Damien Bafile

M036449@tafe.wa.edu.au

Activity 1.7

Contents

[GitHub Account 2](#_Toc12524768)

[Data Structures 2](#_Toc12524769)

[Name, type and purpose of each variable 2](#_Toc12524770)

[Algorithms 2](#_Toc12524771)

[Main Application 2](#_Toc12524772)

[Caluculator.cs 2](#_Toc12524773)

[Arithmetic Library 7](#_Toc12524774)

[ArithmeticMath.cs 7](#_Toc12524775)

[Trigonometric Library 7](#_Toc12524776)

[TrigonometricMath.cs 7](#_Toc12524777)

[Algebraic Library 8](#_Toc12524778)

[AlgebraicMath.cs 8](#_Toc12524779)

[Recommended testing procedure 9](#_Toc12524780)

[Recommendations on upgrades and future enhancements 10](#_Toc12524781)

# GitHub Account

<https://github.com/damien-bafile/Calculator>

# Data Structures

## Name, type and purpose of each variable

|  |  |  |
| --- | --- | --- |
| \_totalAmount | private double | First number to be saved |
| \_tempValue | private double? | Total for Arithmetic operations |
| \_divideButtonClicked | private bool | Flag if divide button was pressed |
| \_minusButtonClicked | private bool | Flag if minus button was pressed |
| \_negativeButtonClicked | private bool | Flag if negative button was pressed |
| \_plusButtonClicked | private bool | Flag if plus button was pressed |
| result | double | Results from an operation |
| \_clear | bool | For checking if the txtDisplay needs clearing |

# Algorithms

## Main Application

### Caluculator.cs

##### FrmCalculator

InitializeComponent

##### void btnEquals\_Click(object sender, EventArgs e)

if plus button pressed totalAmount = tempvalue + textDisplay

if minus Button Clicked totalAmount = tempvalue - textDisplay

if divide Button Clicked totalAmount = tempvalue / textDisplay

if multiply Button Clicked totalAmount = tempvalue \* textDisplay

Display error if totalAmount equals null display totalAmount if number is good

clear textbox set true

reset \_tempValue

Here it tests if the totalAmount is null if it is it means user has tried to divide by zero and displays an error.

##### void btnOne\_Click(object sender, EventArgs e)

If clear equals true

Display = 1

Set clear textbox to false

Else

append 1 to textbox

Set focus to equals button

##### void btnTwo\_Click(object sender, EventArgs e)

If clear equals true

Display = 2

Set clear textbox to false

Else

append 2 to textbox

Set focus to equals button

##### void btnThree\_Click(object sender, EventArgs e)

If clear equals true

Display = 3

Set clear textbox to false

Else

append 3 to textbox

Set focus to equals button

##### void btnFour\_Click(object sender, EventArgs e)

If clear equals true

Display = 4

Set clear textbox to false

Else

append 4 to textbox

Set focus to equals button

##### void btnFive\_Click(object sender, EventArgs e)

If clear equals true

Display = 5

Set clear textbox to false

Else

append 5 to textbox

Set focus to equals button

##### void btnSix\_Click(object sender, EventArgs e)

If clear equals true

Display = 6

Set clear textbox to false

Else

append 6 to textbox

Set focus to equals button

##### void btnSeven\_Click(object sender, EventArgs e)

If clear equals true

Display = 7

Set clear textbox to false

Else

append 7 to textbox

Set focus to equals button

##### btnEight\_Click(object sender, EventArgs e)

If clear equals true

Display = 8

Set clear textbox to false

Else

append 8 to textbox

Set focus to equals button

##### btnNine\_Click(object sender, EventArgs e)

If clear equals true

Display = 9

Set clear textbox to false

Else

append 9 to textbox

Set focus to equals button

##### btnZero\_Click(object sender, EventArgs e)

If clear equals true

Display = 0

Set clear textbox to false

Else

append number to textbox

Set focus to equals button

##### btnPoint\_Click(object sender, EventArgs e)

IF Point not in txtdisplay

Append point

Set focus to equals button

##### void btnClear\_Click(object sender, EventArgs e)

Clear txtdisplay

\_clear set to false

\_totalAmount set to 0

\_totalAmount set to 0

\_plusButtonClicked set to false

\_minusButtonClicked set to false

\_divideButtonClicked set to false

\_multiplyButtonClicked set to false

\_negativeButtonClicked set to false

Set focus to equals button

##### void btnPlus\_Click(object sender, EventArgs e)

If there is a number get text from display

clear the display

set plus button flag true

set minus button flag false

set divide button flag false

set multiply button flag false

Set focus to equals button

##### void void btnMinus\_Click(object sender, EventArgs e)

if the negative button clicked equals false and there is a number in txtdisplay.

clear the display

set plus button flag false

set minus button flag true

set divide button flag false

set multiply button flag false

Set focus to equals button

else

add a minus to display

get negative flag to true

Set focus to equals button

##### void btnMultiply\_Click(object sender, EventArgs e)

If there is a number get text from display

clear the display

set plus button flag false

set minus button flag false

set divide button flag false

set multiply button flag true

Set focus to equals button

##### void btnDivide\_Click(object sender, EventArgs e)

If there is a number get text from display

clear the display

set plus button flag false

set minus button flag false

set divide button flag true

set multiply button flag false

Set focus to equals button

##### void btnTangent\_Click(object sender, EventArgs e)

try

compute tangent with number from display

if result is null

display error

reset tempvalue

else

display result

reset tempvalue

catch

display error

reset tempvalue

clear textbox set true

Set focus to equals button

Here it tests if the output from the library is null and displays an error

##### void btnSine\_Click(object sender, EventArgs e)

try

compute Sine with number from display

if result is null

display error

reset tempvalue

else

display result

reset tempvalue

catch

display error

reset tempvalue

clear textbox set true

Set focus to equals button

##### void btnCosine\_Click(object sender, EventArgs e)

try

compute cosine with number from display

if result is null

display error

reset tempvalue

else

display result

reset tempvalue

catch

display error

reset tempvalue

clear textbox set true

Set focus to equals button

##### void void btnSquareRoot\_Click(object sender, EventArgs e)

compute Square Root with number from display

if is not null

display result

else

display error

reset tempvalue

clear textbox set true

Set focus to equals button

Here it tests if the output from the library is null and displays an error

##### void btnCubeRoot\_Click(object sender, EventArgs e)

compute Cube Root with number from display

display result

reset tempvalue

clear textbox set true

Set focus to equals button

##### void btnInverse\_Click(object sender, EventArgs e)

compute Inverse with number from display

if is not null

display result

else

display error

reset tempvalue

clear textbox set true

Set focus to equals button

Here it tests if the output from the library is null and displays an error

##### void FrmCalculator\_KeyPress(object sender, KeyPressEventArgs e)

if keyboard keypress equals 1

display 1 button

if keyboard keypress equals 2

display 2 button

if keyboard keypress equals 3

display 3 button

if keyboard keypress equals 4

display 4 button

if keyboard keypress equals 5

display 5 button

if keyboard keypress equals 6

display 6 button

if keyboard keypress equals 7

Display 7 button

if keyboard keypress equals 8

display 8 button

if keyboard keypress equals 9

display 9 button

if keyboard keypress equals 0

display 0 button

if keyboard keypress equals +

trigger addition button

if keyboard keypress equals -

trigger minus button

if keyboard keypress equals \*

trigger multiply button

if keyboard keypress equals /

trigger divide button

if trigger keypress equals c or C

trigger cosine

if keyboard keypress equals s or S

trigger sine

if keyboard keypress equals t or T

trigger tangent

if keyboard keypress equals q or Q

trigger cuberoot

if keyboard keypress equals r or R

trigger squareroot

if keyboard keypress equals i or I

trigger inverse button

##### private void FrmCalculator\_KeyDown(object sender, KeyEventArgs e)

if keyboard keypress equals control esc

clear display

if keyboard keypress equals control c

select all text in display

copy text

deselect all text in display

if keyboard keypress equals control v

paste text into buffer

if buffer is a number

output to display

##### bool IsDecimalFormat(string str)

if string is a number

return true

else

return false

## Arithmetic Library

### ArithmeticMath.cs

##### double Addition(double tempvalue, double totalAmount)

return tempvalue plus totalAmount

##### double Subtraction(double tempvalue, double totalAmount)

return tempvalue minus totalAmount

##### double? Divide(double tempvalue, double totalAmount)

if totalAmount is 0

return null;

else

divide tempvalue by totalAmount

return amount divided;

If the dividing number is zero, It returns NULL so it can be tested inside the application and accounted for there.

##### double Multiply(double tempvalue, double totalAmount)

return tempvalue \* amount

## Trigonometric Library

### TrigonometricMath.cs

##### Public static double? Tangent(double amount)

if amount is 90

return null

else

convert number from degrees to radians

compute tanagent on radian number

return result in degrees

If the amount is 90, It returns NULL so it can be tested inside the application and accounted for there.

##### public static double Sine(double amount)

convert number from degrees to radians

compute Sine on radian number

return result in degrees

##### public static double Cosine(double amount)

convert number from degrees to radians

compute Cosine on radian number

return result in degrees

##### private static double ToDegrees(double radians)

compute number from radians to degrees

return degrees;

## Algebraic Library

### AlgebraicMath.cs

##### double? SquareRoot(double amount)

if (amount > 0)

return square root of amount

else

return null;

##### double CubeRoot(double amount)

If amount is less than zero

Make amount positive

Calculate cube root

Make number negative again

If it’s positive just calculate and return the cube root of amount

##### public static double? Inverse(double amount)

if amount is not 0

return 1 divide amount;

else

return null;

Since the inverse of 1 / amount the operator could choose to do and inverse of 0. If the amount is 0, It returns NULL so it can be tested inside the application and accounted for there.

# Recommended testing procedure

Testing should be done on every function and a multitude of inputs. Here is a an example of a test table that will conduct a wide variety of test.

|  |  |  |
| --- | --- | --- |
| Test | Expected | Actual / Comment |
| Addition | | |
| -5 + -5 |  |  |
| -5 + 0 |  |  |
| 0 + 5 |  |  |
| 5 + 5 |  |  |
| Subtraction | | |
| -5 - -5 |  |  |
| 5 - 0 |  |  |
| 0 - 5 |  |  |
| 10 - 1 |  |  |
| Multiplication | | |
| 5 \* 5 |  |  |
| 5 \* 0 |  |  |
| 0 \* 5 |  |  |
| 5 \* -5 |  |  |
| Division | | |
| 10 / 5 |  |  |
| 10 / 3 |  |  |
| 10 / 0 |  |  |
| 0 / 10 |  |  |
| 10 / -2 |  |  |
| Tan | | |
| 0º |  |  |
| 30º |  |  |
| 45o |  |  |
| 60o |  |  |
| 90 o |  |  |
| Sin | | |
| 0º |  |  |
| 30º |  |  |
| 45o |  |  |
| 60o |  |  |
| 90 o |  |  |
| Cos | | |
| 0º |  |  |
| 30º |  |  |
| 45o |  |  |
| 60o |  |  |
| 90 o |  |  |
| Square Root | | |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| -1 |  |  |
| Cube Root | | |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| -1 |  |  |
| Inverse | | |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| -1 |  |  |

# Recommendations on upgrades and future enhancements

* Update Libraries to use tuples for results to simply the code
* Add more arithmetic operations such as mod.
* Add more Trigonometric functions to library such as tan-1, cos-1, sin-1.
* Adding Memory and Memory Clear functionality.
* Add the ability to use radians.
* Parentheses (Order of operations)
* Pi button.