Calc Maestro Design Doc

Created by: Ankit, Teja, Reshil

Project Overview + Core Functionalities:

This project’s goal is to deliver a simple calculator with graphing and various other mathematical functions in addition to performing basic numerical operations. It should be easy to use and return calculations quickly. It will be written mainly in Java and contain a GUI.

Formalized Features (subject to addition):

Simple Numerical Operations

2D graphing and visualization

Trigonometric Functions and reverses

Computation History

Simple GUI in JFrames

Other various functions

Used Libraries:

JFrames and AWT for UI

All other imported functions come from default java packages

Proprietary Tools Used:

Visual Studio Code

MS Paint

Github Desktop

Architecture/Design Philosophy:

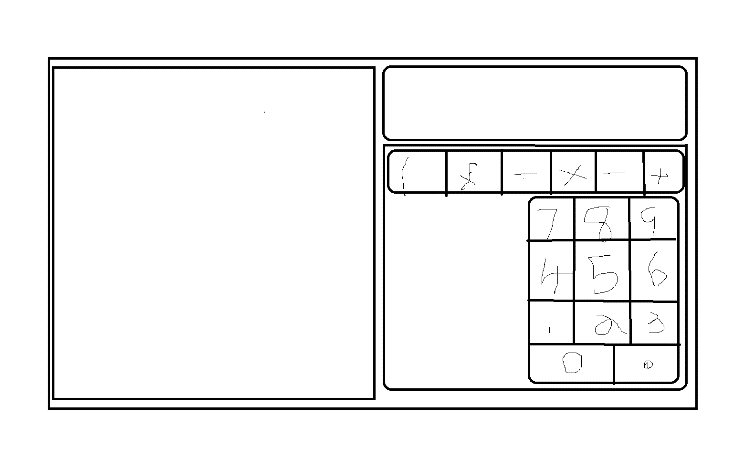
The application will be built in a modular 4-file structure as per follows:

UI.java : Uses the JFrames library to generate a GUI for the end user that passes their input to a generated calculator object and displays the result. Users should run this script to use the calculator. GridBagLayout from AWT will be used as a template.

Calculator.java: Receives a string expression and uses Parser.java and Engine.java to solve it. Also takes in graphing requests in the form of a String expression, double interval, double domain start, and domain end.

Parser.java: Heart of the program. Encodes an expression given into character format using a function called FindNumbers() and stores original data within a list in Engine.java. This character driven approach is mainly to maintain unique identifiers for each result, although the result may be numerically the same, thus allowing easy result reuse and traceability, greatly increasing speed in tasks such as graphing. Upon a time for an operation, the character will be converted into an int index to recall the ‘real’ number from the Engine.java list. Another core function found within this module is pemdasSimply which recursively goes through the encoded result to perform operations according to the rules of pemdas using Engine.java. The answer is then decoded back into a double before returned to the user.

Engine.java: Stores aforementioned list of numbers used for calculation and resultants in double format. Uses this with indexes derived from Parser.java to perform operations.

User interface:

(basic mockup of design)

The interface is to be displayed in a 2-panel format. The left panel will contain output from the calculator that will alternate between the graphing and non-graphing mode of the calculator. The graphing output will be simply, a graph, and the non-graphing will show expression resultants and the history of the calculator.

The right panel will contain an expression field on the top for users to type their wished calculation into. Below this a number pad and operators buttons operators will be there for users to use as an alternate input mechanism. The rest of the space will contain buttons for additional functions such as trigonometry.