2019

Technical Document

GUI CALCULATOR - https://github.com/AUSSIEFIDDY/GUI-Calculator

TOM GREEN

Contents

Data structures

Name, type and purpose of each variable

Name	Type	Purpose
total1	Double	Storing the first number in an Arithmetic calculation.
total2	Double	Storing the total of the calculation.
plusButtonClicked	Boolean	Determining whether the plus button was clicked.
minusButtonClicked	Boolean	Determining whether the minus button was clicked.
divideButtonClicked	Boolean	Determining whether the division button was clicked.
multiplyButtonClicked	Boolean	Determining whether the multiplication button was clicked.

Algorithms

Pseudo code for each method

Add

```
Function Add(Number one, Number two)
{
        Return Number one Plus Number two
}
```

Subtract

```
Function  Sub(Number one, Number two)
{
        Return Number one Minus  Number two
}
```

Divide

```
Function Div(Number one, Number two)
{
    If one is not 0 and two is not 0
        Return Number one Divided by Number two
    Else
        Return Not A Number
}
```

Multiply

```
Function Mult(Number one, Number two)
{
    Return Number one Multiplied by Number two
}
```

Square Root

```
Function Sqrt(Number one)
{
    Variable root is Number one Divided by 3
    For 32 Loops
        Variable root is (Variable root Plus Number one Divided by Variable root)Divided by
        2
    Return Variable root
}
```

Cube Root

Function Cbrt(Number one)

{

    If Number one is 1

        Return Number one

    Variable value is the Square root of the Square root of Number one

    Return Variable value Multiplied by the Cube Root of Variable value

}

Inverse

Function Inv(Number one)

{

    If Number one is not 0

        Return 1 Divided by Number one

    Else

        Return 0

}

Tangent

Function Tan(Number one)

{

    Return the Tangent of Number one

}

Sine

Function Sin(Number one)

{

    Return the Sine of Number one

}

Cosine

Function Cos(Number one)

{

    Return the Cosine of Number one

}

Error handling techniques

There are a few Error handling techniques employed in the GUI Calculator, such as:

- Checking if the display is empty when a function button is clicked.
- Checking that the user doesn't divide by 0.
- Checking that the user doesn't input 0 in an Inverse calculation.
- Checking that the user doesn't input 90 in a Tangent calculation.
- Checking that the user doesn't input a negative number in a Square Root or Cube Root calculation.

Recommended testing procedure

This software should be rigorously tested prior to a commercial release, to do this I would perform both Black Box Testing and White Box Testing and use a test table with a similar structure to the one below:

Test ID	Test	Expected Output	Actual Output	Comments

Recommendations on upgrades and enhancements

The GUI Calculator can certainly be improved upon, a few of the main features I will upgrade are below, sorted from highest to lowest priority:

- Keyboard functionality.
- Calculation overview.
- More methods (such as modulo, logarithm, etc.).
- Making the UI Prettier.