# Technical Document

## Data Structures

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Purpose** |
| result | double | Used to assign the result of the sum as a double |
| num | double | Variable assigned to the user’s input to calculate sums using only 1 number, (square roots, inverses, etc.) |
| num1 | double | Variable assigned to the user’s first numeral input when calculating sums using 2 numbers, (subtractions, multiplications, etc.) |
| num2 | double | Variable assigned to the user’s second numeral input when calculating sums using 2 numbers, (subtractions, multiplications, etc.) |
| plusButtonClicked | boolean | Checks if btnPlus was clicked |
| minusButtonClicked | boolean | Checks if btnMinus was clicked |
| divideButtonClicked | boolean | Checks if btnDivide was clicked |
| multiplyButtonClicked | boolean | Checks if btnMultiply was clicked |

## Algorithms

**Psuedocode:**

BtnZero\_Click()

{

Textbox.Text = textbox.Text + “0”;

}

…

BtnNine\_Click()

{

Textbox.Text = textbox.Text + “9”;

}

BtnClear\_Click()

{

Textbox.Clear();

}

BtnPoint\_Click()

{

Textbox.Text = textbox.Test + “.”;

}

BtnPlus\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.text = num1;

Textbox.Clear();

Bool plusButtonClicked = true;

}

…

BtnMultiply\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.text = num1;

Textbox.Clear();

Bool multiplyButtonClicked = true;

}

BtnEquals\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.text = num2;

If plusButtonClicked is true

Double result = (using BasicMath library) add(num1, num2);

else if minusButtonClicked is true

Double result = (using BasicMath library) minus(num1, num2);

else if divideButtonClicked is true

Double result = (using BasicMath library) divide(num1, num2);

else if multiplyButtonClicked is true

Double result = (using BasicMath library) multiply(num1, num2);

Textbox.Text = result to string;

}

BtnSqrt\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.Text = double num;

If num is not a negative number

Textbox.Text = (using BasicMath library) squareroot(num) to string;

Else

Informative error messagebox

}

ClearToolStripMenuItem\_Click()

{

Textbox.Clear();

}

QuitToolStripMenuItem\_Click()

{

Exit application

}

BtnCbrt\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.Text = double num;

Textbox.Text = (using BasicMath library) cuberoot(num) to string;

}

BtnInv\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.Text = double num;

Textbox.Text = (using BasicMath library) inverse(num) to string;

}

BtnSin\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.Text = double num;

If degrees radiobutton checked

double radians;

Convert input to radians

Textbox.Text = (using BasicMath library) sin(radians) to string

Else if radians radiobutton checked

(no conversion necessary)

Textbox.Text = (using BasicMath library) sin(num) to string

}

…

BtnTan\_Click()

{

--error trap to check if input is valid--

Then:

Textbox.Text = double num;

If degrees radiobutton checked

If num equals to 90 or num equals to 270

Textbox.Text = “Undefined”;

else

double radians;

Convert input to radians

Textbox.Text = (using BasicMath library) sin(radians) to string

Else if radians radiobutton checked

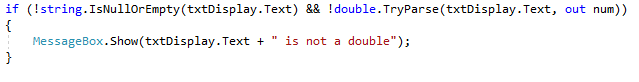
(no conversion necessary)

Textbox.Text = (using BasicMath library) sin(num) to string

}

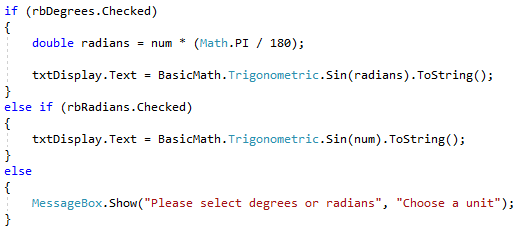
**Error handling techniques:**

To check if the user input was valid I used:



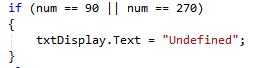
On methods where necessary.

On trigonometric functions I used the code:



To either calculate in radians or degrees. If either option wasn’t selected I gave an informative message box.

For the tangent method I used:



To check if 90 or 270 degrees was entered.

## Testing Procedure

Before commercial release this application should be tested via both black box and white box testing by QA testers and developers. Testing results should be written in a test document.

## Upgrades and Future Enhancements

If in the future, we wish to upgrade/enhance the application. We can add additional calculations to it like going above the square root and cube root and having an option to the power of x, adding an exponential calculation, and adding a % button.