

# Assignment 3: CalculatorGUI

---

## Logistics

- The assignment is meant to be done individually.
- The **deadline** for this assignment is **11:59 PM on Nov. 12, 2023, PDT**.
- Academic dishonesty is unacceptable and will not be tolerated in this course.

## Get Started

For this project, we do not provide template code, you need to start from scratch. Steps for completing this assignment are in general description part.

## Submission Instructions

- During implementing this assignment:
  - **Please follow the steps in general description**

A simple way to check if you are following the steps is to run the sample tests. If every sample test runs smoothly and gets passed, then you are fine.

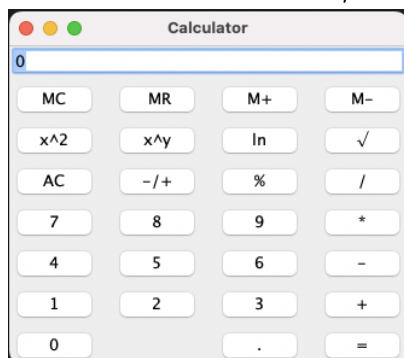
**Fail to obey this rule may lead to reduction of your final score!**

- In this project, you don't have to create a JAR file in your submission. We will use our own script to test your assignment.
- After you generate the JavaDoc, also put a screenshot of the JavaDoc in the zip file. Please do not submit **.rar** file.
- After you complete this assignment, zip up this project folder into **Assignment\_Project.zip**.
- Finally, upload it to Canvas under Assignment 3 submission link.

## General Description

The aim of this assignment is to build a calculator in Java Swing. You need to implement:

- GUI form for the calculator, the GUI will be similar to **(all buttons should be there)**



- Basic operations of calculator, such as
  - A textField to show the results

- Number buttons from 0 to 9
- Addition
- Subtraction
- Multiplication
- Division
- Square
- Power
- Square root
- Percentage
- Natural logarithm
- Decimal point button
- Clear button
- Equal button
- Memory function buttons

The preferred process to work on this assignment is:

1. Create a project with name **CalculatorGUI** (This folder will be zipped when you submit it.)
2. Create a package under src with name **calculator**
3. Create a form under package **calculator** with name **Calculator.form**, check create bound class and it will automatically create **Calculator.java**
4. On **Calculator.form**, design the GUI of calculator. Your calculator should have at least one **JTextField** and 27  **JButton**, the field name of each component should be:

1. **textField**: the textfield to show the operands and results
2. **clearBtn**: the all clear button
3. **equalBtn**: the equal button
4. **signBtn**: the sign button
5. **percentageBtn**: the percentage button
6. **sqrBtn**: the square root button
7. **logBtn**: the natural logarithm button
8. **squareBtn**: the square button
9. **addBtn**: the addition button
10. **minusBtn**: the subtraction button
11. **mulBtn**: the multiplication button
12. **divideBtn**: the division button
13. **digitBtn**: the decimal point button
14. **powerBtn**: the power button
15. **mAddBtn**: the memory add button
16. **mSubBtn**: the memory subtraction button
17. **mRecallBtn**: the memory recall button
18. **mClearBtn**: the memory clear button
19. **zeroBtn**: the number zero
20. **oneBtn**: the number one
21. **twoBtn**: the number two

- 22. **threeBtn**: the number three
- 23. **fourBtn**: the number four
- 24. **fiveBtn**: the number five
- 25. **sixBtn**: the number six
- 26. **sevenBtn**: the number seven
- 27. **eightBtn**: the number eight
- 28. **nineBtn**: the number nine

- 5. Generate the main() method to make the form functional
- 6. Implement Action listener for all 27 button to complete the functionality of calculator. **remind that inheritance is required for this part.** For example, you can define event handler subclass for click a number button: `oneBtnClicked`, `twoBtnClicked`, `threeBtnClicked` ... `nineBtnClicked` which will all extend from the base class: `numberBtnClicked`. **You will lose points if you do not implement inheritance in your code.**
- 7. Try to run your code and see if the GUI looks good and the functionality works well
- 8. Add the following method test() in your `calculator.java`, this is for testing your code

```
public void test( String button){
    switch (button){
        case "0": zeroBtn.doClick();break;
        case "1": oneBtn.doClick();break;
        case "2": twoBtn.doClick();break;
        case "3": threeBtn.doClick();break;
        case "4": fourBtn.doClick();break;
        case "5": fiveBtn.doClick();break;
        case "6": sixBtn.doClick();break;
        case "7": sevenBtn.doClick();break;
        case "8": eightBtn.doClick();break;
        case "9": nineBtn.doClick();break;
        case "%": percentageBtn.doClick();break;
        case "-/+": signBtn.doClick();break;
        case "AC": clearBtn.doClick();break;
        case "*2": squareBtn.doClick();break;
        case "sqr": sqrBtn.doClick();break;
        case "log": logBtn.doClick();break;
        case ".": digitBtn.doClick();break;
        case "+": addBtn.doClick();break;
        case "-": minusBtn.doClick();break;
        case "*": mulBtn.doClick();break;
        case "/": divideBtn.doClick();break;
        case "**": powerBtn.doClick();break;
        case "M+": mAddBtn.doClick();break;
        case "M-": mSubBtn.doClick();break;
        case "MR": mRecallBtn.doClick();break;
        case "MC": mClearBtn.doClick();break;
        case "=": equalBtn.doClick();break;
        case "txt": System.out.println("The result is: " +
textField.getText());break;
```

```

        default: System.out.println("invalid input"); break;
    }
}

```

9. We will use some test samples like following code to test the functionality of your calculator, you can try to test by yourself. **You will lose points if we can not test your code by similar test samples.**

```

public class SampleTest {
    public static void main(String[] args){
        Calculator myCal = new Calculator();

        //test addition
        System.out.println("Testing addition");
        myCal.test("1");
        myCal.test("+");
        myCal.test("2");
        myCal.test("=");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 3");

        //test subtraction
        System.out.println("Testing subtraction");
        myCal.test("1");
        myCal.test(".");
        myCal.test("2");
        myCal.test("-");
        myCal.test("2");
        myCal.test("=");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: -0.8");

        //test multiplication
        System.out.println("Testing multiplication");
        myCal.test("1");
        myCal.test("*");
        myCal.test("2");
        myCal.test("=");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 2");

        //test division
        System.out.println("Testing multiplication");
        myCal.test("1");
        myCal.test(".");
        myCal.test("2");
        myCal.test("/");
        myCal.test("2");
        myCal.test("=");
    }
}

```

```

        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 0.6");

        //test power
        System.out.println("Testing power");
        myCal.test("2");
        myCal.test("**");
        myCal.test("3");
        myCal.test("=");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 8");

        //test square
        System.out.println("Testing square");
        myCal.test("2");
        myCal.test("*2");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 4");

        //test percentage
        System.out.println("Testing percentage");
        myCal.test("1");
        myCal.test("%");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 0.01");

        //test square root
        System.out.println("Testing square root");
        myCal.test("4");
        myCal.test("sqr");
        myCal.test("txt");
        myCal.test("AC");
        System.out.println("The reference result is: 2");
    }
}

```

10. Add some comments by `/** */` to javadoc (at least for constructor() and main()) and generate the JavaDoc.
11. Put a screenshot of the JavaDoc that can show your comments in the project folder.
12. Zip the project file and submit it.

#### Here are some features that you need to implement for this CalculatorGUI:

1. The calculator should support double arithmetics, please use `Double` to store the operands.

2. The default value the app shows are zero, meaning when users run the app, the result text viewer should show 0 before any calculations, and anytime the user clicks AC. After you input a number, the 0 should be cleared. For example if you input 1 in the beginning, it should show 1 instead of 01.
3. The app should at least have 27 buttons including numbers 0 to 9, addition (+), subtraction (-), multiplication (\*), division (/), decimal point (.), equal sign (=), percentage (%), sign (+/-), square root ( $\sqrt{\phantom{x}}$ ), square ( $x^2$ ), power ( $x^y$ ), natural logarithm (ln), memory add (M+), memory subtraction (M-), memory recall (MR), memory clear (MC), and all clear (AC).
4. The order of doing arithmetic operations is:
  1. Input left operand by clicking the buttons, the textfield will show it
  2. Input an operator from addition (+), subtraction (-), multiplication (\*), division (/) and power ( $x^y$ ), the operator will not be shown
  3. Input right operand and the textfield will show the right operand only
  4. Press equal button and only the result will be displayed
  5. **You do not need to handle any error caused by wrong order of the inputting**
5. When the result is equal to an integer, it should be shown as integer. For example,  $1.0 * 2$  will be 2 instead of 2.0.
6. Numbers added after the user hits the decimal point button should be added in the decimal fraction.
7. After hitting the percentage button %, the square root ( $\sqrt{\phantom{x}}$ ), square ( $x^2$ ), natural logarithm (ln), the numbers shown in the calculator (result area) should be directly transformed. For instance, 43 in the result area and hitting % should result in 0.43. Users should be able to see the value without hitting the equal button.
8. The sign button should toggle the sign of the input number, the default sign is positive.
9. If you are not familiar with the memory function, the following link might be helpful.  
<https://www.calculatorsoup.com/calculators/math/basic.php>
10. Please correctly implement all the rest of the arithmetic operations normally as a typical calculator does.
11. The whole set of numbers except the one stored in the memory should be removed upon hitting all clear(AC) and present 0. The memory is cleared only when the MC button is hit.
12. Errors caused by some invalid operations need to be handled. Whenever an invalid operation (e.g., divide 0, sqrt a negative number) has been detected, the result should show "ERROR" after hitting = button.