New Era University
College of Computer Studies

Rm. 247-B, High School Annex B, New Era University
Tel. No.: (+632) 981-4221 loc 3825
E-mail: computerstudies@neu.edu.ph

Project Documentation

CCC112-18

1CICS_Pet

Group 7

Members Name:

John Raiven D. Olazo

John Rey V. Torigossa

Instructor

Prof. Edilberto L. Simbulan Jr.

Scientific Calculator

Project Description and Features

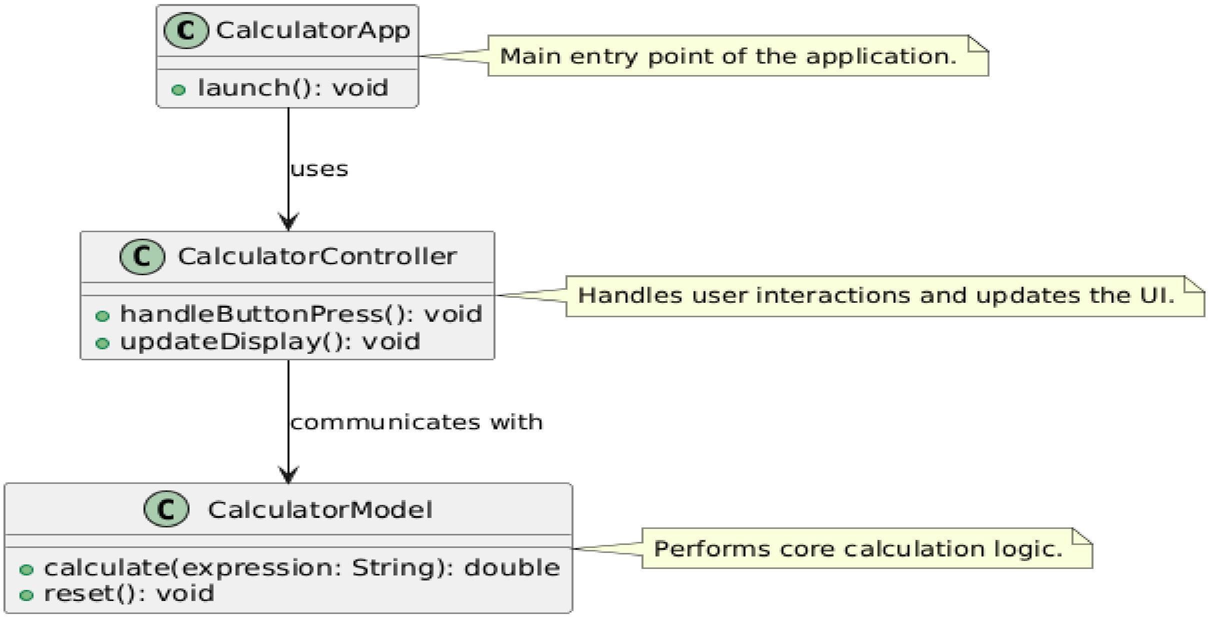
Project Overview

Scientific Calculator FX is a JavaFX-based application developed as an Object-Oriented Programming (OOP) final project. It offers users a comprehensive tool for performing basic arithmetic and advanced scientific calculations. Leveraging JavaFX's capabilities, the application provides a responsive and user-friendly interface.

Key Features

- Basic Arithmetic Operations: Addition, subtraction, multiplication, and division.
- Scientific Function: Trigonometric (sin, cos, tan), logarithmic (log, ln), exponential, square roots, and powers.
- Algebraic Tools: Solve quadratic equations (e.g., 2x^2+4x+2), Factor expressions (e.g., 2x^2+4x+2) or Expand binomials (e.g., (x+2)(x+5)).
  Memory Functions: M+, M-, MR, MC for storing and retrieving values
- Theme Toggle: Easily switch between Light and Dark mode for better visual experience
- Responsive UI: Resizes smoothly and adapts across different screen sizes and window dimensions
- Error Handling: Displays user-friendly "Error" messages when evaluating invalid expressions

Class Diagram


CalculatorApp
launch(): void
Main entry point of the application.
uses
CalculatorController
handleButtonPress(): void
updateDisplay(): void
Handles user interactions and updates the UI.
communicates with
CalculatorModel
calculate(expression: String): double
reset(): void
Performs core calculation logic.

Description of Main Classes and Their Relationships

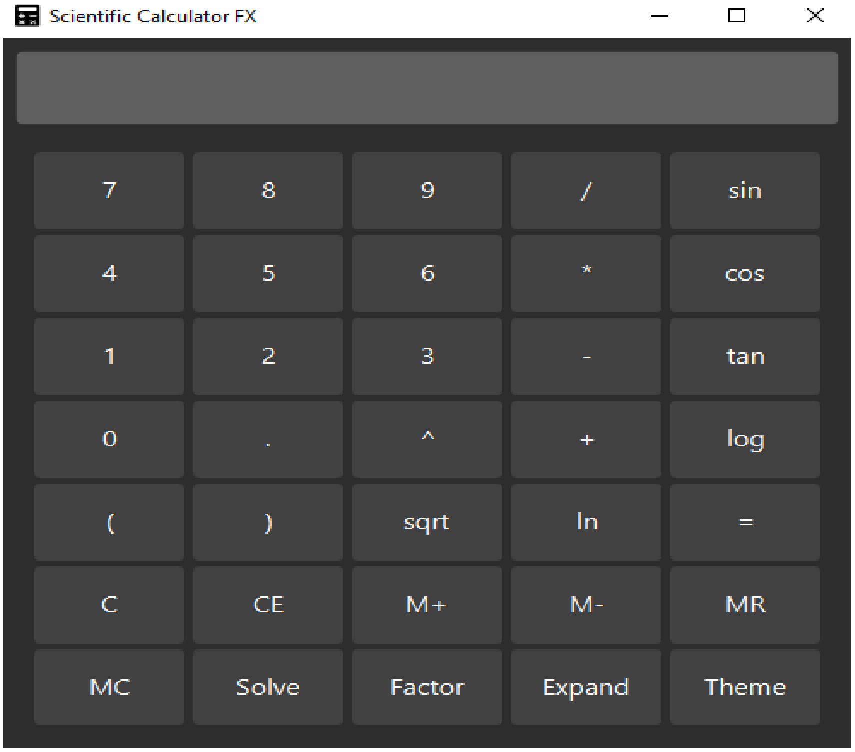
1. CalculatorApp: This is the entry point of the application, extending the `Application` class from JavaFX. It overrides the `start()` method to set up the primary stage and scene, initializing the user interface.
2. CalculatorController: Acting as the intermediary between the user interface and the business logic, this controller manages user interactions. It contains methods like `handleButtonPress()` to process button clicks and `updateDisplay()` to refresh the calculator's display.
3. CalculatorModel: Encapsulating the core calculation logic, this class performs computations based on user input. Methods such as `calculate()` handle the evaluation of expressions, while `reset()` clears the current calculation state.

The relationships among these classes follow the Model-View-Controller (MVC) design pattern:

- CalculatorApp initializes the application and sets up the user interface, acting as the View.
- CalculatorController manages user input and updates the View accordingly.
- CalculatorModel performs the actual computations, representing the Model.

Screenshots

Calculator Main Interface:


Scientific Calculator FX
7
8
9
/
sin
4
5
6
*
cos
1
2
3
-
tan
0
.
^
+
log
(
)
sqrt
ln
=
C
CE
M+
M-
MR
MC
Solve
Factor
Expand
Theme