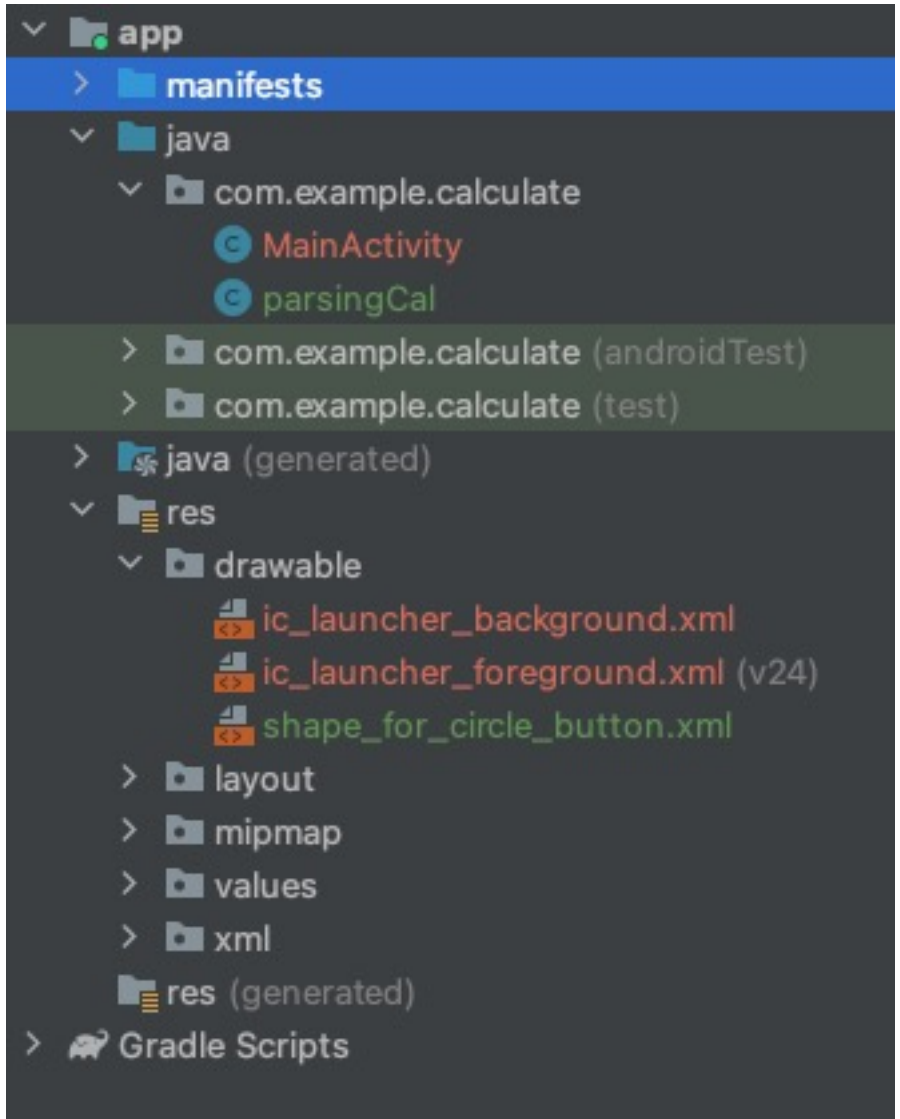
#1. TextView 1개, 버튼 여러개 (숫자, + - * - = 등)

- 기본 기능: 숫자와 더하기 버튼을 누르면 그대로 **TextView**에 나옴.
- = 을 누르면 '식 = 결과' 형태로 나옴 **(ex. 1+1=2)**
- 식이 성립이 안되면 **ERROR** 표시

구조



ParsingCal.java = 문자열 배열을 받아 계산하는 클래스
shape_for_circle_button.xml = 버튼을 원형버튼으로 만들어 주기 위한 xml 파일

```xml
<TextView
    android:id="@+id/TextView"
    android:background="#394045"
    android:layout_width="fill_parent"
    android:layout_height="200px"
    android:layout_marginTop="10dp"
    android:textColor="#10C888"
    android:textSize="35dp"
    android:gravity="right"
    />
```

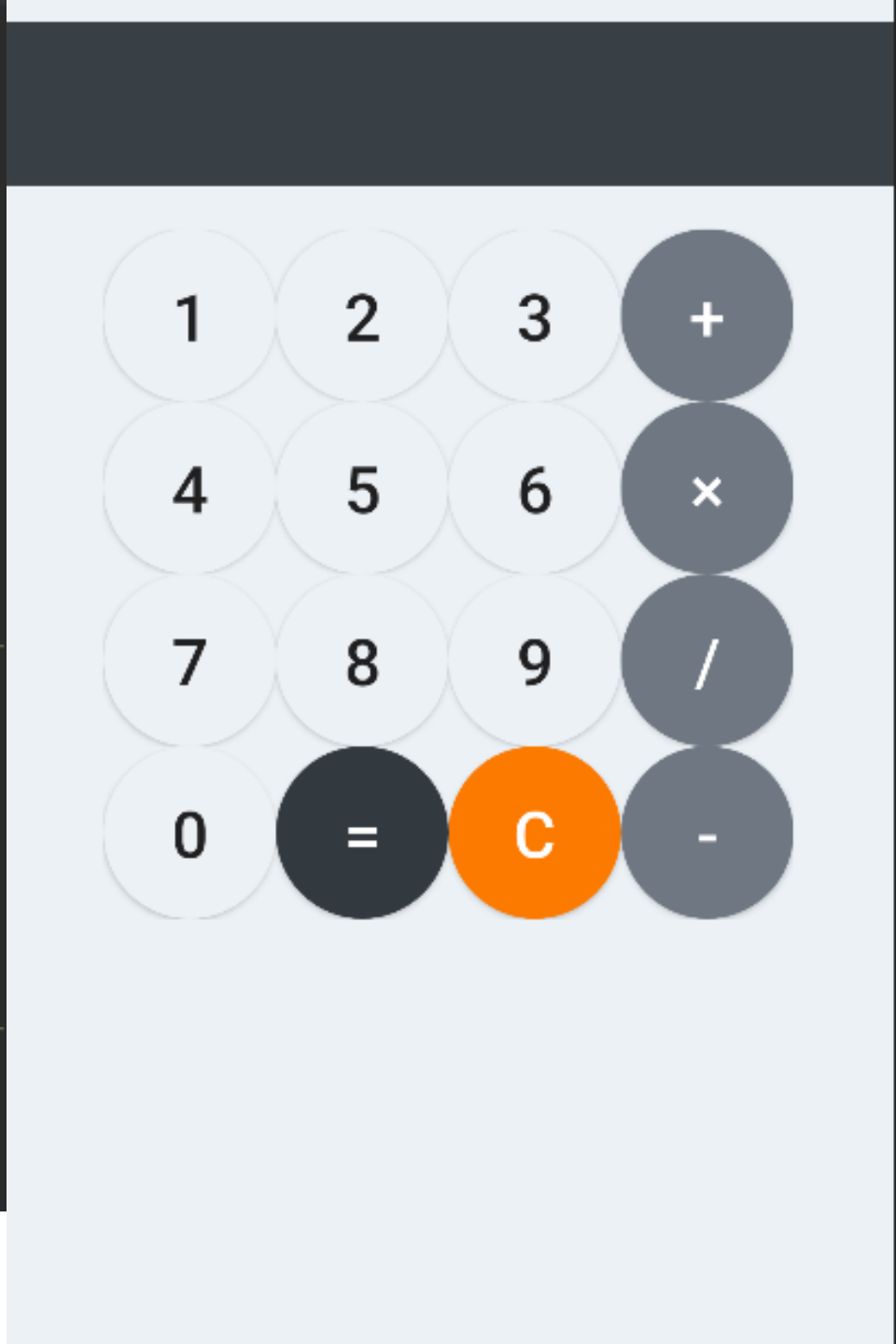
```
<LinearLayout>
        <LinearLayout>
                <Button>
                <Button>
                <Button>
                <Button>
        </LinearLayout>
        <LinearLayout>
                <Button>
                <Button>
                <Button>
                <Button>
        </LinearLayout>
                ....
</LinearLaytou>
```

```xml
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginTop="20dp"
    android:layout_marginRight="10dp"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/button1"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:background="@drawable/shape_for_circle_
            android:backgroundTint="#ECF1F5"
            android:text="1"
            android:textSize="30dp"
            android:onClick="button1Click"/>

        <Button
            android:id="@+id/button2"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:background="@drawable/shape_for_circle_
            android:backgroundTint="#ECF1F5"
            android:text="2"
            android:textSize="30dp"
            android:onClick="button2Click"/>
```



parsingCal.java

```java
public class parsingCal {


    public String calulation(ArrayList<String> splitedString){
        double num1 = Double.parseDouble(splitedString.get(0));
        double num2;
        double temp = 0;
        String operator = "", result = "";

//        입력이 제대로 들어왔는지 확인 하기 위함
        for(int i = 0; i < splitedString.size();i++){
            System.out.println("["+i+"]"+splitedString.get(i));
        }

//        입력값을 루프로 도는데
        for(int i = 1; i < splitedString.size(); i++){
            if(i % 2 == 1){
//                홀수 기준으로 연산자가 들어옴, 들어온 연산자를 저장 후 뒤에 연산자에 따라 비교함
                operator = splitedString.get(i);
                continue;
            }

            num2 = Integer.parseInt(splitedString.get(i));
            switch (operator) {
                case "+":
                    temp = num1 + num2;
                    System.out.println("num1 += " + num1 + num2);
                    break;
                case "-":
                    temp = num1 - num2;
                    System.out.println("num1 -= " + num1+ num2);
                    break;
                case "*":
                    temp = num1 * num2;
                    System.out.println("num1 *= " + num1+ num2);
                    break;
                case "/":
                    System.out.println("num1 /= " + num1+ num2);
                    temp = num1 / num2;
                    break;
            }

            num1 = temp;
        }
        // 소수점 반올림
        return String.format("%.2f",num1);
    }
```

```
I/System.out: [0]3568
I/System.out: [1]*
I/System.out: [2]635
```

MainActivity.java

```java
public class MainActivity extends AppCompatActivity {

    private TextView TV;
//    연산자 버튼을 누루기 전까지 string에 숫자 저장
    private String savedString = "";
//    String에 저장된 연산자와 숫자를 분리하여 .add
    private ArrayList<String> splitedString = new ArrayList<>();
    private boolean countEqual = false;
    private boolean countOperator = false;
    private String finalAnswer = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        }
```

버튼 0~9 리스너

```java
    public void button1Click(View v){
        Button button1 = findViewById(R.id.button1);
        TV=findViewById(R.id.TextView);

        String lang = button1.getText().toString();
        TV.append(lang);

        countOperator = false;
    }

    public void button2Click(View v){
        Button button2 = findViewById(R.id.button2);
        TV=findViewById(R.id.TextView);

        String lang = button2.getText().toString();
        savedString += lang;
        TV.append(lang);
        countOperator = false;
    }
```

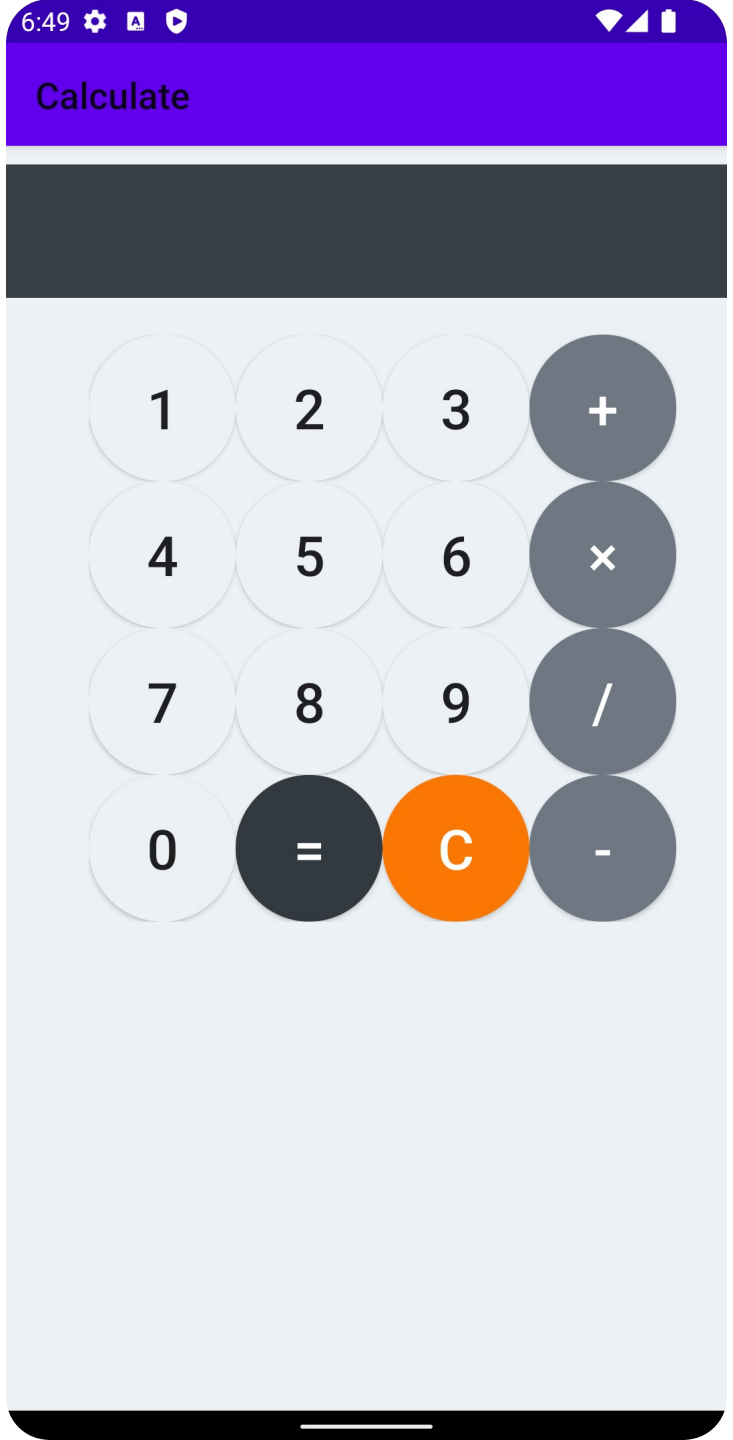
+, -, *, / 연산자 버튼 리스너

```
130     public void buttonPlusClick(View v) {
131         Button buttonPlus = findViewById(R.id.buttonPlus);
132         TV = findViewById(R.id.TextView);
133
134         // 연산자를 두번 연속 누르면 오류 출력
135         if(countOperator){TV.setText("ERROR"); countOperator = false;};
136         // 만약 = 를 누른적이 있다면 청소해줘야 함
137         if(countEqual){TV.setText(finalAnswer); countEqual = false;}
138
139         String lang = buttonPlus.getText().toString();
140         if(!savedString.equals(""))splitedString.add(savedString);
141         splitedString.add("+");
142         savedString = "";
143         TV.append(lang);
144
145         countOperator = true;
146     }
```

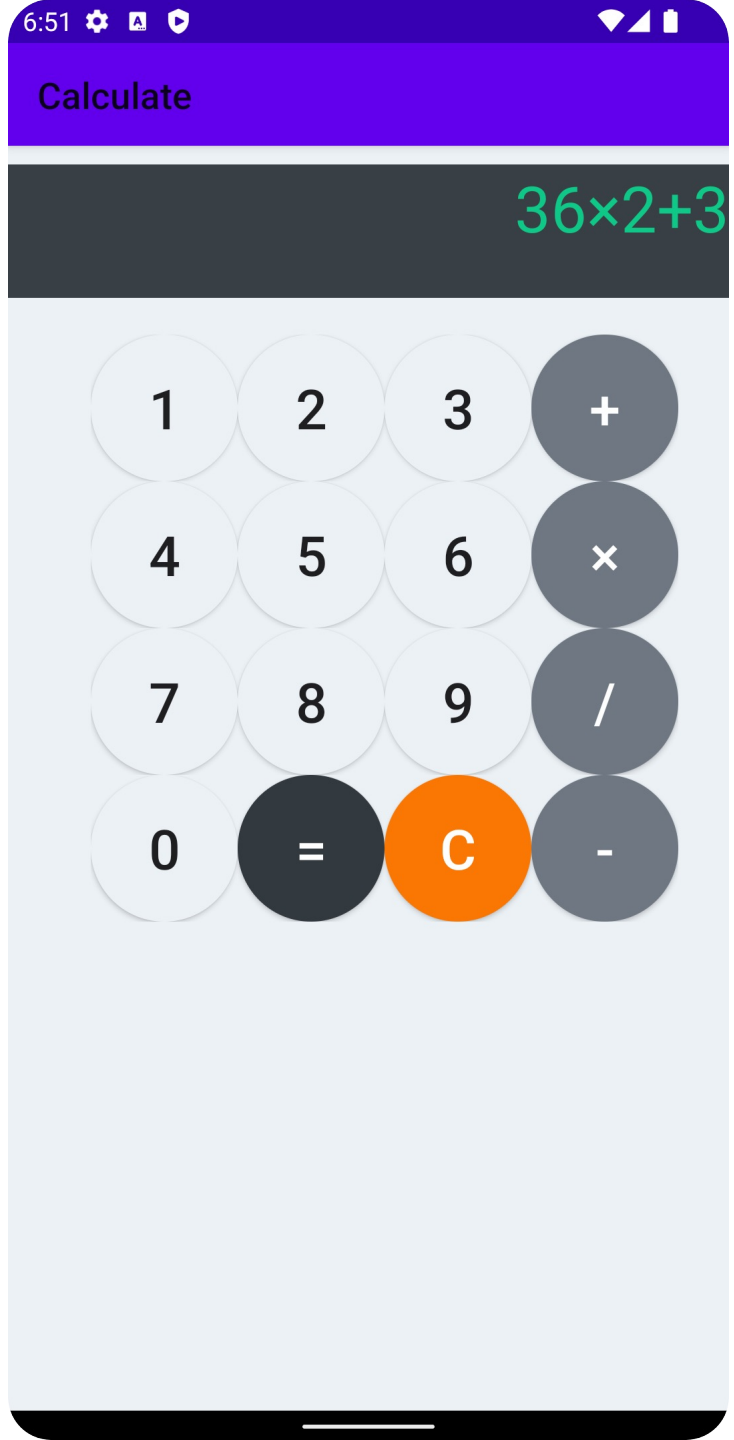
클리어버튼, =버튼 리스너

```
202     public void buttonClearClick(View v){
203         Button buttonClear =  findViewById(R.id.buttonClear);
204         TV=findViewById(R.id.TextView);
205         splitedString.clear();
206         savedString = "";
207         TV.setText("");
208         countOperator = false;
209         countEqual = false;
210     }
211
212     public void buttonEqualClick(View v){
213         Button buttonEqual =  findViewById(R.id.buttonEqual);
214         TV=findViewById(R.id.TextView);
215         splitedString.add(savedString);
216
217         parsingCal result = new parsingCal();
218         finalAnswer = result.calulation(splitedString);
219         TV.append("= "+finalAnswer);
220         splitedString.clear();
221         savedString = "";
222         splitedString.add(finalAnswer);
223
224         countEqual = true;
225     }
```

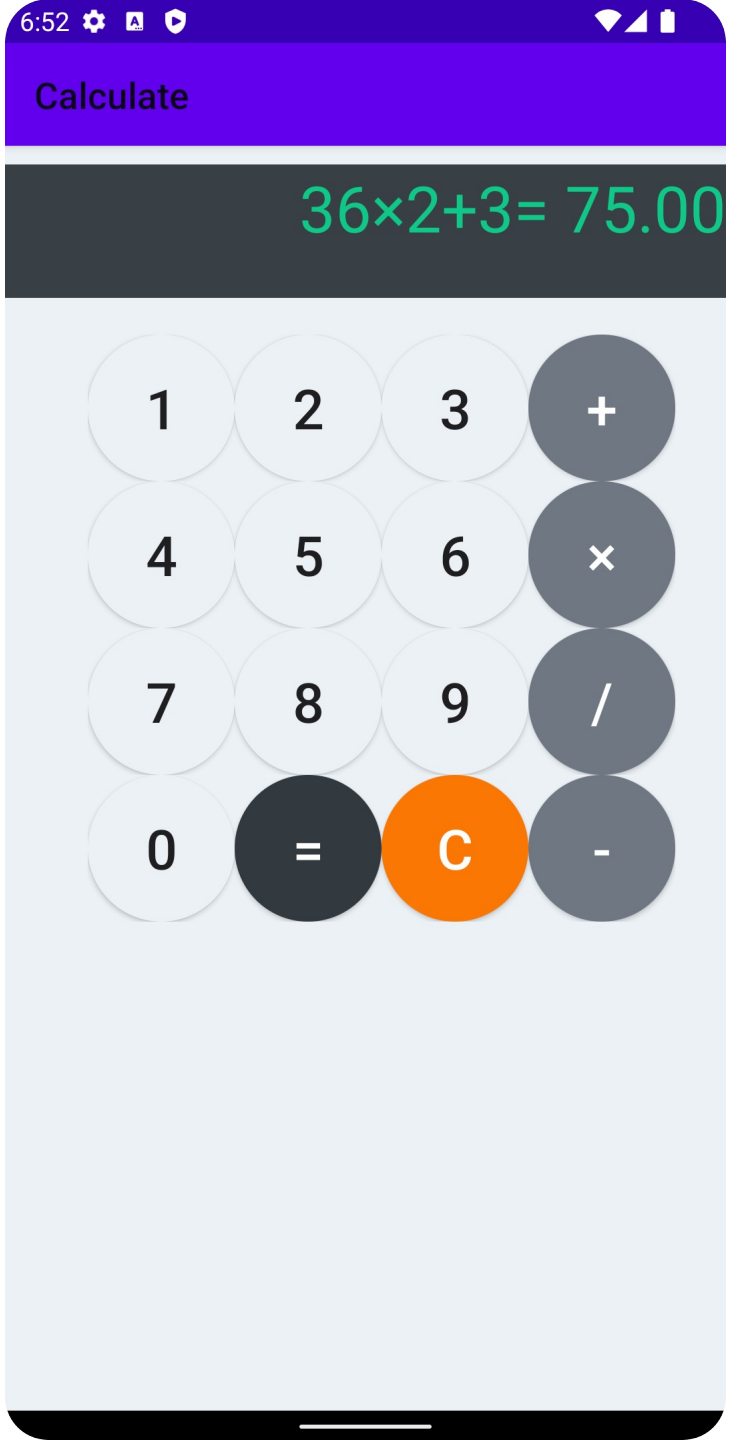
기본


6:49
Calculate
1
2
3
+
4
5
6
×
7
8
9
/
0
=
C
-

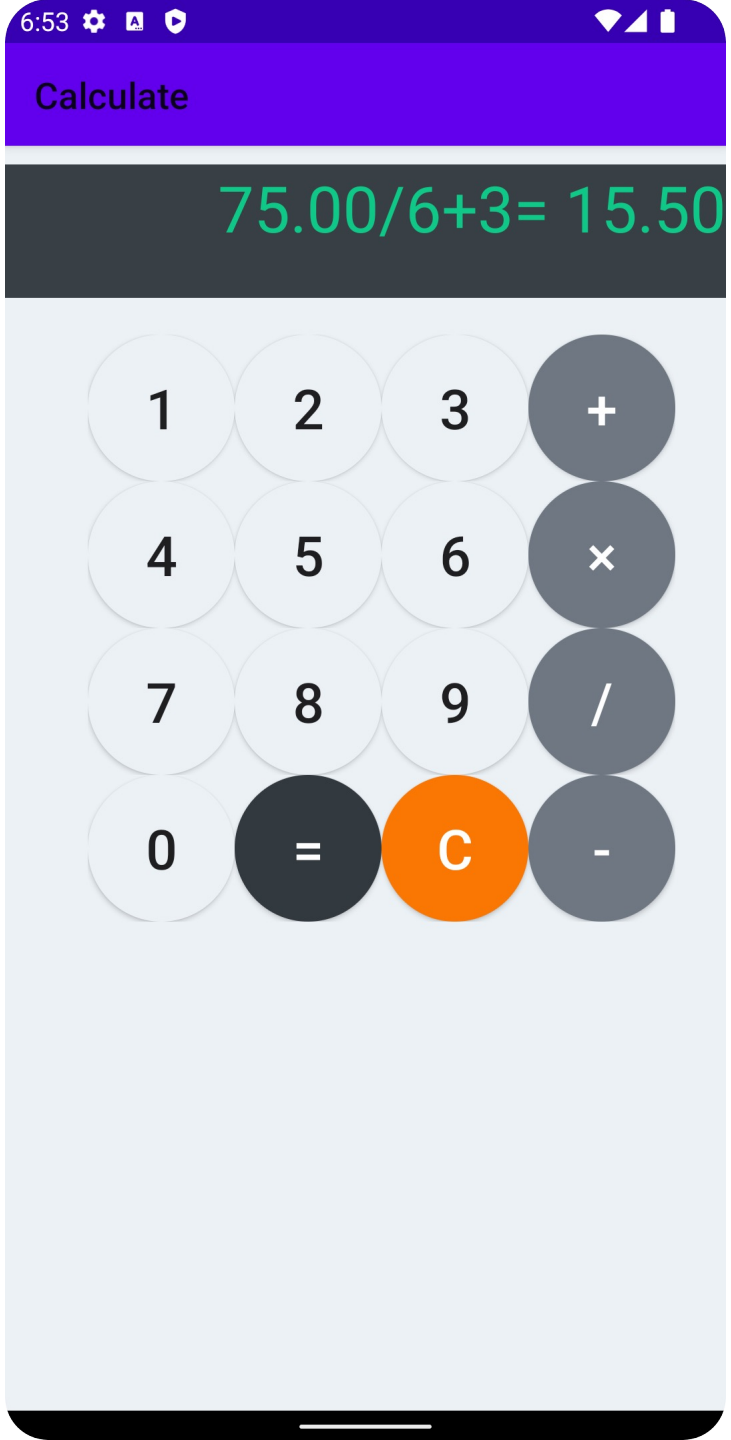
숫자입력


6:51
Calculate
36×2+3
1
2
3
+
4
5
6
×
7
8
9
/
0
=
C
-

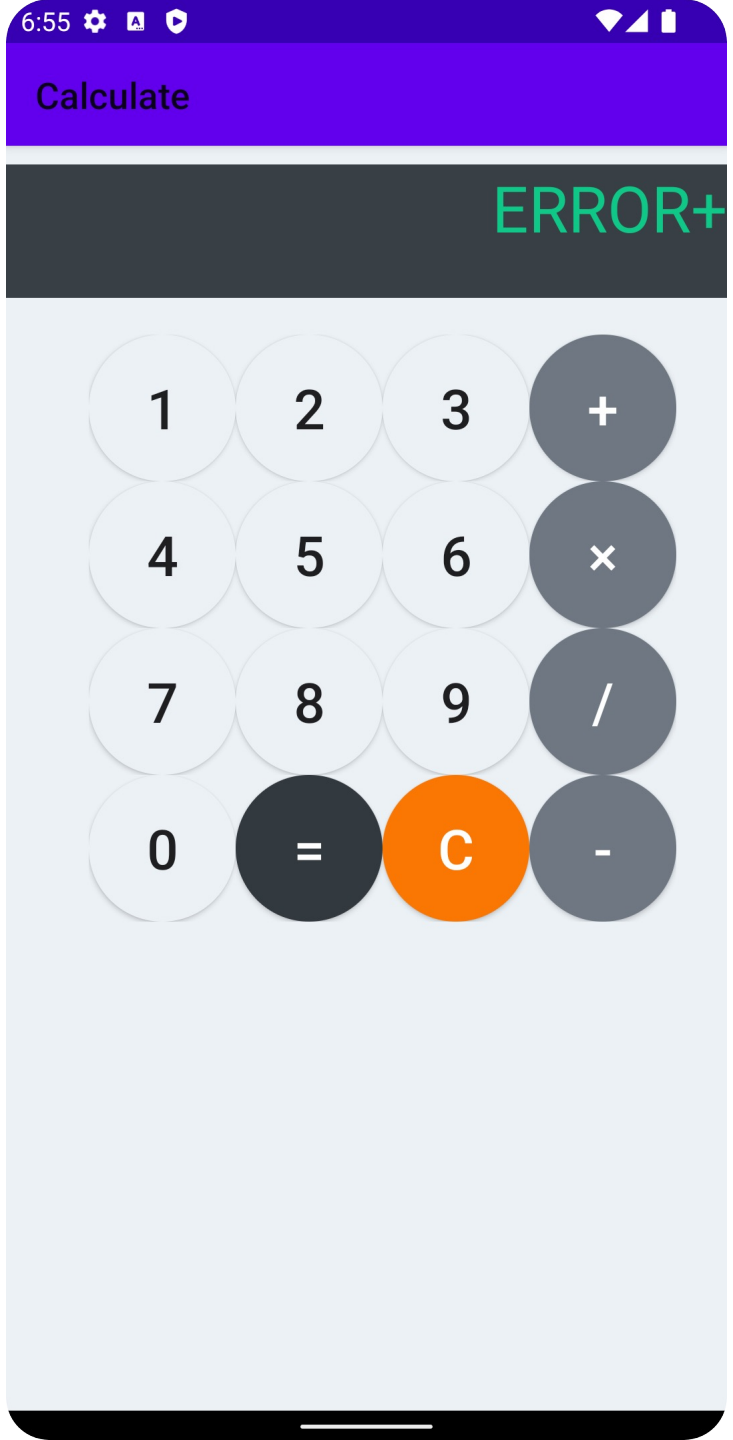
=


6:52
Calculate
36×2+3= 75.00
1
2
3
+
4
5
6
×
7
8
9
/
0
=
C
-

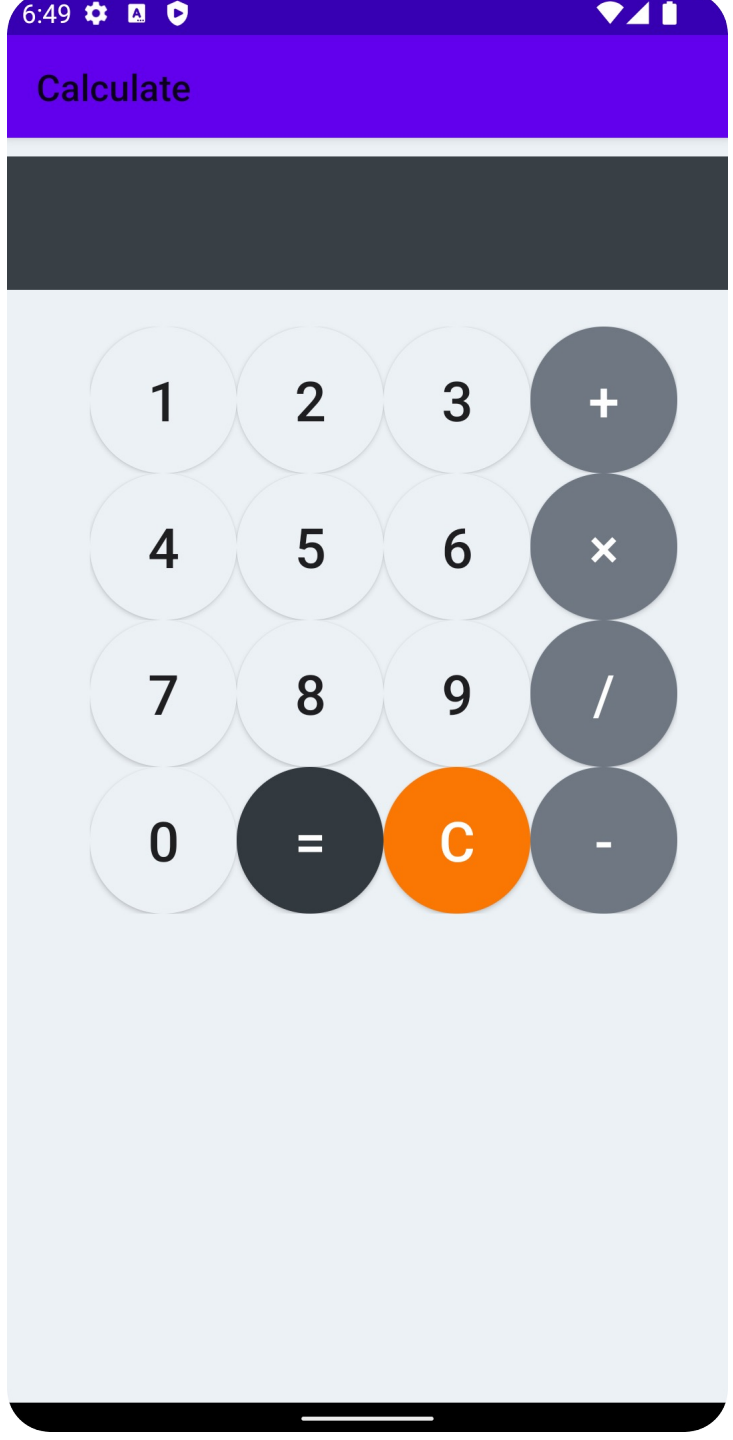
나온 정답에
바로 식 추가


6:53
Calculate
75.00/6+3= 15.50
1
2
3
+
4
5
6
×
7
8
9
/
0
=
C
-

연산자 두번 입력시 오류 출력



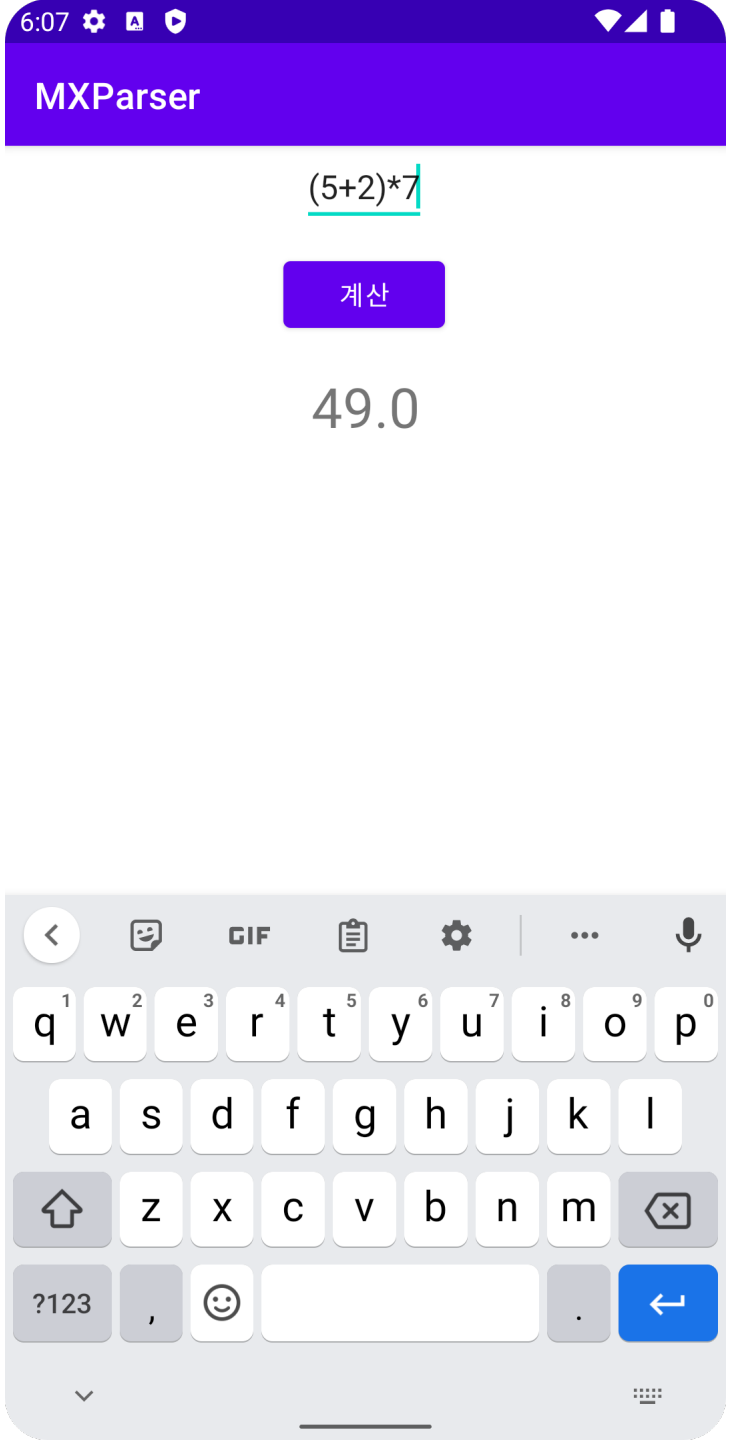
Clear시 전부 초기화



#2. MXParser library 사용(또는 같은 기능 다른 것 사용해도 됨)

- **EditText**에 수식을 치고 '계산' 버튼을 누르면 결과값 나오기
- 자기가 만든 앱 실행 캡쳐, 핵심 코드 캡쳐, 기능 간단하게 설명하는 **ppt** 제출

mXparser는 텍스트로 제공되는 수학적 표현에 매우 유연한 parser. 소프트웨어는 JAVA, C# .NET, TypeScript 및 JavaScript용으로 사용하기 쉬운 API를 제공.



Case 1: Simple calculation

$$2+1$$

```
import org.mariuszgromada.math.mxparser.*;
...
Expression e = new Expression("2+1");
mXparser.consolePrintln("Res: " + e.getExpressionString()
```

```
Res: 2+1 = 3.0
```

Case 2: Changing expression string

$$2-1$$

```
import org.mariuszgromada.math.mxparser.*;
...
e.setExpressionString("2-1");
mXparser.consolePrintln("Res: " + e.getExpressionString()
```

```
Res: 2-1 = 1.0
```

Case 3: Using operators

$$2-\frac{32-4}{23+\frac{4}{5}}-(2-4)(4+6-98.2)+4$$

```
import org.mariuszgromada.math.mxparser.*;
...
Expression e = new Expression("2-(32-4)/(23+4/5)-(2-4)*(4-
mXparser.consolePrintln("Res: " + e.getExpressionString()
```

```
Res: 2-(32-4)/(23+4/5)-(2-4)*(4+6-98.2)+4 = -171.576470588
```

Case 4: Power function

$$2^3+2^3+2^{3^{-4}}$$

```
import org.mariuszgromada.math.mxparser.*;
...
Expression e = new Expression("2^3+2^(-3)+2^3^(-4)");
mXparser.consolePrintln("Res: " + e.getExpressionString()
```

```
Res: 2^3+2^(-3)+2^3^(-4) = 9.133594091576999
```

Case 5: Using numbers in scientific notation

```
import org.mariuszgromada.math.mxparser.*;
...
Expression e1 = new Expression("1.2e2 + 1.2e+2 + 1.2e-2");
mXparser.consolePrintln("Res 1: " + e1.getExpressionString
Expression e2 = new Expression("1.2E2 + 1.2E+2 + 1.2E-2");
mXparser.consolePrintln("Res 2: " + e2.getExpressionString
```

```
[mXparser-v.4.0.0] Res 1: 1.2e2 + 1.2e+2 + 1.2e-2 = 240.01
[mXparser-v.4.0.0] Res 2: 1.2E2 + 1.2E+2 + 1.2E-2 = 240.01
```

Case 6: Percent sign support

```
import org.mariuszgromada.math.mxparser.*;
...
Expression e1 = new Expression("2%");
Expression e2 = new Expression("2% * 100");
Expression e3 = new Expression("pi% * 100");
mXparser.consolePrintln("Res 1: " + e1.getExpressionString
mXparser.consolePrintln("Res 2: " + e2.getExpressionString
mXparser.consolePrintln("Res 3: " + e3.getExpressionString
```

```
[mXparser-v.4.1.0] Res 1: 2% = 0.02
[mXparser-v.4.1.0] Res 2: 2% * 100 = 2.0
[mXparser-v.4.1.0] Res 3: pi% * 100 = 3.141592653589794
```

Case 7: Leading zeros support

```
import org.mariuszgromada.math.mxparser.*;
...
Expression e1 = new Expression("00123");
Expression e2 = new Expression("-00123");
Expression e3 = new Expression("-00000123.123e-10");
mXparser.consolePrintln("Res 1: " + e1.getExpressionString
mXparser.consolePrintln("Res 2: " + e2.getExpressionString
mXparser.consolePrintln("Res 3: " + e3.getExpressionString
```

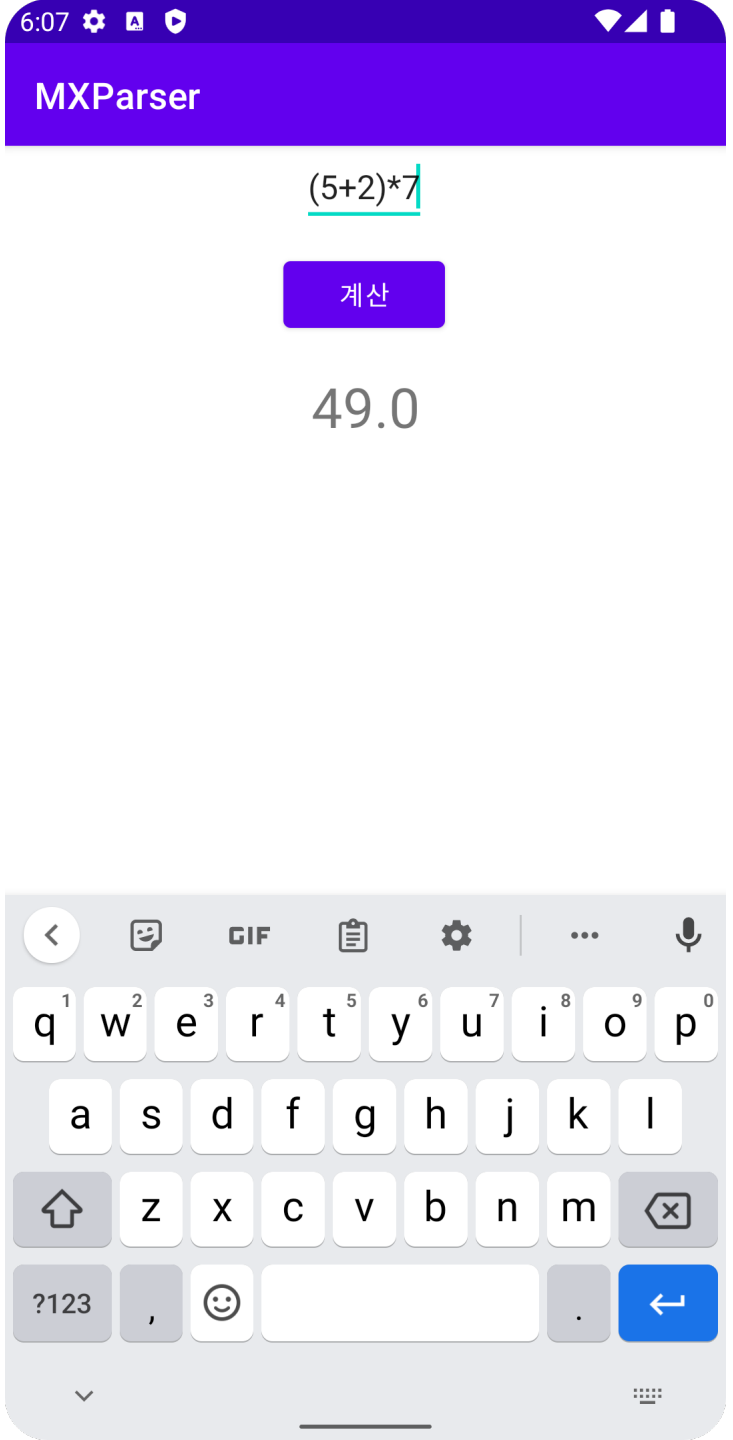
```
[mXparser-v.4.1.0] Res 1: 00123 = 123.0
[mXparser-v.4.1.0] Res 2: -00123 = -123.0
[mXparser-v.4.1.0] Res 3: -00000123.123e-10 = -1.23123E-8
```

Activity_main.xml

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="1+1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="77dp" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:text="계산"
    android:onClick="calculate"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="2"
    android:textSize="30dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button"
    app:layout_constraintVertical_bias="0.031" />
```



```java
package com.example.mxparser;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void calculate(View view){

        EditText editText = findViewById(R.id.editText);
        TextView textView = findViewById(R.id.textView);
        String sik = editText.getText().toString();

        Expression ex = new Expression(sik);
        String result = String.valueOf(ex.calculate());

        textView.setText(result);

    }
}
```

MainActivity.java