Scientific Calculator

Team Members:

Sandesh D Sathyanarayana, Gautham Kashim, Hasil Sharma

Basic Description:

Scientific calculator is an Android Application that implements the scientific functionalities like trigonometric operations, Number systems (bases like Binary, hexadecimal and decimal), Matrix operations, Complex Numbers operations and Combinatorics. It also supports the basic functionalities like addition, subtraction, multiplication, division, logarithms and square roots etc. Few important functionalities in each of the screen is listed below.

1. Trigonometric: sin, cos, tan, sinh, cosh and tanh both in degrees and radians
2. Number System: Three bases binary, hexadecimal and decimal operations and conversions
3. Combinatorics: permutations, combinations, generic powers, logarithms, square roots etc.
4. Matrix Operations: Addition, subtraction, transpose and inverse
5. Complex Numbers: Addition, subtraction, multiplication, inverse and division etc.
6. Compound operations are also supported for all the above mentioned functions.

Description of files:

- app/src: main source code of our application
    - main/ooad: Folder which has the main source code built using the principles of Object-oriented design. It has two parts front end and back end.
        - backend/data/types: This folder has the files needed for all the data types of the backend source code. Ex: double, integer and type information of each object. We have decoupled the all possible data elements we would use in our operations
            - ComplexElem.java: Complex data elements of our code
            - DoubleElem.java: Double data type elements decoupling
            - ElemFactory.java, ElemType.java, ElemTypeEnum.java: Elementary type of elements like integers, enum etc.
            - IntegerElem.java: Integer elements
            - OperatorEnum.java: operators' elements of the code.
        - backend/data/operators: This folder has the operator's information, used in our code. This is where we have decoupled the usage of operators and number of arguments that they take.
        - backend/la/matrix.java: File that has all the functionalities we would use in the matrix operations.
        - backend/la/exceptions: This folder has the files regarding the exceptions we would use in our backend code. We have decoupled the exceptions so they could be independently used in the code. Following are the files for the same reason.
            - IncompatibleMatricesException.java
            - MatrixNonSquareException.java

- NonInvertibleException.java
- SingularMatrixException.java
- frontend/screens/: There are five screen in our application and they share common steps and are captured in "CommonScreenElements.java"
  - CombinatoricsScreen.java: Java code for combinatorics operations
  - ComplexOperationsScreen.java: Complex number systems Java activity file
  - MatrixOperationsScreen.java: Matrix operation activity java code
  - NumberSystemScreen.java: Number System operations activity java code
  - TrignoScreen.java: Trigonometric operations activity java code
- app/src/main/AndroidManifest.xml : Manifest file of Android application to link the activities and there layouts.
- App/main/res: All the resources needed to build the front-end layout of the applications
  - Layout: Directory that has all the front end code of our screen layouts
    - activity_combinatorics.xml: Layout code for Combinatorics operations screen
    - activity_complex.xml: Layout code for Complex numbers operations screen
    - activity_matrixoperations.xml: Layout code for Matrices operations screen
    - activity_number_system.xml: Layout code for Number System with different bases operations screen
    - activity_trigonometric.xml: Layout code for trignometric operations screen
  - app/main/res/values: Folder which has files with all the string constant being defined and used in the layout design.
    - Colors.xml: xml file which has color constants
    - Dimen.xml: xml file which has dimensions constants
    - Strings.xml: xml file which has strings constants
    - Styles.xml: xml file which has style of CSS file constants
  - app/main/mipmap* : Folder that has the images needed to create the basic layout of the main Android application.
  - app/main/res/drawable: Files and images being used in the layout creations.
    - Myborder_grey: grey colored button structurer file used in all layouts
    - Myborder_yellow.xml: grey colored button structurer file used in all layouts
    - myborder_black.xml: Common xml file that defines the border structure for black colored buttons created in all layouts

    - test/java/com/ooad/frontend: Folder that has “ExampleUnitTest.java” that has unit test to test out front end creation of layouts.
    - androidTest/java/com/ooad/frontend: Folder that has “ExampleInstrumentedTest.java” that has unit sample program test for front end creation of layouts.
- Report: Folder that has the report of our project
- Gradle files: Files used to build the android application

Notes on installing or executing:

To run our Android application, we would need actual android cell phone or emulation of cell phone could be done in Android studio. Steps are below.

- Step 1: Download and install Android Studio ID: https://developer.android.com/studio/install
- Step 2: Download our Android code from Git repository : https://github.com/dhawaskar/Scientic_Calculator
- Step 3: In the Android studio click “Open existing project” and open our downloaded application from step 2.
- Step 4: On the top of Android studio, there is run button. Click that and you can run the app either in actual android phone or emulator. Emulator is slow and takes up large resource and android phones are comparatively faster.