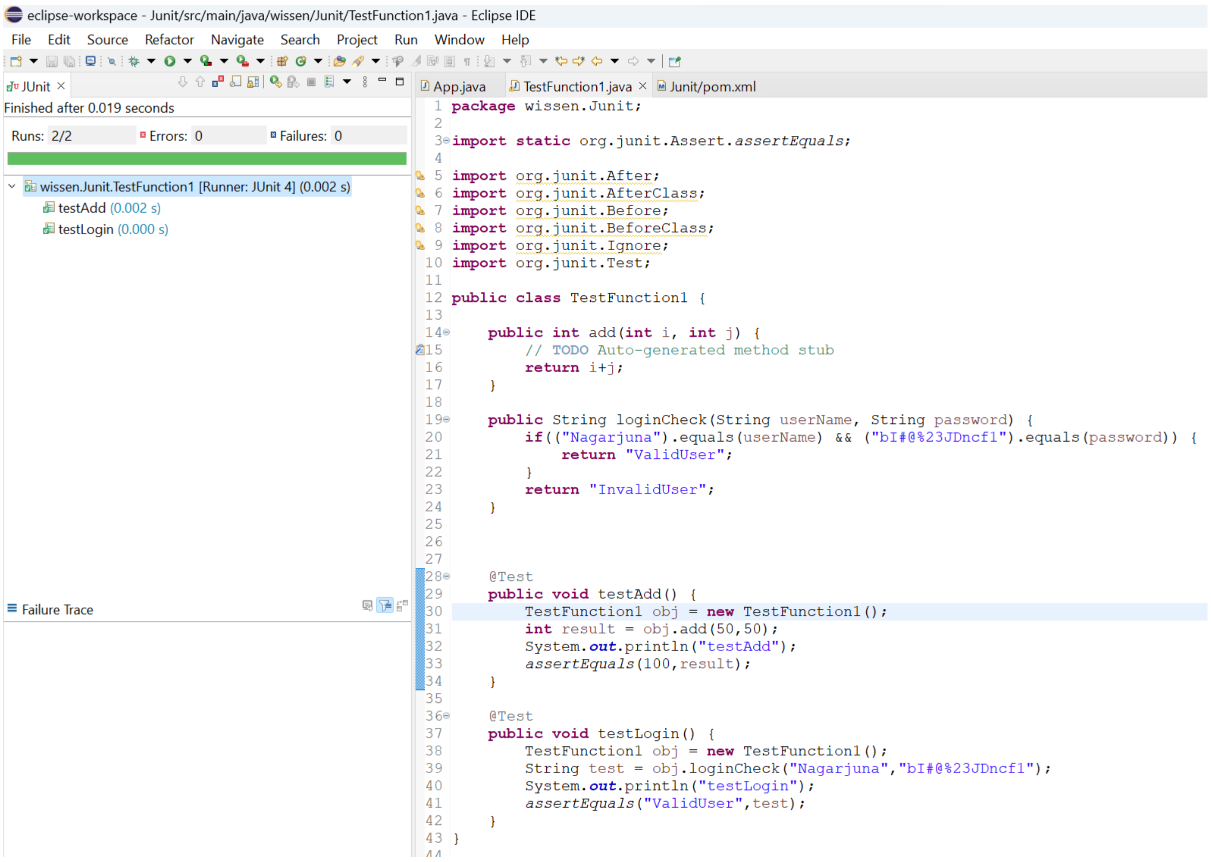
Finished after 0.027 seconds

Runs: 1/1   Errors: 0   Failures: 0

```
wissen.Junit.TestFunction1 [Runner: JUnit 4] (0.000 s)
    testAdd (0.000 s)
```

```java
package wissen.Junit;

import static org.junit.Assert.assertEquals;

public class TestFunction1 {

    public int add(int i, int j) {
        // TODO Auto-generated method stub
        return i+j;
    }

    @Test
    public void testAdd() {
        TestFunction1 obj = new TestFunction1();
        int result = obj.add(50,50);
        assertEquals(100,result);
    }

}
```

TDD Approach for login.



eclipse-workspace - Junit/src/main/java/wissen/Junit/TestFunction1.java - Eclipse IDE

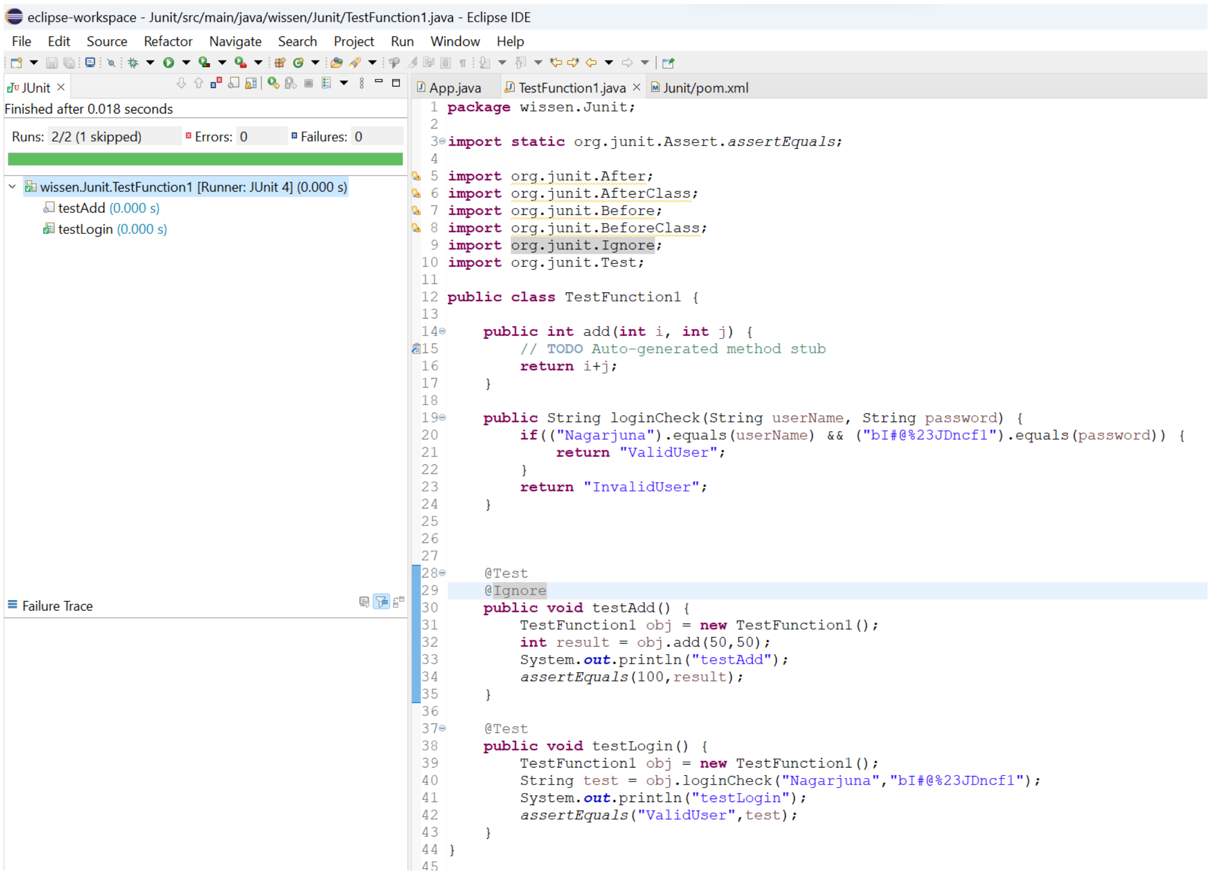
Finished after 0.019 seconds

Runs: 2/2   Errors: 0   Failures: 0

```
wissen.Junit.TestFunction1 [Runner: JUnit 4] (0.002 s)
    testAdd (0.002 s)
    testLogin (0.000 s)
```

```java
package wissen.Junit;

import static org.junit.Assert.assertEquals;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;

public class TestFunction1 {

    public int add(int i, int j) {
        // TODO Auto-generated method stub
        return i+j;
    }

    public String loginCheck(String userName, String password) {
        if(("Nagarjuna").equals(userName) && ("bI#@%23JDncf1").equals(password)) {
            return "ValidUser";
        }
        return "InvalidUser";
    }



    @Test
    public void testAdd() {
        TestFunction1 obj = new TestFunction1();
        int result = obj.add(50,50);
        System.out.println("testAdd");
        assertEquals(100,result);
    }

    @Test
    public void testLogin() {
        TestFunction1 obj = new TestFunction1();
        String test = obj.loginCheck("Nagarjuna","bI#@%23JDncf1");
        System.out.println("testLogin");
        assertEquals("ValidUser",test);
    }
}
```

Ignoring a test case

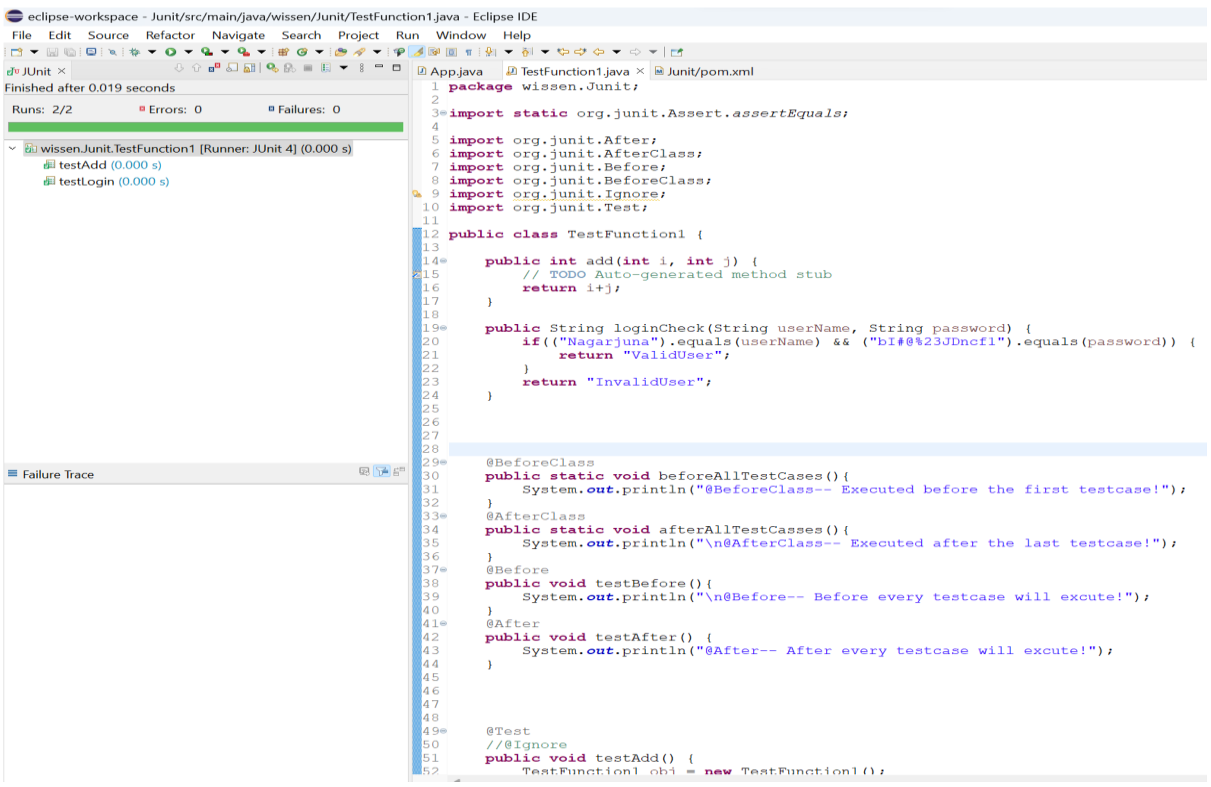


eclipse-workspace - Junit/src/main/java/wissen/Junit/TestFunction1.java - Eclipse IDE

```
JUnit
Finished after 0.018 seconds
Runs: 2/2 (1 skipped)    Errors: 0    Failures: 0

wissen.Junit.TestFunction1 [Runner: JUnit 4] (0.000 s)
    testAdd (0.000 s)
    testLogin (0.000 s)

Failure Trace
```

```java
package wissen.Junit;

import static org.junit.Assert.assertEquals;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;

public class TestFunction1 {

    public int add(int i, int j) {
        // TODO Auto-generated method stub
        return i+j;
    }

    public String loginCheck(String userName, String password) {
        if(("Nagarjuna").equals(userName) && ("bI#@%23JDncf1").equals(password)) {
            return "ValidUser";
        }
        return "InvalidUser";
    }



    @Test
    @Ignore
    public void testAdd() {
        TestFunction1 obj = new TestFunction1();
        int result = obj.add(50,50);
        System.out.println("testAdd");
        assertEquals(100,result);
    }

    @Test
    public void testLogin() {
        TestFunction1 obj = new TestFunction1();
        String test = obj.loginCheck("Nagarjuna","bI#@%23JDncf1");
        System.out.println("testLogin");
        assertEquals("ValidUser",test);
    }
}
```

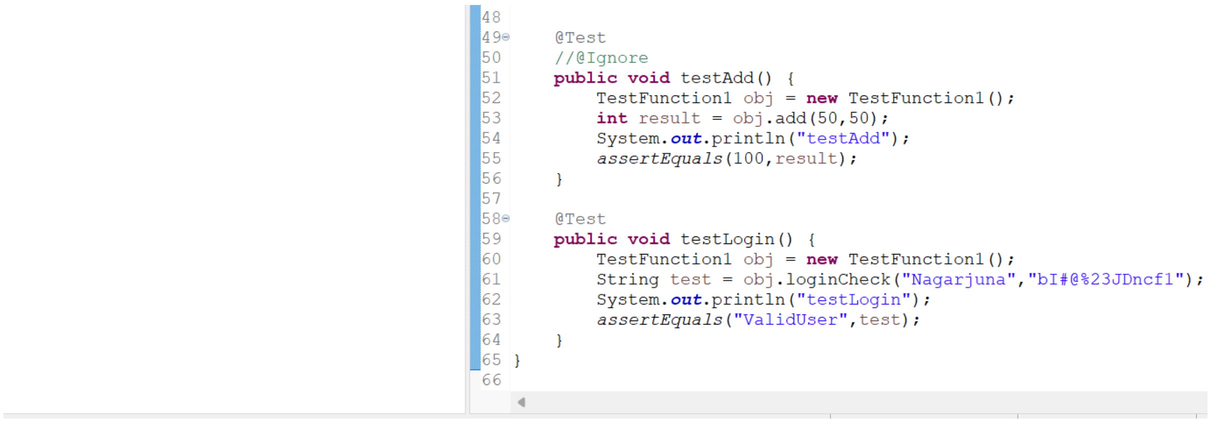
Setup and Teardown functions.



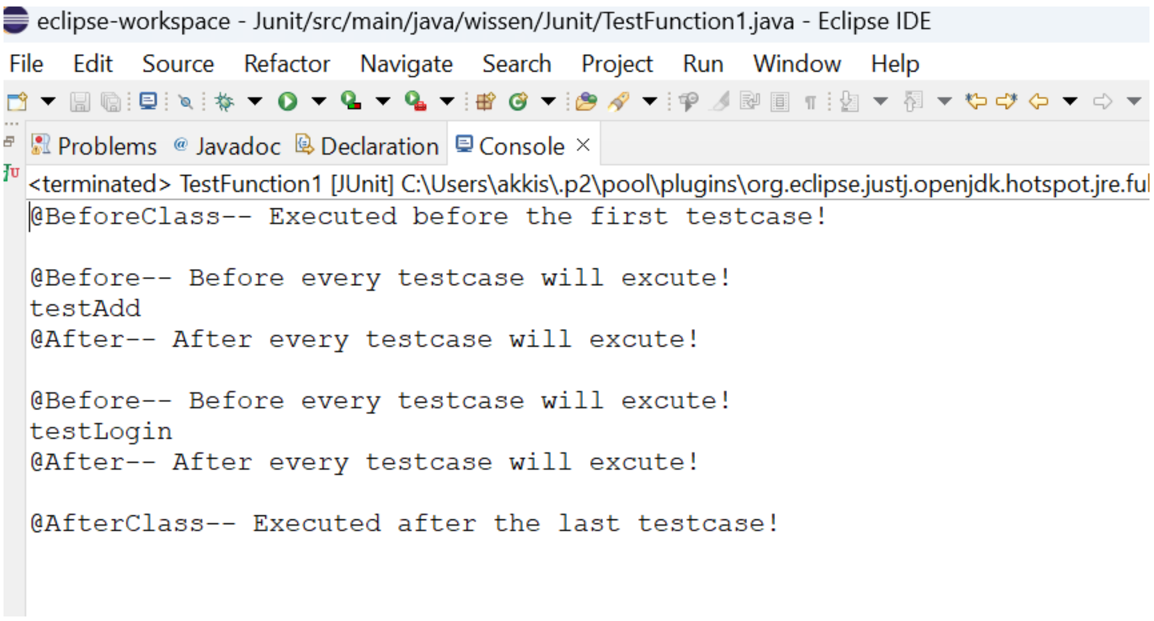
eclipse-workspace - Junit/src/main/java/wissen/Junit/TestFunction1.java - Eclipse IDE

```
JUnit
Finished after 0.019 seconds
Runs: 2/2    Errors: 0    Failures: 0

wissen.Junit.TestFunction1 [Runner: JUnit 4] (0.000 s)
    testAdd (0.000 s)
    testLogin (0.000 s)

Failure Trace
```

```java
package wissen.Junit;

import static org.junit.Assert.assertEquals;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;

public class TestFunction1 {

    public int add(int i, int j) {
        // TODO Auto-generated method stub
        return i+j;
    }

    public String loginCheck(String userName, String password) {
        if(("Nagarjuna").equals(userName) && ("bI#@%23JDncf1").equals(password)) {
            return "ValidUser";
        }
        return "InvalidUser";
    }




    @BeforeClass
    public static void beforeAllTestCases(){
        System.out.println("@BeforeClass-- Executed before the first testcase!");
    }
    @AfterClass
    public static void afterAllTestCasses(){
        System.out.println("\n@AfterClass-- Executed after the last testcase!");
    }
    @Before
    public void testBefore(){
        System.out.println("\n@Before-- Before every testcase will excute!");
    }
    @After
    public void testAfter() {
        System.out.println("@After-- After every testcase will excute!");
    }




    @Test
    //@Ignore
    public void testAdd() {
        TestFunction1 obj = new TestFunction1();
```



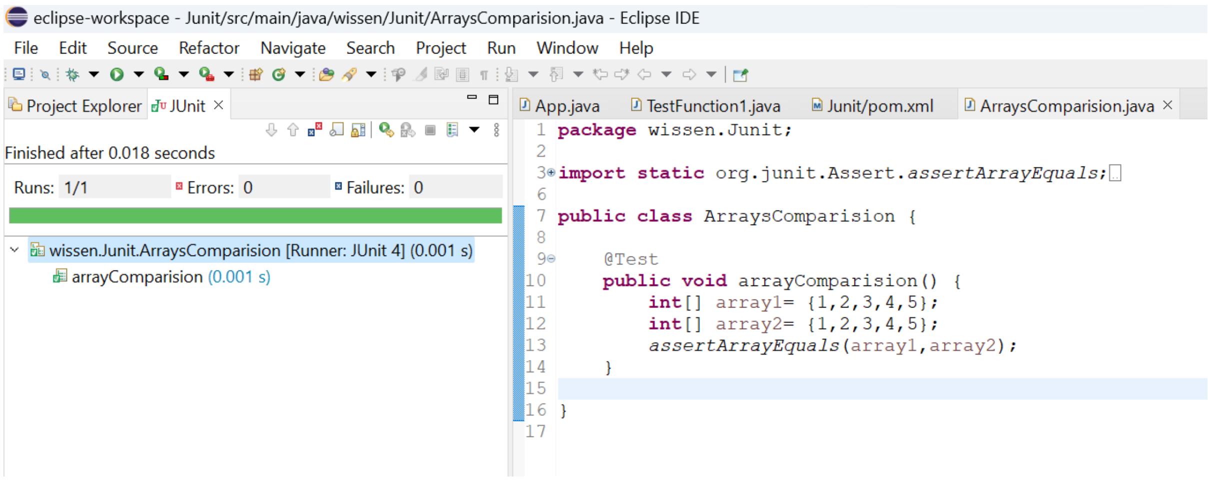
```
    @Test
    //@Ignore
    public void testAdd() {
        TestFunction1 obj = new TestFunction1();
        int result = obj.add(50,50);
        System.out.println("testAdd");
        assertEquals(100,result);
    }

    @Test
    public void testLogin() {
        TestFunction1 obj = new TestFunction1();
        String test = obj.loginCheck("Nagarjuna","bI#@%23JDncf1");
        System.out.println("testLogin");
        assertEquals("ValidUser",test);
    }
}
```



eclipse-workspace - Junit/src/main/java/wissen/Junit/TestFunction1.java - Eclipse IDE

```
<terminated> TestFunction1 [JUnit] C:\Users\akkis\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.ful
@BeforeClass-- Executed before the first testcase!

@Before-- Before every testcase will excute!
testAdd
@After-- After every testcase will excute!

@Before-- Before every testcase will excute!
testLogin
@After-- After every testcase will excute!

@AfterClass-- Executed after the last testcase!
```

Array comparison



eclipse-workspace - Junit/src/main/java/wissen/Junit/ArraysComparision.java - Eclipse IDE

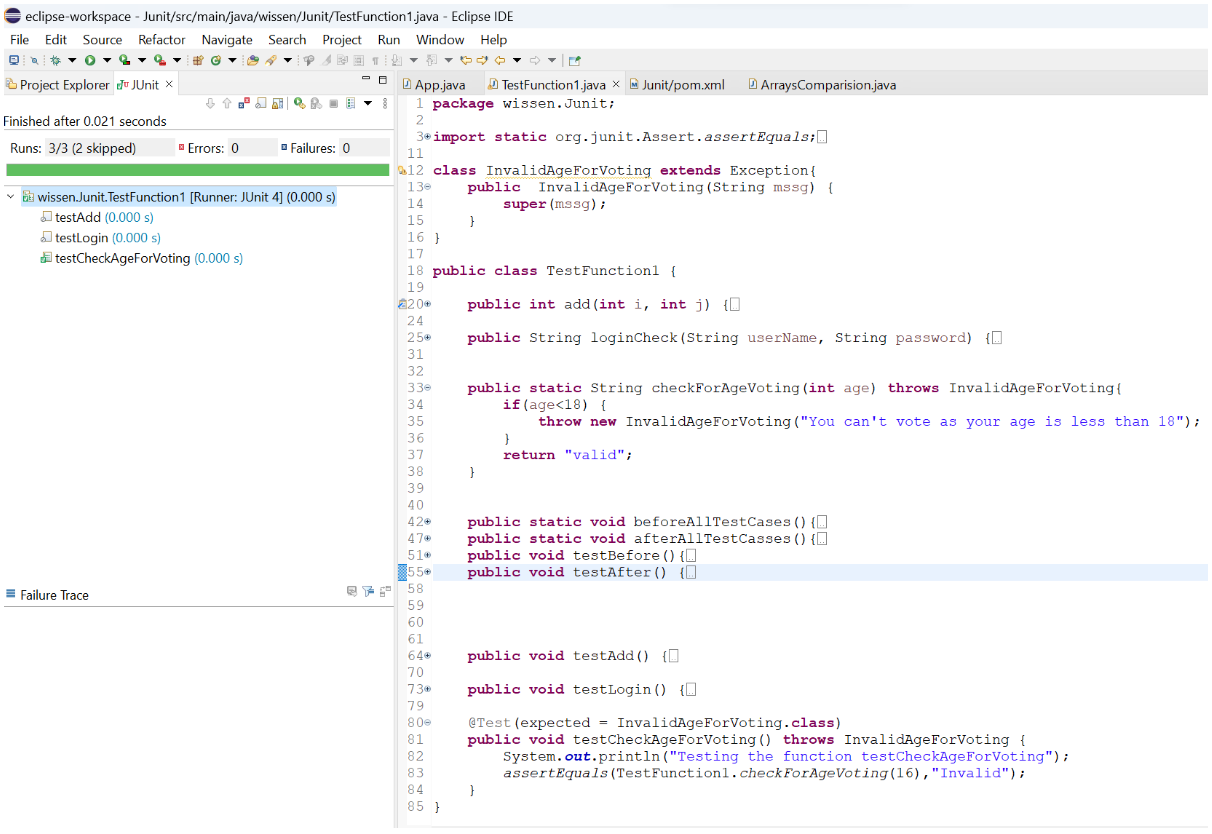
Finished after 0.018 seconds

Runs: 1/1   Errors: 0   Failures: 0

wissen.Junit.ArraysComparision [Runner: JUnit 4] (0.001 s)
    arrayComparision (0.001 s)

```
package wissen.Junit;

import static org.junit.Assert.assertArrayEquals;

public class ArraysComparision {

    @Test
    public void arrayComparision() {
        int[] array1= {1,2,3,4,5};
        int[] array2= {1,2,3,4,5};
        assertArrayEquals(array1,array2);
    }

}
```

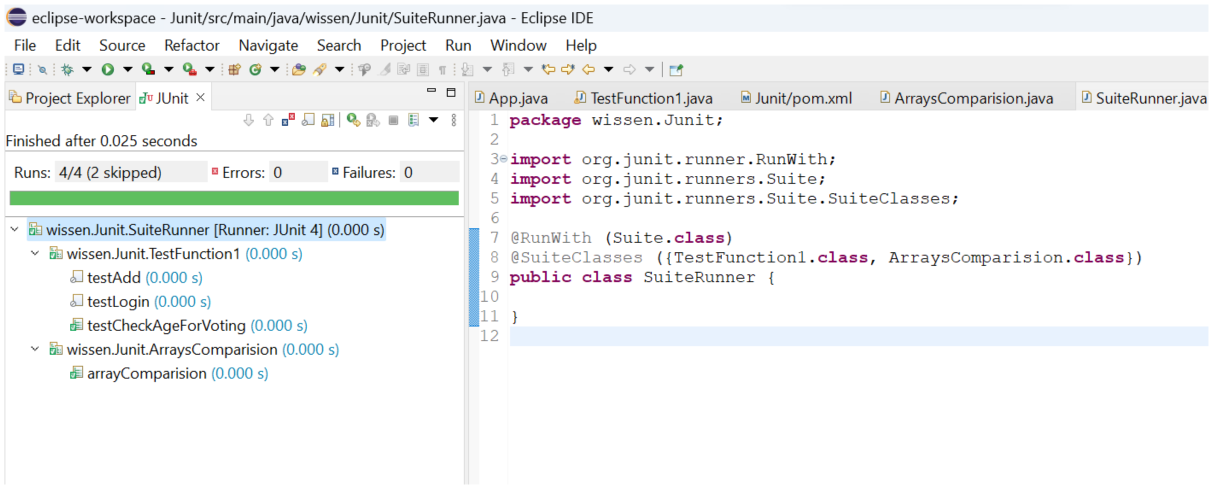
Checking exceptions in JUnit


eclipse-workspace - Junit/src/main/java/wissen/Junit/TestFunction1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
JUnit
App.java
TestFunction1.java
Junit/pom.xml
ArraysComparision.java
Failure Trace

```
Finished after 0.021 seconds
Runs: 3/3 (2 skipped)    Errors: 0    Failures: 0

wissen.Junit.TestFunction1 [Runner: JUnit 4] (0.000 s)
    testAdd (0.000 s)
    testLogin (0.000 s)
    testCheckAgeForVoting (0.000 s)
```

```java
package wissen.Junit;

import static org.junit.Assert.assertEquals;

class InvalidAgeForVoting extends Exception{
    public  InvalidAgeForVoting(String mssg) {
        super(mssg);
    }
}

public class TestFunction1 {

    public int add(int i, int j) {

    public String loginCheck(String userName, String password) {


    public static String checkForAgeVoting(int age) throws InvalidAgeForVoting{
        if(age<18) {
            throw new InvalidAgeForVoting("You can't vote as your age is less than 18");
        }
        return "valid";
    }



    public static void beforeAllTestCases(){
    public static void afterAllTestCasses(){
    public void testBefore(){
    public void testAfter() {




    public void testAdd() {

    public void testLogin() {

    @Test(expected = InvalidAgeForVoting.class)
    public void testCheckAgeForVoting() throws InvalidAgeForVoting {
        System.out.println("Testing the function testCheckAgeForVoting");
        assertEquals(TestFunction1.checkForAgeVoting(16),"Invalid");
    }
}
```

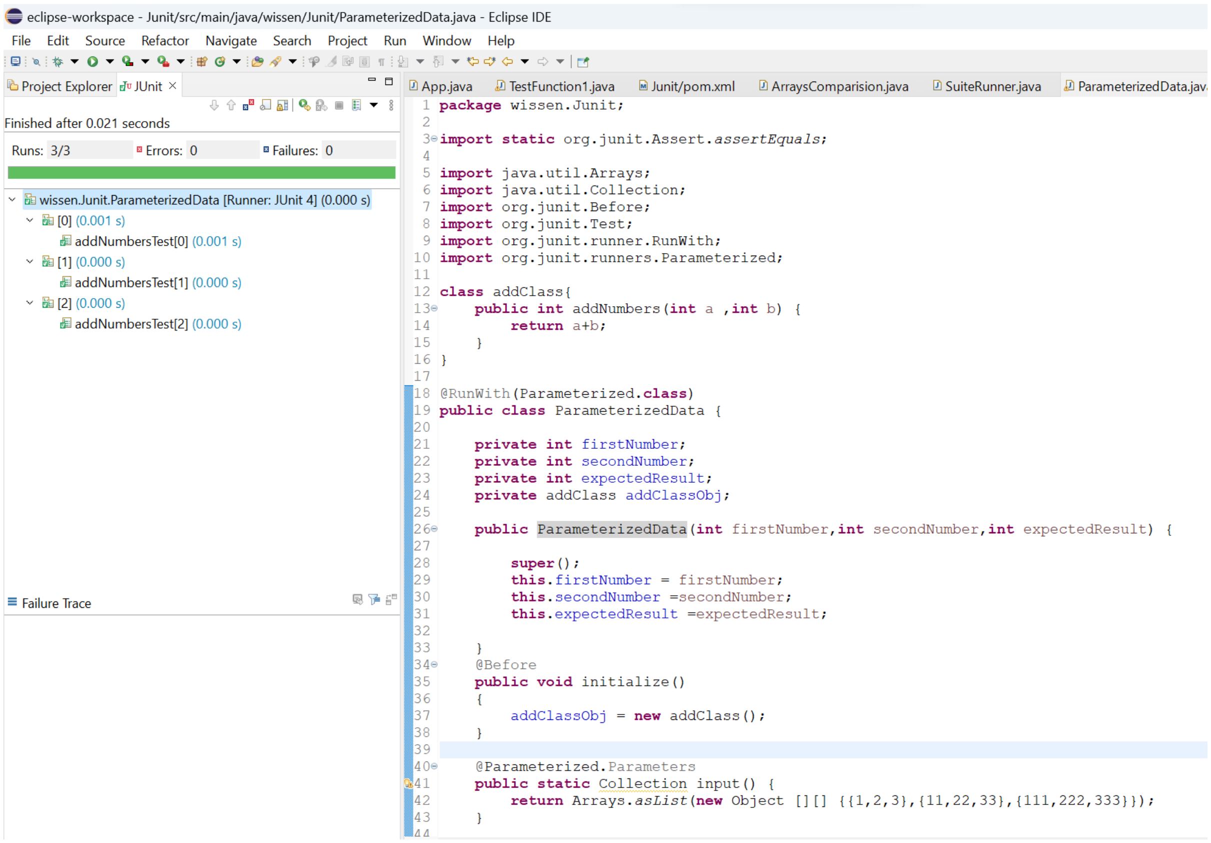
Suite Runner


eclipse-workspace - Junit/src/main/java/wissen/Junit/SuiteRunner.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
JUnit
App.java
TestFunction1.java
Junit/pom.xml
ArraysComparision.java
SuiteRunner.java

```
Finished after 0.025 seconds
Runs: 4/4 (2 skipped)    Errors: 0    Failures: 0

wissen.Junit.SuiteRunner [Runner: JUnit 4] (0.000 s)
    wissen.Junit.TestFunction1 (0.000 s)
        testAdd (0.000 s)
        testLogin (0.000 s)
        testCheckAgeForVoting (0.000 s)
    wissen.Junit.ArraysComparision (0.000 s)
        arrayComparision (0.000 s)
```

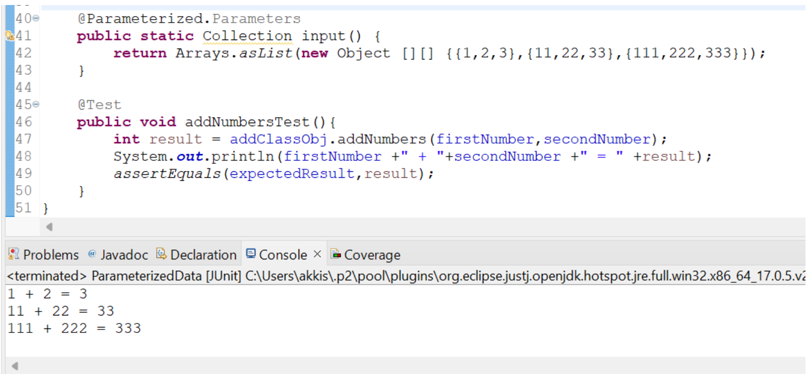
```java
package wissen.Junit;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
import org.junit.runners.Suite.SuiteClasses;

@RunWith (Suite.class)
@SuiteClasses ({TestFunction1.class, ArraysComparision.class})
public class SuiteRunner {

}

```

Parameterized Data


eclipse-workspace - Junit/src/main/java/wissen/Junit/ParameterizedData.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
JUnit
App.java
TestFunction1.java
Junit/pom.xml
ArraysComparision.java
SuiteRunner.java
ParameterizedData.jav
Failure Trace

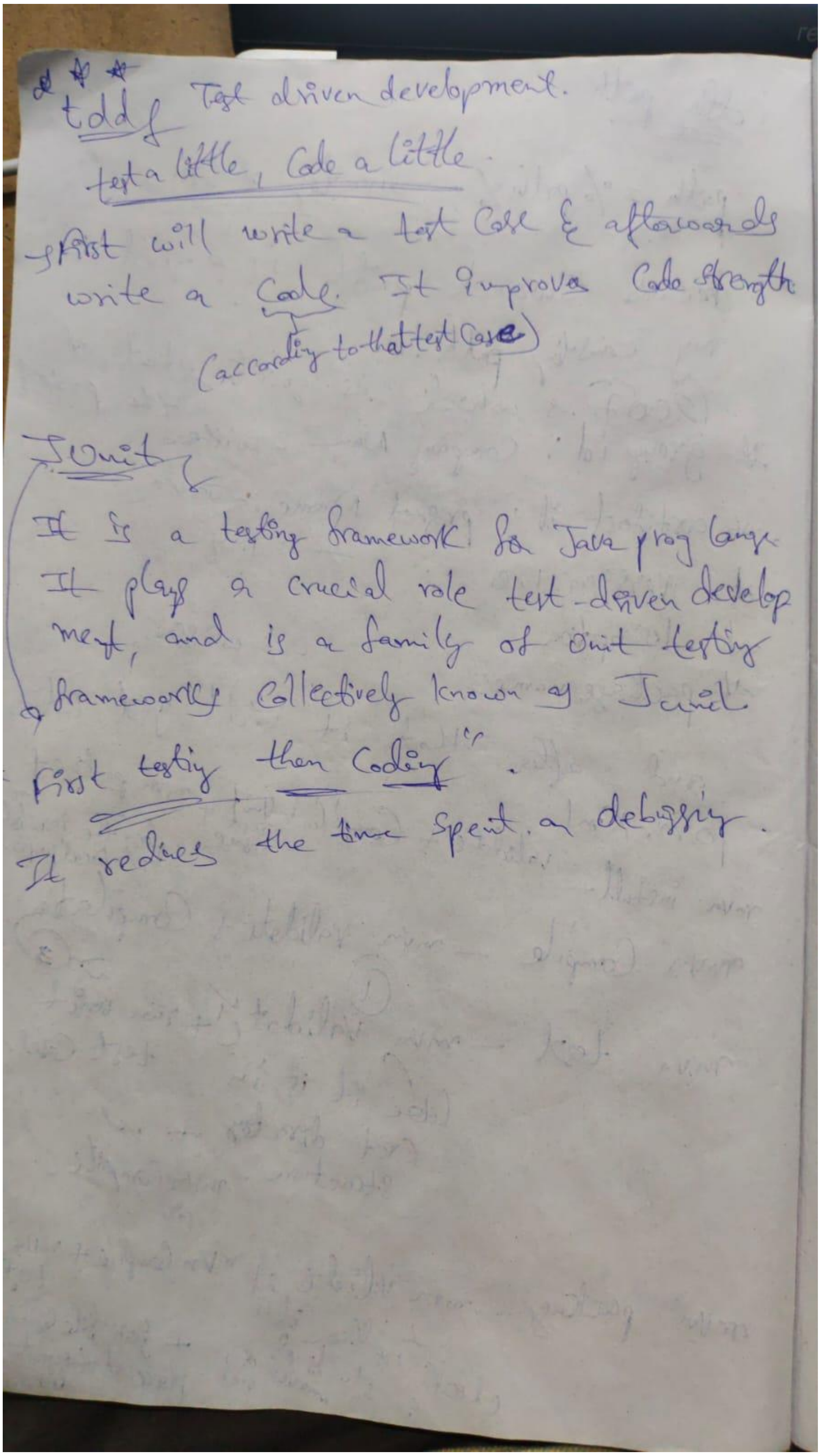
```
Finished after 0.021 seconds
Runs: 3/3    Errors: 0    Failures: 0

wissen.Junit.ParameterizedData [Runner: JUnit 4] (0.000 s)
  [0] (0.001 s)
    addNumbersTest[0] (0.001 s)
  [1] (0.000 s)
    addNumbersTest[1] (0.000 s)
  [2] (0.000 s)
    addNumbersTest[2] (0.000 s)
```

```java
package wissen.Junit;

import static org.junit.Assert.assertEquals;

import java.util.Arrays;
import java.util.Collection;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

class addClass{
    public int addNumbers(int a ,int b) {
        return a+b;
    }
}

@RunWith(Parameterized.class)
public class ParameterizedData {

    private int firstNumber;
    private int secondNumber;
    private int expectedResult;
    private addClass addClassObj;

    public ParameterizedData(int firstNumber,int secondNumber,int expectedResult) {

        super();
        this.firstNumber = firstNumber;
        this.secondNumber =secondNumber;
        this.expectedResult =expectedResult;

    }
    @Before
    public void initialize()
    {
        addClassObj = new addClass();
    }

    @Parameterized.Parameters
    public static Collection input() {
        return Arrays.asList(new Object [][] {{1,2,3},{11,22,33},{111,222,333}});
    }
```



```java
    @Parameterized.Parameters
    public static Collection input() {
        return Arrays.asList(new Object [][] {{1,2,3},{11,22,33},{111,222,333}});
    }

    @Test
    public void addNumbersTest(){
        int result = addClassObj.addNumbers(firstNumber,secondNumber);
        System.out.println(firstNumber +" + "+secondNumber +" = " +result);
        assertEquals(expectedResult,result);
    }
}
```

```
<terminated> ParameterizedData [JUnit] C:\Users\akkis\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v
1 + 2 = 3
11 + 22 = 33
111 + 222 = 333
```



* * * tdd{ Test driven development.

test a little, Code a little.

-First will write a test Case & afterwards write a Code. It improves Code strength
(according to that test case)

JUnit{

It is a testing framework for Java prog lang. It plays a crucial role test-driven development, and is a family of Unit testing frameworks collectively known as Junit

"First testing then Coding".

It reduces the time spent on debugging.