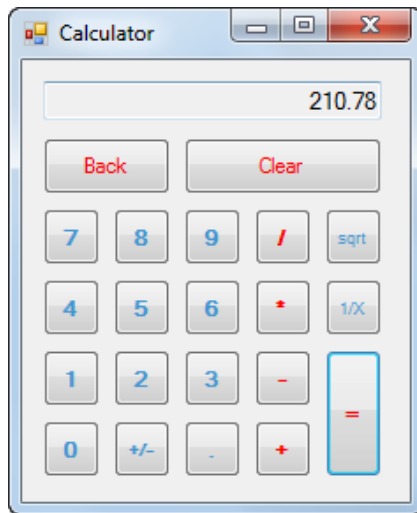


Homework 5 - Calculator

GUI

Description

For this assignment we will create a form that lets the user perform the operations provided by a basic calculator. We will solve this using a Calculator Class which performs the required operations. You will need to research how to create ENUM in C# for this assignment. They are simple to implement.



Specifications

You will need to complete the following:

1. Create a new Project called Calculator.
2. To perform an addition, subtraction, multiplication, or division operation, the user clicks the first number, followed by the appropriate operator key (+, -, *, /), followed by the second number and the equals key (=).
3. To perform an addition, subtraction, multiplication, or division operation on the result of a previous operation, the user clicks another operator key, followed by another number and the equals key. The user can also repeat the previous operation on the result by clicking the equals keys without first clicking another operator and number.
4. To calculate the square root or the reciprocal of a number or to change the sign of a number, the user clicks the number followed by the appropriate operator key (sqrt, 1/X, +/-).
5. To calculate the square root or the reciprocal of the result of a previous operation, the user clicks the appropriate operator key.
6. Each time the user clicks a number key, the number is displayed in the text box at the top of the form. This text box also displays the result of an operation when the user clicks the sqrt, 1/X, +/-, or = key.
7. To erase the last digit entered, the user clicks the Back key.

- ~~8. To clear all the values entered, the user clicks the Clear key.~~
- ~~9. Create a class named Calculator that implements the functions of the calculator. A design for the class is given below.~~
- ~~10. The Calculator class should accept double parameters and provide a double result for its calculated values.~~
- ~~11. If the user tries to divide a number by zero, the calculator should display an error message in the text box. Remember how we compare doubles.~~
- ~~12. **NOTE**** - One of the behaviors of the Windows calculator is that you can perform a calculation by clicking an operator key other than the = key. In other words, you can add a list of numbers like this using the Windows calculator: 3 + 12 + 14 + 8 you're your calculator should behave the same. The user should NOT have to perform each calculation separately by clicking the = key like this: 3 + 12 = + 14 = + 8 =. Make sure your implementation works like the Windows calculator. This can be a bit difficult because you must maintain the state of the calculator, Given this operation some thought, this will be a complicated task.~~
13. Make sure you do not forget to put all XML comments and a comment at the top of the code file which contains your name and the assignment.

Design of the Calculator Class

Private field or Property	Description
CurrentValue	A double that stores the result currently displayed by the calculator.
Operand1	A double that stores the value of the first operand.
Operand2	A double that stores the value of the second operand.
Op	An Operator type that stores a member of the Operator enumeration.
Operator	An enumeration with these constants: Add, Subtract, Multiply, Divide, and None.
Constructor	Description
()	Creates a Calculator object with default values. The default value for the op field is Operator.None.
Property	Description
CurrentValue	Gets the value of the currentValue field.
Method	Description
Add(displayValue)	Sets the operand1 and currentValue fields to the value that's passed to it, and sets the op field to Operation.Add.
Method	Description
Subtract(displayValue)	Sets the operand1 and currentValue fields to the value that's passed to it, and sets the op field to Operation.Subtract.
Multiply(displayValue)	Sets the operand1 and currentValue fields to the value that's passed to it, and sets the op field to Operation.Multiply.
Divide(displayValue)	Sets the operand1 and currentValue fields to the value that's passed to it, and sets the op field to Operation.Divide.
Equals()	Performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the operand1 field.
Equals(displayValue)	Sets the operand2 field to the value that's passed to it. Then, performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the operand1 field.
Clear()	Sets the private fields to their default values.

Documentation

You will create a document (.docx, .rtf, .pdf) which contains the following:

- Your name and assignment.
- A screenshot of your form running with at least one test case per operation (Make sure you include division by zero).
- How did you solve the problem of making your calculator behave like the Windows calculator?

- Explain in detail the following concepts –
 - Fields
 - Properties (what problem do they solve)
 - Auto-Implemented Properties
 - Constructors
 - Object Initializers (what problem do they solve).
 - Expression-Bodied Members

What to Submit

You need to submit your entire solution folder (zipped) and your document. **DO NOT** zip your document. Make sure your document is in the correct format and all your files include your name and assignment.