


```
6  android {
10     defaultConfig {
16
17         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
18     }
19
20     buildTypes {
21         release {
22             isMinifyEnabled = false
23             proguardFiles(
24                 getDefaultProguardFile("name: "proguard-android-optimize.txt"),
25                 "proguard-rules.pro"
26             )
27         }
28     }
29     compileOptions {
30         sourceCompatibility = JavaVersion.VERSION_1_8
31         targetCompatibility = JavaVersion.VERSION_1_8
32     }
33     kotlinOptions {
34         jvmTarget = "1.8"
35     }
36
37     buildFeatures {
38         viewBinding = true
39     }
40 }
41
42 dependencies {
43     implementation(libs.exp4j)
44
45     implementation(libs.androidx.core.ktx)
46     implementation(libs.androidx.appcompat)
47     implementation(libs.material)
48     implementation(libs.androidx.activity)
49     implementation(libs.androidx.constraintlayout)
50     testImplementation(libs.junit)
51     androidTestImplementation(libs.androidx.junit)
52     androidTestImplementation(libs.androidx.espresso.core)
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="purple_200">#FFBB86FC</color>
4     <color name="purple_500">#FF6200EE</color>
5     <color name="purple_700">#FF3700B3</color>
6     <color name="teal_200">#FF03DAC5</color>
7     <color name="teal_700">#FF018786</color>
8     <color name="black">#FF000000</color>
9     <color name="white">#FFFFFFFF</color>
10    <color name="red">#FF3131</color>
11    <color name="pink">#FFC0CB</color>
12    <color name="orange">#FFA500</color>
13    <color name="neon_green">#39FF14</color>
14    <style name="Button_Style" parent="Widget.AppCompat.Button.Colored">
15        <item name="android:background">@color/white</item>
16        <item name="android:textSize">24sp</item>
17        <item name="android:textColor">@color/black</item>
18        <item name="android:gravity">center</item>
19    </style>
20 </resources>
```

activity_main.xml
MainActivity.kt
build.gradle.kts (:app)
colors.xml

```

1 package com.example.myapplication
2
3 > import ...
16
17 class MainActivity : AppCompatActivity() {
18
19     private lateinit var binding: ActivityMainBinding
20
21     override fun onCreate(savedInstanceState: Bundle?) {
22         super.onCreate(savedInstanceState)
23         binding = ActivityMainBinding.inflate(layoutInflater)
24         setContentView(binding.root)
25
26         setUpView() // Установка представлений
27     }
28
29     private fun setUpView() {
30         // Обработка нажатий кнопок и добавление значений в поле ввода
31         binding.buttonClear.setOnClickListener {
32             binding.input.text = "" // Очистка поля ввода
33             binding.output.text = "" // Очистка поля вывода
34         }
35         binding.buttonBracketLeft.setOnClickListener {
36             addToInputText(value: "(") // Добавление открывающей скобки
37         }
38         binding.buttonBracketRight.setOnClickListener {
39             addToInputText(value: ")") // Добавление закрывающей скобки
40         }
41         binding.button0.setOnClickListener { addToInputText(value: "0") }
42         binding.button1.setOnClickListener { addToInputText(value: "1") }
43         binding.button2.setOnClickListener { addToInputText(value: "2") }
44         binding.button3.setOnClickListener { addToInputText(value: "3") }
45         binding.button4.setOnClickListener { addToInputText(value: "4") }
46         binding.button5.setOnClickListener { addToInputText(value: "5") }
47         binding.button6.setOnClickListener { addToInputText(value: "6") }
48         binding.button7.setOnClickListener { addToInputText(value: "7") }
49         binding.button8.setOnClickListener { addToInputText(value: "8") }
50         binding.button9.setOnClickListener { addToInputText(value: "9") }
51         binding.buttonDot.setOnClickListener { addToInputText(value: ".") } // Добавление десятичной точки
52
53         // Обработка нажатий математических операций
54         binding.buttonDivision.setOnClickListener { addToInputText(value: "/" ) }
55         binding.buttonMultiply.setOnClickListener { addToInputText(value: "*" ) }
56         binding.buttonSubtraction.setOnClickListener { addToInputText(value: "-" ) }

```

```
17 class MainActivity : AppCompatActivity() {
18     binding.input.append(value)
19 }
20
21 // Функция получения строки выражения
22 private fun getInputExpression(): String {
23     return binding.input.text.toString()
24 }
25
26 // Функция вывода результата
27 private fun showResult() {
28     try {
29         // Вычисление выражения
30         val expression = getInputExpression().replace(oldValue: "%", newValue: "/100")
31         val result = ExpressionBuilder(expression).build().evaluate()
32
33         // Форматирование результата
34         val formattedResult = DecimalFormat(pattern: "0.#####").format(result).toString()
35         binding.output.text = formattedResult
36         binding.output.setTextColor(ContextCompat.getColor(context: this, R.color.neon_green))
37     } catch (e: ParseException) {
38         Log.e(tag: "Calculator", msg: "Ошибка при парсинге выражения", e)
39         binding.output.text = "Ошибка: неверное выражение"
40         binding.output.setTextColor(ContextCompat.getColor(context: this, R.color.red))
41     } catch (e: ArithmeticException) {
42         Log.e(tag: "Calculator", msg: "Ошибка при вычислении результата", e)
43         binding.output.text = "Ошибка: деление на ноль"
44         binding.output.setTextColor(ContextCompat.getColor(context: this, R.color.red))
45     } catch (e: NumberFormatException) {
46         Log.e(tag: "Calculator", msg: "Ошибка при форматировании результата", e)
47         binding.output.text = "Ошибка: некорректное значение"
48         binding.output.setTextColor(ContextCompat.getColor(context: this, R.color.red))
49     } catch (e: Exception) {
50         Log.e(tag: "Calculator", msg: "Неизвестная ошибка", e)
51         binding.output.text = "Ошибка"
52         binding.output.setTextColor(ContextCompat.getColor(context: this, R.color.red))
53     }
54 }
55 }
```