Test Cases for EvaluateIt.jar

Tests Designed by: Sebastian Rincon Aguirre
Tests Designed date: 18/03/2023.
Tests Executed by: Sebastian Rincon Aguirre
Tests Execution date: 01/04/2023.
Student ID:
Course: Business Analysis (334)

Introduction

The aim of this report is to present a set of comprehensive test cases for the EvaluateIt.jar file, a calculator program designed to evaluate mathematical expressions in accordance with a defined syntax. Our testing approach covers a wide range of possible operations between numbers to ensure that any potential defects are identified early in the development process, thus preventing any significant delays or holdbacks down the line. This calculator is a crucial component of our company's software strategy, intended to streamline complex and simple calculations for individuals in the production area, enabling them to perform tasks more efficiently and accurately than with a standard calculator.

Solutions Assessment & Validation - Assignment

Tests to pass cases.

Test case #	Test title	Test Summary	Testing steps	Test Data	Expected Results	Actual Results	Status
TTP - 01	Addition of two positive integers	Adding two integers to test the functionality of the addition operation	Choose two integers of 1 digit each and add them using the addition operation	5 + 8	13	13	Pass
TTP - 02	Subtraction of two positive integers	Subtracting two integers to test the functionality of the addition operation	Choose two integers of 1 digit each and add them using the subtraction operation	7 - 2	5	5	Pass
TTP - 03	Multiplication of two positive integers	Multiplying two positive integers to test the functionality of the addition operation	Choose two integers of 1 digit each and add them using the multiplication operation	4 * 6	24	24	Pass
TTP - 04	Division of two positive integers	Dividing two positive integers to test the functionality of the addition operation	Choose two integers and add them using the division operation	10 / 5	2	2	Pass
TTP - 05	Exponentiation of a positive integer	Use exponential function to perform a basic operation	Choose two single numbers and use the exponentiation expression	2 ^ 3	8	8	Pass
TTP - 06	Modulus of two positive integers	Check if the modulus operator shows if a number is odd or even	Choose two single numbers and use the modulus expression	9 % 4	1	1	Pass

TTP - 07	Absolute value of a negative integer	Check if the absolute value operation works properly	write down the abs expression and add a single digit number positive or negative in the brackets	abs(-6)	6	6	Pass
TTP - 08	Nearest upper integer of a decimal number	Check if the number in rounded to the nearest upper integer	write down the ceil expression and add a positive decimal number in the brackets	ceil(3.2)	4	4	Pass
TTP - 09	Nearest lower integer of a decimal number	Check if the number in rounded to the nearest lower integer	write down the floor expression and add a positive decimal number in the brackets	floor(3.9)	3	3	Pass
TTP - 10	Rounding a decimal number to the nearest integer	Check if the number in rounded to the nearest integer	write down the round expression and add a positive decimal number in the brackets	round(4.5)	5	5	Pass
TTP - 11	Sine of an angle in radians	check if the sine of an angle is converted correctly to radians	write down the sine expression and add a positive decimal number in the brackets	sin(0.5)	0,00873	0,00873	Pass

TTP - 12	Cosine of an angle in radians	check if the cosine of an angle is converted correctly to radians	write down the cosine expression and add a positive decimal number in the brackets	cos(0.5)	0,99996	0,99996	Pass
TTP - 13	Tangent of an angle in radians	check if the tangent of an angle is converted correctly to radians	write down the tangent expression and add a positive decimal number in the brackets	tan(0.5)	0.5463	0,5463	Pass
TTP - 14	Natural logarithm of a positive number	Check the natural logarithm expression of a positive number - integer	write down the natural logarithm expression and add a positive decimal number in the brackets	ln(2)	0.6931	ln not recognized as a valid expression	Fail
TTP - 15	Base 10 logarithm of a positive number	Check the Base 10 logarithm expression of a positive number - integer	write down the Base 10 logarithm expression and add a positive decimal number in the brackets	log(100)	2	4,605	Fail
TTP - 16	Base 2 logarithm of a positive number	Check the Base 2 logarithm expression of a positive number - integer	write down the Base 2 logarithm expression and add a positive decimal number in the brackets	log2(8)	3	3	Pass

TTP - 17	Maximum of multiple arguments	Using a string of numbers, we are testing if the max function selects the highest number of the string	write down a string of 5 integers inside of the bracket using the max function	max(3, 5, 1, 7, 2)	7	1	Fail
TTP - 18	Minimum of multiple arguments	Using a string of numbers, we are testing if the min function selects the highest number of the string	write down a string of 5 integers inside of the bracket using the min function	min(3, 5, 1, 7, 2)	1	1	Pass
TTP - 19	Average of multiple arguments	Using a string of numbers, we are testing if the average function calculates the average	write down a string of 5 integers inside of the bracket using the average function	average(3, 5, 1, 7, 2)	3,6	3,6	Pass
TTP - 20	Sum of multiple arguments	Using a string of numbers, we are testing if the sum function calculates the average	write down a string of 5 integers inside of the bracket using the sum function	sum(3, 5, 1, 7, 2)	18	18	Pass
TTP - 21	Addition and Subtraction	Evaluate the expression 2 + 3 - 1	Enter the expression "2 + 3 - 1" and press enter	2 + 3 - 1	4	4	Pass
TTP - 22	Multiplication and Division	Evaluate the expression 10 / 2 * 3	Enter the expression "10 / 2 * 3" and press enter	10 / 2 * 3	15	15	Pass
TTP - 23	Parentheses and Exponents	Evaluate the expression (4 + 3) ^ 2	Enter the expression "(4 + 3) ^ 2" and press enter	(4 + 3) ^ 2	49	49	Pass

TTP - 24	Mixed Operations	Evaluate the expression 5 * 2 + 10 / 2 - 3	Enter the expression "5 * 2 + 10 / 2 - 3" and press enter	5 * 2 + 10 / 2 - 3	12	12	Pass
TTP - 25	Trigonometric Functions	Evaluate the expression sin(45) + cos(45)	Set the radians mode enter the expression "sin(45) + cos(45)" and press enter	sin(45) + cos(45)	1,3762	1,3762	Pass
TTP - 26	Logarithmic Functions	Evaluate the expression log(100) + ln(e)	Enter the expression "log(100) + ln(e)" and press enter	log(100) + ln(e)	3	ln not recognized as a valid expression	Fail
TTP - 27	Random Function	Evaluate the expression random(10)	Enter the expression "random(10)" and press enter	random(10)	random number between 0 and 10	random numbers between 0 and 10	Pass

Tests to Fail.

Test case #	Test title	Test Summary	Testing steps	Test Data	Expected Results	Actual Results	Status
TTF - 01	Division by zero	Test division by zero	Input "10/0"	10/0	Error message "division by zero"	Infinity	Pass

TTF - 02	Invalid operator	Test invalid operator	Input "10@2"	10@2	Error message "invalid operator"	10@2 not recognized as a valid expression	Pass
TTF - 03	Missing operand	Test missing operand	Input "10+"	10+	blank	blank	Pass
TTF - 04	Invalid function name	Test invalid function name	Input "sq(4)"	sq(4)	sq not recognized as a valid expression	sq not recognized as a valid expression	Pass
TTF - 05	Invalid function arguments	Test invalid function arguments	Input "sin(abc)"	sin(abc)	abc not recognized as a valid expression	abc not recognized as a valid expression	Pass
TTF - 06	Missing closing parenthesis	Test missing closing parenthesis	Input "2*(3+4"	2*(3+4	Parentheses mismatched	Parentheses mismatched	Pass
TTF - 07	Invalid decimal precision	Test invalid decimal precision	Input "round(10.123, 15)"	round(10.123, 15)	Invalid argument count for round	Invalid argument count for round	Pass
TTF - 08	Invalid conversion function	Test invalid conversion function	Input "toinch (5)"	toinch (5)	toinch not recognized as a valid expression	toinch not recognized as a valid expression	Pass
TTF - 09	Invalid constant name	Test invalid constant name	Input "pir"	pir	pir not recognized as a valid expression	pir not recognized as a valid expression	Pass
TTF - 10	Missing multiplication operator	Test missing multiplication operator	Input "2(3+4)"	2(3+4)	blank	blank	Pass
TTF - 11	Invalid input format	Test invalid input format	Input "10++2"	10++2	Error message "invalid input format"	12	Fail
TTF - 12	Invalid exponent operator	Test invalid exponent operator	Input "2^*2"	2^*2	blank	blank	Pass

TTF - 13	Invalid modulus operator	Test invalid modulus operator	Input "10%2%"	10%2%	blank	blank	Pass
TTF - 14	Invalid degree value	Test invalid degree value	Input "sin ()"	sin ()	Invalid argument count for sin	Invalid argument count for sin	Pass
TTF - 15	Invalid logarithm base	Test invalid logarithm base	Input "log0(10)"	log0(10)	log0 not recognized as a valid expression	log0 not recognized as a valid expression	Pass
TTF - 16	Invalid argument type	Test invalid argument type	Input "sum (2, 'three')"	sum (2, 'three')	three' not recognized as a valid expression	three' not recognized as a valid expression	Pass
TTF - 17	Invalid random value	Test invalid random value	Input "random (-5)"	random (-5)	random (-5) not recognized as a valid expression	Random number between -1 and -5	Fail
TTF - 18	Division with invalid arguments	Test division with invalid arguments	Input "10	10/+	blank	blank	Pass
TTF - 19	Logarithm of negative number	Test logarithm function with a negative number	Enter the expression "-log (-3)" into the calculator	-3	Invalid argument passed to log	Invalid argument passed to log	Pass

Defects

Defect ID	Test case ID	Test Title	Summary	Testing Steps	Test Data	Expected Results	Actual Results	Status
1	TTP - 14	Natural logarithm of a positive number	Check the natural logarithm expression of a positive number - integer	Write down the natural logarithm expression and add a positive decimal number in the brackets (ln (2))	2	0.6931	ln not recognized as a valid expression	Fail
2	TTP - 15	Base 10 logarithm of a positive number	Check the Base 10 logarithm expression of a positive number - integer	Write down the Base 10 logarithm expression and add a positive decimal number in the brackets (log (100))	100	2	4.605	Fail

3	TTP - 17	Maximum of multiple arguments	Using a string of numbers, we are testing if the max function selects the highest number	Write down a string of 5 integers inside of the bracket using the max function (max (3, 5, 1, 7, 2))	3, 5, 1, 7, 2	7	1	Fail
4	TTF - 11	Invalid input format	Test invalid input format	Input "10++2"	10++2	Error message "invalid input format"	12	Fail
5	TTF - 14	Invalid degree value	Test invalid degree value	Input "sin ()"	sin ()	Invalid argument count for sin	Invalid argument count for sin	Fail
6	TTF - 17	Invalid random value	Test invalid random value	Input "random(-5)"	random(-5)	random(-5) not recognized as a valid expression	Random number between -1 and -5	Fail

Test coverage plan.

To thoroughly evaluate the calculator application, we have identified a range of test cases that cover a variety of scenarios. The following sections outline the tests we have identified and the rationale behind each test.

Functional Tests

We have identified a range of functional tests to ensure that the calculator functions as expected in different scenarios. These tests cover basic arithmetic operations, as well as more complex functions such as logarithms, trigonometric functions, and random number generation. We will also assess for expected error messages when invalid input is entered.

Boundary Tests

To ensure that the calculator can manage extreme values, we have included a range of boundary tests. These tests include inputting the largest and smallest possible numbers for each data type, as well as testing for edge cases such as dividing by zero and calculating the modulus of zero.

Usability Tests

We will also conduct usability tests to ensure that the calculator is easy to use and understand. These tests will involve observing users as they perform basic operations and recording any difficulties or confusion they encounter. We will also test the effectiveness of the calculator's user interface, including the layout and visibility of buttons and input fields.

Test Execution

We will execute these tests manually, recording the results and any issues encountered during the testing process. We will also use automated testing tools where possible to increase efficiency and reduce the potential for human error.

Test Coverage Analysis

We will analyze our test coverage to ensure that all critical features and functions of the calculator have been thoroughly evaluated. We will track the number of tests executed, the number of tests passed and failed, and any issues or bugs identified during testing. Based on the results of our analysis, we may identify areas where additional testing is required to achieve complete coverage.