Matrix Calculator

Overview

Matrix Calculator is an Android application that performs various matrix operations using native C++ code with the Eigen library. The app allows users to perform addition, subtraction, multiplication, element-wise division, and proper mathematical division using matrix inverse.

Features

- Matrix operations: add, subtract, multiply, element-wise divide, and divide using inverse
- Support for matrices of any dimension
- Real-time matrix preview as you type
- Multi-format input: space-separated or new-line formatted
- Elegant UI with Material Design components
- Clear visual presentation of input matrices and results
- Comprehensive error handling with helpful messages

Technical Implementation

The app uses a hybrid approach combining Kotlin for the Android UI and C++ with the Eigen library for the mathematical operations:

Android/Kotlin

- Implements the UI and input handling
- Provides real-time matrix previews
- Manages data validation and error handling
- Interfaces with native code through JNI

C++/Eigen

- Performs efficient matrix calculations
- Handles matrix operations: addition, subtraction, multiplication, element-wise division, and division using inverse
- Uses the powerful Eigen library for vector and matrix operations

How to Use

Input Matrices

1. Enter the dimensions (rows and columns) for both matrices
2. Enter the elements for each matrix in the provided text fields

- Elements can be entered as space-separated values in a single line
- Alternatively, use a new line for each row of the matrix

3. A live preview will show how your matrix looks as you type

Perform Operations

- **Add**: Adds matrices element by element (requires same dimensions)
- **Subtract**: Subtracts the second matrix from the first (requires same dimensions)

- **Multiply**: Performs matrix multiplication (requires columns of first = rows of second)
- **Divide (Element)**: Performs element-wise division (requires same dimensions)
- **Divide ($A \times B^{-1}$)**: Performs mathematical matrix division by multiplying by the inverse (requires second matrix to be square and invertible)
- **Clear All**: Resets all inputs and results

View Results

- The result appears in the "Result" section at the bottom
- Results are formatted for easy readability with proper matrix notation
- Error messages appear when operations cannot be performed

Implementation Notes

- Uses JNI (Java Native Interface) to call C++ code from Kotlin
- Implements proper error handling for matrix operations
- Provides real-time validation of inputs