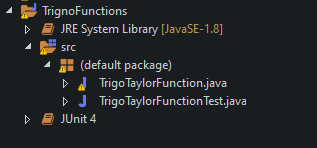
**Lab Assignment – 7**

**Team 11**

The GitHub link mentioned below includes source code, different versions under commit history, Word Document that has test cases with its output in JUnit, and final successful result.

<https://github.com/sonikalpit/TrignoJunit>

**Junit Exercise:** This assigment includes implementation of trigonometric functions such as, Sin, Cos, Tan by using Taylor series formula and the methods that are required to perform the execution.



TrigoTaylorFunction.java has many functions which are listed below:

1. TrignoSin()
2. TrignoCos()
3. TrignoTan()
4. Factorial()
5. powerResult()
6. RadNormalize()
7. DegToRad()

TrigoTaylorFunctionTest.java is Junit test that performs testing of the methods which have been implemented in above class.

**Test Cases:**

1. Sin – Different test strategies will be carried out to check its accuracy whether it generates same value as Math.sin() calculates and those test cases are listed below for Sin function

* Values from first quadrant such as, between 0 to π/2.
* Values from second quadrant such as, between π/2 to π.
* Values from third quadrant such as, between π to 3π/2.
* Values from fourth quadrant such as, between 3π/2 to 2π.

1. Cos – Different test strategies will be carried out to check its accuracy whether it generates same value as Math.cos() calculates and those test cases are listed below for Cos function

* Values from first quadrant such as, between 0 to π/2.
* Values from second quadrant such as, between π/2 to π.
* Values from third quadrant such as, between π to 3π/2.
* Values from fourth quadrant such as, between 3π/2 to 2π.

1. Tan – Different test strategies will be carried out to check its accuracy whether it generates same value as Math.tan() calculates and those test cases are listed below for Tan function

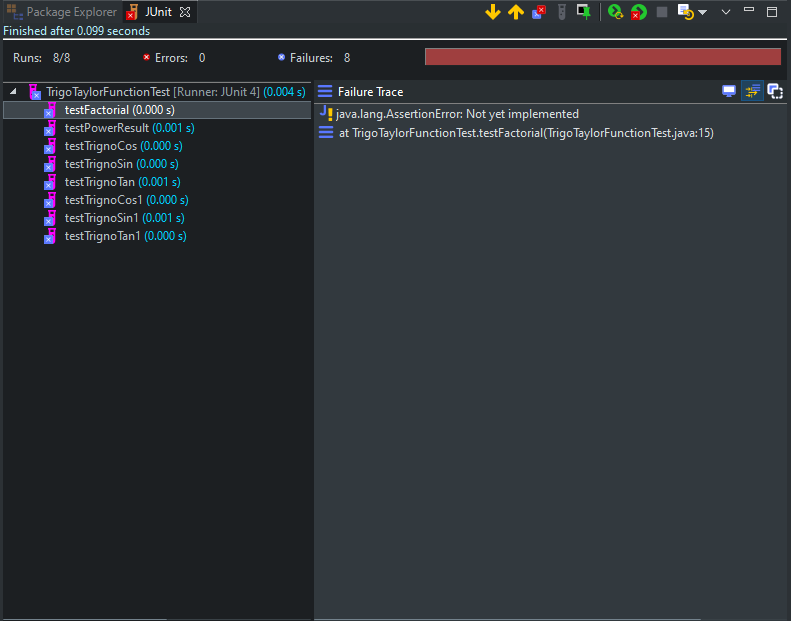
* Values from first quadrant such as, between 0 to π/2.
* Values from second quadrant such as, between π/2 to π.
* Values from third quadrant such as, between π to 3π/2.
* Values from fourth quadrant such as, between 3π/2 to 2π.

|  |  |  |  |
| --- | --- | --- | --- |
| SIN | Expected | Actual | Result |
| Radians | | | |
| 30 | -0.9880 | -0.9880 | Pass |
| 150 | -0.7148 | -0.7148 | Pass |
| 270 | -0.1760 | -0.1760 | Pass |
| 360 | 0.9589 | 0.9588 | Pass |
| Degree | | | |
| 30 | 0.5000 | 0.4999 | Pass |
| 120 | 0.8660 | 0.8660 | Pass |
| 270 | -1.000 | -1.0001 | Pass |
| 360 | 0.0000 | 0.0000 | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| COS | Expected | Actual | Result |
| Radians | | | |
| 30 | 0.1542 | 0.1542 | Pass |
| 120 | 0.8141 | 0.8141 | Pass |
| 270 | 0.9843 | 0.9843 | Pass |
| 360 | -0.2836 | -0.2836 | Pass |
| Degree | | | |
| 30 | 0.8660 | 0.8660 | Pass |
| 120 | -0.5000 | -0.5000 | Pass |
| 270 | 0.0000 | 0.0000 | Pass |
| 360 | 1.0000 | 1.0000 | Pass |

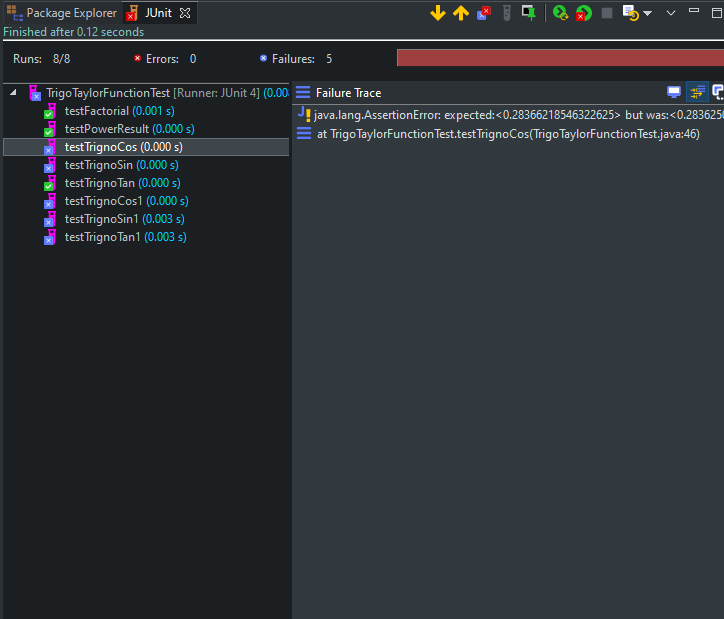
|  |  |  |  |
| --- | --- | --- | --- |
| TAN | Expected | Actual | Result |
| Radians | | | |
| 30 | -6.4053 | -6.4053 | Pass |
| 120 | 0.7131 | 0.7131 | Pass |
| 270 | -0.1788 | -0.1788 | Pass |
| 360 | -3.3801 | -3.3800 | Pass |
| Degree | | | |
| 30 | 0.5773 | 0.5773 | Pass |
| 120 | -1.7320 | -1.7320 | Pass |
| 225 | 1.0000 | 1.0000 | Pass |
| 360 | 0.0000 | 0.0000 | Pass |

* The above table shows the comparison of Math.\*() ; \* = Sin,Cos, Tan and implemented functions and it has passed all the testcases for given values either in Radians or Degree.
* Version 1.0 has all methods with no implementation inside the block so the output of that test suit is failure for all the tests included.



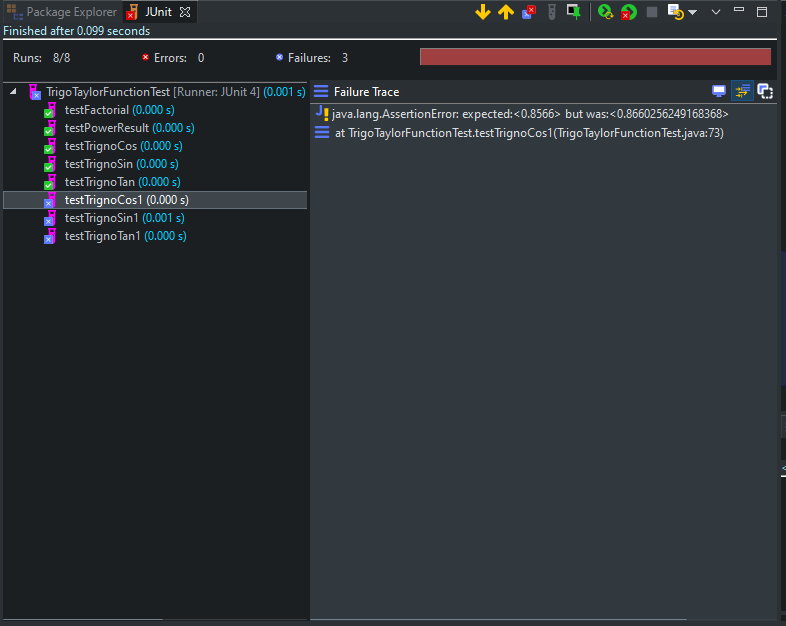
Version 1

* Version 2.0 is having implementation of Sin, Cos, Tan, and its required functions in Taylor series such as calculating power, factorial. It is clear that Factorial, Power and some other test cases produces accurate results. Although Sin and Cos has been implemented in this version but they are unable to produce accurate result as Math functions gives.



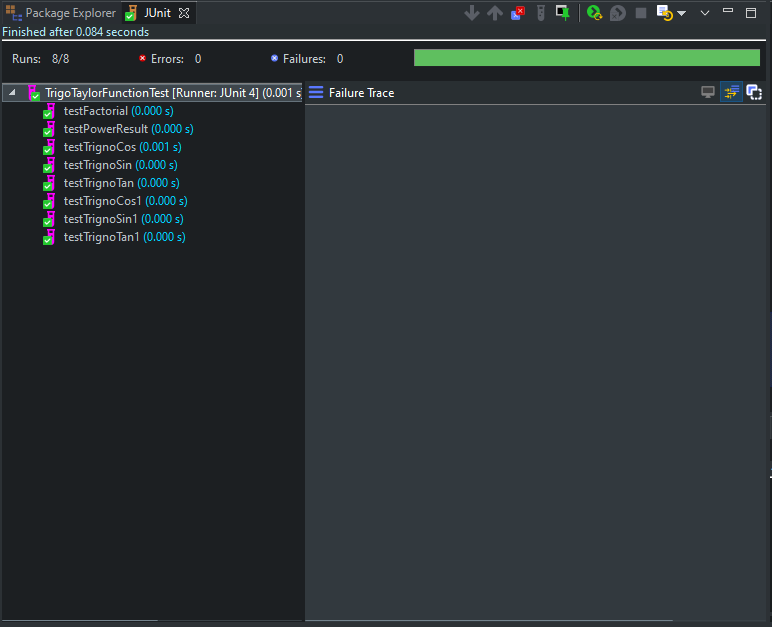
Version 2

* This version is refined version of Version 2.0 as it did not have accurate results so after modifying some of the code, it can now produce almost same value as Math functions generates. However, it cannot calculate accurate value for Degree input in this version.



Version 3

* This final version has all the capabilities to calculate trigonometric functions in radians and degree input as well and to support both functionality, additional DegToRad function has been implemented before. To check the accuracy, different values from various quadrant has been given as an input and some exceptions has been handled to overcome of logical interruption.
* To conclude, it has passed all the tests for given input with accuracy.

Version 4 (Final Version)