New Era University
College of Computer Studies
Res. 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.

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

Scientific Calculator

Project Description and Features

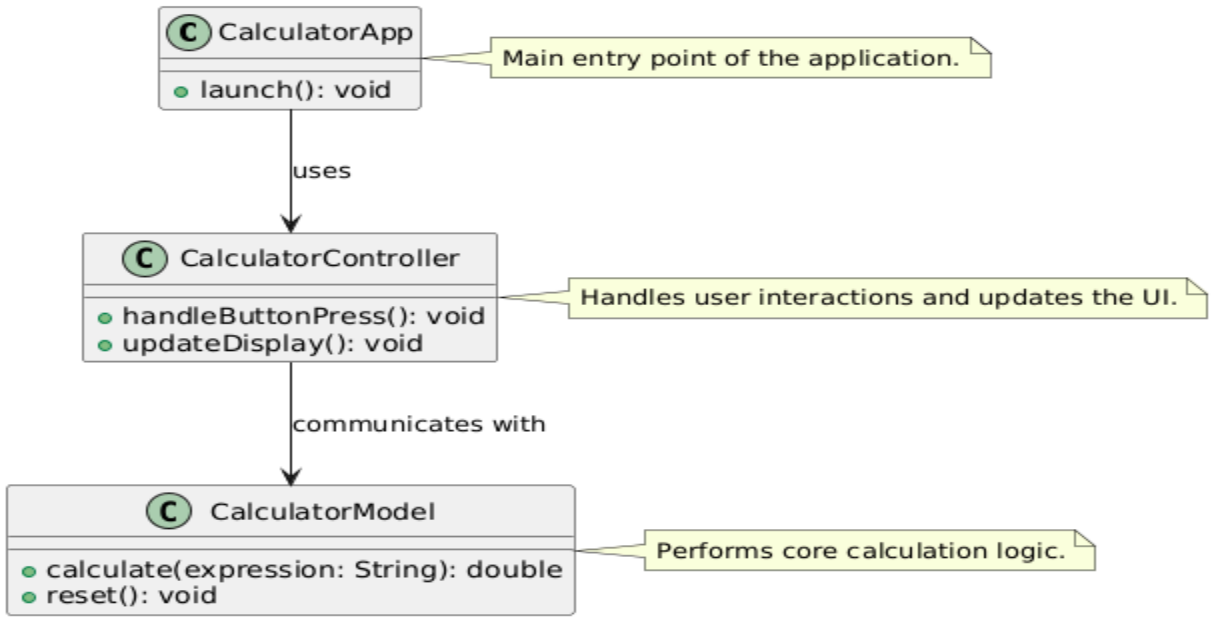
Project Overview

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: Supports addition, subtraction, multiplication, and division.
- Scientific Functions: Includes trigonometric (sine, cosine, tangent), logarithmic, and exponential calculations.
- Responsive User Interface: Utilizes JavaFX to ensure compatibility across various screen sizes and devices.
- Object-Oriented Design: Employs OOP principles for modularity and ease of maintenance.

Class Diagram



Description of Main Classes and Their Relationships

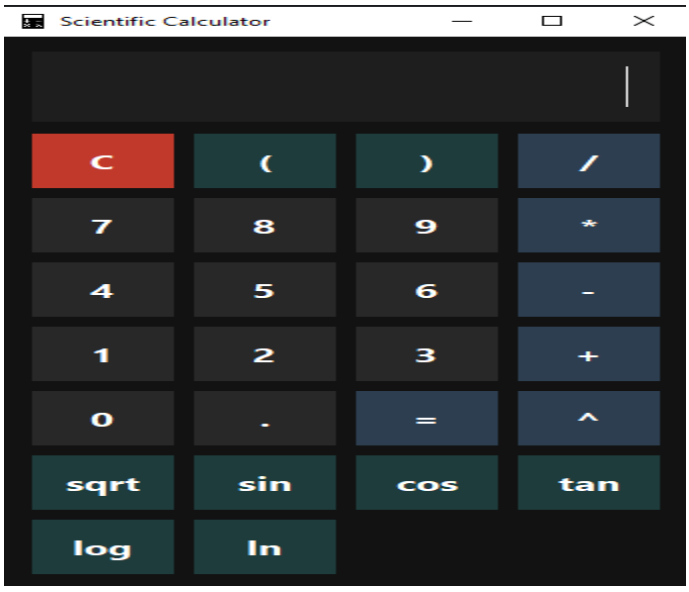
1. CalculatorApp: This is the entry point of the application, extending the `Application` class from JavaFX. It overrides the `start()` method to set up the primary stage and scene, initializing the user interface.
2. CalculatorController: Acting as the intermediary between the user interface and the 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

1. **Calculator Main Interface:**



2. **Scientific Functions Panel:**