Calculator Application

M203210 | 9/10/2018

Technical Documentation

Craig Baldry

2018

Contents

[Introduction 2](#_Toc526856167)

[Data Structures 3](#_Toc526856168)

[Algorithms 4](#_Toc526856169)

[Pseudo Code 4](#_Toc526856170)

[CalculatorForm.cs 4](#_Toc526856171)

[Arithmetic.cs 8](#_Toc526856172)

[Algebraic.cs 8](#_Toc526856173)

[Trignometric.cs 8](#_Toc526856174)

[Error handling techniques 9](#_Toc526856175)

[Recommended testing procedure 9](#_Toc526856176)

[Recommendations 10](#_Toc526856177)

# Introduction

This report will provide information related to the Calculator application from developing this application. In the report it will cover the data structures regarding the name type and purpose of each variable, the algorithms used with pseudo code for each method and error handling techniques, recommended testing procedure and recommendations on upgrades and future enhancements. To see this repository please click [here](https://github.com/CraigBaldry/BasicCalculator).

# Data Structures

Here is a table with all the variables that were used in this application:

|  |  |  |
| --- | --- | --- |
| Name | Type | Purpose |
| inputA | Double | When using arithmetic this will be our first value after the uses any arithmetic controls |
| inputB | Double | When using arithmetic this will be our second value after the uses any arithmetic controls |
| output | Double | this will be used to display the output of the calculation to user |
| isPlusButtonClicked | Boolean | Used to check if user clicked plus control |
| isMinusButtonClicked | Boolean | Used to check if user clicked minus control |
| isDividebuttonClicked | Boolean | Used to check if user clicked divide control |
| isMultiplyButtonClicked | Boolean | Used to check if user clicked multiply control |
| btnClear | Button | Clears txtDisplay to default 0 |
| txtDisplay | TextBox | User can type or use control to add numeric values, decimal or negative |
| btn0 – btn9 | Button | Control for numeric values |
| btnDecimal | Button | Control for displaying decimal |
| btnMakesNegative | Button | Makes it negative |
| btnAdd | Button | This adds inputA with inputB |
| btnMinus | Button | This subtracts inputA with inputB |
| btnDivide | Button | This divides inputA with inputB |
| btnMultiply | Button | This multiplies inputA with inputB |
| btnEqual | Button | This computes inputA and inputB |
| btnSQRT | Button | Squares the txtDisplay text |
| btnCube | Button | Cubes the txtDisplay text |
| btnInverse | Button | Inverses the txtDisplay text |
| btnTan | Button | Finds the tangent value |
| btnSin | Button | Finds the sine value |
| btnCos | Button | Finds the cosine value |
| menu | MenuStrip | Contains the menu option |
| fileMenuItem | ToolStipMenuItem | Contains Quit and Clear |
| quitMenuItem | ToolStipMenuItem | Allows user close program |
| clearMenuItem | ToolStripMenuItem | Allows user to clear the text |

# Algorithms

The following algorithms will be supplied using pseudo code, to allow for better understanding of what is happening. Will also include how to handle any problems that may arise from in proper use.

## Pseudo Code

Here are all the pseudo code for each classes.

### CalculatorForm.cs









### Arithmetic.cs



### Algebraic.cs



### Trignometric.cs

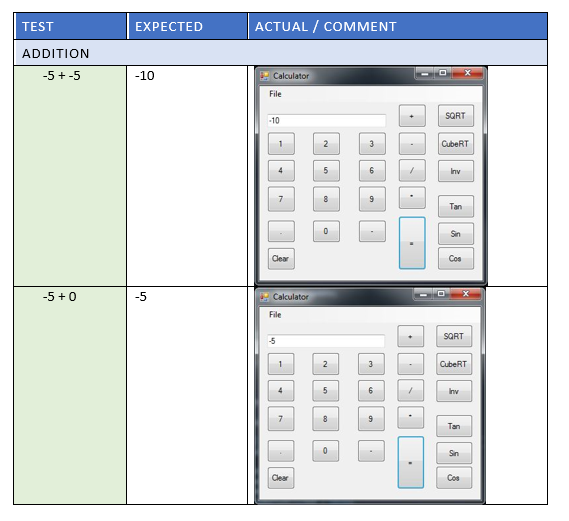


## Error handling techniques

With this program the exception that was found that could occur was the FormatException. By eliminating the possible use of characters that aren’t numeric it helps to eliminate the possibility of failure. There is however the chance the user can clear the text resulting in a null value which will throw this exception. To catch this exception all the methods are surrounded with a try block, in the catch block I have left empty so that the user isn’t interrupted, and the program fails and shutdown.

# Recommended testing procedure

To test this program, it would be recommended that a test table for each method with a range of valid and invalid data to be tested on the program. Here is an example of a data table that could be used to test this program:



# Recommendations

Here is a list of recommended upgrades or future enhancements:

* Expand the menu items to include different functions
* Include a pi button to input in the calculator
* Include bracket buttons
* Include backspace button
* Include other types of functions such as currency convertor, time, volume and maybe even speed.
* An option to switch between a standard GUI interface with basic arithmetic and algebraic controls, to a scientific calculator that includes the tan, sin and cos controls
* The negative button could be a toggle like feature to make the number negative or positive