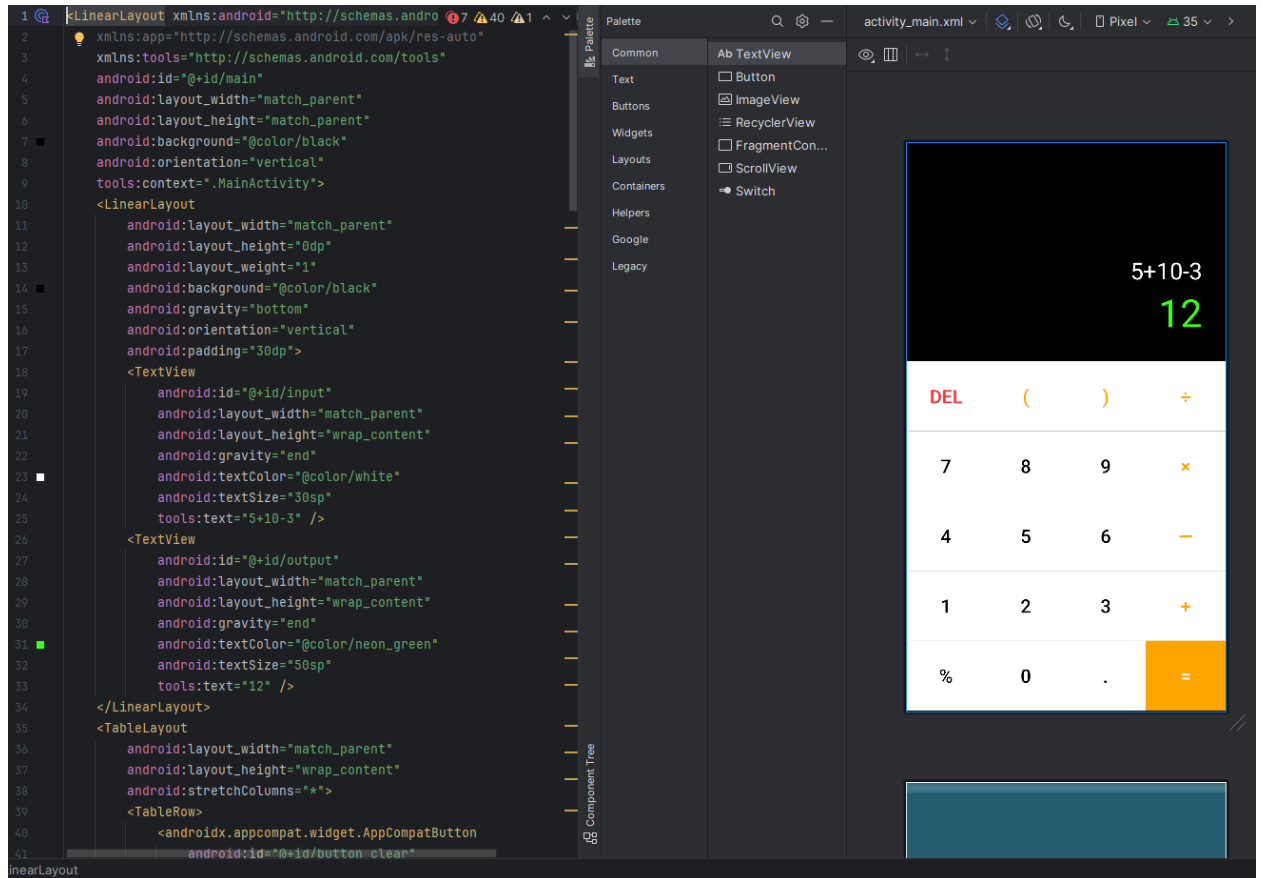


Практическая работа 9

1. Главная активность



2. Подключаем ViewBinding и библиотеку exp4j

```
buildFeatures{  
    viewBinding=true  
}
```

```
41 dependencies {  
42     implementation ("net.objecthunter:exp4j:0.4.8")  
43     implementation(libs.androidx.core.ktx)  
44     implementation(libs.androidx.appcompat)  
45     implementation(libs.material)  
46     implementation(libs.androidx.activity)  
47     implementation(libs.androidx.constraintlayout)  
48     testImplementation(libs.junit)  
49     androidTestImplementation(libs.androidx.junit)  
50     androidTestImplementation(libs.androidx.espresso.core)  
51 }
```

3. MainActivity, обработка нажатия на кнопки, решение выражения и вывод результата, если выражение некорректное, то программа выводит ошибку

```

3  > import ...
13
14  </> class MainActivity : AppCompatActivity() {
15      private lateinit var binding: ActivityMainBinding
16      @Override fun onCreate(savedInstanceState: Bundle?) {
17          super.onCreate(savedInstanceState)
18          enableEdgeToEdge()
19          binding = ActivityMainBinding.inflate(layoutInflater)
20          setContentView(binding.root)
21          ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
22              val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
23              v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
24              insets
25          }
26          binding.buttonClear.setOnClickListener{
27              binding.input.text = " "
28              binding.output.text = " "
29          }
30          binding.buttonBracketLeft.setOnClickListener {
31              addToInputText( value: "(")
32          }
33          binding.buttonBracketLeft.setOnClickListener {
34              addToInputText( value: ")")
35          }
36          binding.button0.setOnClickListener {
37              addToInputText( value: "0")
38          }
39          binding.button1.setOnClickListener {
40              addToInputText( value: "1")
41          }
42          binding.button2.setOnClickListener {
43              addToInputText( value: "2")
44          }
45          binding.button3.setOnClickListener {
46              addToInputText( value: "3")
47          }
48          binding.button4.setOnClickListener {
49              addToInputText( value: "4")
50          }
51          binding.button5.setOnClickListener {

```

```

52         addToInputText( value: "5")
53     }
54     binding.button6.setOnClickListener {
55         addToInputText( value: "6")
56     }
57     binding.button7.setOnClickListener {
58         addToInputText( value: "7")
59     }
60     binding.button8.setOnClickListener {
61         addToInputText( value: "8")
62     }
63     binding.button9.setOnClickListener {
64         addToInputText( value: "9")
65     }
66     binding.buttonDot.setOnClickListener {
67         addToInputText( value: ".")
68     }
69     binding.buttonDivision.setOnClickListener {
70         addToInputText( value: "÷")
71     }
72     binding.buttonMultiply.setOnClickListener {
73         addToInputText( value: "x")
74     }
75     binding.buttonSubtraction.setOnClickListener {
76         addToInputText( value: "-")
77     }
78     binding.buttonAddition.setOnClickListener {
79         addToInputText( value: "+")
80     }
81     binding.buttonEquals.setOnClickListener {
82         showResult()
83     }
84     binding.buttonPercent.setOnClickListener {
85         addToInputText( value: "%")
86     }
87 }
88 // Функция для добавления текста к полю ввода
89 private fun addToInputText(value: String) {
90     binding.input.append(value)// Добавляем переданное значение в конец поля ввода

```

```

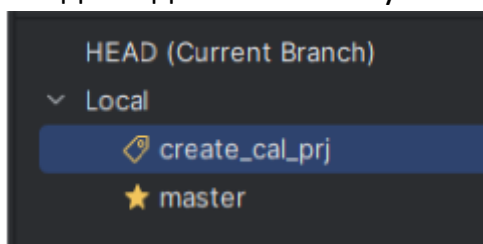
1 }
2 // Функция для получения строки ввода
3 private fun getInputExpression(): String {
4     return binding.input.text.toString()
5 }
6 // Функция для показа результата вычислений
7 private fun showResult() {
8     try {
9         val expression = getInputExpression().replace( oldValue: "%", newValue: "/100") //Заменяем % на соответствующее значение для деления
10        val result = ExpressionBuilder(expression).build().evaluate()
11        binding.output.text =
12            DecimalFormat( pattern: "0.#####").format(result).toString()
13        binding.output.setTextColor(
14            ContextCompat.getColor( context: this,
15                R.color.neon_green))
16    } catch (e: Exception) {
17        binding.output.text = "Ошибка"
18        binding.output.setTextColor(ContextCompat.getColor( context: this,
19            R.color.red))
20    }
21 }
22 }

```

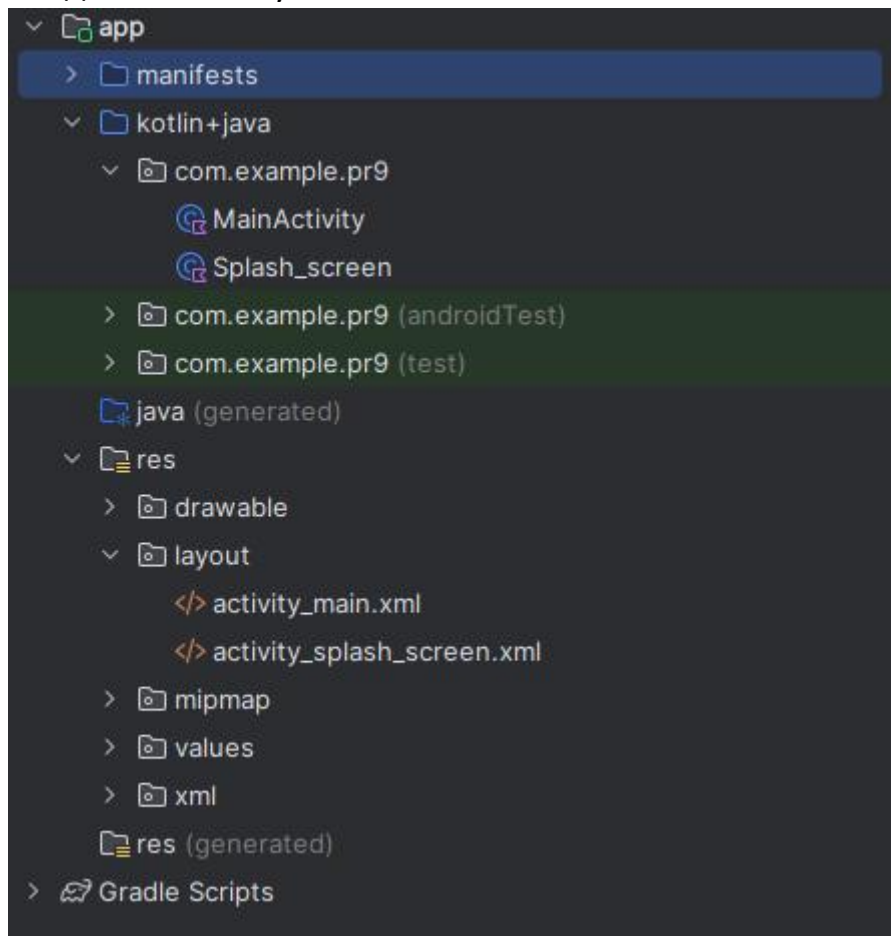
4. Цвета и оформление кнопки

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <style name="Button_Style" parent="Widget.AppCompat.Button.Colored">
4         <item name="android:background">@color/white</item>
5         <item name="android:textSize">24sp</item>
6         <item name="android:textColor">@color/black</item>
7         <item name="android:gravity">center</item>
8     </style>
9     <color name="purple_200">#FFBB86FC</color>
10    <color name="purple_500">#FF6200EE</color>
11    <color name="purple_700">#FF3700B3</color>
12    <color name="teal_200">#FF03DAC5</color>
13    <color name="teal_700">#FF018786</color>
14    <color name="black">#FF000000</color>
15    <color name="white">#FFFFFFFF</color>
16    <color name="red">#FF3131</color>
17    <color name="pink">#FFC0CB</color>
18    <color name="orange">#FFA500</color>
19    <color name="neon_green">#39FF14</color>
20 </resources>
21
22
```

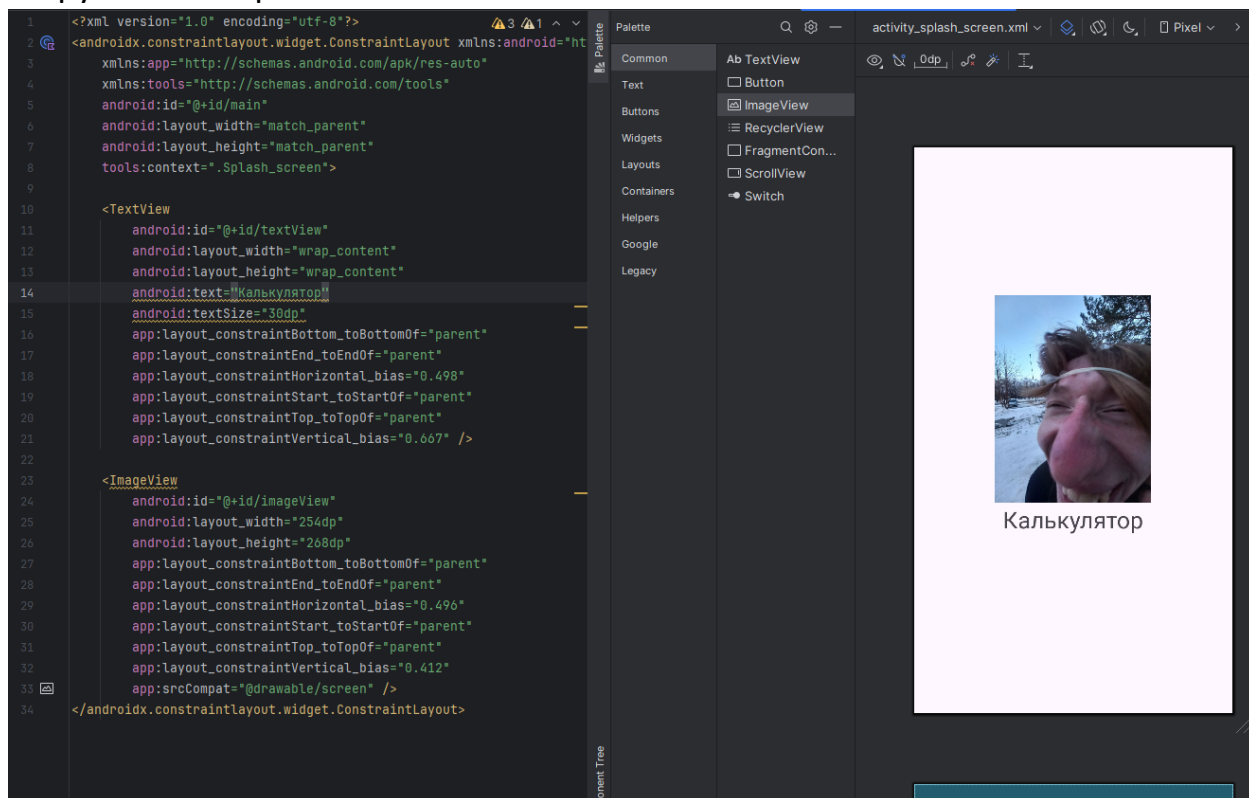
5. Создаем дополнительную ветку, в ней мы будем создавать заставку



6. Создаем заставку



7. Загрузочный экран



8. Редактируем 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="@string/app_name"
11     android:roundIcon="@mipmap/ic_launcher_round"
12     android:supportRtl="true"
13     android:theme="@style/Theme.AppCompat"
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>
```

9. Результат

