

Практика 9

Мэин активити

```
package com.example.ract3

import ...

class MainActivity : AppCompatActivity() {
    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
                insets
        }

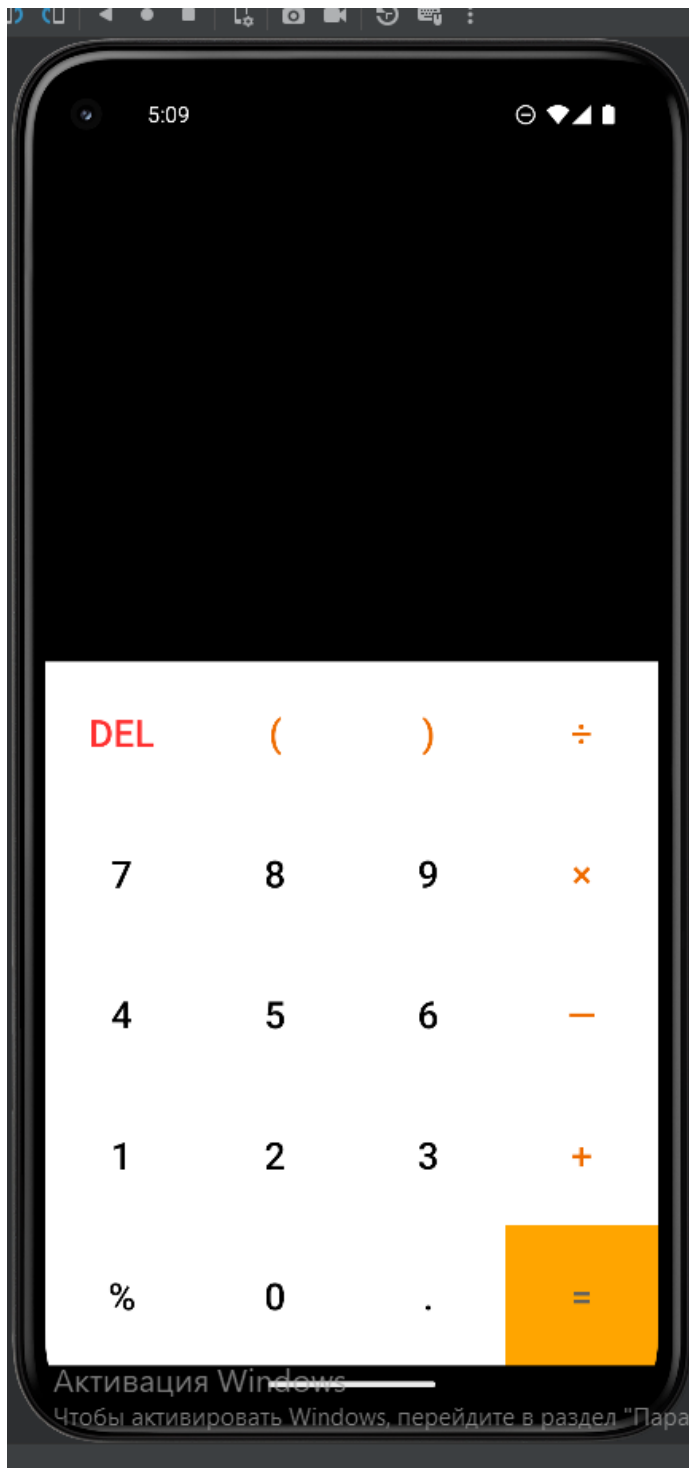
        binding.buttonClear.setOnClickListener{
            binding.input.text = ""
            binding.output.text = ""
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( value: "(")
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( 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")
        }
    }
}
```

```
private fun showResult() {
    try {
        val expression = getInputExpression().replace( oldValue: "%", newValue: "/100")
        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))
    }
}

private fun addToInputText(value: String) {
    binding.input.append(value)
}

private fun getInputExpression(): String {
    return binding.input.text.toString()
}
```

Кулькүлөтөр



1. Добавление цветов: Цветовая палитра для интерфейса приложения была добавлена в файл colors.xml. Это включает основные цвета для фона, текста и кнопок, а также стили для кнопок.
2. Создание интерфейса: В файле activity_main.xml был использован GridLayout для построения структуры калькулятора. Внесены текстовые поля для ввода и вывода, а также кнопки для цифр и операций.
3. Применение ViewBinding: Подключили ViewBinding в файле Gradle и инициализировали его в MainActivity для упрощения доступа к элементам интерфейса.

4. Обработка нажатий кнопок: Для каждой кнопки были настроены обработчики нажатий. При нажатии на кнопку соответствующий символ добавляется в поле ввода.

5. Интеграция библиотеки `exp4j`: Включили зависимость `exp4j` в `build.gradle.kts`, чтобы обрабатывать математические выражения в виде строк.

6. Вычисление результата: Реализовали метод `showResult()`, использующий `exp4j` для вычисления введенного выражения. Метод обрабатывает как корректные, так и некорректные выражения, показывая результат или сообщение об ошибке.

7. Создание заставки: Была создана новая активность `Splash_screen`, отображающая заставку перед запуском основного калькулятора. Заставка длится 2 секунды, после чего происходит переход к `MainActivity`.

8. Изменения в манифесте: Внесены корректировки в `AndroidManifest.xml`, чтобы указать, что `Splash_screen` является стартовой активностью приложения.