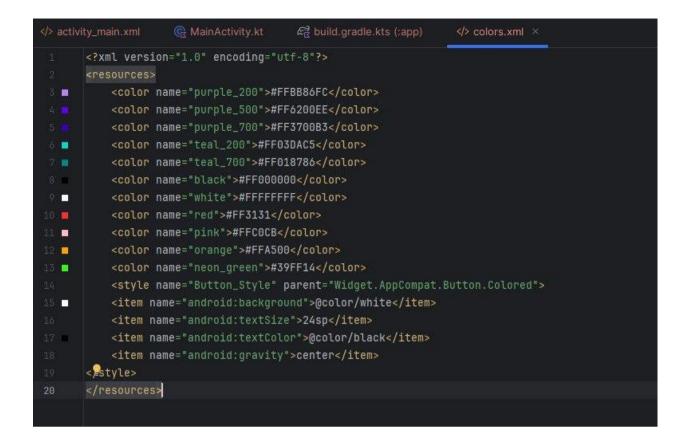


```
Project Alt+1 xml
                     @ MainActivity.kt
                                         £ build.gradle.kts (:app) ×
      android {
          defaultConfig {
               testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
          buildTypes {
                   <u>isMinifyEnabled</u> = false
                   proguardFiles(
                       getDefaultProguardFile( name: "proguard-android-optimize.txt"),
                       "proguard-rules.pro"
          compileOptions {
               sourceCompatibility = JavaVersion.VERSION_1_8
               targetCompatibility = JavaVersion.VERSION_1_8
          kotlinOptions {
               jvmTarget = "1.8"
          buildFeatures {
      H
      dependencies {
          implementation (libs.exp4j)
          implementation(libs.androidx.core.ktx)
          implementation(libs.androidx.appcompat)
          implementation(libs.material)
          implementation(libs.androidx.constraintlayout)
          testImplementation(libs.junit)
          androidTestImplementation(libs.androidx.junit)
          androidTestImplementation(libs.androidx.espresso.core)
```



```
@ MainActivity.kt × & build.gradle.kts (:app)
        package com.example.myapplication
17 > 
class MainActivity : AppCompatActivity() {
            private lateinit var binding: ActivityMainBinding
                binding = ActivityMainBinding.inflate(layoutInflater)
                    binding.input.text = " "// Очистка поля ввода
                     binding.output.text = " "// Очистка поля вывода
                     addToInputText( value: "(") // Добавление открывающей скобки
                binding.button3.setOnClickListener { addToInputText( value: "3") }
                binding.button4.setOnClickListener { addToInputText( value: "4") }
                 binding.button6.setOnClickListener { addToInputText( value "6") }
                 binding.button7.setOnClickListener { addToInputText( value: "7") }
                 binding.button8.setOnClickListener { addToInputText( value: "8") }
                 binding.button9.setOnClickListener { addToInputText( value: "9") }
                 binding.buttonDot.setOnClickListener { addToInputText( value: ".") } // Добавление десятичной точки
                 binding.buttonMultiply.setOnClickListener { addToInputText( value: "*") }
                                 transfirm catfunflicklicanan & andInnuttavtf value = = 1
```

```
@ MainActivity.kt × & double.kts (:app)
class MainActivity : AppCompatActivity() {
           binding.input.append(value)
     // Функция получения строки выражения
     private fun getInputExpression(): String {
           return binding.input.text.toString()
                // Вычисление выражения
                val result = ExpressionBuilder(expression).build().evaluate()
                val formattedResult = DecimalFormat( pattern: *0.######*).format(result).toString()
                \underline{\texttt{binding}}. \texttt{output.setTextColor}(\texttt{ContextCompat.getColor}(\underline{\texttt{context}}, \texttt{this}, \texttt{R.color}.\underline{\texttt{neon\_green}}))
                Log.e( tag: "Calculator", Imsg: "Ошибка при парсинге выражения", е) binding.output.text = "Ошибка: неверное выражение"
                \underline{\texttt{binding}}. \texttt{output}. \\ \texttt{setTextColor}(\texttt{ContextCompat}. \\ \texttt{getColor}(\texttt{context}. \\ \texttt{this}, \texttt{R.color}. \\ \underline{\textit{red}}))
                Log.e( tag: "Calculator", msg: "Ошибка при вычислении результата", е) binding.output.text = "Ошибка: деление на ноль"
                Log.e( tag: "Calculator", msg: "Ошибка при форматировании результата", е) binding.output.text = "Ошибка: некорректное значение"
                \underline{\texttt{binding}}. \texttt{output}. \\ \texttt{setTextColor}(\texttt{ContextCompat}. \\ \texttt{getColor}(\texttt{context}, \texttt{this}, \texttt{R.color}. \\ \underline{\textit{red}}))
                Log.e( tag: "Calculator", msg: "Неизвестная ошибка", е)
                 binding.output.setTextColor(ContextCompat.getColor(context this, R.color.red))
```