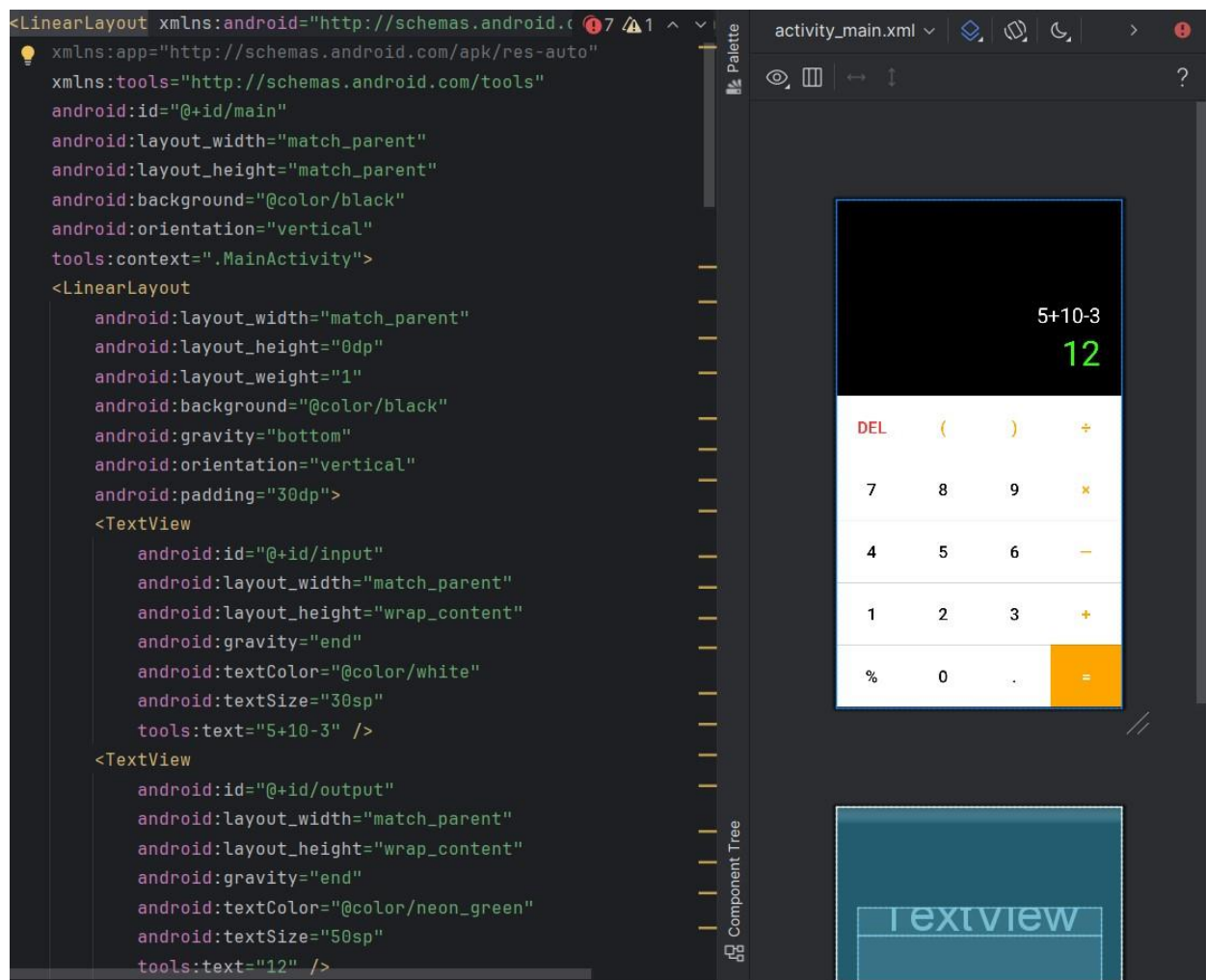


№9(Калькулятор)

1. Гл. активность



2. Добавляю библиотеку exr4j и ViewBinding

```
dependencies {  
    implementation ("net.objecthunter:exp4j:0.4.8")  
    implementation(libs.androidx.core.ktx)  
    implementation(libs.androidx.appcompat)  
    implementation(libs.material)  
    implementation(libs.androidx.activity)  
    implementation(libs.androidx.constraintlayout)  
    testImplementation(libs.junit)  
    androidTestImplementation(libs.androidx.junit)  
    androidTestImplementation(libs.androidx.espresso.core)  
}
```

3. MainActivity

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
        binding.buttonClear.setOnClickListener{
            binding.input.text = " "
            binding.output.text = " "
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( value: "(")
        }
        binding.buttonBracketRight.setOnClickListener {
            addToInputText( value: ")")
        }
        binding.button0.setOnClickListener {
            addToInputText( value: "0")
        }
        binding.button1.setOnClickListener {
            addToInputText( value: "1")
        }
        binding.button2.setOnClickListener {
            addToInputText( value: "2")
        }
        binding.button3.setOnClickListener {
            addToInputText( value: "3")
        }
    }
}
```

```
binding.button4.setOnClickListener {
    addToInputText( value: "4")
}
binding.button5.setOnClickListener {
    addToInputText( value: "5")
}
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: ".")
}
binding.buttonDivision.setOnClickListener {
    addToInputText( value: "/")
}
binding.buttonMultiply.setOnClickListener {
    addToInputText( value: "x")
}
binding.buttonSubtraction.setOnClickListener {
    addToInputText( value: "-")
}
binding.buttonAddition.setOnClickListener {
    addToInputText( value: "+")
}
```

```

    }
    binding.buttonEquals.setOnClickListener {
        showResult()
    }
    binding.buttonPercent.setOnClickListener {
        addToInputText( value: "%")
    }
}

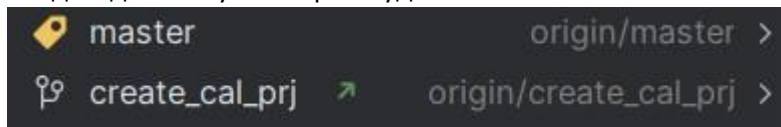
// Функция для добавления текста к полю ввода
private fun addToInputText(value: String) {
    binding.input.append(value) // Добавляем переданное значение в конец поля ввода
}

// Функция для получения строки ввода
private fun getInputExpression(): String {
    return binding.input.text.toString()
}

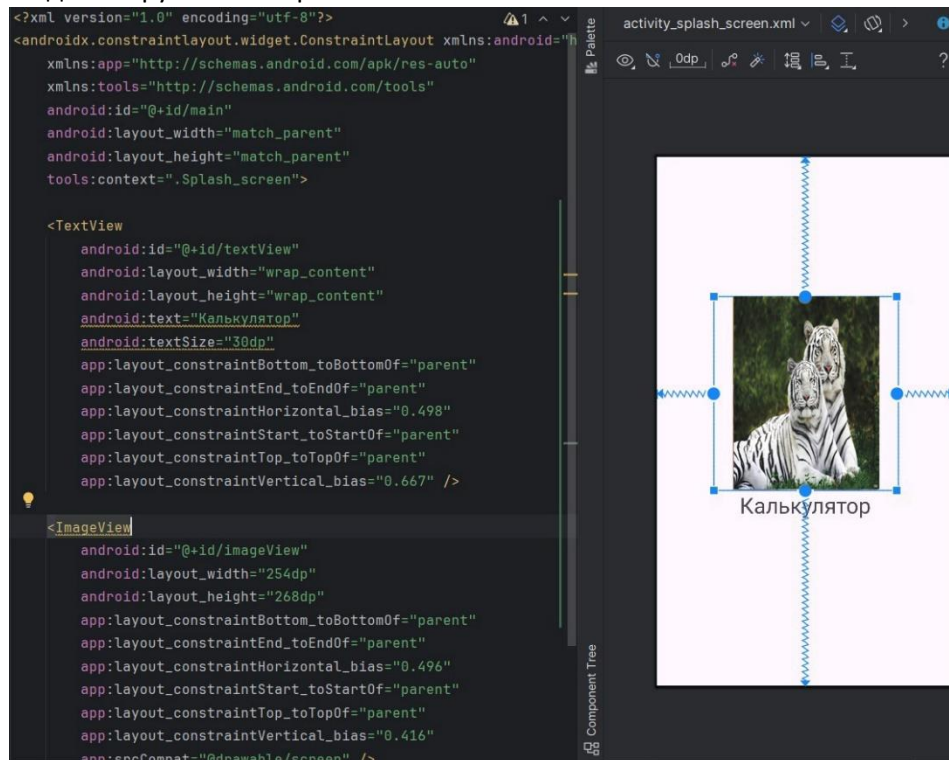
// Функция для показа результата вычислений
private fun showResult() {
    try {
        val expression = getInputExpression().replace( oldValue: "%", newValue: "/100") //Заменяем % на соответствующий символ
        val result = ExpressionBuilder(expression).build().evaluate()
        binding.output.text =
            DecimalFormat( pattern: "0.#####").format(result).toString()
        binding.output.setTextColor(
            ContextCompat.getColor( context: this,
                R.color.neon_green))
    } catch (e: Exception) {
        binding.output.text = "Ошибка"
        binding.output.setTextColor(ContextCompat.getColor( context: this,
            R.color.red))
    }
}
}

```

4. Создаю доп ветку в которой будет заставка



5. Создаю загрузочный экран



6. Редактирую манифест


```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools">
4
5      <application
6          android:allowBackup="true"
7          android:dataExtractionRules="@xml/data_extraction_rules"
8          android:fullBackupContent="@xml/backup_rules"
9          android:icon="@mipmap/ic_launcher"
10         android:label="Pr9"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportRtl="true"
13         android:theme="@style/Theme.Pr9"
14         tools:targetApi="31">
15         <activity
16             android:name=".MainActivity"
17             android:exported="false" />
18         <activity
19             android:name=".Splash_screen"
20             android:exported="true">
21             <intent-filter>
22                 <action android:name="android.intent.action.MAIN" />
23
24                 <category android:name="android.intent.category.LAUNCHER" />
25             </intent-filter>
26         </activity>
27     </application>
28
29 </manifest>

```

7. Результат

