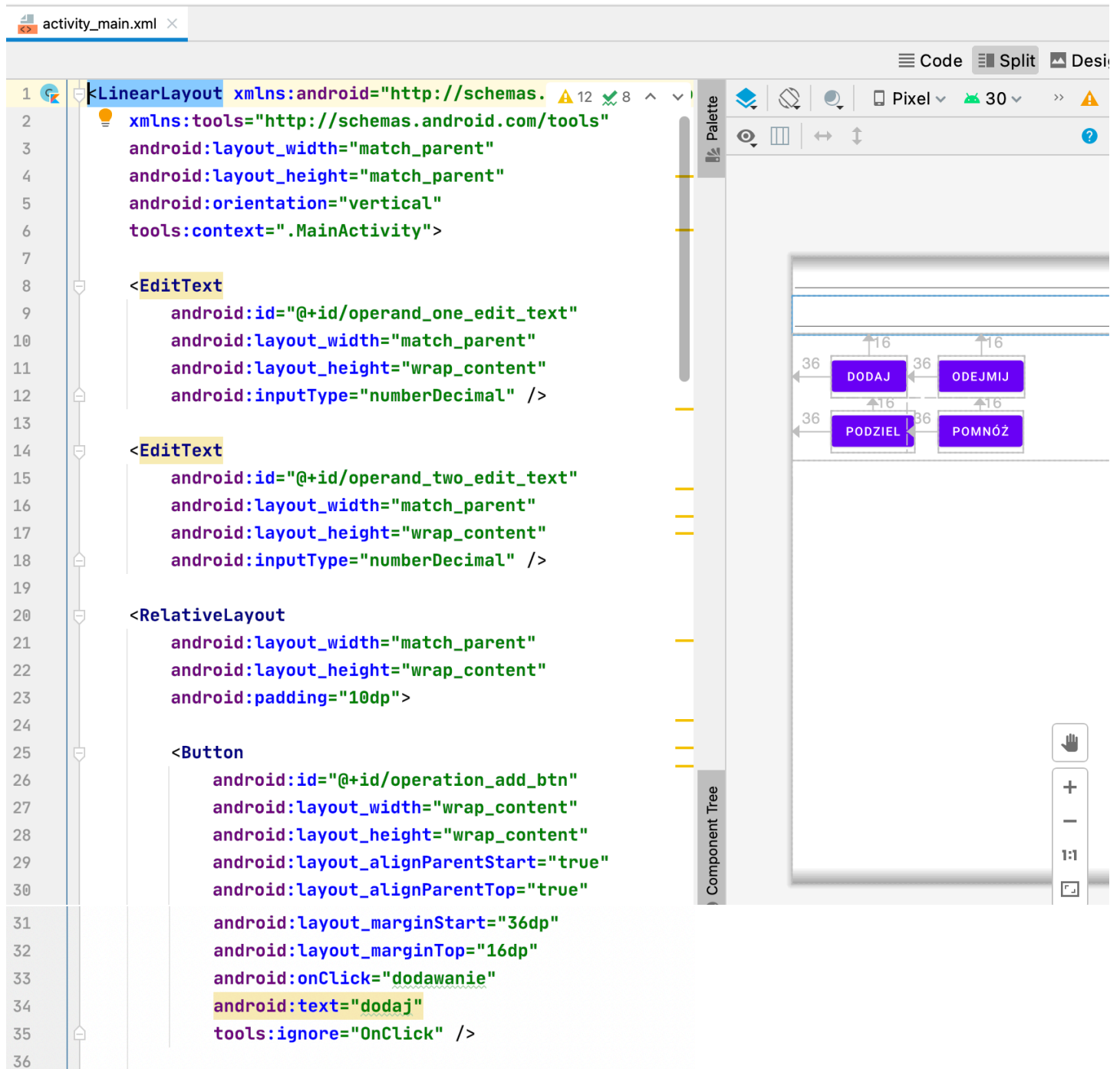


## Praktyczne zastosowanie Android Studio - Kalkulator

Projekt ten można wykonać także przy pomocy fragmentów (jak każdą aplikację). Wykonamy prosty kalkulator wykonujący podstawowe działania na 2-ch liczbach:

Tworzymy nowy projekt z pustą aktywnością. Projektujemy interfejs kalkulatora w pliku XML:



```
activity_main.xml
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical"
6   tools:context=".MainActivity">
7
8   <EditText
9     android:id="@+id/operand_one_edit_text"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:inputType="numberDecimal" />
13
14   <EditText
15     android:id="@+id/operand_two_edit_text"
16     android:layout_width="match_parent"
17     android:layout_height="wrap_content"
18     android:inputType="numberDecimal" />
19
20   <RelativeLayout
21     android:layout_width="match_parent"
22     android:layout_height="wrap_content"
23     android:padding="10dp">
24
25     <Button
26       android:id="@+id/operation_add_btn"
27       android:layout_width="wrap_content"
28       android:layout_height="wrap_content"
29       android:layout_alignParentStart="true"
30       android:layout_alignParentTop="true"
31
32       android:layout_marginStart="36dp"
33       android:layout_marginTop="16dp"
34       android:onClick="dodawanie"
35       android:text="dodaj"
36       tools:ignore="OnClick" />
```

The Design view on the right shows a visual representation of the layout. It features two input fields at the top, followed by a grid of four buttons: 'DODAJ' (Add), 'ODEJMIJ' (Subtract), 'PODZIEL' (Divide), and 'POMNÓŻ' (Multiply). Dimensions are indicated for the buttons and their spacing.

```
37 <Button
38     android:id="@+id/operation_sub_btn"
39     android:layout_width="wrap_content"
40     android:layout_height="wrap_content"
41     android:layout_alignParentTop="true"
42     android:layout_marginStart="36dp"
43     android:layout_marginLeft="36dp"
44     android:layout_marginTop="16dp"
45     android:layout_toEndOf="@id/operation_add_btn"
46     android:layout_toRightOf="@id/operation_add_btn"
47     android:onClick="odejmowanie"
48     android:text="odejmij"
49     tools:ignore="OnClick" />
50
51 <Button
52     android:id="@+id/operation_div_btn"
53     android:layout_width="wrap_content"
54     android:layout_height="wrap_content"
55     android:layout_below="@id/operation_add_btn"
56     android:layout_alignParentStart="true"
57     android:layout_marginStart="36dp"
58     android:layout_marginTop="16dp"
59     android:onClick="dzielenie"
60     android:text="podziel"
61     tools:ignore="OnClick" />
62
63 <Button
64     android:id="@+id/operation_mul_btn"
65     android:layout_width="wrap_content"
66     android:layout_height="wrap_content"
67     android:layout_below="@id/operation_add_btn"
68     android:layout_marginStart="36dp"
69     android:layout_marginLeft="36dp"
70     android:layout_marginTop="16dp"
71     android:layout_toEndOf="@id/operation_add_btn"
72     android:layout_toRightOf="@id/operation_add_btn"
73     android:onClick="mnozenie"
74     android:text="pomnóż"
75     tools:ignore="OnClick" />
76
77 </RelativeLayout>
78
79 <TextView
80     android:id="@+id/operation_result_text_view"
81     android:layout_width="wrap_content"
82     android:layout_height="wrap_content"
83     android:layout_marginTop="20dp" />
84
85 </LinearLayout>
```

Przy okazji nadaliśmy odpowiednie nazwy funkcją wykonującą odpowiednie działania.

Teraz można oprogramować zachowanie aplikacji w pliku Main\_Activity.kt :

```
MainActivity.kt x
1 package com.example.kalkulator
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import android.widget.EditText
7 import android.widget.TextView
8 import android.widget.Toast
9
10 class MainActivity : AppCompatActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14     }
15     // //funkcje matematyczne
16     fun dodawanie(view: View){
17         val x = findViewById<EditText>(R.id.operand_one_edit_text).text.toString().toFloat()
18         val y = findViewById<EditText>(R.id.operand_two_edit_text).text.toString().toFloat()
19         val wynik = findViewById<TextView>(R.id.operation_result_text_view)
20         wynik.setText((x+y).toString())
21     }
22     fun odejmowanie(view:View){
23         val x = findViewById<EditText>(R.id.operand_one_edit_text).text.toString().toFloat()
24         val y = findViewById<EditText>(R.id.operand_two_edit_text).text.toString().toFloat()
25         val wynik = findViewById<TextView>(R.id.operation_result_text_view)
26         wynik.setText((x-y).toString())
27     }
28     fun dzielenie(view:View){
29         val x = findViewById<EditText>(R.id.operand_one_edit_text).text.toString().toFloat()
30         val y = findViewById<EditText>(R.id.operand_two_edit_text).text.toString().toFloat()
31         val wynik = findViewById<TextView>(R.id.operation_result_text_view)
32         if (y == 0f){
33             Toast.makeText(applicationContext, text: "Przez zero nie wolno dzielić", Toast.LENGTH_SHORT).show()
34         } else {
35             wynik.setText((x/y).toString())
36         }
37     }
38     fun mnozenie(view:View){
39         val x = findViewById<EditText>(R.id.operand_one_edit_text).text.toString().toFloat()
40         val y = findViewById<EditText>(R.id.operand_two_edit_text).text.toString().toFloat()
41         val wynik = findViewById<TextView>(R.id.operation_result_text_view)
42         wynik.setText((x*y).toString())
43     }
44 }
```

Teraz można przejść do przetestowania aplikacji.