LAB 7

SE 317

Test methods

* Functionality
  + testAddition
  + testDivision
  + testDivideByZero
  + testSquareRootOfNEgativeNumber
  + testMultiplication
  + testSubtraction
* Features
  + testGetDecimalButton
  + testGetEqualsButton
  + testGetNumber Buttons
  + testGetAddButton
  + testGetSubtractionButton
  + testGetMultiplyButton
  + testGetDivideButton
  + testGetSqrtButton
  + testGetSqaureButton
  + testGetMemoryAddButton
  + testGetMemorySubtractionButton
  + testGetMemoryRecallButton
  + testGetClearButton
  + testGetDeleteButton
  + testGetDisplay
* Bounderies
  + testOverflow
  + testOverflow2

Test Criterion for CalculatorController, CalculatorModel, and CalculatorView:

Arithmetic Operations:

Verify that addition, subtraction, multiplication, and division operations produce correct results for various operand combinations.

Test with positive, negative, and zero operands.

Test with operands close to the maximum and minimum representable values for doubles.

Single Operand Operations:

Test square root and square operations to ensure they return correct results.

Include test cases with positive, negative, and zero operands.

Verify that square root operation throws an exception for negative operands.

Memory Operations:

Test adding and subtracting from memory and verify that memory is updated correctly.

Check that memory operations handle large numbers appropriately.

Test recalling memory to ensure it retrieves the correct value.

Test clearing memory to ensure it resets properly.

Display Manipulation:

Test clearing the display to ensure it removes all content.

Test deleting the last character to ensure it removes one character at a time.

Verify that the display is updated correctly after each operation.

Test display of memory value on the calculator display.

Button Functionality:

Test clicking each button (number buttons, operation buttons, memory buttons, delete, clear, etc.) to ensure they perform the correct actions.

Check that the buttons are enabled/disabled as expected based on the current state of the calculator.

Verify that all buttons are properly linked to their corresponding actions in the controller.

Layout and Usability:

Test the layout of the calculator GUI to ensure it is visually appealing and functional.

Verify that the calculator components are properly aligned and spaced.

Conduct usability testing to ensure ease of use and intuitiveness of the calculator interface.

Integration Testing:

Test the interaction between CalculatorController, CalculatorModel, and CalculatorView to ensure they work together correctly.

Verify that the calculations performed by the controller are consistent with the model's calculations and displayed correctly in the view.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Test Method | Test Criterion | Test input Values | Test Expected output | Test Actual Output | Success/Fail |
| 1 | Intra-testing | testAddition | (5, +,3) | 8 | 8 | Success |
| 2 | Intra-testing | testDivision | (10,/,2) | 5 | 5 | Success |
| 3 | Intra-testing | testDivideByZero | (5,/,0) | Divide by zero error | Divide by zero error | Success |
| 4 | Intra-testing | testSquareRootOfNegativeNumber | (-4,sqrt,0) | Cannot take the square root of a negative number | Cannot take the square root of a negative number | Success |
| 5 | Intra-testing | testMultiplaction | (10,\*,10) | 100 | 100 | Success |
| 6 | Intra-testing | testSubtraction | (30,-,6) | 24 | 24 | Success |
| 7 | Intra-testing | testMemoryOperation | 10,5,0 |  |  | Success |

A screen shot of a computer program

Description automatically generated