Osama Abou Hajar – C00220135

Security Application Project Documentation

Introduction

Aim of this assignment is to develop java calculator application that would perform basic functions and would have appropriate user interface to use it.

Use of correct Java conventions, graphic user interface and functionality are the most important criteria while achieving final result. Main requirements are listed bellow.

Requirements

Main requirements are to develop graphic user interface that would allow user to perform following actions with the application:

* Addition
* Subtraction
* Multiplication
* Division
* Clear function
* Memory function
* History function

These are the most important function that must be included in application. Acceptable user interface is an advantage.

Graphic User Interface

GUI development was very simple process as it is just selecting items from menu and placing them on screen in appropriate way. NetBeans software automatically generates code such as variable name, and its constant position on screen. Graphic user interface look like

A screenshot of a cell phone

Description generated with very high confidence

1. Calculator Screen: to show the calculations operations.
2. Memory Screen: to display the date stored in the memory.
3. History Screen: to show the operations history.
4. Back Space: to delete one single number at the time.
5. Memory Re-Call: to print the number stored in the memory on the memory screen.
6. Memory subtraction: to subtract the value on the main screen from the memory.
7. Memory addition: to add the value on the main screen from the memory.
8. Memory Clear: to clear the memory and set it value to zero.
9. Clear: to clear the main screen and set its value to zero.

Calculator logic:

If %,C,◀ pressed

If +,-, \*, / pressed

Press equal, answer is displayed, proceed with next operator

Enter Second digit using keypad

Proceed with next operation

Move first digit to top of screen, display operator sign in middle

Display result at bottom part of screen

Press operator button (+, -, \*, / , %,C,◀)

Enter first digit using keypad

Java code

**private JTextField tfRes;**

// the main screen for the operations, has been initial to zero as a start value

**double num1;**

//to hold the value of the first number chosen.

**double num2;**

//to hold the value of the second number chosen.

**double result;**

// to hold the Double value of the result after the operation applied

**String operation;**

//to hold the operation sign.

**String answer;**

//to hold the result as String value and print on the screen.

**double memory;**

//to hold the memory value.

**private JTextField tfMemory;**

// to show the value stored in the memory.

**private JTextArea tfHistory;**

// to show the operations history.

**JScrollPane scrollPane;**

//to make the history screen scrollable.

**boolean clear = false;**

// a conditional variable to clear the screen after each fully operation.

**If statements for “=” button.**

**if (operation == "+")**

**{**

**result = num1 + num2 ;**

**answer = String.format("%.2f", result);**

**tfRes.setText(answer);**

**textArea.setText(textArea.getText() + "\n " + num1 +" "+ operation +" "+ num2 + " = " + result );**

**}**

**else if (operation == "-")**

**{**

**result = num1 - num2 ;**

**answer = String.format("%.2f", result);**

**tfRes.setText(answer);**

**textArea.setText(textArea.getText() + "\n " + num1 +" "+ operation +" "+ num2 + " = " + result );**

**}**

**else if (operation == "\*")**

**{**

**result = num1 \* num2 ;**

**answer = String.format("%.2f", result);**

**tfRes.setText(answer);**

**textArea.setText(textArea.getText() + "\n " + num1 +" "+ operation +" "+ num2 + " = " + result );**

**}**

**else if (operation == "/")**

**{**

**result =num1 / num2 ;**

**try {**

**result = divide(num1, num2);**

**} catch (DividebyZeroException dbze) {**

**JOptionPane.showMessageDialog( null, dbze,**

**"Error Message", JOptionPane.ERROR\_MESSAGE );**

**}**

**answer = String.format("%.2f", result);**

**tfRes.setText(answer);**

**textArea.setText(textArea.getText() + "\n " + num1 +" "+ operation +" "+ num2 + " = " + result );**

**}**

**else if (operation.equals("C")) {**

**tfRes.setText("0");**

**}**

Error log and available improvements

Two handling has been created:   
 1 – **DividebyZeroException**

**public class DividebyZeroException extends ArithmeticException {**

**public DividebyZeroException ()**

**{**

**super("Attemp divid by Zero");**

**}**

**public DividebyZeroException (String message)**

**{**

**super(message);**

**}**

2 – **NumberFormatException**

**try {**

**num1 = Double.parseDouble(tfRes.getText());**

**} catch (NumberFormatException ne) {**

**JOptionPane.showMessageDialog( null, ne,"Error Message", JOptionPane.ERROR\_MESSAGE );**

**}**

**PSEUDOCODE**

Constructor pseudocode

* Create master panel to hold components (masterJPanel)
* Set the layout to BorderLayout
* Set the gap between components to be 6 pixels horizontally and vertically
* Set background colour to grey
* Create display screen (displayJTextField)
* Display 0 as the default
* Set font Right align text
* Set background colour to white
* Do not allow text to be edited
* Display all pixels within the boundary of the display screen
* Set border to lowered bevel
* Add display screen to northern section of master panel
* Create an array to hold the button values for the keypad
* Create a panel to hold the buttons
* Set the layout to GridLayout
* 25 buttons to be used arranged 5 columns by 5 rows
* Set gap between buttons to be 2 pixels
* Set background colour to lightGray
* Create and add the buttons
* Set the text of the buttons
* Set the border of the buttons to raised bevel
* Set background colour of buttons to lightGray
* Register the buttons with a listener
* Add buttons to button panel
* Set the text colour for the CA, CE and = buttons to red.
* Add button panel to master panel
* Add master panel to frame.

Testing Documentation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test No** | **Test Objective/Scenario** | **Test Cases (input data)** | **Expected Result** | **Actual Result** | **Test Date** | **Status** |
| 1 | User clicks on a number button | Number | The number appears in the text field | Number appears | 06/04/18 | Complete |
| 2 | User enters a number, clicks on the addition button, enters another number and clicks the equals button | Number | The calculator adds the 2 numbers and returns the result | Result of addition is correct | 06/04/18 | Complete |
| 3 | User enters a number, clicks on the minus button, enters another number and clicks the equals button | Number | The calculator subtracts the second number from the first and returns the result | Result of subtraction was incorrect | 06/04/18 | Incomplete |
| 4 | User enters a number, clicks on the multiplication button, enters another number and clicks the equals button | Number | The calculator multiplies the 2 numbers and returns the result | Result of multiplication was incorrec | 06/04/18 | Incomplete |
| 5 | User enters a number, clicks on the division button, enters another number and clicks the equals button | Number | The calculator divides the 2 numbers and returns the result | Result of division was incorrect | 06/04/18 | Incomplete |
| 6 | User clicks the clear button to clear the text field | None | The text field appears empty | The text field is empty | 06/04/18 | Complete |