**Calculator Project Documentation**

**Overview**

This project is a Java-based calculator application built using JavaFX for the user interface. It supports basic arithmetic operations, trigonometric functions, and uses external file handling to store the calculation history.

**Classes**

**1. Calculator**

This is the main class that initializes and manages the calculator's GUI.

* **Fields**:
  + **static final double PI**: Constant value for π.
  + **static boolean firstTIME**: A boolean flag, unused in the current implementation.
* **Methods**:
  + **public static void main(String[] args)**: Launches the JavaFX application.
  + **public void start(Stage primaryStage)**: Initializes the UI components, including the display and buttons, and handles stage configuration.

**Key Functionality:**

* Configures a grid-based layout (**GridPane)** to organize buttons.
* Handles button initialization by calling **Buttons.createButton.**
* Defines the application scene and sets it to the primary stage.

**2. Buttons**

This class is responsible for creating and styling the buttons used in the calculator.

* **Methods**:
  + **static Button createButton(String text, Label displayLabel, GridPane gridPane)**: Creates a button with specific text, styles it, and defines its behavior.
* **Button Actions**:
  + **C**: Clears the display.
  + **=**: Evaluates the current expression using **exp4j** library and updates the display. The result is saved to a file using **FileHandling.writeFile**.
  + **DEL**: Deletes the last character from the display.
  + **π**: Appends the value of π to the display.
  + Other buttons: Appends the respective character to the display.
* **Hover Effects**:
  + Changes button colors when the mouse enters or exits the button area.

**3. FileHandling**

This class manages reading, writing, and deleting the calculation history stored in a text file.

* **Fields**:
  + **static final String FILE\_PATH**: Path to the file storing calculation history.
* **Methods**:
  + **static void writeFile(String expression)**: Appends a given expression to the history file.
  + **static ArrayList<String> readFile()**: Reads the history file and returns its content as an ArrayList of strings.
  + **static void deleteFile()**: Deletes the history file.

**Key Features**

1. **Graphical User Interface**:
   * Designed using JavaFX with a clean and intuitive layout.
   * Styled using inline CSS for a modern look.
2. **Basic Arithmetic Operations**:
   * Addition, subtraction, multiplication, and division.
3. **Trigonometric Functions**:
   * SIN, COS, and TAN buttons can be extended to calculate trigonometric values (not implemented in the provided code).
4. **Persistent History**:
   * Stores all evaluated expressions and results in a file.
   * Allows users to view past calculations by reading the file.
5. **Error Handling**:
   * Displays "Error" on the calculator display for invalid expressions.
6. **Hover Effects**:
   * Provides visual feedback when hovering over buttons.

**Dependencies**

* **JavaFX**: Used for creating the graphical user interface.
* **exp4j Library**: Used for evaluating mathematical expressions.
* **Java I/O**: For file handling (reading and writing history).

**How to Extend**

1. **Add Scientific Functions**:
   * Enhance the **createButton** method in the **Buttons** class to handle scientific calculations (e.g., logarithms, exponents).
2. **History Display**:
   * Add a button to display the calculation history using **FileHandling.readFile**.
3. **Error Handling Improvements**:
   * Provide more descriptive error messages for invalid inputs.