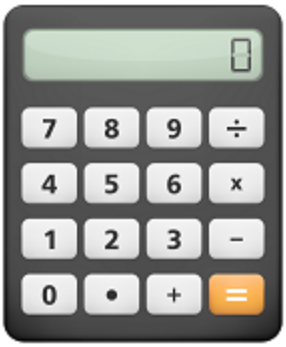
Java FX Part #2

Re-submit Assignment

Due Jan 31 by 11:59pm **Points** 100 **Submitting** a website url

Java FX Part #2



This is a continuation of part #1 of the calculator assignment, which can be found here: Java FX Part #1. In this part we will write the underlying logic of the calculator and connect all events between the UI and logic layers.

Objectives

Course:

- Implement a program using a modern software framework that leverages architectural patterns, such as model-view-controller, object-relational-mapping or dependency injection.

Module:

- To write a graphical program using a modern Java framework.
- To identify existing design patterns in a modern framework.
- To apply the model-view-controller paradigm in a GUI-based application.

MVC in Action

Adhering to the MVC paradigm, we will not include any logic in our CalculatorUI class (our view). Instead we will write a new Calculator class (our model) that contains all state information and event handlers for interacting with the calculator. The Calculator class should manage the following:

Data

- Stores the operands for an operation
- Stores the operator for an operation

Event Handlers

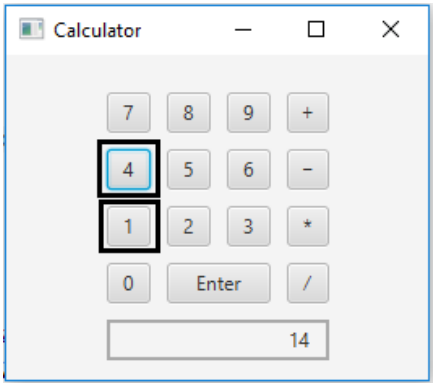
- Responds to numeric entry to the calculator
- Responds to operator entry (+, -, *, /)
- Responds to the enter key

Actions

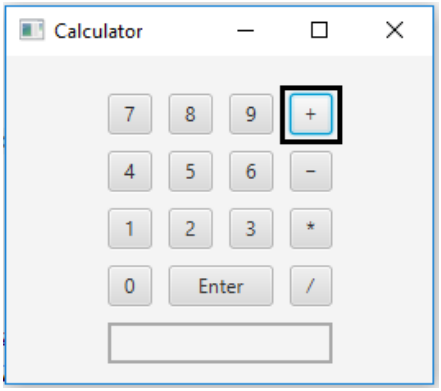
- After a change has been made through the event handlers above
- Updates the UI to show the current output of the calculator, as described below

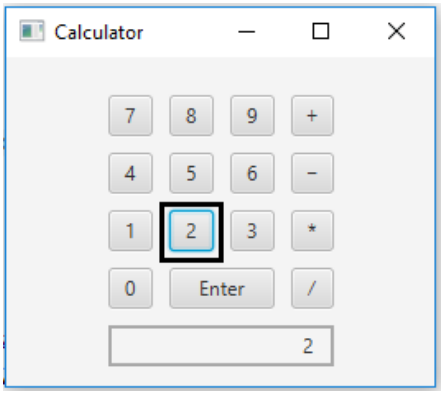
Expected Behavior of the Calculator

1. Clicking on numeric buttons should show their output on the calculator from right-to-left.

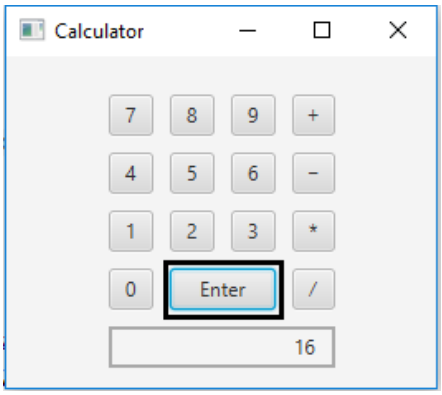


2. Clicking on an operator should save the operator to the calculator, clear the output on the calculator and let the user enter a second number.

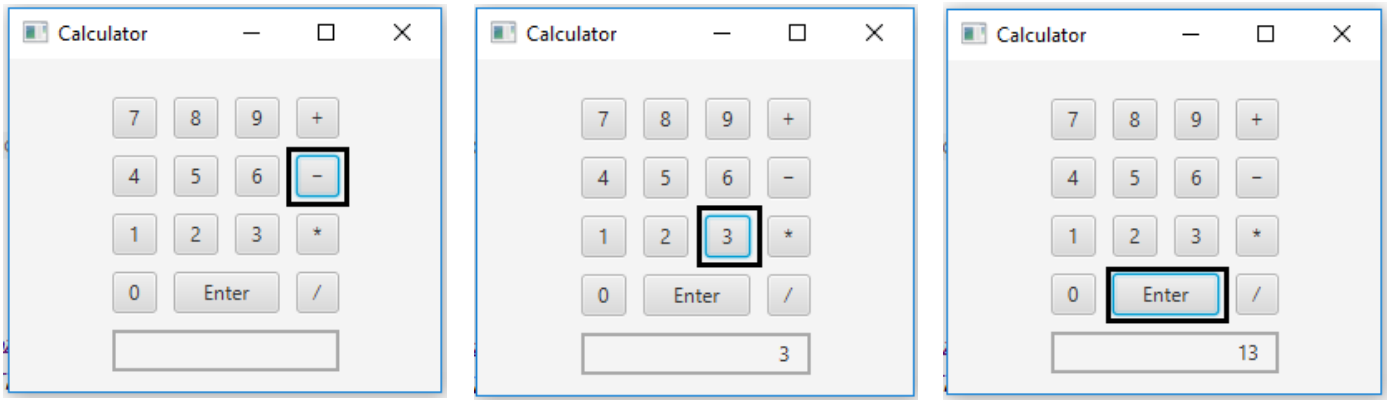




3. Clicking enter should display the results of both operands and the saved operator.



4. If a user then clicks on another operator your program should return to step #2 above and let the user use the current result as the first operator. This can be seen in the screenshots below.



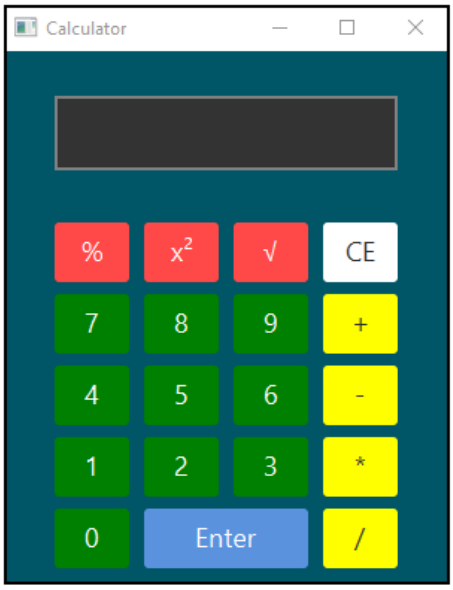
Submission

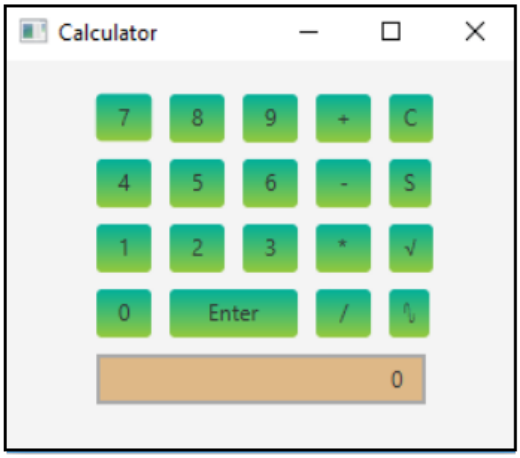
- All projects must be completed through the IntelliJ IDE
- Your project should be a Maven project with appropriate settings
- Your project files should be saved to a private GIT repository which is shared with myself
- All files must have proper documentation (comments and full Javadocs)
- All files must follow our style guidelines

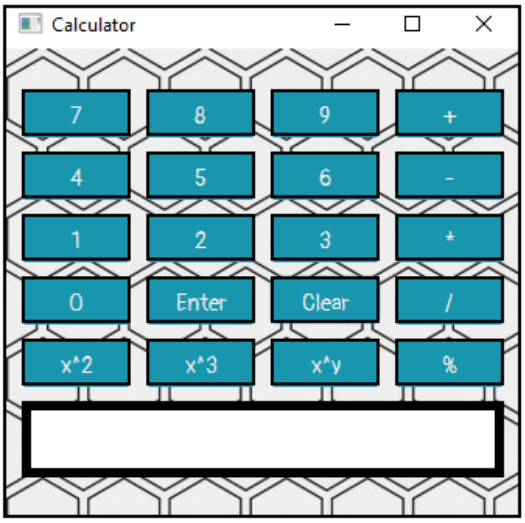
Extra Credit (5 points)

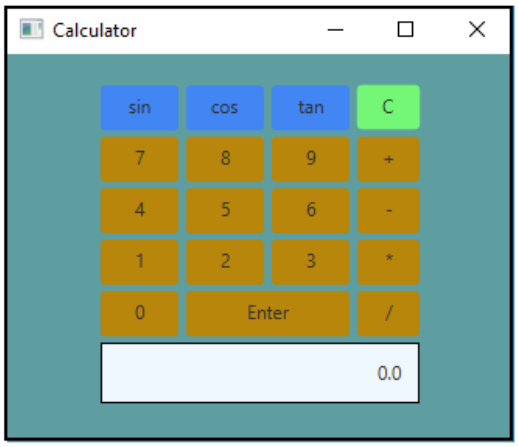
Add an extra feature to the calculator. Points will be awarded based on effort. Here are some examples, feel free to think up other features:

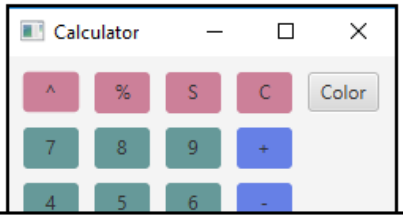
- A reset button (often denoted as C or CE)
- A store button that saves a result for later use
- Adding trigonometric buttons (sin(x), cos(x), tan(x), ...)
- Adding a square root button

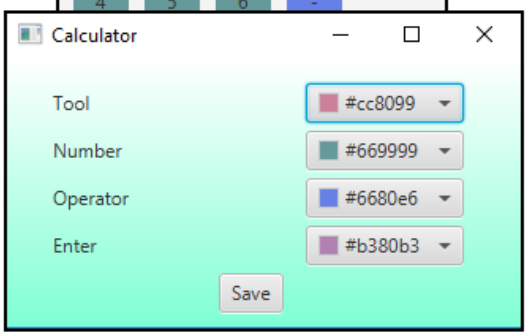


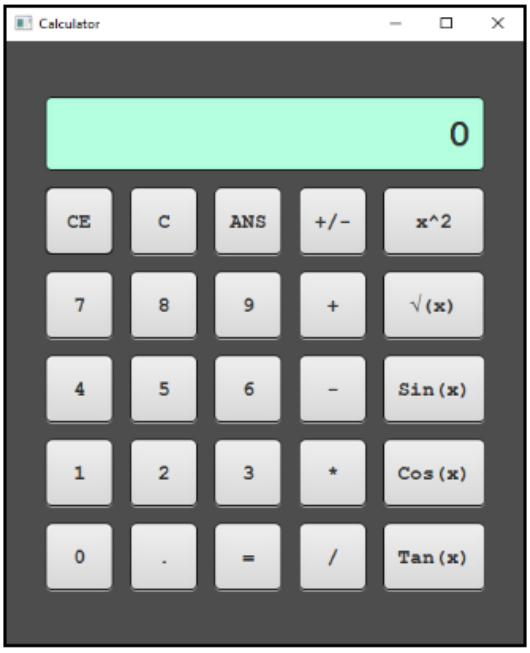












Java FX Assn. Part #2 Rubric (1)

Criteria	Ratings		Pts
Numeric buttons can be pressed and entries are displayed in the results label at the bottom of the screen. Numeric characters are displayed from left-to-right, as they are entered.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Pressing an operator key saves the current numeric entry and operator to the calculator. This also clears the output of the calculator.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Clicking enter after entering both operands and an operator displays the results of the calculation to the calculator.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
After clicking enter a user can then enter a new operator and operand and seeing a new result using the previous calculation results.	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Your program adheres to the MVC paradigm. All logic and data for the calculator is located outside of the CalculatorUI class.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
IntelliJ project is set up correctly as a Maven project. All project files are saved to GIT in a private repository.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
Full Javadocs are provided for all classes and public methods.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
All code follows the style guide provided.	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
			Total Points: 100.0