**JUNIT TESTING**

A JUnit test is a method contained in a class which is only used for testing. This is called a Test class. To define that a certain method is a test method, annotate it with the @Test annotation. This method executes the code under test. You use an assert method, provided by JUnit or another assert framework, to check an expected result versus the actual result. These method calls are typically called asserts or assert statements. You should provide meaningful messages in assert statements. That makes it easier for the user to identify and fix the problem. This is especially true if someone looks at the problem, who did not write the code under test or the test code.

**JUnit Assert Methods**

* This class provides a number of assertion methods, which are useful for writing the test cases. We are recording only failed assertions by using this method. In JUnit, all the assertions contain the assert class.
* At the time of testing applications by using assert methods we are using the fail method to test the application.

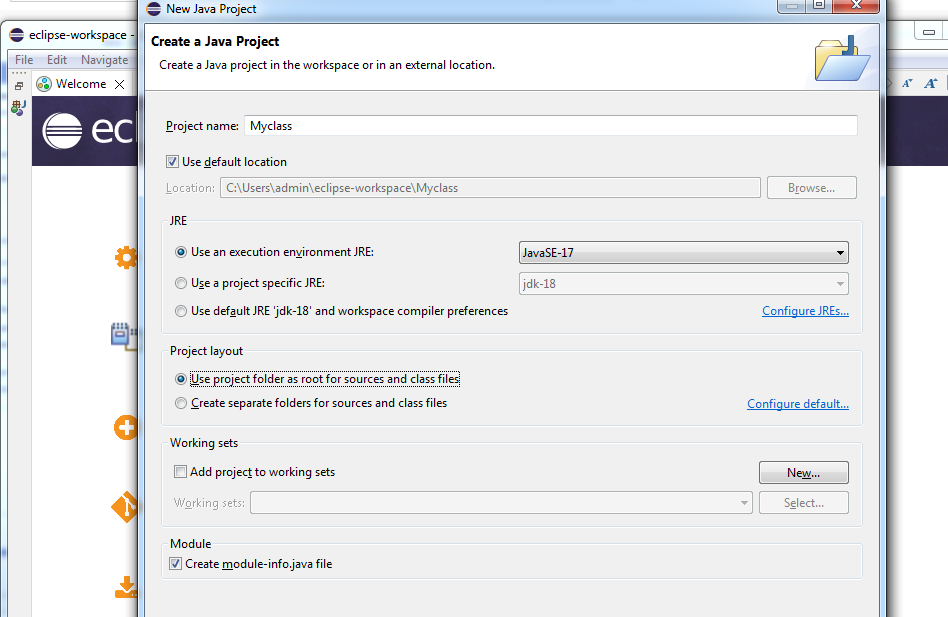
|  |  |  |
| --- | --- | --- |
| **Methods to assert Test results** | | |
| **1** | **assertEquals(expected, actual)** | Tests that expected, actual results are the same |
| **2** | **assertTrue(boolean condition)** | Checks that the boolean condition is true |
| **3** | **assertFalse(boolean condition)** | Checks that the boolean condition is false. |

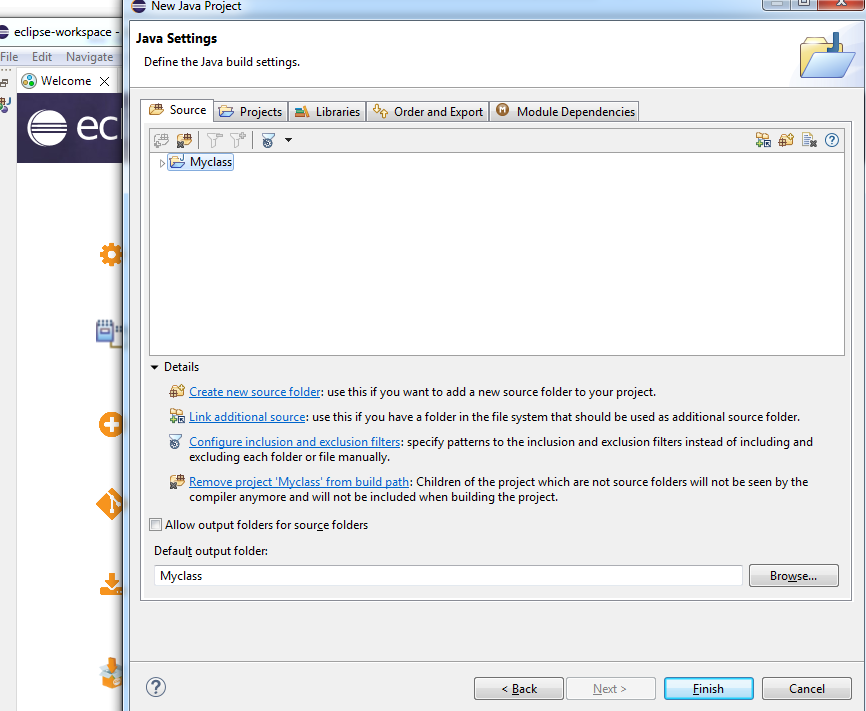
**JUNIT TESTING USING ECLIPSE:**

**Step 1:** Files🡪 New🡪 Java Project

Give the Project name as “Myclass1” and Select Project layout as “Use project folder as root for sources and class files”

Click **Next** and Click **Finish**

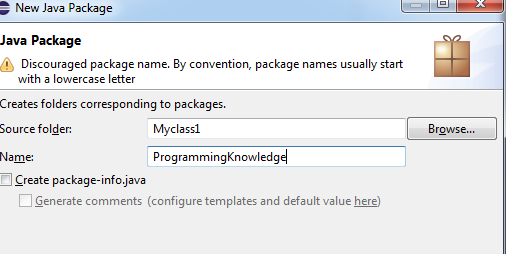


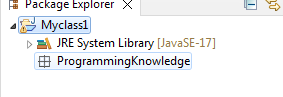


**Step 2:** Right click project i.e “Myclass1”🡪 New 🡪 package

Give the package name as “ProgrammingKnowledge”

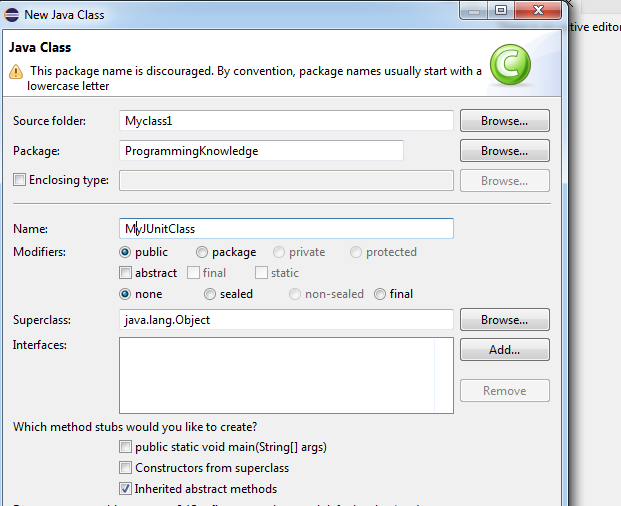
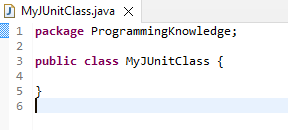
Click **finish**





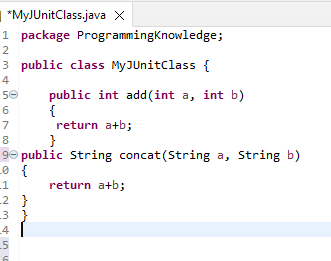
**Step 3**: Right click on package ”Programmingknowledge” 🡪 New🡪 Class

Give the classname as ”MyJunitClass”

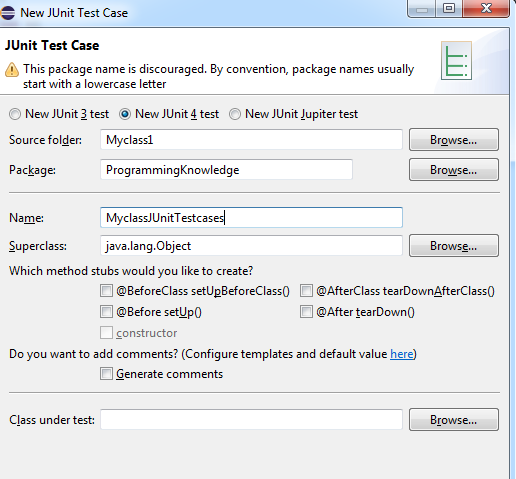
**Step 4:**

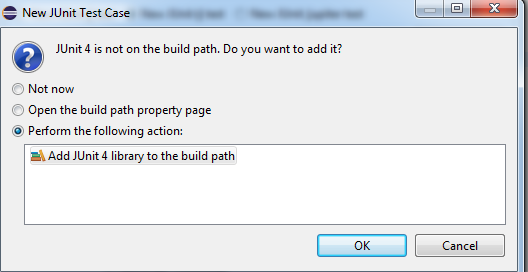
Write the following functions



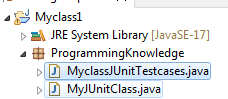
**Step 5:** Right click on Myclass1🡪New🡪 Junit Test Case

Give the name as “MyclassJUnitTestcases” 🡪 Click on finish🡪 then ok



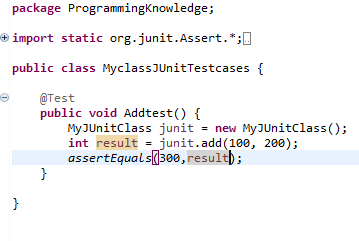


**Following java files are created after step 5**



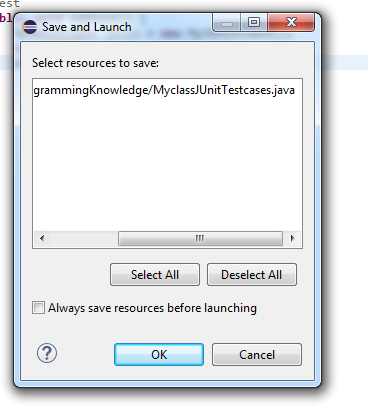
**Step 6:**

Click on MyclassJUnitTestcases.java and Update the code:



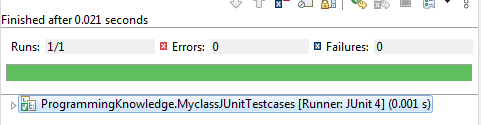
**Right Click on MyclassJUnitTestcases.java🡪 Debug as 🡪 Select Junit Test**

**Click ok**



Junit Result as below

**SUCCESS CASE**

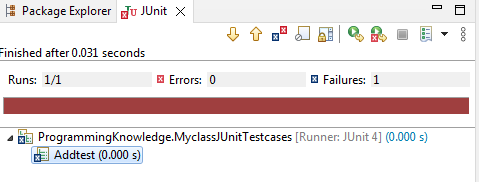


**Failure Case**

**Update expected result with wrong value as shown below**



**Right Click on MyclassJUnitTestcases.java🡪 Debug as 🡪 Select Junit Test**

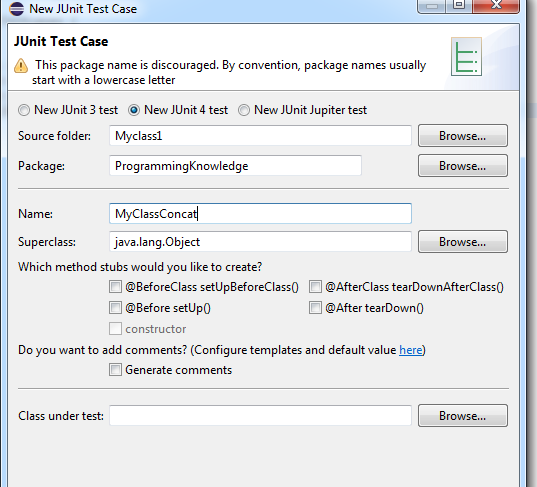


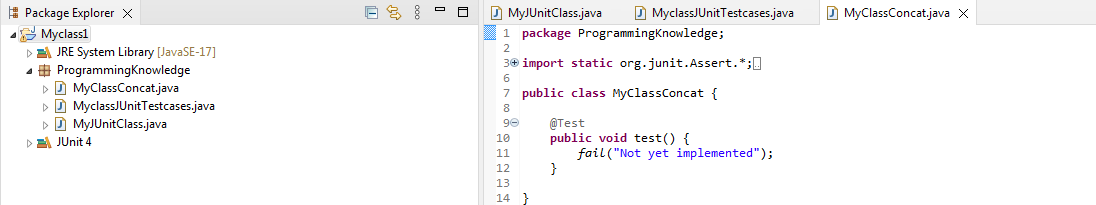
**Step 7:**

**Repeat Step 5 for String concat**

Right click on Myclass1🡪New🡪 Junit Test Case

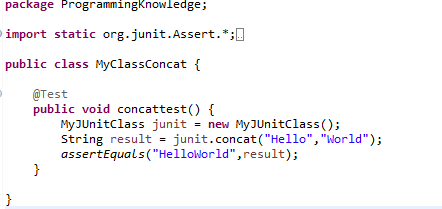
Give the name as “Myclassconcat” 🡪 Click on finish🡪 then ok





**Step 8:**

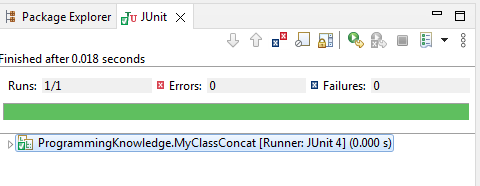
Update the code to concat two strings as below:



### *assertEquals*

The assertEquals assertion verifies that the expected and actual values are equal

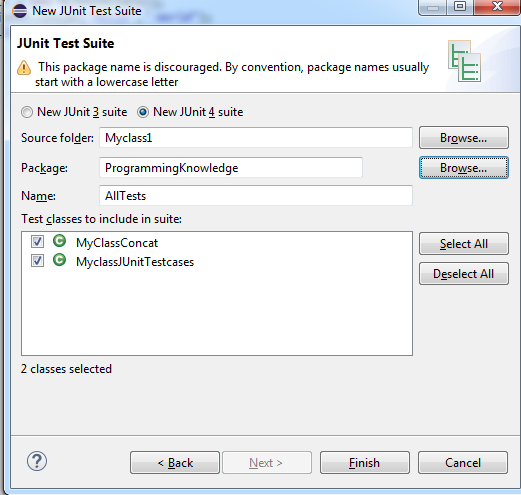
**Right Click on Myclassconcat.java🡪 Debug as 🡪 Select Junit Test**



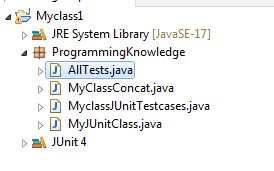
**Step 9: Create test suite for both add and concat cases**

**Right click on Myclass🡪 New🡪 other🡪 Under Junit Select Junit Test Suite**

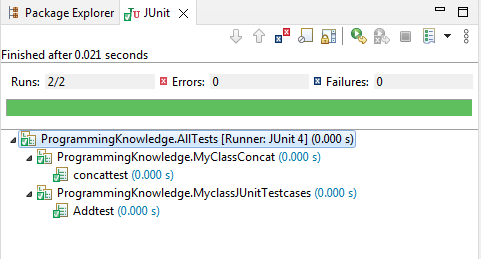
**Click finish**



**AllTests.java gets created**



**Right click on AllTests.java🡪 Debug As🡪 Select Junit Test**



https://www.youtube.com/watch?v=tkzJsP7NP54

**assertTrue()** method checks if supplied boolean condition is true. In case, condition is false, it will through AssertError.

1. public static void assertTrue(boolean condition)
2. public static void assertTrue(boolean condition, Supplier messageSupplier)

**1. public static void assertTrue(boolean condition)**

**Under Step-4 In MyJUnitClass.java add the below function**

public boolean isPrime(int number)

    {

        for (int i = 2; i <=Math.sqrt(number); i++) {

            if(number%i==0)

                return false;

        }

        return true;

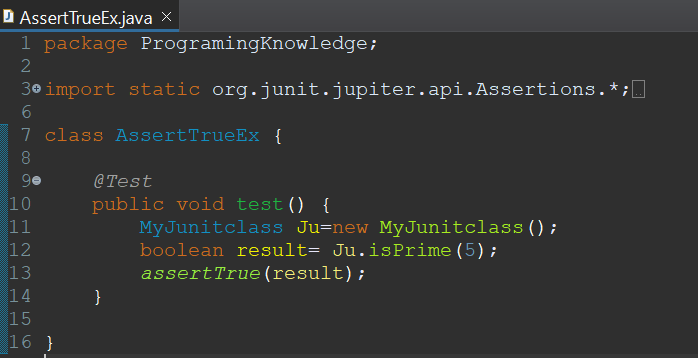
    }

**Repeat Step 5 for isPrime**

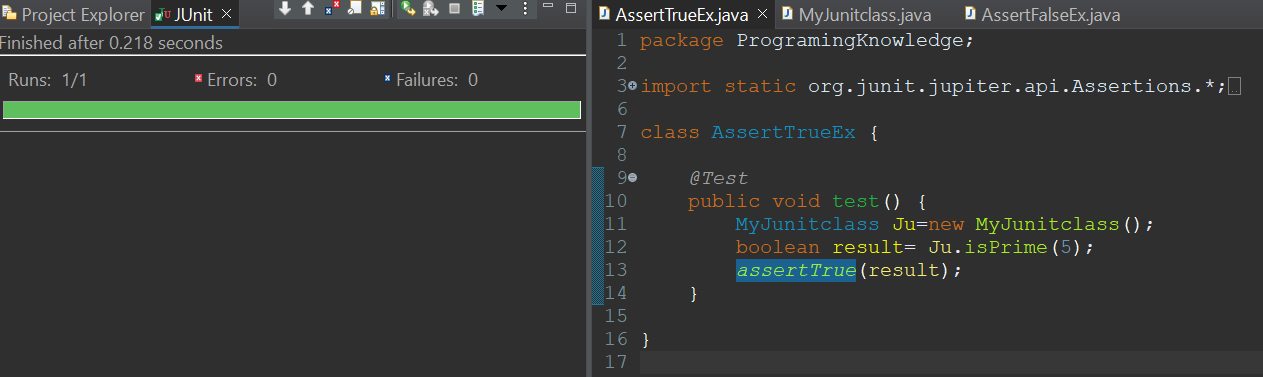
Right click on Myclass1🡪New🡪 Junit Test Case

Give the name as “AssertTrueEx” 🡪 Click on finish🡪 then ok

Update the code to test function as below:



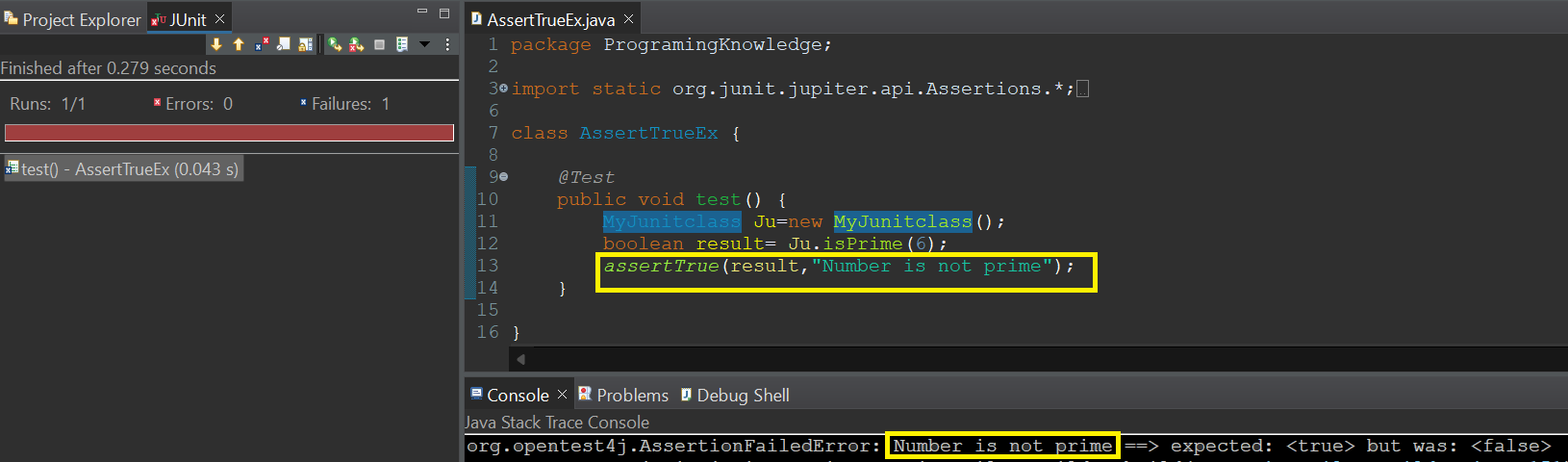
**Right Click on AssertTrueEx.java🡪 Debug as 🡪 Select Junit Test**



Input passed through the calling function is prime hence it return true, As it returns true, executed testcase will be pass.

**2. public static void assertTrue(boolean condition, Supplier messageSupplier)**

If we use above method by passing some message as second parameter, it displays the message in case of test failure.

****

number passed through the isPrime function is not a prime number, Hence function returns false. As it returns false , executed testcase will be fail.

**assertFalse** – AssertFalse() checks if supplied boolean condition is false. In case, condition is true, it will through AssertError.

1. public static void assertFalse(boolean condition)
2. public static void assertFalse(boolean condition, Supplier messageSupplier)

**1. public static void assertFalse(boolean condition)**

**Under Step-4 In MyJUnitClass.java add the below function**

public boolean isPalindrome(String str)

{

for (int i = 0,j=str.length()-1; i <str.length()/2; i++,j--) {

if(str.charAt(i)!=str.charAt(j))

{

return false;

}

 }

return true;

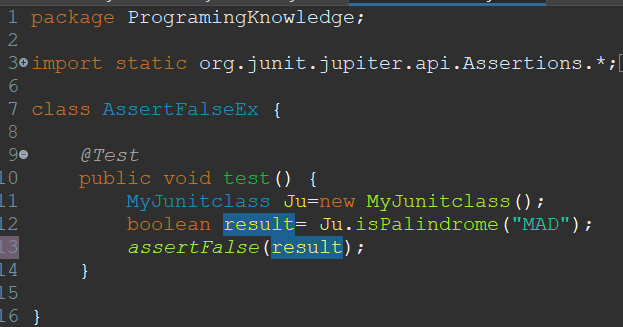
}

**Repeat Step 5 for isPalindrome**

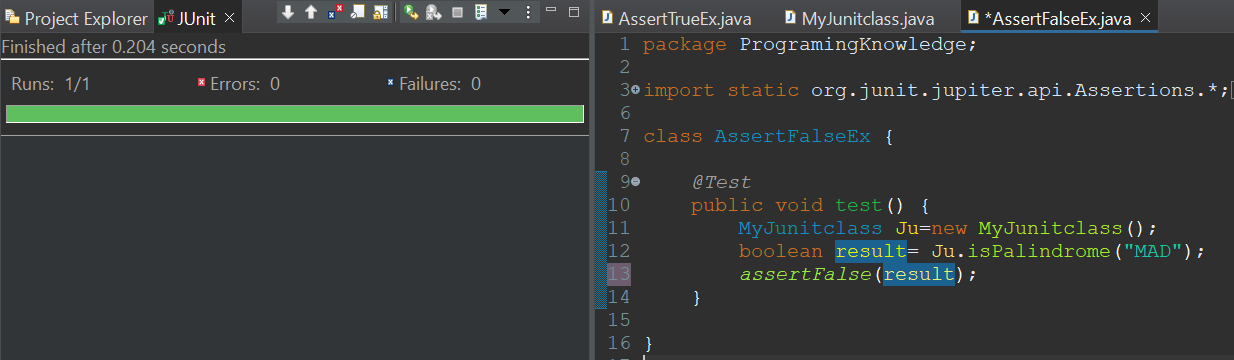
Right click on Myclass1🡪New🡪 Junit Test Case

Give the name as “AssertFalseEx” 🡪 Click on finish🡪 then ok

Update the code to test function as below:

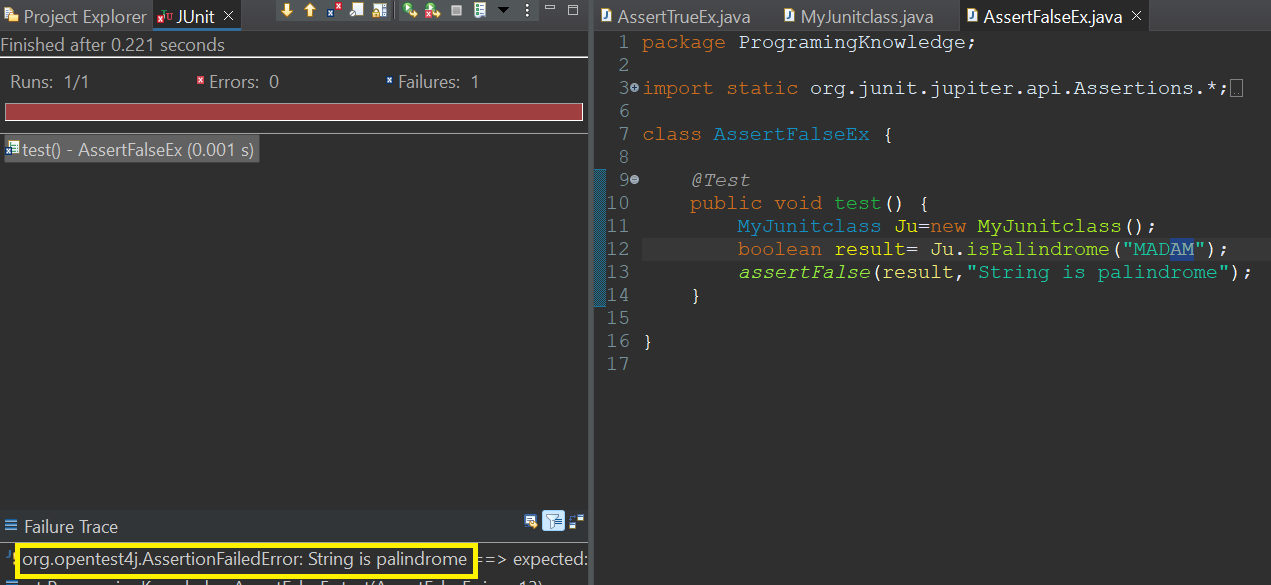


**Right Click on AssertFalseEx.java🡪 Debug as 🡪 Select Junit Test**

  
string passed through the isPalindrome function is not a palindrome, hence function returns false. As it returns false , executed testcase will be pass.

**2. public static void assertFalse(boolean condition, Supplier messageSupplier)**

string passed through the isPalindrome function is a palindrome, Hence function returns true. As it returns True, executed testcase will be fail and also it displays the message which has passed through the assertFalse Function.

****