Alexander Scholz

South Metropolitan TAFE  [Company address]

A1.7 Technical Documentation

Contents

[Data Structure 2](#_Toc478558487)

[Algorithms 2](#_Toc478558488)

[Psuedo Code 2](#_Toc478558489)

[Maths.dll 2](#_Toc478558490)

[Btn\_Click 3](#_Toc478558491)

[btnClear 3](#_Toc478558492)

[btnAdd 3](#_Toc478558493)

[btnMinus 3](#_Toc478558494)

[btnMulitply 4](#_Toc478558495)

[btnDivide 4](#_Toc478558496)

[btnEqu 5](#_Toc478558497)

[btnSquRoot 5](#_Toc478558498)

[btnCubeRoot 6](#_Toc478558499)

[btnInvert 6](#_Toc478558500)

[btnSin 6](#_Toc478558501)

[btnCos 7](#_Toc478558502)

[btnTan 7](#_Toc478558503)

[Error handling 7](#_Toc478558504)

[Recommended Testing Procedure 7](#_Toc478558505)

[Recommendation on upgrades and future enhancements 7](#_Toc478558506)

# Data Structure

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Type | Description | Value |
| A | Double | Store the first number input in text box upon an equation button being clicked | 0 |
| B | Double | Store second number input in text box upon equBtn being clicked | 0 |
| Result | Double | Value is the result of math equation to be put in text box | 0 |
| addBtnClicked | Boolean | True if the add button is clicked, used to determine which arithmetic equation is used | false |
| subBtnClicked | Boolean | True if the subtract button is clicked, used to determine which arithmetic equation is used | false |
| divBtnClicked | Boolean | True if the divide button is clicked, used to determine which arithmetic equation is used | false |
| multBtnClicked | Boolean | True if the multiply button is clicked, used to determine which arithmetic equation is used | false |

# Algorithms

## Psuedo Code

### Maths.dll

Arithmetic {

Public double add { a + b };

Public double sub { a – b };

Public double mult { a \* b };

Public double div { a / b };

}

Algebraic {

Public double SquareRoot { Math.Round ( Math.SquareRoot ( a ) ) };

Public double CubeRoot { Math.Pow ( a, ( 1 / 3 ) ) };

Public double invert { a – ( a \* 2 ) ) };

}

Trigonometric {

Public double sin { Math.Sin( Math.Pi \* a / 180 ) };

Public double cos { Math.Cos( Math.Pi \* a / 180 ) };

Public double tan { Math.Tan( Math.Pi \* a / 180 ) };

}

### Btn\_Click

Private btn\_Click( object sender, eventArguments e ) {

Button clickedButton = sender button;

If clickedButton = “.” & txtDisplay.contains( “.” ) {

return;

} else {

txtDisplay += clickedButton;

}

}

### btnClear

Private btnClear\_Click( object sender, eventArguments e ) {

txtDisplay.Clear;

A = B = Result = 0;

}

### btnAdd

Private btnAdd (object sender, eventArguments e) {

if txtDisplay.Text != " " {

A += double.Parse txtDisplay.Text ;

txtDisplay.Clear;

addBtnClicked = true;

subBtnClicked = false;

divBtnClicked = false;

multBtnClicked = false;

}

else {

MessageBox.Show( error message );

}

}

### btnMinus

Private btnMinus (object sender, eventArguments e) {

if txtDisplay.Text != " " {

A += double.Parse txtDisplay.Text ;

txtDisplay.Clear;

addBtnClicked = false;

subBtnClicked = true;

divBtnClicked = false;

multBtnClicked = false;

}

else {

MessageBox.Show( error message );

}

}

### btnMulitply

Private btnMultiply (object sender, eventArguments e) {

if txtDisplay.Text != " " {

A += double.Parse txtDisplay.Text ;

txtDisplay.Clear;

addBtnClicked = false;

subBtnClicked = false;

divBtnClicked = false;

multBtnClicked = true;

}

else {

MessageBox.Show( error message );

}

}

### btnDivide

Private btnDivide (object sender, eventArguments e) {

if txtDisplay.Text != " " {

A += double.Parse txtDisplay.Text ;

txtDisplay.Clear;

addBtnClicked = false;

subBtnClicked = false;

divBtnClicked = true;

multBtnClicked = false;

}

else {

MessageBox.Show( error message );

}

}

### btnEqu

Private btnEqu ( object sender, eventArguments e ) {

if txtDisplay.Text = "" {

txtDisplay = A;

}

if addBtnClicked = true {

B = double.Parse( txtDisplay );

Result = Maths.Class1.Arithmetic.Add( A, B );

}

else if subBtnClicked = true {

B = double.Parse( txtDisplay );

Result = Maths.Class1.Arithmetic.Sub( A, B );

}

else if multBtnClicked = true {

B = double.Parse( txtDisplay );

Result = Maths.Class1.Arithmetic.Mult( A, B );

}

else if divBtnClicked = true {

B = double.Parse( txtDisplay );

Result = Maths.Class1.Arithmetic.Div( A, B );

}

txtDisplay = Result;

A = B = Result = 0;

}

### btnSquRoot

Private btnSquRoot ( object sender, eventArguments e ) {

if txtDisplay.Text != " " {

A = double.Parse( txtDisplay );

Result = Maths.Class1.Algebraic.SQRT( A );

txtDisplay = Result;

}

else {

MessageBox.Show( error message );

}

}

### btnCubeRoot

Private btnCubeRoot ( object sender, eventArguments e ) {

if txtDisplay.Text != " " {

A = double.Parse( txtDisplay );

Result = Maths.Class1.Algebraic.CBRT( A );

txtDisplay = Result;

}

else {

MessageBox.Show( error message );

}

}

### btnInvert

Private btnInvert ( object sender, eventArguments e ) {

if txtDisplay.Text != " " {

A = double.Parse( txtDisplay );

Result = Maths.Class1.Algebraic.Invert( A );

txtDisplay = Result;

}

else {

MessageBox.Show( error message );

}

}

### btnSin

Private btnSin ( object sender, eventArguments e ) {

if txtDisplay.Text != " " {

A = double.Parse( txtDisplay );

Result = Maths.Class1.Trigonometric.Sin( A );

txtDisplay = Result;

}

else {

MessageBox.Show( error message );

}

}

### btnCos

Private btnCos ( object sender, eventArguments e ) {

if txtDisplay.Text != " " {

A = double.Parse( txtDisplay );

Result = Maths.Class1.Trigonometric.Cos( A );

txtDisplay = Result;

}

else {

MessageBox.Show( error message );

}

}

### btnTan

Private btnTan ( object sender, eventArguments e ) {

if txtDisplay.Text != " " {

A = double.Parse( txtDisplay );

Result = Maths.Class1.Trigonometric.Tan( A );

txtDisplay = Result;

}

else {

MessageBox.Show( error message );

}

}

## Error handling

I used if statements in the Arithmetic, Algebraic and Trigonometric buttons to check if a number is in the textbox, if text box is empty, the click events end or, on the equBtn click event, the space has the value of A. I also used an IF statement to see if a decimal point was used, if not then add one or if there was a decimal then don’t add a another decimal. To prevent incorrect input such as letters, the display text box was made read only.

# Recommended Testing Procedure

The testing procedure was to first test the number and decimal buttons, along with the Arithmetic buttons to check the reference to Maths.dll and the arithmetic algorithms. Then test the Algebraic buttons with the numbers, and then the Trigonometric buttons. Last, applied error checking and testing all the buttons work with the error checking in place.

# Recommendation on upgrades and future enhancements

Recommended upgrades include:

-Adding more math algorithms for things like ‘to the power of…’ or percentages.

-Use a listbox to show earlier inputted equations until the clear button is pressed.