

Activity_main.xml

Input/output

```
20      <TextView
21          android:id="@+id/input"
22          android:layout_width="match_parent"
23          android:layout_height="wrap_content"
24          android:gravity="end"
25          android:textColor="@color/white"
26          android:textSize="30sp"
27          tools:text="5+10-3" />
28
29      <TextView
30          android:id="@+id/output"
31          android:layout_width="match_parent"
32          android:layout_height="wrap_content"
33          android:gravity="end"
34          android:textColor="@color/neon_green"
35          android:textSize="50sp"
36          tools:text="12" />
```

<tablerow> кнопки калькулятора

```
72      <TableRow>
73          <androidx.appcompat.widget.AppCompatButton
74              android:id="@+id/button_7"
75              style="@style/Button_Style"
76              android:layout_width="wrap_content"
77              android:layout_height="90dp"
78              android:text="7" />
79          <androidx.appcompat.widget.AppCompatButton
80              android:id="@+id/button_8"
81              style="@style/Button_Style"
82              android:layout_width="wrap_content"
83              android:layout_height="90dp"
84              android:text="8" />
85          <androidx.appcompat.widget.AppCompatButton
86              android:id="@+id/button_9"
87              style="@style/Button_Style"
88              android:layout_width="wrap_content"
89              android:layout_height="90dp"
90              android:text="9" />
91          <androidx.appcompat.widget.AppCompatButton
92              android:id="@+id/button_multiply"
93              style="@style/Button_Style"
94              android:layout_width="wrap_content"
95              android:layout_height="90dp"
96              android:text="x"
97              android:textColor="@color/orange" />
```

```

72     <TableRow>
73         <androidx.appcompat.widget.AppCompatButton
74             android:id="@+id/button_7"
75             style="@style/Button_Style"
76             android:layout_width="wrap_content"
77             android:layout_height="90dp"
78             android:text="7" />
79         <androidx.appcompat.widget.AppCompatButton
80             android:id="@+id/button_8"
81             style="@style/Button_Style"
82             android:layout_width="wrap_content"
83             android:layout_height="90dp"
84             android:text="8" />
85         <androidx.appcompat.widget.AppCompatButton
86             android:id="@+id/button_9"
87             style="@style/Button_Style"
88             android:layout_width="wrap_content"
89             android:layout_height="90dp"
90             android:text="9" />
91         <androidx.appcompat.widget.AppCompatButton
92             android:id="@+id/button_multiply"
93             style="@style/Button_Style"
94             android:layout_width="wrap_content"
95             android:layout_height="90dp"
96             android:text="x"

```

Mainactivity.kt

```
AndroidManifest.xml  activity_main.xml  MainActivity.kt  styles.xml  strings.xml
1  package com.example.lab9
2
3  import android.os.Bundle
4  import android.widget.Button
5  import androidx.activity.enableEdgeToEdge
6  import android.widget.TextView
7  import androidx.appcompat.app.AppCompatActivity
8  import androidx.core.content.ContextCompat
9  import androidx.core.view.ViewCompat
10 import androidx.core.view.WindowInsetsCompat
11 import java.text.DecimalFormat
12 import net.objecthunter.exp4j.ExpressionBuilder
13
14
15 class MainActivity : AppCompatActivity() {
16
17     private lateinit var input: TextView
18     private lateinit var output: TextView
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         enableEdgeToEdge()
23         setContentView(R.layout.activity_main)
24
25         input = findViewById(R.id.input)
26         output = findViewById(R.id.output)
27
28         val btnClear: Button = findViewById(R.id.button_clear)
29     }
```

```
val btnClear: Button = findViewById(R.id.button_clear)

btnClear.setOnClickListener {
    input.text = ""
    output.text = ""
}

findViewById<Button>(R.id.button_bracket_left).setOnClickListener { addToInputText("(") }
findViewById<Button>(R.id.button_bracket_right).setOnClickListener { addToInputText(")") }
findViewById<Button>(R.id.button_0).setOnClickListener { addToInputText("0") }
findViewById<Button>(R.id.button_1).setOnClickListener { addToInputText("1") }
findViewById<Button>(R.id.button_2).setOnClickListener { addToInputText("2") }
findViewById<Button>(R.id.button_3).setOnClickListener { addToInputText("3") }
findViewById<Button>(R.id.button_4).setOnClickListener { addToInputText("4") }
findViewById<Button>(R.id.button_5).setOnClickListener { addToInputText("5") }
findViewById<Button>(R.id.button_6).setOnClickListener { addToInputText("6") }
findViewById<Button>(R.id.button_7).setOnClickListener { addToInputText("7") }
findViewById<Button>(R.id.button_8).setOnClickListener { addToInputText("8") }
findViewById<Button>(R.id.button_9).setOnClickListener { addToInputText("9") }
findViewById<Button>(R.id.button_dot).setOnClickListener { addToInputText(".") }
findViewById<Button>(R.id.button_division).setOnClickListener { addToInputText("÷") }
findViewById<Button>(R.id.button_multiply).setOnClickListener { addToInputText("x") }
findViewById<Button>(R.id.button_subtraction).setOnClickListener { addToInputText("-") }
findViewById<Button>(R.id.button_addition).setOnClickListener { addToInputText("+") }
findViewById<Button>(R.id.button_equals).setOnClickListener { showResult() }
findViewById<Button>(R.id.button_percent).setOnClickListener { addToInputText("%") }
```

```

55         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
56             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
57             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
58             insets
59         }
60     }
61
62     private fun addToInputText(value: String) {
63         input.append(value)
64     }
65
66     private fun getInputExpression(): String {
67
68         return input.text.toString()
69     }
70
71     private fun showResult() {
72         try {
73             val expression = getInputExpression().replace("%", "/100")
74             val result = ExpressionBuilder(expression).build().evaluate()
75             output.text = DecimalFormat("0.#####").format(result).toString()
76             output.setTextColor(ContextCompat.getColor(this, R.color.neon_green))
77         } catch (e: Exception) {
78             output.text = "Ошибка"
79             output.setTextColor(ContextCompat.getColor(this, R.color.red))
80         }

```

Build.gradle.kts

```

42     dependencies {
43         implementation("net.objecthunter:exp4j:0.4.8")

```

Запуск

33+22

55

DEL

(

)

÷

7

8

9

×

4

5

6

−

1

2

3

+

%

0

.

=