Министерство образования Республики Беларусь

Учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет компьютерного проектирования

Дисциплина: Компьютерные сети

**ЛАБОРАТОРНАЯ РАБОТА № 1**

Android - калькулятор

Выполнил:

студент гр. 714301

Гельдымурадов С.

Проверил:

Минск 2020

Используется язык Kotlin. Для отображения истории испольуется RecyclerView.

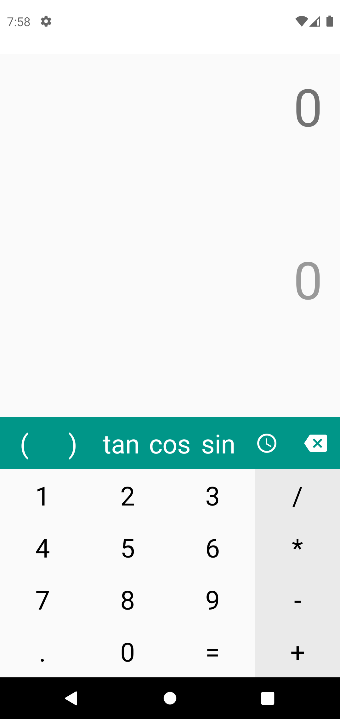


рисунок 1. Портретный режим.

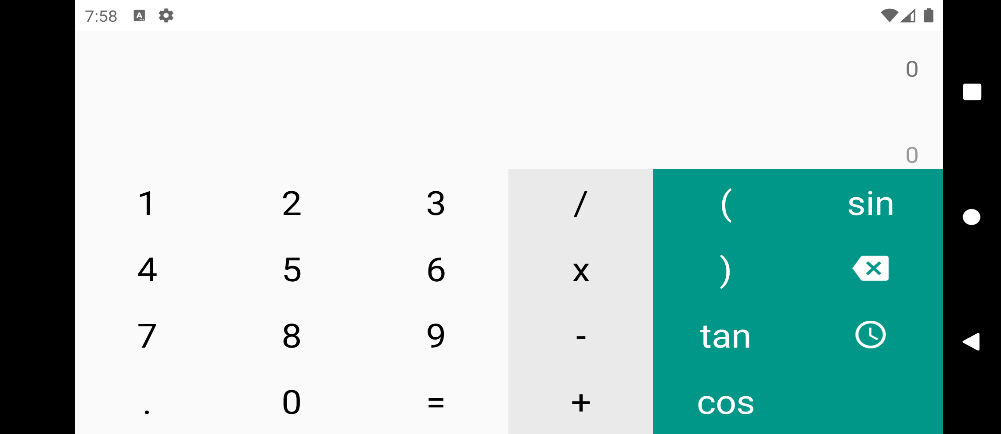


рисунок 2. альбомный режим.

Решения для выражения отображается и обновляется сразу после ввода. При неверном выражении отображается NaN.

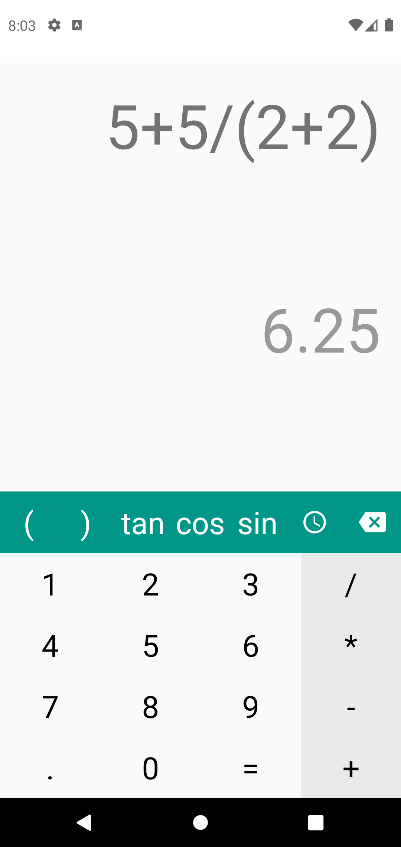


рисунок 3. Выражение.

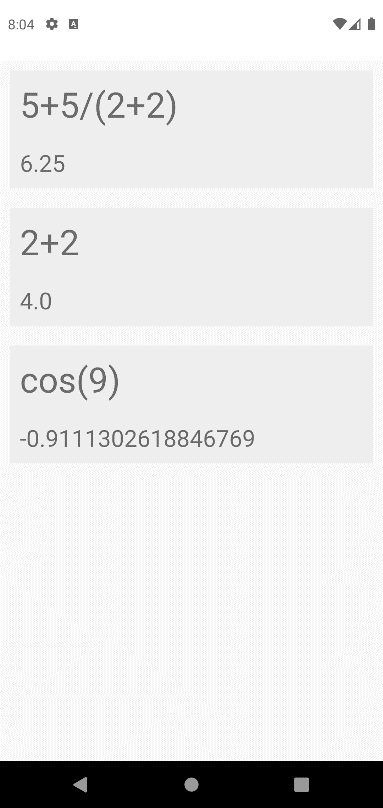


рисунок 4. История введенных выражений.

**Листинг кода.**

**https://github.com/bsuir-temp/ol1**

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

if (savedInstanceState == null) {

supportFragmentManager

.beginTransaction()

.add(R.id.content, Main.newInstance(), "dogList")

.commit()

}

}

}

data class Expression(val expression: String,val result: String)

class CalcVH {

val engine: ScriptEngine = ScriptEngineManager()?.getEngineByName("rhino")

var number1: TextView? = null

var number2: TextView? = null

var number3: TextView? = null

var number4: TextView? = null

var number5: TextView? = null

var number6: TextView? = null

var number7: TextView? = null

var number8: TextView? = null

var number9: TextView? = null

var number0: TextView? = null

var operationCClose: TextView? = null

var operationCOpen: TextView? = null

var operationPlus: TextView? = null

var operationMinus: TextView? = null

var operationMultiple: TextView? = null

var operationDevide: TextView? = null

var operationExec: TextView? = null

var operationDelete: ImageView? = null

var operationHistory: ImageView? = null

var operationTan: TextView? = null

var operationCos: TextView? = null

var operationSin: TextView? = null

var calcScreen: TextView? = null

var calcResultScreen: TextView? = null

fun map(root: View, fm: FragmentManager) {

number0 = root.findViewById(R.id.button0)

number1 = root.findViewById(R.id.button1)

number2 = root.findViewById(R.id.button2)

number3 = root.findViewById(R.id.button3)

number4 = root.findViewById(R.id.button4)

number5 = root.findViewById(R.id.button5)

number6 = root.findViewById(R.id.button6)

number7 = root.findViewById(R.id.button7)

number8 = root.findViewById(R.id.button8)

number9 = root.findViewById(R.id.button9)

operationTan = root.findViewById(R.id.buttonTA)

operationCos = root.findViewById(R.id.buttonCO)

operationSin = root.findViewById(R.id.buttonSI)

operationCClose = root.findViewById(R.id.buttonCS)

operationCOpen = root.findViewById(R.id.buttonOS)

calcScreen = root.findViewById(R.id.screen)

calcResultScreen = root.findViewById(R.id.screenResult)

operationDelete = root.findViewById(R.id.buttonC)

operationHistory = root.findViewById(R.id.buttonHI)

operationDevide = root.findViewById(R.id.buttonD)

operationExec = root.findViewById(R.id.buttonR)

operationMinus = root.findViewById(R.id.buttonM)

operationMultiple = root.findViewById(R.id.buttonX)

operationPlus = root.findViewById(R.id.buttonP)

val listener = View.OnClickListener { v -> if (v is TextView)

settext("" + calcScreen?.getText() + v.text) }

number1?.setOnClickListener(listener)

number2?.setOnClickListener(listener)

number3?.setOnClickListener(listener)

number4?.setOnClickListener(listener)

number5?.setOnClickListener(listener)

number6?.setOnClickListener(listener)

number7?.setOnClickListener(listener)

number8?.setOnClickListener(listener)

number9?.setOnClickListener(listener)

number0?.setOnClickListener(listener)

operationDelete?.setOnClickListener(View.OnClickListener {

settext(calcScreen?.getText().toString().substring(0, calcScreen?.text!!.length - 1))

})

operationDelete?.setOnLongClickListener(OnLongClickListener {

settext(null)

true

})

val operation = View.OnClickListener {

it as TextView

settext(calcScreen?.getText().toString() + it.getText())

}

val mathOperation = View.OnClickListener {

it as TextView

settext(calcScreen?.getText().toString() + it.getText() + "(")

}

operationTan?.setOnClickListener(mathOperation)

operationCos?.setOnClickListener(mathOperation)

operationSin?.setOnClickListener(mathOperation)

operationDevide?.setOnClickListener(operation)

operationPlus?.setOnClickListener(operation)

operationMinus?.setOnClickListener(operation)

operationMultiple?.setOnClickListener(operation)

operationCClose?.setOnClickListener(operation)

operationCOpen?.setOnClickListener(operation)

operationHistory?.setOnClickListener {

fm.beginTransaction()

.replace(R.id.content,History(),"History")

.addToBackStack(null).commit()

}

operationExec?.setOnClickListener {

History.list.add(Expression(calcScreen?.text.toString(),calcResultScreen?.text.toString()))

}

}

private fun settext(intext: String?) {

var text = intext

if (text != null) while (text!!.startsWith("0")) text = text.substring(1)

if (text == null || text.isEmpty()) text = 0.toString()

calcScreen!!.text = text

text = text.toString().replace("cos","Math.cos")

text = text.toString().replace("tan","Math.tan")

text = text.toString().replace("sin","Math.sin")

try {

calcResultScreen?.text = engine.eval(text).toString()

}catch (th:Throwable){calcResultScreen?.text = "NaN"}

}

}

class History : Fragment() {

companion object {

var list:ArrayList<Expression> = ArrayList()

fun newInstance(): History {

return History()

}

}

lateinit var recyclerView: RecyclerView

override fun onCreateView(inflater: LayoutInflater,

container: ViewGroup?,

savedInstanceState: Bundle?): View? {

val view = inflater.inflate(R.layout.fragment\_history, container, false)

val viewManager = LinearLayoutManager(context)

recyclerView = view.findViewById<RecyclerView>(R.id.list) .apply {

setHasFixedSize(true)

layoutManager = viewManager

adapter = HistoryAdapter(list)

}

return view

}

}

class Main : Fragment() {

companion object {

fun newInstance(): Main {

return Main()

}

}

override fun onCreateView(inflater: LayoutInflater,

container: ViewGroup?,

savedInstanceState: Bundle?): View? {

val view: View = inflater.inflate(R.layout.fragment\_main, container, false)

val holder = CalcVH()

holder.map(view,fragmentManager!!)

return view

}

}

class HistoryAdapter(private val list: ArrayList<Expression>) :

RecyclerView.Adapter<HistoryAdapter.ExpressionVH>() {

class ExpressionVH(root: View) : RecyclerView.ViewHolder(root)

{

val expression = root.findViewById<TextView>(R.id.expression)

val result = root.findViewById<TextView>(R.id.result)

}

override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): HistoryAdapter.ExpressionVH {

val root = LayoutInflater.from(parent.context).inflate(R.layout.fragment\_history\_item, parent, false)

return ExpressionVH(root)

}

override fun onBindViewHolder(holder: ExpressionVH, position: Int) {

holder.expression.text = list[position].expression

holder.result.text = list[position].result

}

override fun getItemCount() = list.size

}