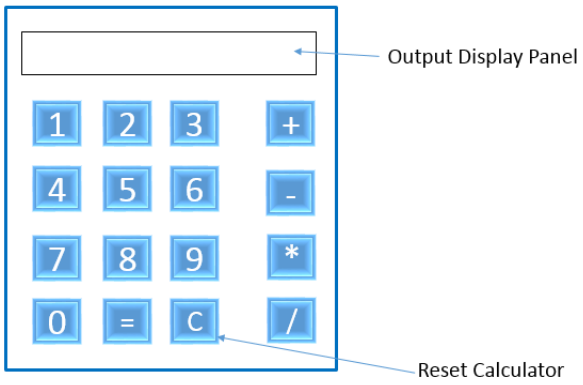
SE 311: Software Architecture II, Winter 2018-2019

Homework Assignment 4

Design and Implement a simple calculator client and a log server.

1. Requirements: The user should interact with a simple calculator with a Graphical User Interface as follows:



Using this GUI, the user should be able to run the following functions:

R1. At the beginning, the Out Display Panel is empty

R2. The user should be able to conduct simple math calculation by clicking the buttons, and the result should be displayed in the Output Display Panel. Following are some example input and output that should be supported:

Index	Precondition	User action	Post condition
C1	Initialized or reset	The user clicks buttons “3”, “+”, “5”, and “=” consecutively.	The system displays “8”,
C2	Initialized or reset	The user clicks buttons “3”, “3”, “+”, “5”, “5” and “=” consecutively.	The system displays “88”
C3	Initialized or reset	The user clicks buttons “4”, “+”, “5”, “*” (at this point, the system should display “9”), then clicks “6” and “=”	The system displays “54”.
C4	Initialized or reset	The user clicks a non-digit button	The system remains unchanged
C5	Any state	The user clicks “C” bottom	The system resets and displays nothing
C6	A non-digit button is clicked	The user clicks a non-digit button other than “C”	The system displays and error message with two choices, “reset” or “discard”. ---If the user choses “reset”, the system reset; ---If the user choses
C7	Initialized or reset	The user clicks a digit and then clicks “=”	

			"discard", the system is set back to the previous state
C8	A previous calculation was successful and the previous result was displayed	The user clicks a digit	The system gets ready for a new calculation
C9		The user clicks a non-digit	The system resets and displays nothing

R3. Whenever a math calculation is successfully conducted, that is, the "=" button is clicked, the client will send a message to the server, which will record the successful math equation, such as "3+5+8=16"

R4. The server should be able to display 1) the total number of all the successful calculated math equations, 2) the list all the equations.

2. Instructions
   (1) You must use a MVC combined with Socket-based Client-Server architecture
   (2) You must use the following design patterns:
      a. Composite pattern
      b. Visitor Pattern
      c. Observer pattern
      d. State Pattern

You need to turn in the following items from bbLearning:

Submission 1: UML Class diagram

Turn in a ***1-page PDF*** UML class diagram showing the design of the calculator client. The diagram has to be generated from a computer program. Do not hand write your UML.

Submission 2: UML Component diagram

Turn in a ***1-page PDF*** UML component diagram showing Client component, Server Component, and their interfaces. The diagram has to be generated from a computer program. Do not hand write your UML.

In the component diagram, please make it clear which classes belong to which component. In the client component, please model clearly which classes belong to Model, View and Controller, which should be the inner components within Client.

Submission 3: source code in a Java project

Turn in two java projects, one for the server component and another for the client component, which we can open, view the source code, and run the program. You should also include a readme.txt to indicate how to configure and run the program.

Appendix:

The state transition diagram of the calculator: