**Assignment 3**

**Member: 22F-3644,22F-3669**

**Documentation**

**Introduction**

The Calculator Project is a Java-based application that provides a graphical user interface (GUI) to perform basic arithmetic operations. Designed with simplicity and ease of use, the project includes robust functionality for addition, subtraction, multiplication, division, and decimal operations. The application is built using the **Swing Framework** for the GUI and adheres to modern programming principles.

## ****Features****

### **Functional Features:**

1. **Basic Arithmetic Operations**:
   * Addition
   * Subtraction
   * Multiplication
   * Division
2. **Decimal Support**: Handle decimal-based arithmetic.
3. **Error Handling**:
   * Prevention of division by zero.
   * Handling boundary values and exceptions.
4. **Clear Functionality**: Reset all calculations and display.

### **Interface Features:**

1. User-friendly GUI with a dark theme.
2. Responsive buttons with rounded edges.
3. Support for real-time result updates.
4. Multi-threaded test cases for performance validation.

## ****Code Overview****

### **1. Main Application**

The **Calculator** class provides:

* GUI initialization (JFrame and JPanel setup).
* Event handling via the ActionListener interface.
* Arithmetic logic encapsulated in the calculateResult() method.
* Helper methods for operator handling (handleOperator).

**Key Components:**

* **Display Panel**: Shows the current value.
* **Button Panel**: Includes number buttons (0-9) and operation buttons.

### **2. Custom Button Design**

The RoundedButtonUI class ensures visually appealing, rounded buttons with anti-aliased rendering.

## ****Testing and Validation****

The **CalculatorTest** class uses JUnit 5 to validate all core functionalities.

### **Test Categories:**

1. **Arithmetic Operations**:
   * Tests for addition, subtraction, multiplication, and division.
   * Includes cases for positive, negative, and decimal numbers.
2. **Boundary Value Analysis (BVA)**:
   * Handles extreme values like Integer.MAX\_VALUE and Integer.MIN\_VALUE.
3. **Equivalence Class Partitioning (ECP)**:
   * Tests across valid and invalid value ranges.
4. **Error Scenarios**:
   * Division by zero.
   * Unexpected input handling.
5. **CSV-Driven Tests**:
   * Uses external CSV files to validate batch operations.

## ****Key Classes and Methods****

### **Calculator**

| **Method** | **Description** |
| --- | --- |
|  |  |
| createButton() | Creates styled buttons with custom properties. |
| handleOperator() | Captures operator button click and stores state. |
| calculateResult() | Executes the arithmetic operation based on stored values. |

### **CalculatorTest**

| **Method** | **Description** |
| --- | --- |
| testAdditionPositiveNumbers() | Verifies addition of two positive numbers. |
| testSubtractionBoundaryValues() | Tests subtraction with boundary values. |
| testDivisionByZero() | Validates that division by zero throws an error. |
| testOperationsFromCSV() | Executes operations defined in an external CSV. |

**Technologies Used**

1. **Programming Language**: Java
2. **Frameworks**:
   * Swing for GUI
   * JUnit 5 for testing
3. **Tools**:
   * IDE: IntelliJ IDEA / Eclipse
   * Version Control: Git
   * Testing: JUnit Console and IDE Integration

## ****How to Run****

### **Prerequisites:**

1. **Java Development Kit (JDK)**: Version 8 or above.
2. IDE with support for Java (e.g., IntelliJ IDEA, Eclipse).
3. CSV files (optional for test cases).

### **Steps:**

1. **Clone the repository:**

bash

Copy code

git clone <repository-url>

1. **Compile the project:**

bash

Copy code

javac Calculator.java

1. **Run the application:**

bash

Copy code

java Calculator

1. **Execute Tests:**

bash

Copy code

mvn test

**Future Enhancements**

1. **Scientific Calculator Features**:
   * Trigonometric and logarithmic functions.
2. **Localization**:
   * Multi-language support for the UI.
3. **Advanced Error Handling**:
   * Detailed messages for user errors.
4. **Mobile Compatibility**:
   * Port the project to Android/iOS.

## ****Conclusion****

This project demonstrates a robust implementation of a calculator application, emphasizing usability and reliability. The accompanying test cases ensure thorough validation of the codebase, making it a reliable and extensible solution.