

CSE 223 Programming Assignment #3 Spring 2019

May 2, 2019

Introduction

For your third programming assignment, you're going to implement a GUI-based calculator in Java using Swing. You should use Eclipse+Windowbuilder, as discussed in class. You must use the Java JDK 1.8 or 1.9.

The design of the calculator is mostly up to you. There's flexibility in how it looks and works; and there are options as far as the specs of what it is capable of (see below).

Important Note: Do not use any online sources to look up code for a calculator. There are a lot of tutorials and examples related to this type of project. If you use any of these, you will not learn as much, and this will be a problem down the road. Ask questions in class, on Canvas, or during office hours. Stay away from Google and other sites, and try to work through the difficult spots with trial and error, test code, and so on.

Requirements

Your program should display something like a “standard” calculator: buttons 0-9 for inputting numbers; operators including + - * / = and a display for showing numbers as they're entered as well as results. Output must use GUI components: printing to stdout is not acceptable. Neither is doing non-GUI things, even if they happen in a GUI (for example, displaying in a GUI “Enter first number, hit enter when done” and having the user type in a number and then hit enter). If you're in doubt about your design, *ask*.

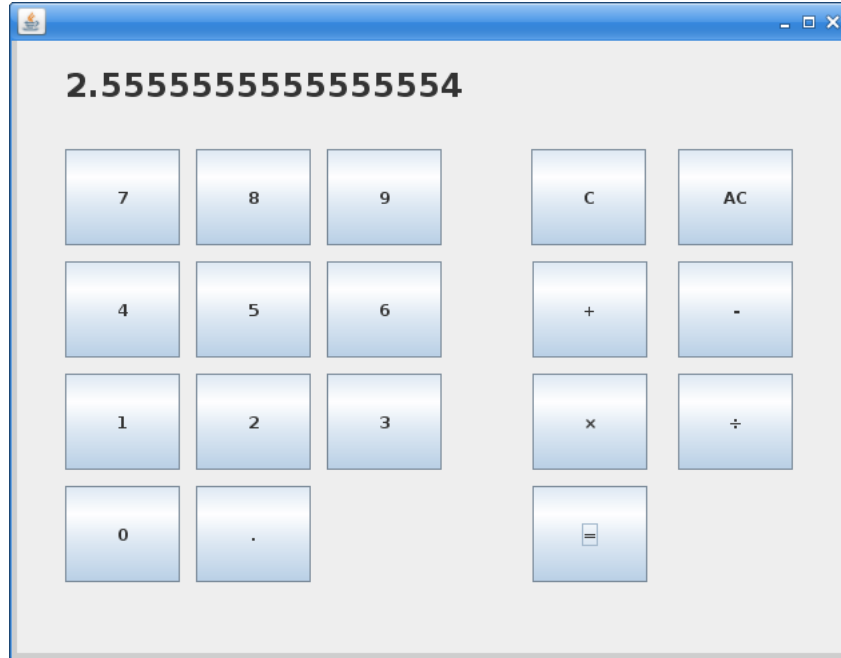
Grading

This assignment is worth 100 points, and the maximum you can score is 100 points. You may choose how you wish to build up your grade, using the following grading table:

- Documentation: the usual grading for comments, comment block in header, etc: 10 points max
 - Correct submission, meaning file names, method of exporting the zip file, etc: 5 points
 - Functional behavior and design: 85 points max, awarded as follows:
 - Basic frame looks like a calculator, has buttons and a display: 15 points
 - Buttons let you enter a number which is displayed properly: 20 points
 - + - * / work correctly on integers: 30 points
 - Clear button (clears last entry): 5 points
 - Constants (e.g. 2+3=(displays 5) 6= (displays 8): 5 points
 - Works with real numbers (numbers containing a decimal point): 15 points
 - Scientific functions (at least 4, e.g. sin, cos, sqrt, log, exp, etc): 5 points
 - Handles division by 0 in a reasonable way: 5 points
 - Resizable: everything still looks well-proportioned when you resize the frame: 5 points
- DEDUCTIONS**
- Doesn't treat the "0" key correctly (e.g. hitting 007 displays 007 instead of 0): -5 points
 - (If you implement a decimal point): hitting double decimal points causes numbers like 1.5: -5 points
 - Program crashes (throws an exception): -15 points

Sample Appearance

Here's a sample of what your frame might look like (this is only an example):



Suggested Approach

We'll discuss this in class.

Submission

Export your file from Eclipse as described in class, and upload the zip file to Canvas by the deadline. Be sure to download your file into a clean directory and import it as a new project into Eclipse (and then test it) to make sure your submission is correct. This is not a simple zip file of your code: it's an Eclipse-specific format, and must be created as described in class.