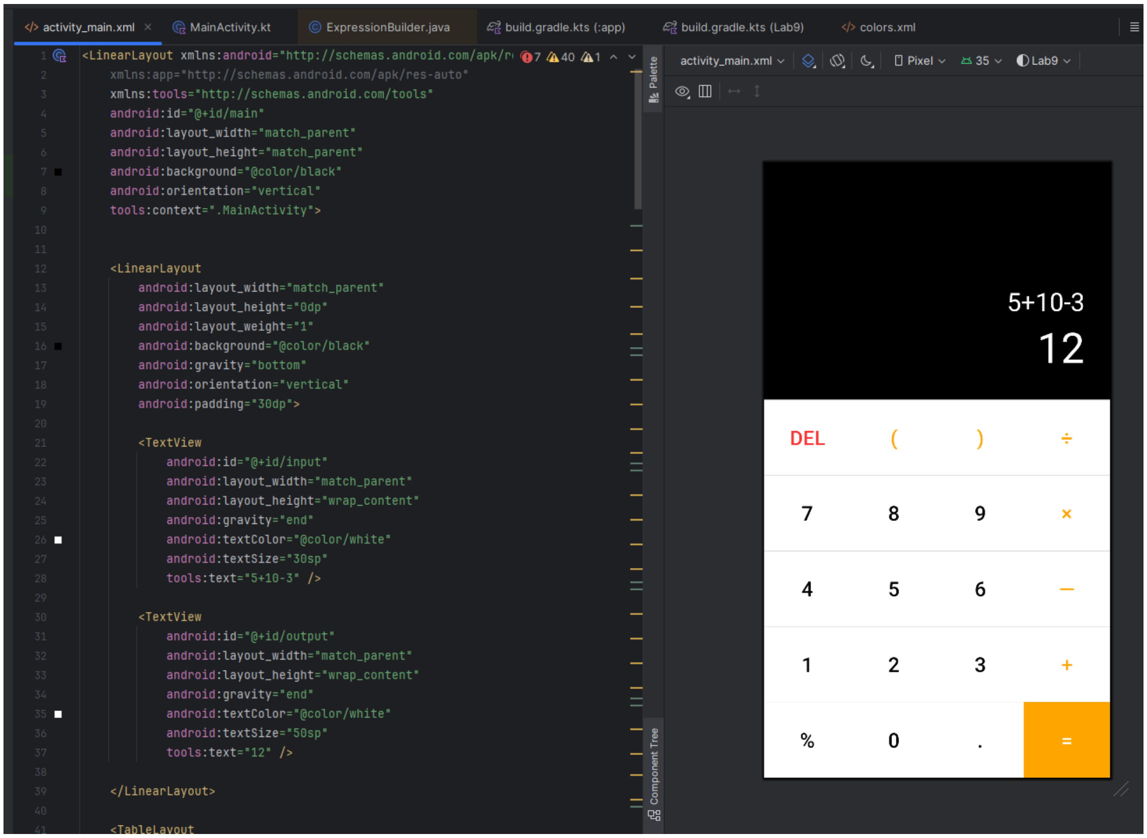
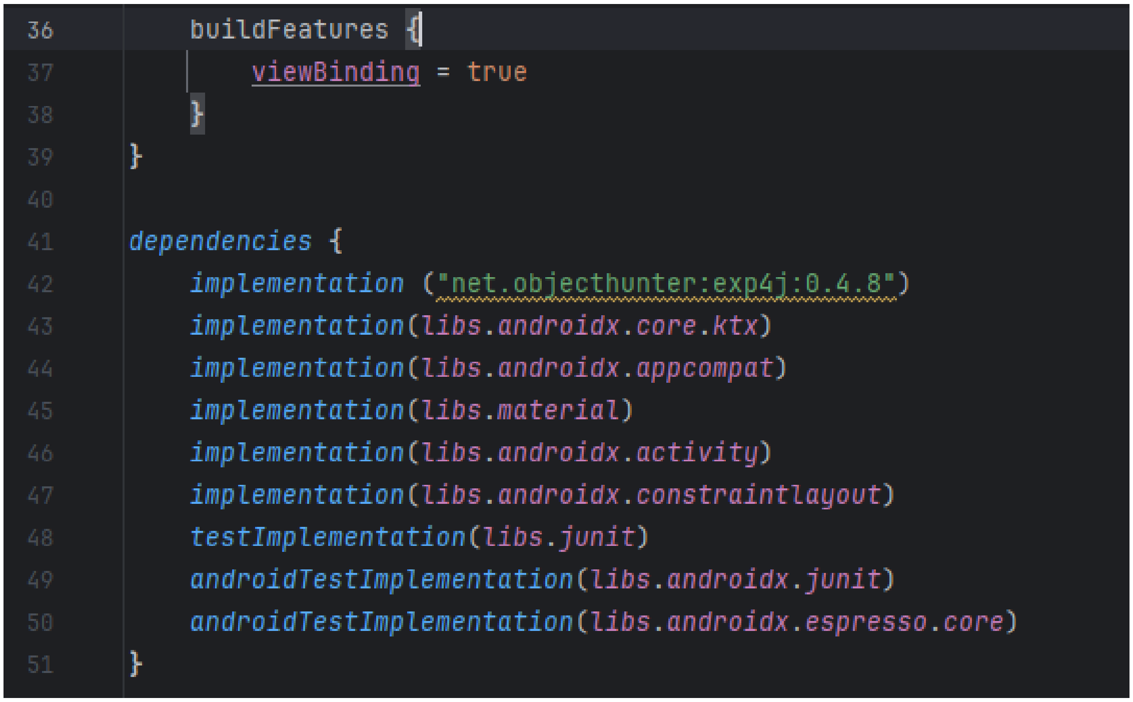
Практическая работа 9

1. Главная активность



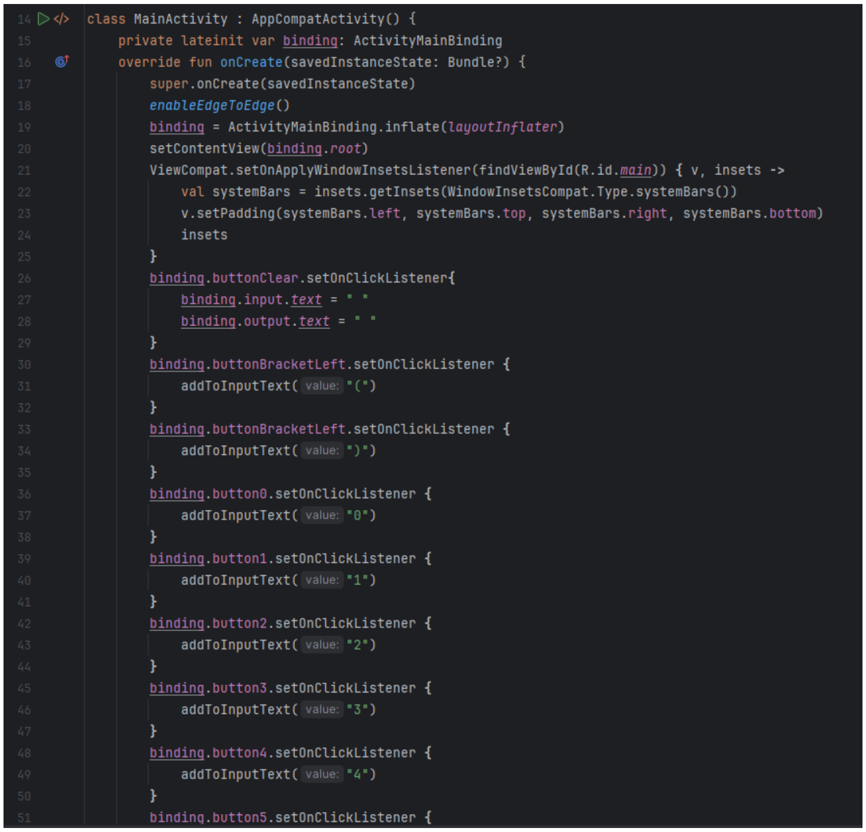
```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/r
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/black"
    android:orientation="vertical"
    tools:context=".MainActivity">


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="@color/black"
        android:gravity="bottom"
        android:orientation="vertical"
        android:padding="30dp">

        <TextView
            android:id="@+id/input"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="end"
            android:textColor="@color/white"
            android:textSize="30sp"
            tools:text="5+10-3" />

        <TextView
            android:id="@+id/output"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="end"
            android:textColor="@color/white"
            android:textSize="50sp"
            tools:text="12" />

    </LinearLayout>

    <TableLayout
```

2. Подключаем ViewBinding и библиотеку exp4j

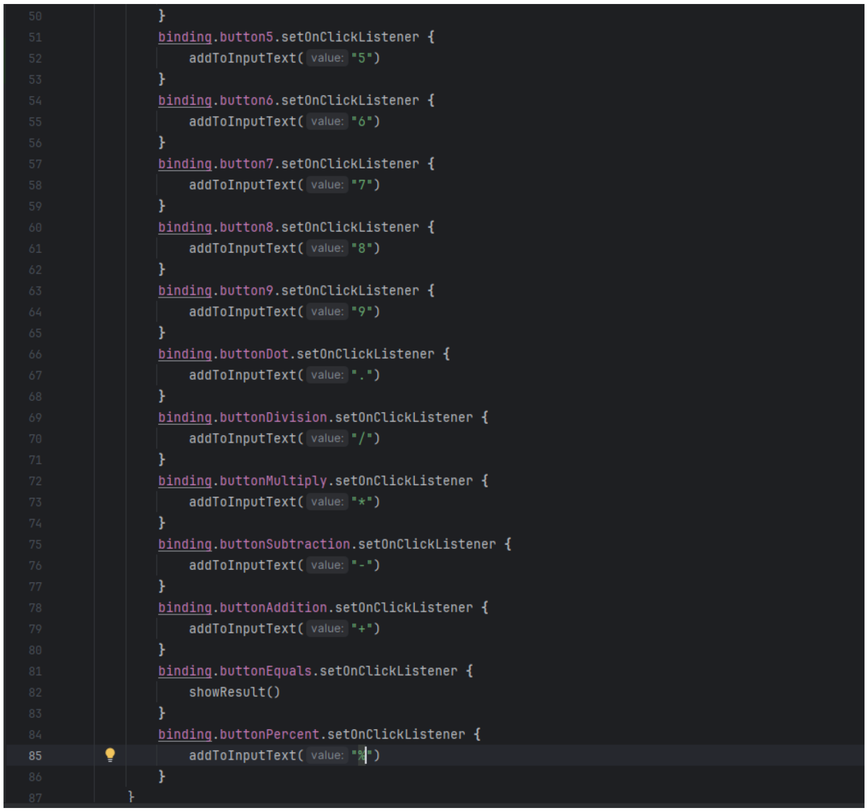


```kotlin
        buildFeatures {
            viewBinding = true
        }
}

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)
}
```

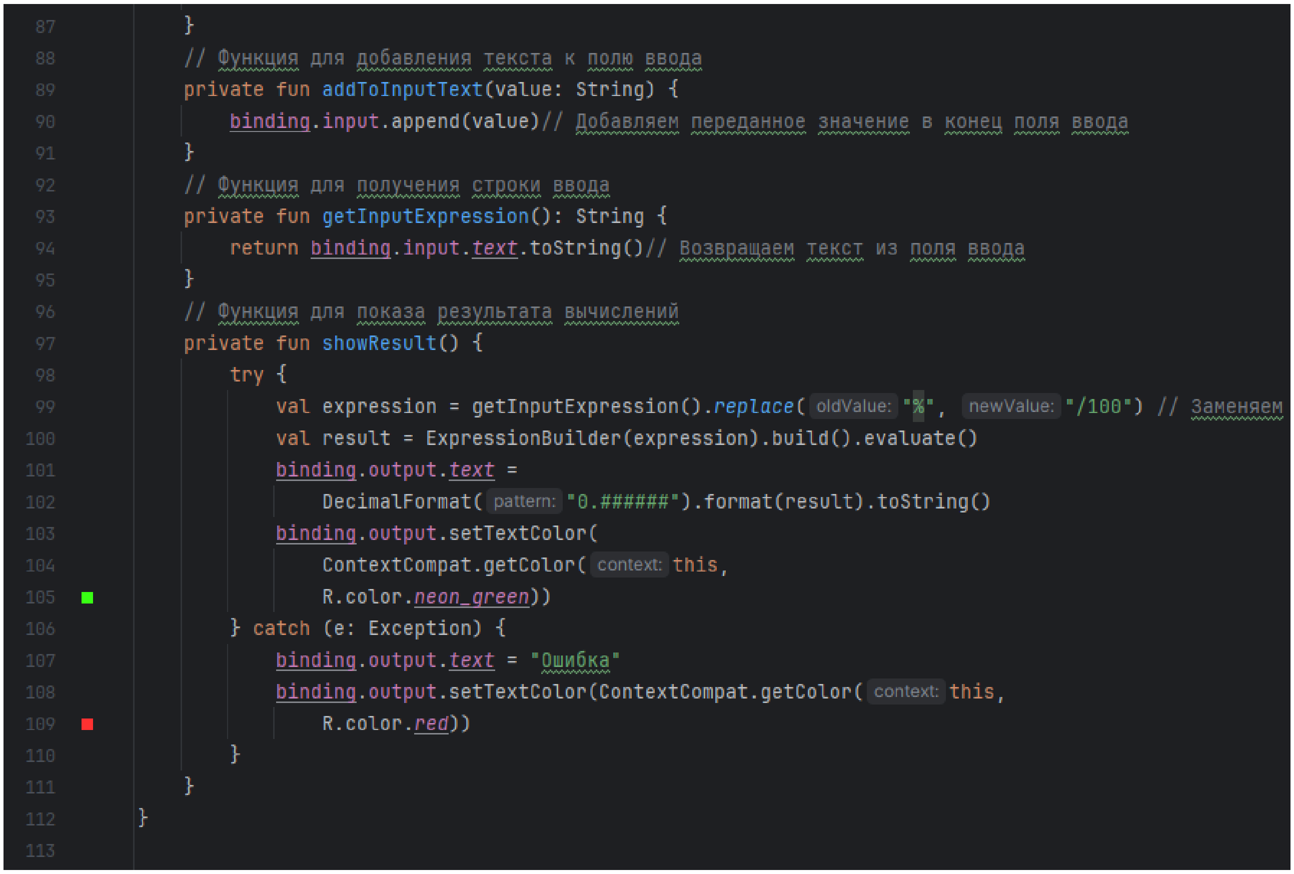
3. MainActivity, обработка нажатия на кнопки, решение выражения и вывод результата, если выражение некорректное, то программа выводит ошибку



```
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.buttonBracketLeft.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 {
```

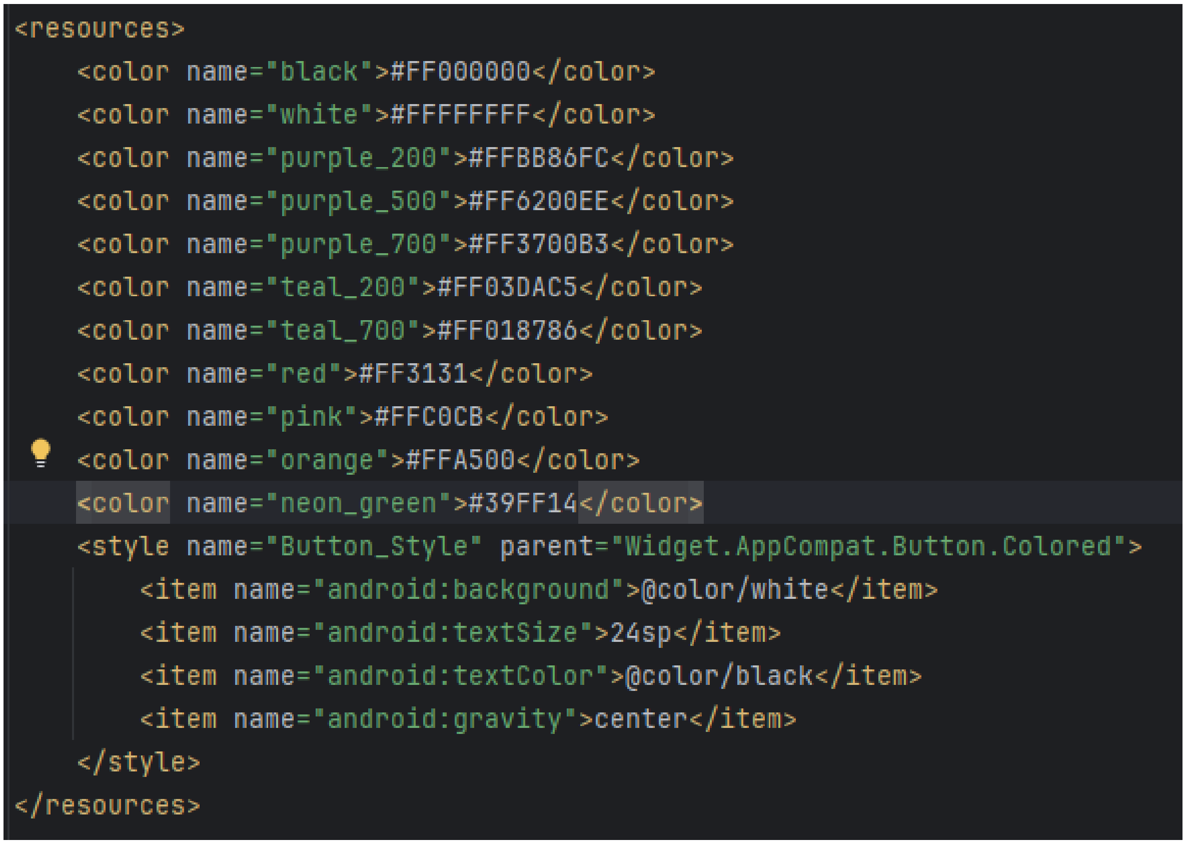


```
        }
        binding.button5.setOnClickListener {
            addToInputText( value: "5")
        }
        binding.button6.setOnClickListener {
            addToInputText( value: "6")
        }
        binding.button7.setOnClickListener {
            addToInputText( value: "7")
        }
        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: "*")
        }
        binding.buttonSubtraction.setOnClickListener {
            addToInputText( value: "-")
        }
        binding.buttonAddition.setOnClickListener {
            addToInputText( value: "+")
        }
        binding.buttonEquals.setOnClickListener {
            showResult()
        }
        binding.buttonPercent.setOnClickListener {
            addToInputText( value: "%")
        }
    }
```



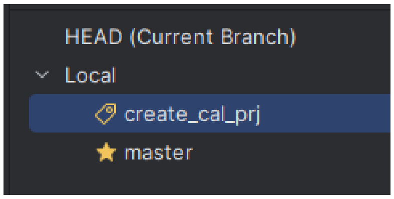
```
    }
    // Функция для добавления текста к полю ввода
    private fun addToInputText(value: String) {
        binding.input.append(value)// Добавляем переданное значение в конец поля ввода
    }
    // Функция для получения строки ввода
    private fun getInputExpression(): String {
        return binding.input.text.toString()// Возвращаем текст из поля ввода
    }
    // Функция для показа результата вычислений
    private fun showResult() {
        try {
            val expression = getInputExpression().replace( oldValue: "%",  newValue: "/100") // Заменяем
            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. Цвета и оформление кнопки



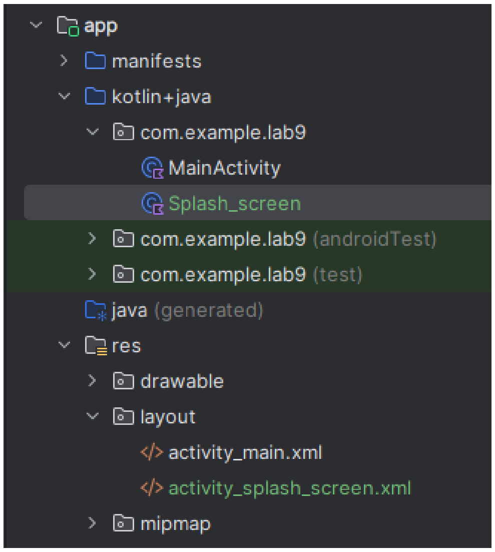
```
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="red">#FF3131</color>
    <color name="pink">#FFC0CB</color>
    <color name="orange">#FFA500</color>
    <color name="neon_green">#39FF14</color>
    <style name="Button_Style" parent="Widget.AppCompat.Button.Colored">
        <item name="android:background">@color/white</item>
        <item name="android:textSize">24sp</item>
        <item name="android:textColor">@color/black</item>
        <item name="android:gravity">center</item>
    </style>
</resources>
```

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



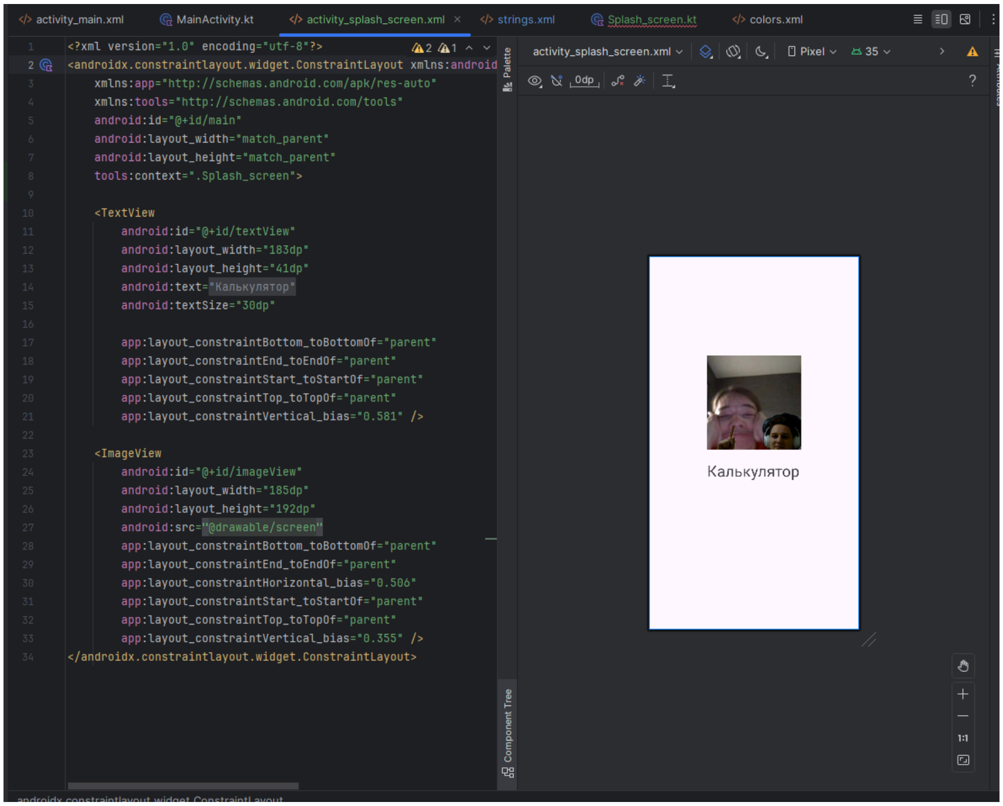
```
HEAD (Current Branch)
Local
    create_cal_prj
    master
```

6. Создаем заставку



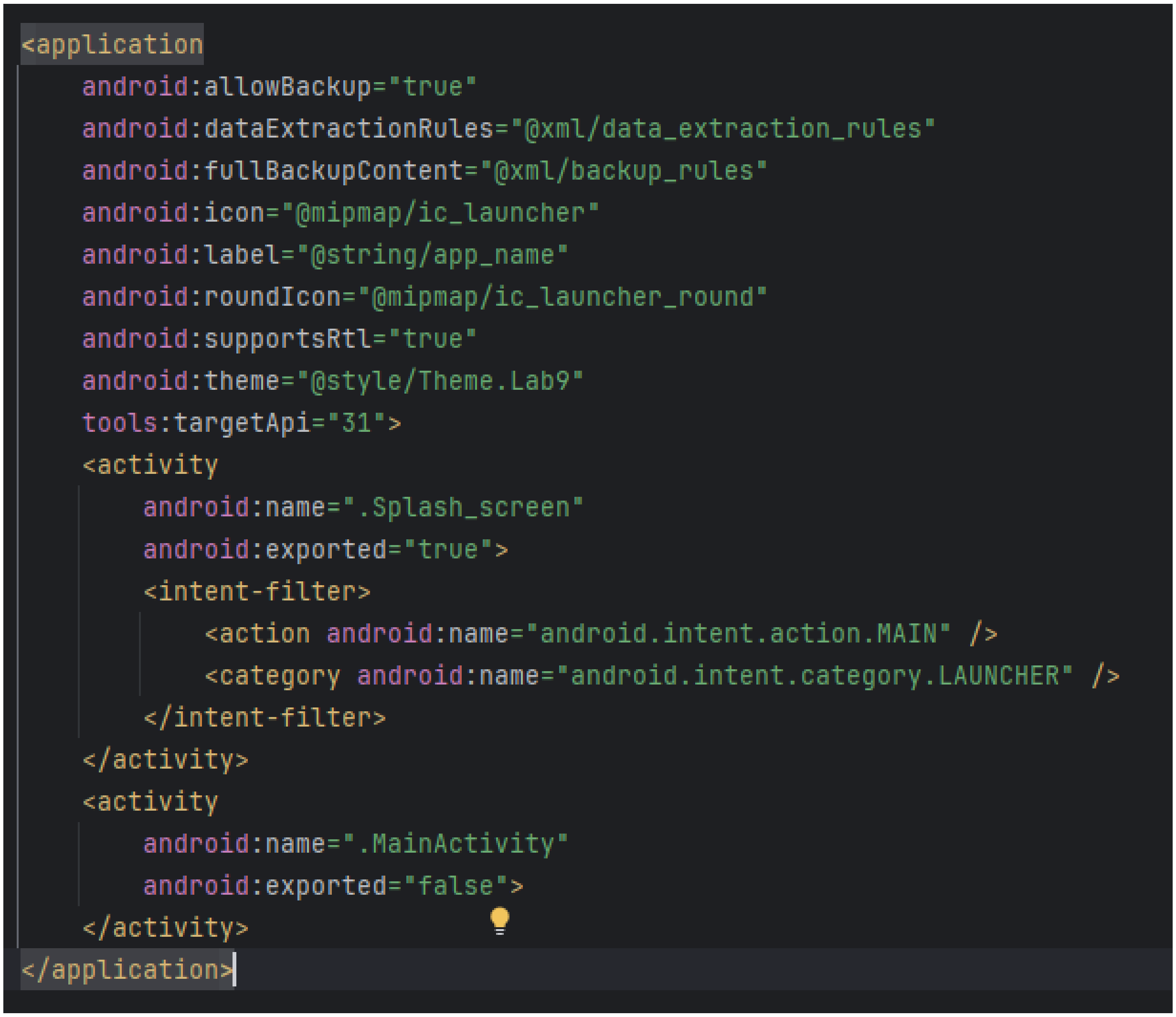
```
app
    manifests
    kotlin+java
        com.example.lab9
            MainActivity
            Splash_screen
        com.example.lab9 (androidTest)
        com.example.lab9 (test)
    java (generated)
    res
        drawable
        layout
            activity_main.xml
            activity_splash_screen.xml
        mipmap
```

7. Загрузочный экран



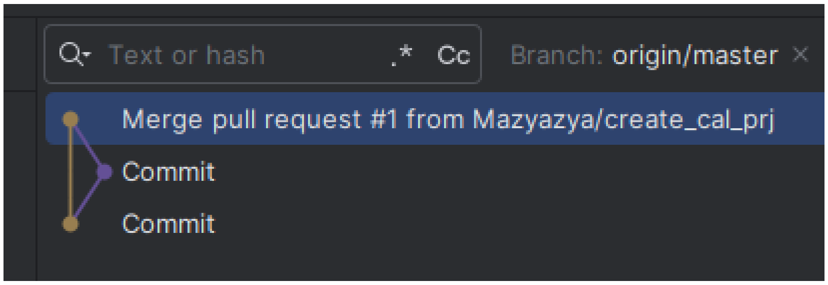
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Splash_screen">

    <TextView
        android:id="@+id/textView"
        android:layout_width="183dp"
        android:layout_height="41dp"
        android:text="Калькулятор"
        android:textSize="30dp"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.581" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="185dp"
        android:layout_height="192dp"
        android:src="@drawable/screen"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.506"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.355" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

8. Редактируем манифест



```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Lab9"
    tools:targetApi="31">
    <activity
        android:name=".Splash_screen"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".MainActivity"
        android:exported="false">
    </activity>
</application>
```

9. Делаем Pull-Request в основную ветку

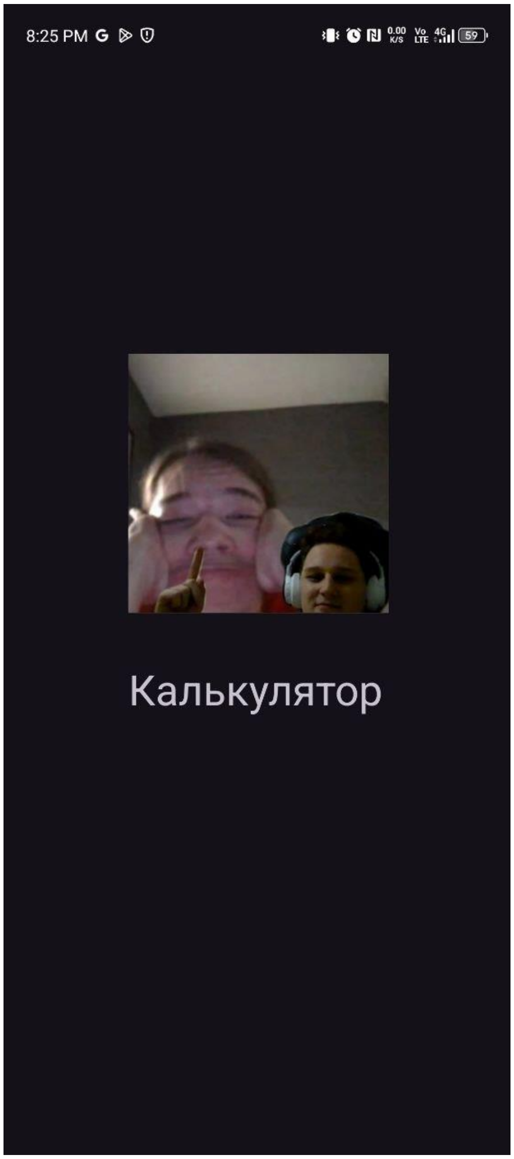


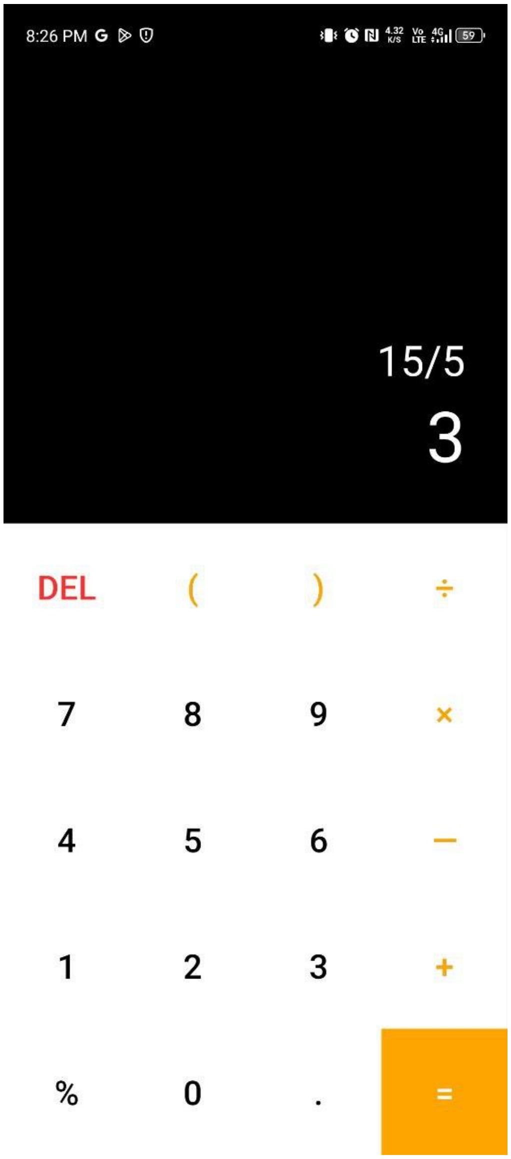
Merge pull request #1 from Mazyazya/create_cal_prj

Commit

Commit

10. Результат:


8:25 PM
Калькулятор


8:26 PM
15/5
3
DEL
(
)
÷
7
8
9
×
4
5
6
−
1
2
3
+
%
0
.
=