**Calculator Application Documentation**

**Overview**

The **Calculator Application** is a simple Windows Forms application built using C# that provides basic arithmetic operations (+, -, \*, /) on floating-point numbers. It features a graphical user interface (GUI) with number buttons, operation buttons, and functionalities like clearing the display and handling decimal points.

**Features**

1. **Basic Arithmetic Operations:**
   * Addition (+)
   * Subtraction (-)
   * Multiplication (\*)
   * Division (/)
2. **User Interface:**
   * Display area (labelResult) to show the current number or result.
   * Number buttons (0 to 9) for inputting numbers.
   * Operation buttons (+, -, \*, /) to select arithmetic operations.
   * Special buttons:
     + C: Clears all inputs and resets the calculator.
     + ±: Toggles the sign of the current number.
     + .: Adds a decimal point to the current number.

**Functionality**

* **Initialization (Reset Method):** Initializes all variables and resets the calculator display to zero.
* **Input Handling (GetNumber Method):** Appends digits to currentNumber and updates the display accordingly when number buttons are clicked.
* **Operation Selection (GetOperation Method):** Sets the selected operation (+, -, \*, /) and prepares for the next number input.
* **Calculation (Calculate Method):** Performs the selected arithmetic operation on firstNumber and secondNumber based on the chosen operation. Displays the result and updates variables for further calculations.
* **Error Handling:** Handles division by zero errors and other exceptions gracefully by displaying "Error" and resetting the calculator.

**Implementation Details**

* **Variables:**
  + currentNumber: String to store the current number being input.
  + operation: String to store the selected operation (+, -, \*, /).
  + firstNumber, secondNumber: Float variables to store operands for calculations.
  + result: Float variable to store the result of calculations.
  + isDuplicate: Boolean flag to manage consecutive operations without pressing equals.
* **Event Handling:**
  + NumberButton\_Click: Handles clicks on number buttons.
  + OperationButton\_Click: Handles clicks on operation buttons.
  + btnPN\_Click: Toggles the sign of the current number.
  + btnDot\_Click: Adds a decimal point to the current number.
  + btnC\_Click: Resets the calculator to its initial state.
  + btnEqual\_Click: Initiates the calculation process.

**Usage**

1. **Launching the Application:**
   * Run the application (calculatorApp.exe).
   * The calculator interface will appear.
2. **Performing Calculations:**
   * Click on number buttons to input numbers.
   * Click on operation buttons to select an arithmetic operation.
   * Click on = to compute the result.
   * Use C to clear the display and reset the calculator.
3. **Error Handling:**
   * Division by zero errors are handled by displaying "Error" and resetting the calculator.

**Future Improvements**

1. **Enhanced User Interface:**
   * Styling and layout improvements for better user experience.
   * History functionality to show previous calculations.
2. **Advanced Operations:**
   * Support for more advanced mathematical functions (e.g., square root, exponentiation).
3. **Unit Tests:**
   * Implement unit tests to ensure accurate calculations and robust error handling.

**Conclusion**

The **Calculator Application** provides a basic yet functional tool for performing arithmetic operations on floating-point numbers. It demonstrates fundamental concepts of C# programming and Windows Forms development.