ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | С.В. Щекин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| Создание приложения с различными интерфейсными элементами |
| по курсу: Программирование мобильных устройств |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Д.В. Самарин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

**Задание:**

Создать простое приложение c использованием различных интерфейсных элементов, выполнить сборку и запуск приложения. Убедиться в корректном отображении интерфейсных элементов в основном окне. Описать результаты в отчете (в т.ч. тексты xml 12 для activity).

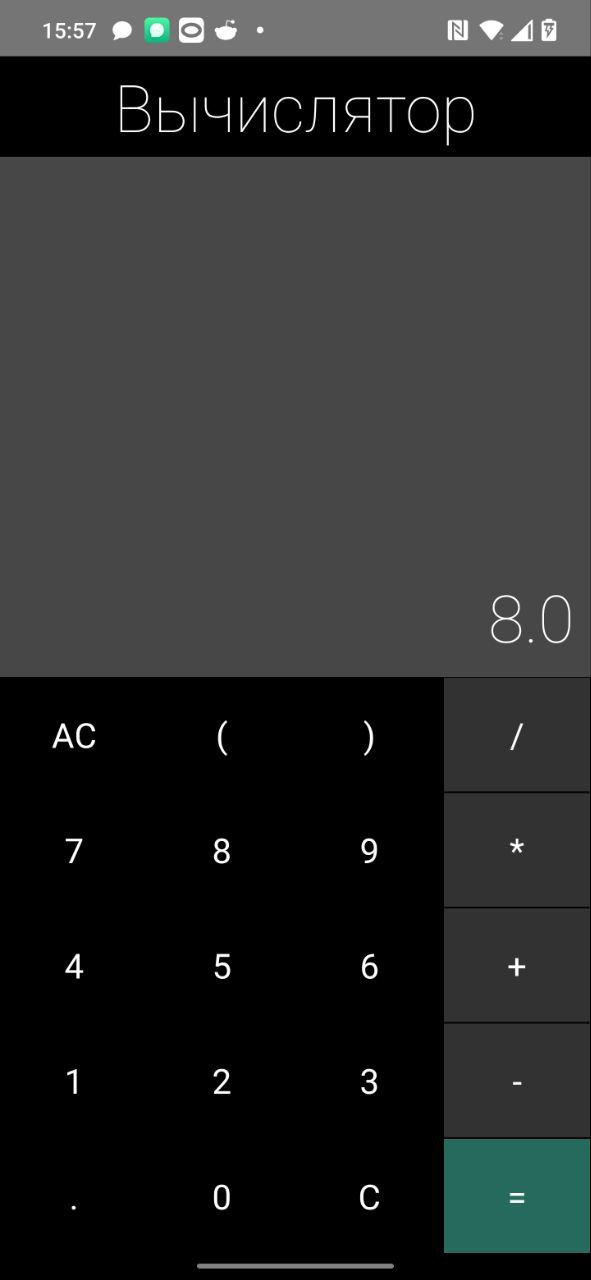
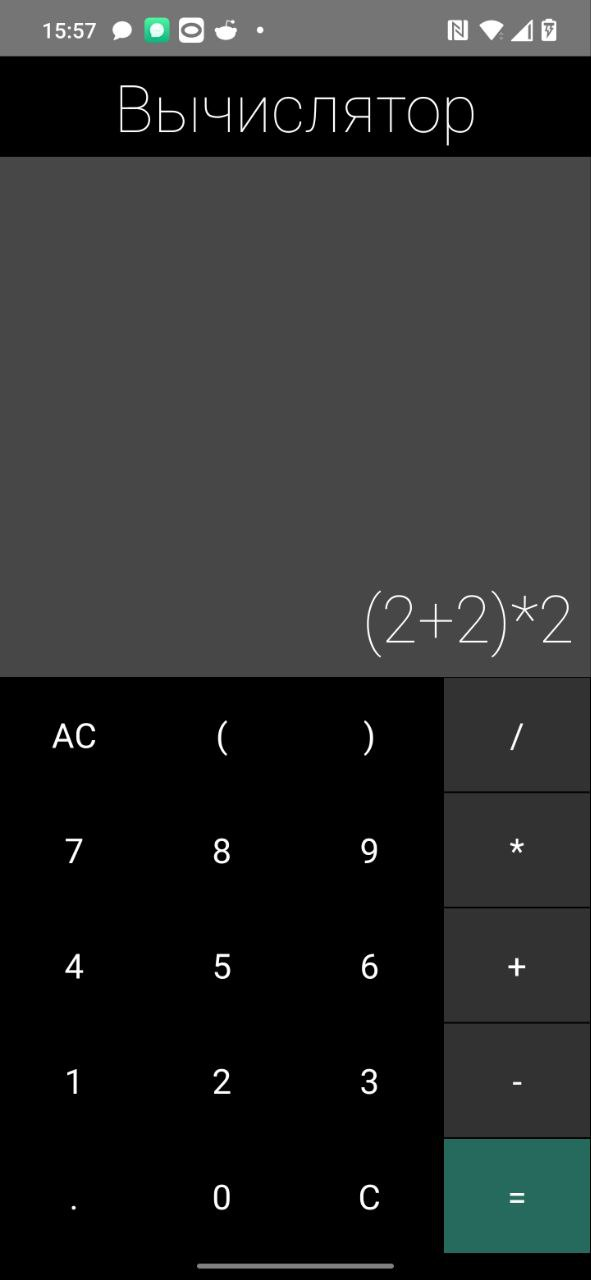
**Цель работы:**

Освоение возможностей создания приложения с различными типами

интерфейсных элементов для мобильного устройства

**Ход работы:**

Созданное приложение работает в качестве калькулятора.



**Листинг:**

**MainAktivity.kt**

|  |
| --- |
| package com.example.lab3calc import android.widget.TextView import android.os.Bundle import androidx.activity.ComponentActivity import androidx.activity.compose.setContent import androidx.compose.foundation.layout.fillMaxSize import androidx.compose.material3.MaterialTheme import androidx.compose.material3.Surface import androidx.compose.material3.Text import androidx.compose.runtime.Composable import androidx.compose.ui.Modifier import androidx.compose.ui.tooling.preview.Preview import com.example.lab3calc.ui.theme.LAB3calcTheme import net.objecthunter.exp4j.ExpressionBuilder   class MainActivity : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  findViewById<TextView>(R.id.*btn\_0*).setOnClickListener**{**setTextFields("0")**}** findViewById<TextView>(R.id.*btn\_AC*).setOnClickListener **{** findViewById<TextView>(R.id.*math\_operation*).*text* = ""  findViewById<TextView>(R.id.*math\_operation*).*text* = ""  **}** findViewById<TextView>(R.id.*btn\_right*).setOnClickListener **{** setTextFields("(") **}** findViewById<TextView>(R.id.*btn\_left*).setOnClickListener **{** setTextFields(")") **}** findViewById<TextView>(R.id.*btn\_palka*).setOnClickListener **{** setTextFields("/") **}** findViewById<TextView>(R.id.*btn\_7*).setOnClickListener **{** setTextFields("7") **}** findViewById<TextView>(R.id.*btn\_8*).setOnClickListener **{** setTextFields("8") **}** findViewById<TextView>(R.id.*btn\_9*).setOnClickListener **{** setTextFields("9") **}** findViewById<TextView>(R.id.*btn\_zvezda*).setOnClickListener **{** setTextFields("\*") **}** findViewById<TextView>(R.id.*btn\_4*).setOnClickListener **{** setTextFields("4") **}** findViewById<TextView>(R.id.*btn\_5*).setOnClickListener **{** setTextFields("5") **}** findViewById<TextView>(R.id.*btn\_6*).setOnClickListener **{** setTextFields("6") **}** findViewById<TextView>(R.id.*btn\_plus*).setOnClickListener **{** setTextFields("+") **}** findViewById<TextView>(R.id.*btn\_1*).setOnClickListener **{** setTextFields("1") **}** findViewById<TextView>(R.id.*btn\_2*).setOnClickListener **{** setTextFields("2") **}** findViewById<TextView>(R.id.*btn\_3*).setOnClickListener **{** setTextFields("3") **}** findViewById<TextView>(R.id.*btn\_minus*).setOnClickListener **{** setTextFields("-") **}** findViewById<TextView>(R.id.*btn\_t*).setOnClickListener **{** setTextFields(".") **}** findViewById<TextView>(R.id.*btn\_C*).setOnClickListener **{** val str = findViewById<TextView>(R.id.*math\_operation*).*text*.toString()  if (str.*isNotEmpty*()) {  findViewById<TextView>(R.id.*math\_operation*).*text* = str.*substring*(0, str.length - 1)   }  **}** findViewById<TextView>(R.id.*btn\_ravno*).setOnClickListener **{** val str = findViewById<TextView>(R.id.*math\_operation*).*text*.toString()  try {  val result = ExpressionBuilder(str).build().evaluate()  findViewById<TextView>(R.id.*math\_operation*).*text* = result.toString()  } catch (e: Exception) {  findViewById<TextView>(R.id.*math\_operation*).*text* = "Ошибка"  }  **}** }  fun setTextFields(str: String) {  findViewById<TextView>(R.id.*math\_operation*).append(str)  } } |

Вывод:

Были освоены возможности создания приложения с различными типами

интерфейсных элементов для мобильного устройства