Kotlin Calculator Application - Detailed Documentation

1. Introduction

This document provides an in-depth explanation of the Kotlin-based Calculator application developed for Android. The calculator performs basic arithmetic operations and is designed with simplicity, usability, and maintainability in mind. The documentation includes features, user interface design, detailed code breakdown, flow of operations, and potential improvements.

2. Features

• Basic arithmetic operations: Addition (+), Subtraction (-), Multiplication (*), and Division (/).
• A clear button to reset inputs and results.
• Dynamic display updates with user input.
• Handles consecutive calculations without restarting the app.
• Prevents crashes during division by zero (returns 0.0 in such cases).

3. Application Flow

Step 1: User launches the app, sees the display and buttons.
Step 2: User enters a number using digit buttons.
Step 3: User selects an operator (+, -, *, /).
Step 4: User enters the second number.
Step 5: User presses '=' to calculate the result.
Step 6: Result is displayed and can be used for new calculations.
Step 7: 'C' button clears the input and resets all variables.

4. UI Design

The UI consists of: • A display area (TextView) for numbers and results.
• Ten number buttons (0–9).
• Four operator buttons (+, -, *, /).
• An equal button (=) for performing calculations.
• A clear button (C) for resetting the app state.

The layout is created using ConstraintLayout in activity_main.xml.

5. Code Structure and Explanation

5.1 MainActivity Class

The MainActivity class is the entry point for the application. It extends AppCompatActivity to provide backward compatibility.

5.2 Variable Declarations

• tvDisplay (TextView): Displays user input and calculation results.
• currentInput (String): Stores the current number input by the user.
• firstNumber (Double): Stores the first operand.
• operator (String): Stores the selected arithmetic operator.
• isNewOperation (Boolean): A flag to indicate if the next input is a new operation.

5.3 onCreate() Method

This method initializes the activity, sets the layout using setContentView, and connects UI components with their corresponding Kotlin variables using findViewById. It also sets click listeners for all buttons.

5.4 Number Buttons Handling

Number buttons (0–9) are added to a list for efficient handling. A for loop iterates through the list, assigning an onClickListener to each button. When pressed: • If isNewOperation is true, the current input is reset. • The button's index is appended to currentInput. • The display is updated with the new input.

5.5 Operator Buttons Handling

The operator buttons (+, -, *, /) call the setOperator() function with their respective symbol. This stores the first number, sets the operator, and flags the next input as a new operation.

5.6 Equal Button Logic

When '=' is clicked: • The second number is converted from currentInput to Double.
• The calculation is performed using a when expression based on the operator.
• Division by zero is handled by returning 0.0.
• The result is displayed and stored as currentInput for further calculations.

5.7 Clear Button

The clear button resets all variables and updates the display to '0'.

5.8 setOperator() Function

Stores the first number and selected operator. Sets isNewOperation to true so that the next input replaces the current input.

6. Future Enhancements

• Add support for decimal inputs and floating-point precision.
• Include advanced operations (square root, percentage, memory functions).
• Improve UI with Material Design components.
• Add input validation and detailed error messages.