ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РЕСПУБЛИКИ ХАКАСИЯ

«ХАКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07

«Информационные системы и программирование»

Практическое занятие 1

Руководитель:

\_\_\_\_\_\_\_\_ И. Н. Брюханова

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка, дата)

Выполнил:

Студент группы ИС(ПРО)-41

\_\_\_\_\_\_\_\_ А. А. Мальцева

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Абакан 2025 г.

**ЗАДАНИЕ №1**

**Задание:**

Создайте простое Android-приложение «Счётчик» с тремя кнопками ("+", "-", "Сброс") и текстовым полем для отображения числа, где кнопки увеличивают, уменьшают и обнуляют значение счётчика.

Для выполнения задания используйте стандартные View-компоненты (TextView, Button) и обработчики кликов setOnClickListener в Activity

Рисунок 1 – Требование к внешнему виды программы

**Решение:**

**Результат:**

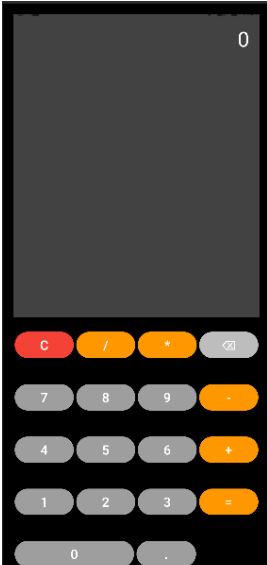
Рисунок 3 – Компилирование задания 1

**ЗАДАНИЕ №2**

**Задание:**

Разработай калькулятор с базовыми арифметическими операциями (+, -, \*, /) и кнопкой "=", используя TextView для отображения ввода и результата, цифровые кнопки от 0 до 9, кнопки операций и очистки, реализуя логику вычислений в Activity с обработчиками кликов для каждой кнопки и отображением промежуточных результатов в текстовом поле. Для компоновки UI используйте Linearlayout.

Примечание: уже можно было упростить до взятие двух чисел и простого счета.

Рисунок 4 – Требование к внешнему виду программы

**Решение:**

Листинг задания 1.1 - MainActivity.kt

package com.example.maa\_calulate //использование пакета проетка,путь к проекту

import android.annotation.SuppressLint

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() { //вход в основнйо класс входа в программу

private lateinit var editTextNumber1: EditText //поля которые инициализируют объекты и их тип

private lateinit var editTextNumber2: EditText

private lateinit var textViewResult: TextView

private lateinit var buttonAdd: Button

private lateinit var buttonSubtract: Button

private lateinit var buttonMultiply: Button

private lateinit var buttonDivide: Button

private lateinit var buttonEquals: Button

private lateinit var buttonClear: Button

private var currentOperation: String? = null //хранит операцию которая выполняется

private var number1: Double = 0.0 // тип поля для ввода числа 1

private var number2: Double = 0.0 // тип поля для ввода числа 2 плавающая точка, удвоенное значение float

override fun onCreate(savedInstanceState: Bundle?) //функция для создания активити

{

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main) //использование файла xml activity\_main

initViews()

setupClickListeners()

}

private fun initViews() //инициализация элементов в View (xml. они переходят по ID объекта)

{

editTextNumber1 = findViewById(R.id.Num1)

editTextNumber2 = findViewById(R.id.Num2)

textViewResult = findViewById(R.id.textViewResult)

buttonAdd = findViewById(R.id.plus)

buttonSubtract = findViewById(R.id.mines)

buttonMultiply = findViewById(R.id.multiply)

buttonDivide = findViewById(R.id.divide)

buttonEquals = findViewById(R.id.equality)

buttonClear = findViewById(R.id.drop)

}

private fun setupClickListeners() //функция отклика действий при нажатии на кнопки

{

buttonAdd.setOnClickListener { setOperation("+") }

buttonSubtract.setOnClickListener { setOperation("-") }

buttonMultiply.setOnClickListener { setOperation("\*") }

buttonDivide.setOnClickListener { setOperation(":") }

buttonEquals.setOnClickListener { calculateResult() } // вызов функции выводящей результат

buttonClear.setOnClickListener { clearAll() } // вызов функции очищающий поля num1 и num 2, а так-же результат

}

@SuppressLint("SetTextI18n") //отключение проблем инициализации

private fun setOperation(operation: String)

{

if (validateInput())

{

currentOperation = operation //передает в новую перменную выбранную операцию

number1 = editTextNumber1.text.toString().toDouble() //перевод текста из стркои в тип double

number2 = editTextNumber2.text.toString().toDouble() //перевод текста из стркои в тип double

textViewResult.text = "ヾ(\*'▽'\*) для вывода операции $operation , нажмите '=' " //выводит текст с опрацией

}

}

@SuppressLint("SetTextI18n")

private fun calculateResult()

{

if (currentOperation == null) //если не выбрана операция

{

showToast("Сначала выберите операцию ヾ(`ヘ´)ﾉﾞ") //текст о том что бы пользователь ввел операцию

return

}

if (!validateInput()) //провекра кода на валидацию, если не успешно

{

return //выход из функции

}

number1 = editTextNumber1.text.toString().toDouble() //перевод текста в тип с плавающи йтчкой (двойной float)

number2 = editTextNumber2.text.toString().toDouble() //перевод текста в тип с плавающи йтчкой (двойной float)

val result = when (currentOperation) //счет операции, когда выбрана операция, запись результата в result

{

"+" -> number1 + number2

"-" -> number1 - number2

"\*" -> number1 \* number2

":" -> {

if (number2 != 0.0) //провекра на то, что бы число на которое будет делиться - не равно 0

{

number1 / number2 // если не ноль то разделяем

}

else

{

showToast("Σ(O\_O) Вы делите на ноль, вы что!?А ну исправляйте вычисление!") //если все-тки разделили на 0

return

}

}

else -> 0.0 //иначе 0

}

textViewResult.text = "(￣▽￣)ノ и ваш результат вычислений: $result" //установке текста с показом результата вычисления

}

private fun clearAll() //функция сброса

{

editTextNumber1.text.clear() //очищение поля с номером 1

editTextNumber2.text.clear() //очищение поля с номером 2

textViewResult.text = "( ・・)つ―●○◎- все сброшено и ждет расчета! Накорми калькулятор вычислениями!" //вывод текста в результат как результат

currentOperation = null //сброс операции присовоением нового значения

number1 = 0.0 //присвоение значения для ввода цифр 0

number2 = 0.0 //присвоение значения для ввода цифр 0

}

private fun validateInput(): Boolean //функция на проверку заполнения

{

val num1Text = editTextNumber1.text.toString() //принимает текст и переводит в строковый тип при вводе чисел в поле

val num2Text = editTextNumber2.text.toString()

if (num1Text.isEmpty() || num2Text.isEmpty()) //проверка что оба поля под числа заняты, и если 1 или 2 не введено

{

showToast("Вы числа считать пришли, или пустоты вычислять? (￣□￣」)")

return false

}

//далее организован блок проверки как в C#

try//попытка перевести введеные числа в тип с плавающей точкой

{

num1Text.toDouble()

num2Text.toDouble()

}

catch (e: NumberFormatException) //вылавливание числел с неверным форматом

{

showToast("(ᓀ \_ᓀ) введите корректные числа...")

return false //возвращает false

}

return true// если все прошло возвращает true

}

private fun showToast(message: String) //функция вывода строки для сообщений

{

Toast.makeText(this, message, Toast.LENGTH\_SHORT).show() //создание текста и показ

}

}

Листинг задания 1.2 – activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<EditText

android:id="@+id/Num1"

android:layout\_width="381dp"

android:layout\_height="61dp"

android:layout\_marginBottom="8dp"

android:hint="Первое число"

android:inputType="numberDecimal" />

<EditText

android:id="@+id/Num2"

android:layout\_width="383dp"

android:layout\_height="67dp"

android:layout\_marginBottom="16dp"

android:hint="Второе число"

android:inputType="numberDecimal" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginBottom="16dp">

<Button

android:id="@+id/plus"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="+"

android:backgroundTint="@color/design\_default\_color\_secondary"

android:layout\_marginEnd="4dp" />

<Button

android:id="@+id/mines"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="-"

android:backgroundTint="@color/design\_default\_color\_secondary"

android:layout\_marginEnd="4dp" />

<Button

android:id="@+id/multiply"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="\*"

android:backgroundTint="@color/design\_default\_color\_secondary"

android:layout\_marginEnd="4dp" />

<Button

android:id="@+id/divide"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:backgroundTint="@color/design\_default\_color\_secondary"

android:text=":" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<Button

android:id="@+id/equality"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="="

android:layout\_marginEnd="8dp"

android:backgroundTint="@color/design\_default\_color\_secondary" />

<Button

android:id="@+id/drop"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Сброс"

android:backgroundTint="@color/design\_default\_color\_secondary"

tools:ignore="HardcodedText" />

</LinearLayout>

<TextView

android:id="@+id/textViewResult"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="24dp"

android:text="результат вычисления"

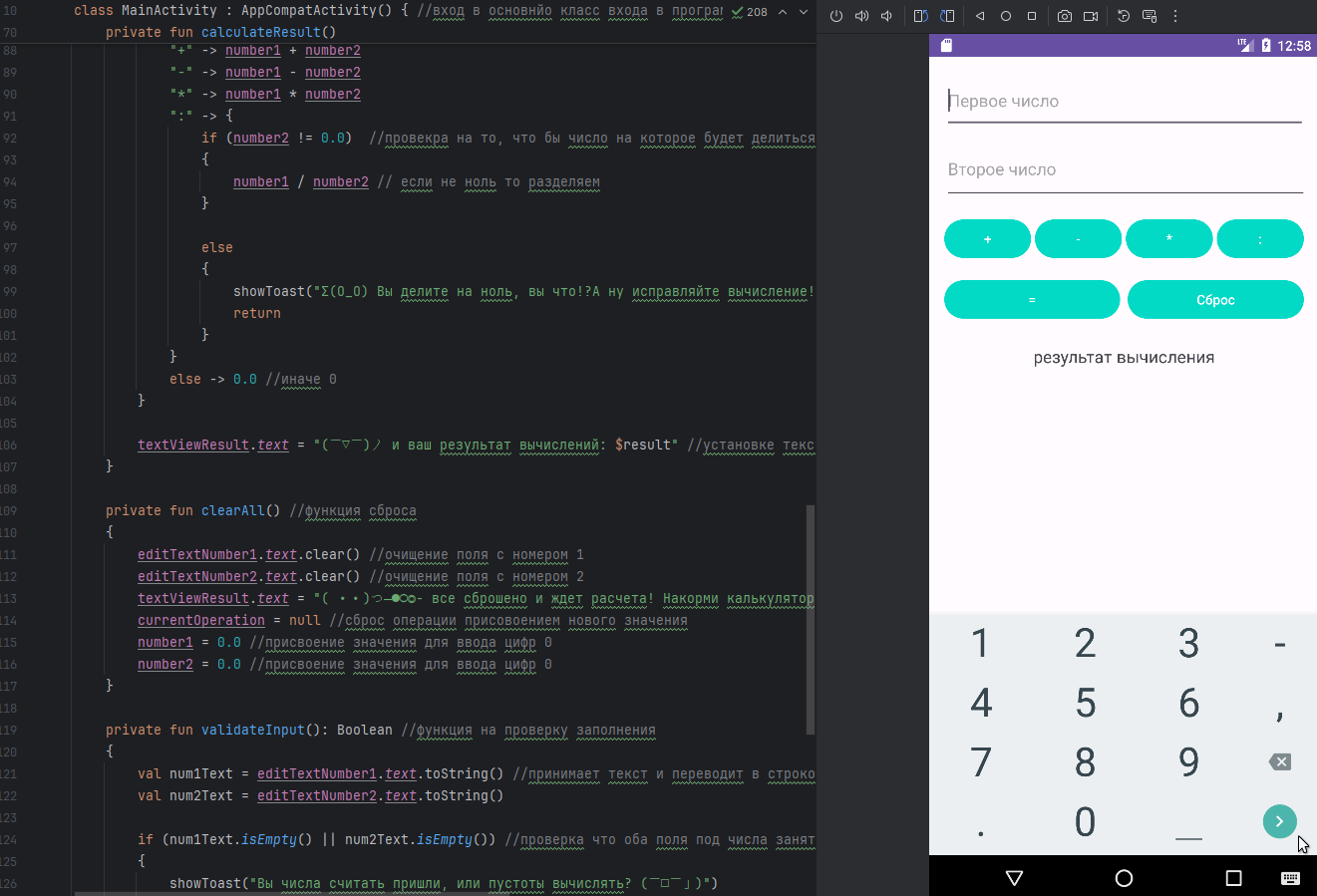
android:textSize="18sp"

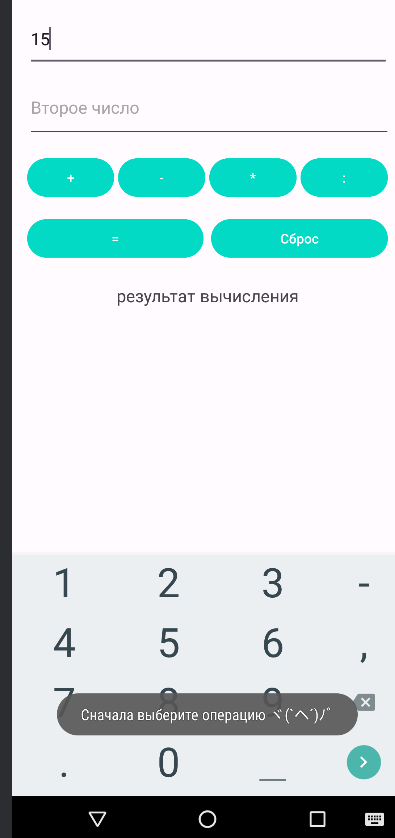
android:gravity="center"

tools:ignore="HardcodedText" />

</LinearLayout>

**Результат:**

Рисунок 8 – Компилирование задания 2

Рисунок - Ошибка при нажатии =

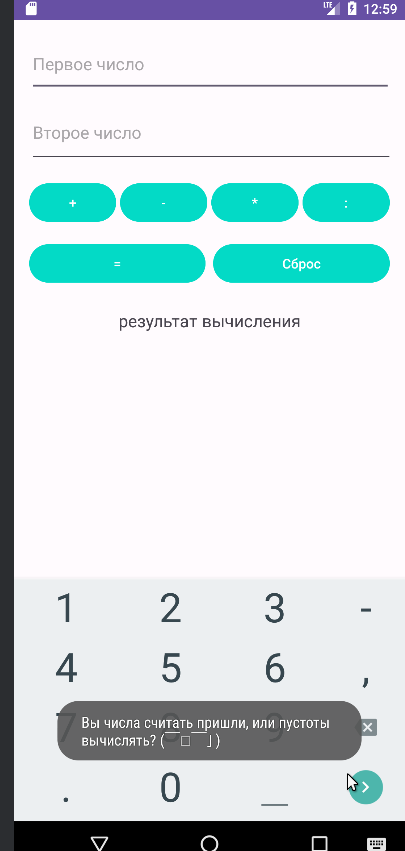
**Р**исунок - Ошибка при выборе операции с пустыми значениями

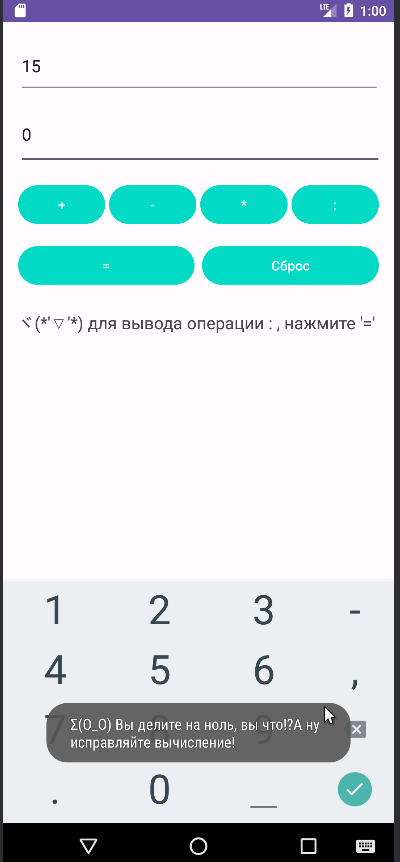
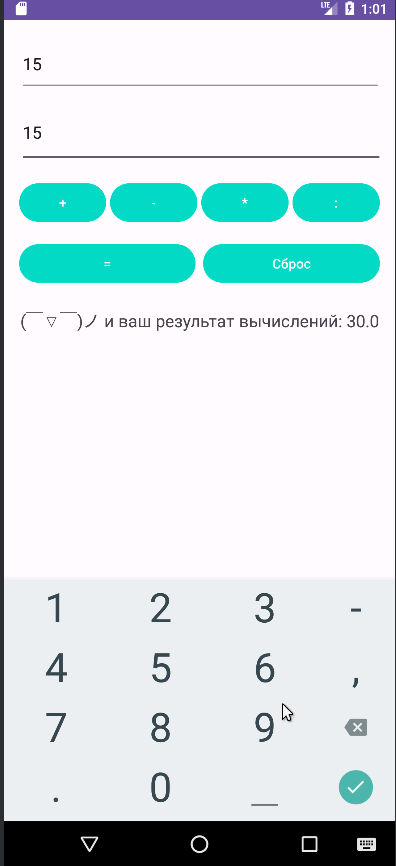
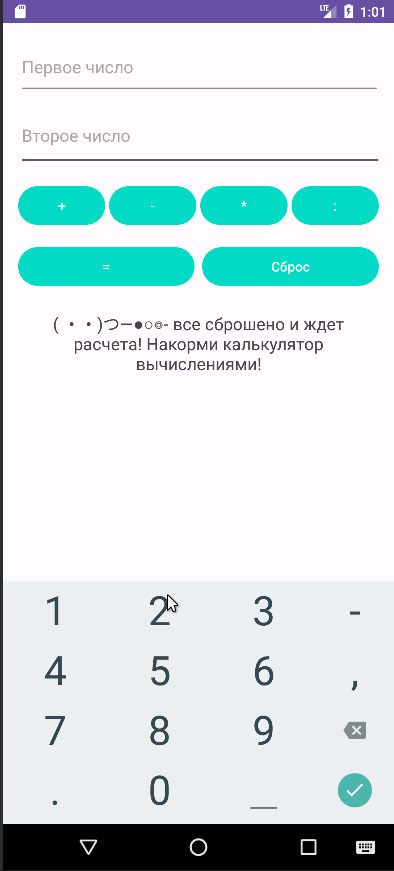
Рисунок - Ошибка при делении на 0 после нажатия =

Рисунок - Вычисление любой операции если все корректно

Рисунок - Вывод после нажатия кнопки сброса

**ВЫВОД:**