

## **Lab-1 Calculator application**



**SAN JOSÉ STATE**  
**UNIVERSITY**

Author: Vivek Maran (012556895)

Submitted to: Prof. Hungwen Li

Date: Wednesday, February 13, 2019

## TABLE OF CONTENTS

<b>1. LAB OVERVIEW .....</b>	<b>3</b>
<b>2. GLOBAL APPLICATION SETTINGS (APP NAME, THEME).....</b>	<b>3</b>
<b>2.1 SETTING THE APPLICATION NAME.....</b>	<b>3</b>
<b>2. 2 SETTING THE THEME .....</b>	<b>3</b>
<b>3. LAYOUT .....</b>	<b>3</b>
<b>3. 1 TEXT VIEW .....</b>	<b>3</b>
<b>3. 2 KEYPAD.....</b>	<b>4</b>
<b>4. CONTROLLER .....</b>	<b>5</b>
<b>4. 1 REGISTERING ONCLICKLISTENERS .....</b>	<b>6</b>
<b>4. 2 LIST OF ONCLICKLISTENERS.....</b>	<b>6</b>
<b>4. 2.1 Listeners for numeric keys.....</b>	<b>6</b>
<b>4. 2.2 Listeners for operator keys.....</b>	<b>6</b>
<b>4. 2.3 Listeners for clear key.....</b>	<b>7</b>
<b>4. 3 COMPUTING THE RESULT OF THE ARITHMETIC EXPRESSION.....</b>	<b>7</b>
<b>5. QUESTIONS.....</b>	<b>7</b>
<b>6. SCREENSHOTS FROM ANDROID MONITOR.....</b>	<b>9</b>
<b>7. APP SCREENSHOTS .....</b>	<b>9</b>
<b>8. REFERENCES.....</b>	<b>10</b>

## 1. Lab overview

A calculator application for android platform has been developed in this lab. The application accepts an arithmetic expression through its built-in keypad, evaluates it and displays the result in its text view. The implementation consists of a single activity with its corresponding layout and controller. Following is the compatibility information of the application.

Target SDK version	API 28: Android 9.0(Pie)
Minimum SDK version	15
Tested version	Android 8.0.0

## 2. Global application settings (App name, theme)

### 2.1 Setting the application name

This section describes the method used for configuring the application settings like the application name and theme. The ‘AndroidManifest.xml’ file uses the application name from the ‘strings.xml’ file present in the ‘\$PROJECT\_ROOT/res/values/’ folder. Following changes were made in the ‘strings.xml’ file to change the application name

```
<resources>
    <string name="app_name">CALCULATOR</string>
</resources>
```

### 2.2 Setting the theme

This ‘AndroidManifest.xml’ uses the theme from the ‘AppTheme’ attribute of the ‘styles.xml’ file. This attribute is configured to use the ‘Theme.AppCompat’ for setting a black theme to the application.

```
<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat">
    <!-- Customize your theme here. -->
</style>
```

## 3. Layout

This section describes the layout settings of the activity. The application uses a relative layout with vertical orientation as the container in which the text view and keypad is placed. The text view is for displaying user input/output, and the keypad is for entering the input.

### 3.1 Text view

Following snippet shows the configuration of the text view which is used to display user input and the calculated output.

```
<TextView
    android:id="@+id/txtScreen"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```

android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:gravity="right|center_vertical"
android:maxLength="16"
android:padding="50dp"
android:textAppearance="?android:attr/textAppearanceLarge"
android:textSize="30sp"
android:typeface="serif" />

```

- **android:id = "@+id/text\_screen"** The parameter sets the id for this view as 'text\_screen' and is used in the controller to operate on this resource.
- **android:layout\_width="match\_parent"** sets the width to match the parent container
- **android:layout\_height="wrap\_content"** sets the height just enough to fill the content
- **android:layout\_alignParentTop="true"** aligns the top border of this child with the top border of the parent.
- **android:layout\_centerHorizontal="true"** aligns this child to the center(horizontal( of the parent
- **android:gravity="right|center\_vertical"** Aligns the contents within the text field to be horizontally right and vertically center aligned
- **android:maxLength="16"** Constrains the text length to be 16
- **android:padding="50dp"** Adds a padding of 50dp to this view.
- **android:textAppearance="?android:attr/textAppearanceLarge"** Uses the android theme attribute to set the text appearance to be large
- **android:textSize="30sp"** Sets text size to 30 pixels
- **android:typeface="serif"** Sets font to 'serif'

### 3.2 Keypad

The keypad is contained within a vertically oriented linear layout container (which is the also child of the relative layout containing the text view). This container is subdivided into multiple linear layouts of horizontal orientation. Each of these represents a row of buttons in the keypad. This layout is depicted in the picture below

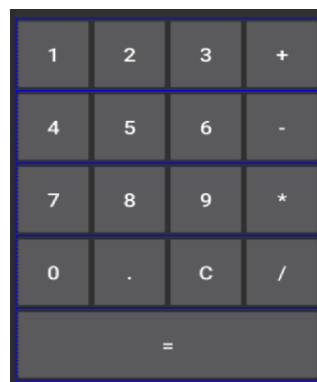


Figure 1: Multiple linear layouts of buttons contained in a linear layout

Following snippet shows the configuration of each button in the keypad.

```
<Button
    android:id="@+id/btnOne"
    android:text="1"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:textSize="30sp"/>
```

- **android:id = "@+id/text\_screen"** The parameter sets the id for this view as 'btnOne' and is used in the controller to operate on this resource.
- **android:text=1"** Sets the text for this button as "1"
- **android:layout\_width="0dp"** Width "0dp" is used in conjunction with weight="1" to set the widget to occupy 100% of the available space within the layout.
- **android:layout\_height="match\_parent"** sets the width to match the parent container
- **android:layout\_weight="1"** Makes the widget to occupy 100% of the available space within the layout.
- **android:textSize="30sp"** Sets text size to 30 pixels

## 4. Controller

The controller module implements the logic to handle the user inputs aggregate it, evaluate, and display the result on the text view. The 'MainActivity.Java' class implements this logic by registering and implementing multiple event listeners.

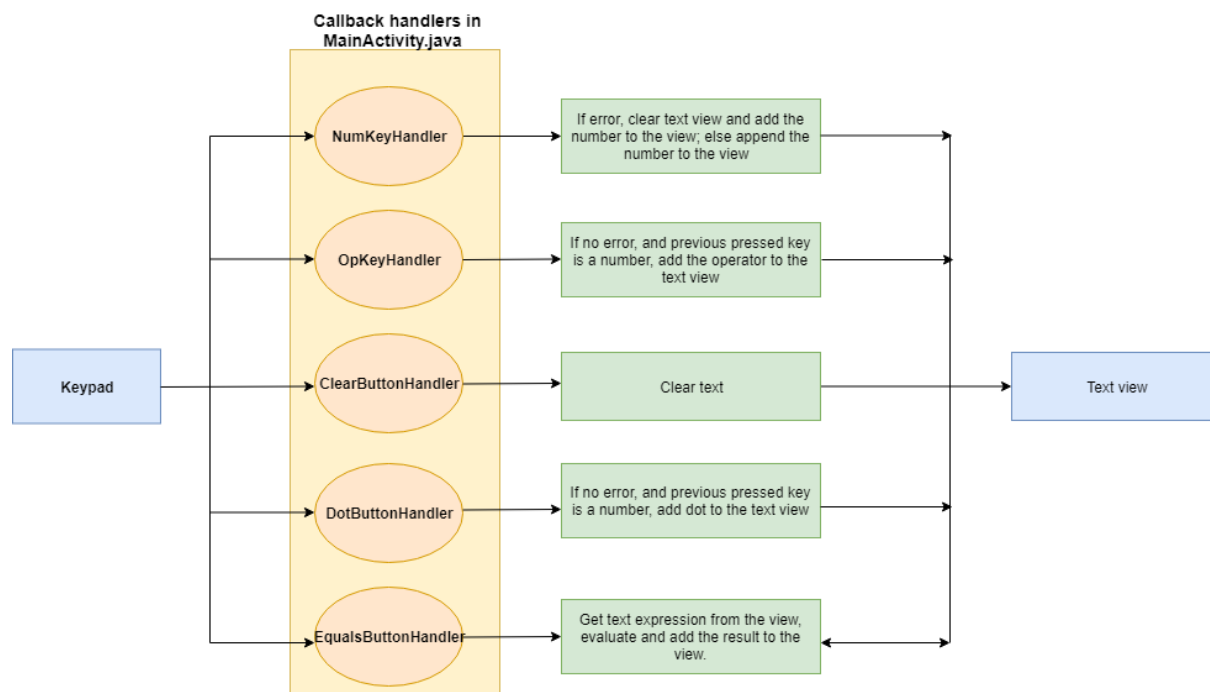


Figure 2: Controller architecture

The following section describes the list of listeners registered by the activity and their functions.

#### 4.1 Registering OnClickListener

The first task of the controller is to receive inputs corresponding to the button press of the user. To do this, the controller registers ‘OnClickListener’ for all the keys in the keypad. The registration is done by implementing the ‘onClick’ method of the ‘view.OnClickListener’ interface, and passing the instance of the same to the ‘setOnClickListener’ method of the corresponding view (button/key). Following code snippet shows an example of registering an ‘OnClickListener’

```
private void setHandlersForNumKeys()
{
    View.OnClickListener listener = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Button button = (Button) v;
            if (isError) {
                txtScreen.setText(button.getText());
                isError = false;
            } else {
                txtScreen.append(button.getText());
            }
            isLastPressedNum = true;
        }
    };
    for (int id : num_keys) {
        findViewById(id).setOnClickListener(listener);
    }
}
```

#### 4.2 List of OnClickListener

##### 4.2.1 Listeners for numeric keys

All the numeric keys use a common handler. Upon press of any numeric key, the state of the controller is first checked. If it is an error state, the text view is cleared, and the pressed key is added to the view. Else, the pressed key is appended to the view.

##### 4.2.2 Listeners for operator keys

All the operator keys, except “=”, use a common handler. Upon press of any operator key, the state of the controller is first checked. If it is not in an error state and the last pressed key was a numeric key, the pressed key is added to the view. Else, the pressed key is ignored.

#### 4. 2.3 Listeners for clear key

Upon press of clear key, the text view is cleared.

#### 4. 3 Computing the result of the arithmetic expression

The above listeners aggregate the user entered expression on to the text view. Finally when the user presses '=' key, the aggregated expression is fetched from the text view and evaluated using the 'exp4j' library and the result is displayed back to the text view.

### 5. Questions

What is difference between Gradle and Maven? Why is Gradle used extensively in Android project developments?

#### Answer

Gradle and Maven both are build management tools popularly used in Java Based projects. Based on the performance and capabilities that Gradle offers, Google created the 'Android Gradle plugin' packaged with 'Android Studio' for building android application. It's the reason of extensive use of Gradle in Android project developments. The distinguishing features of Gradle are:

1. Unlike XML based configuration for Maven, Gradle uses a Domain Specific Language based on Groovy for writing build scripts. This allows the scripts to be concise and easy to develop.
2. Groovy offers build performance improvements like incremental compilation, compiler-avoidance to reduce the compile time
3. Robust dependency management
4. Allows integration with other build tools.

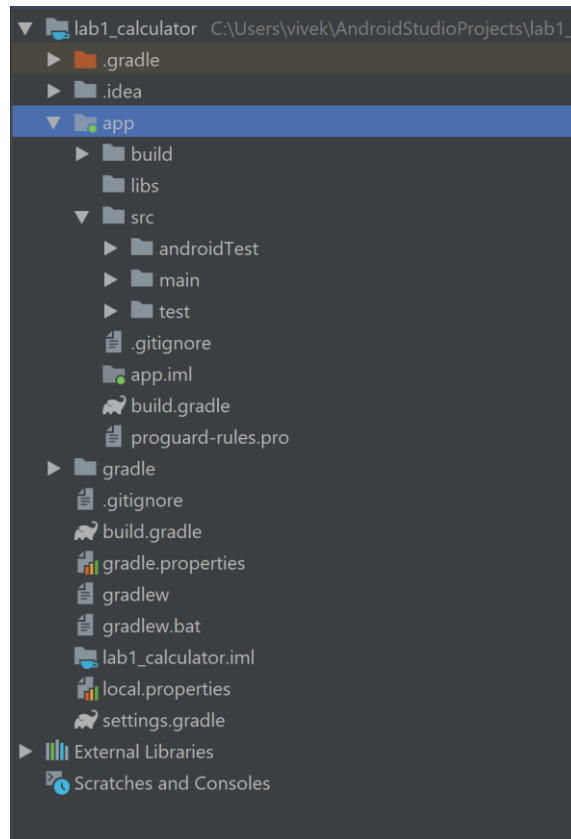
What is project structure in Android Studio? List down the folder structure created for your application.

#### Answer

Project structure in Android studio is the directory tree representing the source files, resources, build scripts and generated outputs of the project (app in the case of android studio). The following list describes the details on each of these directories.

1. **'lab1\_calculator' (Main project):** Represents the entire project context, which may contain apps and modules. It is named by the project name
2. **'idea':** Contains the metadata files for this project
3. **'app':** Folder corresponding to the application (contained in the project)
  - a. **'build':** Contains the output of the 'make' process including compiled classes, dex files, resources. The generated R.java files will also reside in this directory
  - b. **'libs':** Contains private libraries
  - c. **'src':** Contains the source files (.java) files and the resource files (assets, layouts, etc..)

4. **‘gradle’**: Contains the gradle scripts for building' this project
5. **‘External libraries’**: Not a folder but indicates references to the external libraries.



What is “adb” and where is it located? What is the importance of “adb”?

#### Answer

1. ‘adb’ (Android Device Bridge) is a command line tool that allows the development machine to communicate with the Android device.
2. ‘adb’ is located in the **‘android\_sdk/platform-tools/’**. In my case, it is at “C:\Users\vivek\AppData\Local\Android\Sdk\platform-tools”
3. Importance of adb: It facilitates development machine to debug and install applications on the target device. It also provides access to the device shell in which multiple commands are executed.
4. The components of ‘adb’ are:
  - a. ‘adb client’: Runs on the development machine
  - b. ‘adb daemon’ (adb daemon) which runs command on the device, runs as a background process
  - c. ‘Server’: Runs as a background process on the device, and facilitates communication between ‘adb’ and the ‘adb daemon’



## 6. Screenshots from android monitor



Figure 3:Profiler

## 7. App screenshots

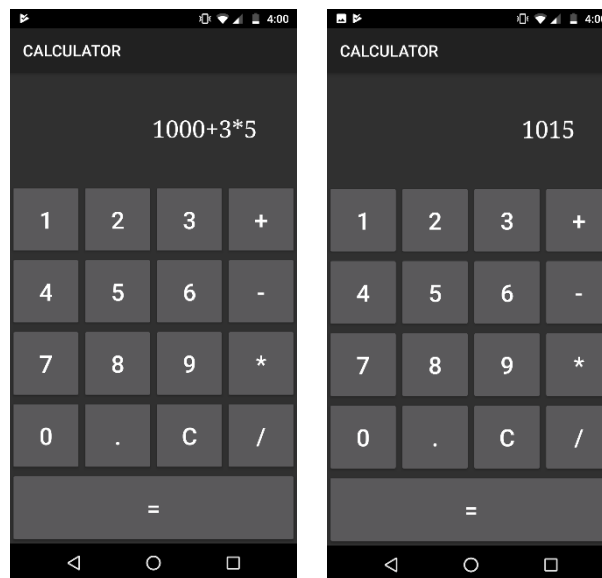


Figure 4: Application screenshots

## 8. References

- [1] <https://developer.android.com/guide/topics/manifest/manifest-intro>
- [2] <https://medium.com/androiddevelopers/picking-your-compilesdkversion-minsdkversion-targetsdkversion-a098a0341ebd>
- [3] <https://developer.android.com>
- [4] <https://stackoverflow.com/questions/6975607/what-does-layout-height-0dp-mean>
- [5] <https://blog.gradle.org/incremental-compiler-avoidance>
- [6] <https://technologyconversations.com/2014/06/18/build-tools/>
- [7] <https://developer.android.com/studio/releases/gradle-plugin>
- [8] <https://stackoverflow.com/questions/16754643/what-is-gradle-in-android-studio>
- [9] <https://stackoverflow.com/questions/17431838/android-studio-project-structure-v-s-eclipse-project-structure>
- [10] <https://www.javahelps.com/2015/03/android-simple-calculator.html>