**package** com.example.pract1calci;  
  
**import** android.os.Bundle;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** net.objecthunter.exp4j.Expression;  
**import** net.objecthunter.exp4j.ExpressionBuilder;  
  
  
**public class** MainActivity **extends** AppCompatActivity {  
 *// IDs of all the numeric buttons* **private int**[] **numericButtons** = {R.id.***button0***, R.id.***button1***, R.id.***button2***, R.id.***button3***, R.id.***button4***, R.id.***button5***, R.id.***button6***, R.id.***button7***, R.id.***button8***, R.id.***button9***};  
 *// IDs of all the operator buttons* **private int**[] **operatorButtons** = {R.id.***buttonadd***, R.id.***buttonsub***, R.id.***buttonmul***, R.id.***buttondiv***};  
 *// TextView used to display the output* **private** TextView **txtScreen**;  
 *// Represent whether the lastly pressed key is numeric or not* **private boolean lastNumeric**;  
 *// Represent that current state is in error or not* **private boolean stateError**;  
 *// If true, do not allow to add another DOT* **private boolean lastDot**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 *// Find the TextView* **this**.**txtScreen** = (TextView) findViewById(R.id.***txtScreen***);  
 *// Find and set OnClickListener to numeric buttons* setNumericOnClickListener();  
 *// Find and set OnClickListener to operator buttons, equal button and decimal point button* setOperatorOnClickListener();  
 }  
  
 */\*\*  
 \* Find and set OnClickListener to numeric buttons.  
 \*/* **private void** setNumericOnClickListener() {  
 *// Create a common OnClickListener* View.OnClickListener listener = **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *// Just append/set the text of clicked button* Button button = (Button) v;  
 **if** (**stateError**) {  
 *// If current state is Error, replace the error message* **txtScreen**.setText(button.getText());  
 **stateError** = **false**;  
 } **else** {  
 *// If not, already there is a valid expression so append to it* **txtScreen**.append(button.getText());  
 }  
 *// Set the flag* **lastNumeric** = **true**;  
 }  
 };  
 *// Assign the listener to all the numeric buttons* **for** (**int** id : **numericButtons**) {  
 findViewById(id).setOnClickListener(listener);  
 }  
 }  
  
 */\*\*  
 \* Find and set OnClickListener to operator buttons, equal button and decimal point button.  
 \*/* **private void** setOperatorOnClickListener() {  
 *// Create a common OnClickListener for operators* View.OnClickListener listener = **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *// If the current state is Error do not append the operator  
 // If the last input is number only, append the operator* **if** (**lastNumeric** && !**stateError**) {  
 Button button = (Button) v;  
 **txtScreen**.append(button.getText());  
 **lastNumeric** = **false**;  
 **lastDot** = **false**; *// Reset the DOT flag* }  
 }  
 };  
 *// Assign the listener to all the operator buttons* **for** (**int** id : **operatorButtons**) {  
 findViewById(id).setOnClickListener(listener);  
 }  
 *// Decimal point* findViewById(R.id.***button10***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (**lastNumeric** && !**stateError** && !**lastDot**) {  
 **txtScreen**.append(**"."**);  
 **lastNumeric** = **false**;  
 **lastDot** = **true**;  
 }  
 }  
 });  
 *// Clear button* findViewById(R.id.***buttonC***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **txtScreen**.setText(**""**); *// Clear the screen  
 // Reset all the states and flags* **lastNumeric** = **false**;  
 **stateError** = **false**;  
 **lastDot** = **false**;  
 }  
 });  
 *// Equal button* findViewById(R.id.***buttoneql***).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 onEqual();  
 }  
 });  
 }  
  
 */\*\*  
 \* Logic to calculate the solution.  
 \*/* **private void** onEqual() {  
 *// If the current state is error, nothing to do.  
 // If the last input is a number only, solution can be found.* **if** (**lastNumeric** && !**stateError**) {  
 *// Read the expression* String txt = **txtScreen**.getText().toString();  
 *// Create an Expression (A class from exp4j library)* Expression expression = **new** ExpressionBuilder(txt).build();  
 **try** {  
 *// Calculate the result and display* **double** result = expression.evaluate();  
 **txtScreen**.setText(Double.*toString*(result));  
 **lastDot** = **true**; *// Result contains a dot* } **catch** (ArithmeticException ex) {  
 *// Display an error message* **txtScreen**.setText(**"Error"**);  
 **stateError** = **true**;  
 **lastNumeric** = **false**;  
 }  
 }  
 }  
 }

BUILD GRADLE(MODULE APP)

apply plugin: **'com.android.application'**android {  
 compileSdkVersion 29  
 buildToolsVersion **"29.0.0"** defaultConfig {  
 applicationId **"com.example.pract1calci"** minSdkVersion 15  
 targetSdkVersion 29  
 versionCode 1  
 versionName **"1.0"** testInstrumentationRunner **"androidx.test.runner.AndroidJUnitRunner"** }  
 buildTypes {  
 release {  
 minifyEnabled **false** proguardFiles getDefaultProguardFile(**'proguard-android-optimize.txt'**), **'proguard-rules.pro'** }  
 }  
}  
  
dependencies {  
 implementation fileTree(dir: **'libs'**, include: [**'\*.jar'**])  
 implementation **'androidx.appcompat:appcompat:1.0.2'** implementation **'androidx.constraintlayout:constraintlayout:1.1.3'** testImplementation **'junit:junit:4.12'** androidTestImplementation **'androidx.test:runner:1.2.0'** androidTestImplementation **'androidx.test.espresso:espresso-core:3.2.0'** compile fileTree(dir: **'libs'**, include: [**'\*.jar'**])  
 compile **'com.android.support:appcompat-v7:21.0.3'** compile **'net.objecthunter:exp4j:0.4.4'**}

gradle scripts 🡺 build gradle(module: app)::

last 3 lines ::

compile fileTree(**dir**: **'libs'**, **include**: [**'\*.jar'**])  
 compile **'com.android.support:appcompat-v7:21.0.3'** compile **'net.objecthunter:exp4j:0.4.4'**}