Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 5

з курсу: “Розробка програмного забезпечення мобільних пристроїв”

**Виконав:**  
студент 4-го курсу,  
групи ТВ-11

Гудзовський Марк Юрійович

Посилання на GitHub репозиторій:

<https://github.com/Chivas1717/kpi-kotlin-lab-5>

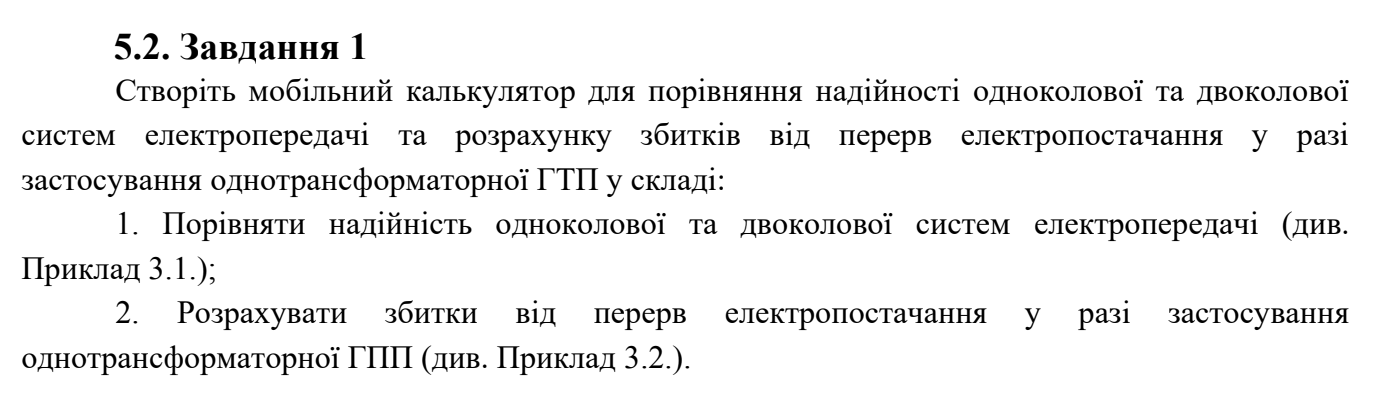
**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

Лабораторна робота № 5

**Завдання:**



**Хід виконання:**

Перед тим, як обчислювати значення функії/шукати розв'язки рівняння, потрібно ініцілізувати змінні та зчитати їх з інпутів в додатку:

MainActivity.kt

package com.example.kpikotlinlab5

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.\*

import androidx.compose.runtime.\*

import androidx.compose.ui.Modifier

import androidx.compose.ui.tooling.preview.Preview

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

*setContent* **{**

MainScreen()

**}**

}

}

@OptIn(ExperimentalMaterial3Api::class)

@Composable

fun MainScreen() {

// стейт для індексу обраного табу

var selectedTabIndex by remember **{** *mutableStateOf*(0) **}**

// визначаємо таби

Scaffold(

topBar = **{**

TabRow(

selectedTabIndex = selectedTabIndex,

modifier = Modifier.*fillMaxWidth*()

) **{**

// Define tabs

Tab(

selected = selectedTabIndex == 0,

onClick = **{** selectedTabIndex = 0 **}**,

text = **{** Text("Калькулятор 1") **}**

)

Tab(

selected = selectedTabIndex == 1,

onClick = **{** selectedTabIndex = 1 **}**,

text = **{** Text("Калькулятор 2") **}**

)

**}**

**}**

) **{** innerPadding **->**

// показуємо composable в залежності від обраного табу

when (selectedTabIndex) {

0 -> CalculatorOne(modifier = Modifier.*padding*(innerPadding))

1 -> CalculatorTwo(modifier = Modifier.*padding*(innerPadding))

}

**}**

}

@Preview(showBackground = true)

@Composable

fun MainScreenAppPreview() {

MainScreen()

}

В цьому файлі виконуються навігація та ініціалізація усіх калькуляторів.

**Завдання 1**

CalculatorOne.kt

package com.example.kpikotlinlab5

import androidx.compose.foundation.clickable

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.Spacer

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.height

import androidx.compose.foundation.layout.padding

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.verticalScroll

import androidx.compose.material3.OutlinedTextField

import androidx.compose.material3.Button

import androidx.compose.material3.DropdownMenu

import androidx.compose.material3.DropdownMenuItem

import androidx.compose.material3.ExperimentalMaterial3Api

import androidx.compose.material3.MaterialTheme

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.setValue

import androidx.compose.ui.platform.*LocalFocusManager*

import androidx.compose.ui.unit.dp

import kotlin.math.sqrt

import androidx.compose.ui.Modifier

data class EquipmentReliability(

val failureRate: Double,

val averageRepairTime: Int,

val frequency: Double,

val averageRecoveryTime: Int?

)

val *data* = *mapOf*(

"T-110 kV" *to* EquipmentReliability(0.015, 100, 1.0, 43),

"T-35 kV" *to* EquipmentReliability(0.02, 80, 1.0, 28),

"T-10 kV (кабельна мережа 10 кВ)" *to* EquipmentReliability(0.005, 60, 0.5, 10),

"T-10 kV (повітряна мережа 10 кВ)" *to* EquipmentReliability(0.05, 60, 0.5, 10),

"B-110 kV (елегазовий)" *to* EquipmentReliability(0.01, 30, 0.1, 30),

"B-10 kV (малолойний)" *to* EquipmentReliability(0.02, 15, 0.33, 15),

"B-10 kV (вакуумний)" *to* EquipmentReliability(0.05, 15, 0.33, 15),

"Збірні шини 10 кВ на 1 приєднання" *to* EquipmentReliability(0.03, 2, 0.33, 15),

"АВ-0,38 кВ" *to* EquipmentReliability(0.05, 20, 1.0, 15),

"ЕД 6,10 кВ" *to* EquipmentReliability(0.1, 50, 0.5, 0),

"ЕД 0,38 кВ" *to* EquipmentReliability(0.1, 50, 0.5, 0),

"ПЛ-110 кВ" *to* EquipmentReliability(0.007, 10, 0.167, 35),

"ПЛ-35 кВ" *to* EquipmentReliability(0.02, 8, 0.167, 35),

"ПЛ-10 кВ" *to* EquipmentReliability(0.02, 10, 0.167, 35),

"КЛ-10 кВ (траншея)" *to* EquipmentReliability(0.03, 44, 1.0, 9),

"КЛ-10 кВ (кабельний канал)" *to* EquipmentReliability(0.005, 18, 1.0, 9)

)

@OptIn(ExperimentalMaterial3Api::class)

@Composable

fun CalculatorOne(modifier: Modifier = Modifier) {

val focusManager = *LocalFocusManager*.current

val amountMap = *mapOf*(

"T-110 kV" *to* remember **{** *mutableStateOf*("1") **}**,

"T-35 kV" *to* remember **{** *mutableStateOf*("0") **}**,

"T-10 kV (кабельна мережа 10 кВ)" *to* remember **{** *mutableStateOf*("0") **}**,

"T-10 kV (повітряна мережа 10 кВ)" *to* remember **{** *mutableStateOf*("0") **}**,

"B-110 kV (елегазовий)" *to* remember **{** *mutableStateOf*("1") **}**,

"B-10 kV (малолойний)" *to* remember **{** *mutableStateOf*("1") **}**,

"B-10 kV (вакуумний)" *to* remember **{** *mutableStateOf*("0") **}**,

"Збірні шини 10 кВ на 1 приєднання" *to* remember **{** *mutableStateOf*("6") **}**,

"АВ-0,38 кВ" *to* remember **{** *mutableStateOf*("0") **}**,

"ЕД 6,10 кВ" *to* remember **{** *mutableStateOf*("0") **}**,

"ЕД 0,38 кВ" *to* remember **{** *mutableStateOf*("0") **}**,

"ПЛ-110 кВ" *to* remember **{** *mutableStateOf*("10") **}**,

"ПЛ-35 кВ" *to* remember **{** *mutableStateOf*("0") **}**,

"ПЛ-10 кВ" *to* remember **{** *mutableStateOf*("0") **}**,

"КЛ-10 кВ (траншея)" *to* remember **{** *mutableStateOf*("0") **}**,

"КЛ-10 кВ (кабельний канал)" *to* remember **{** *mutableStateOf*("0") **}**

)

var Woc by remember **{** *mutableStateOf*("") **}**

var Tvoc by remember **{** *mutableStateOf*("") **}**

var Kaoc by remember **{** *mutableStateOf*("") **}**

var Kpoc by remember **{** *mutableStateOf*("") **}**

var Wdk by remember **{** *mutableStateOf*("") **}**

var Wds by remember **{** *mutableStateOf*("") **}**

// розмітка застосунку

Column(

modifier = Modifier

.*fillMaxSize*()

.*padding*(16.*dp*)

.*verticalScroll*(rememberScrollState())

) **{**

Spacer(modifier = Modifier.*height*(56.*dp*))

OutlinedTextField(

value = amountMap["T-110 kV"]?.value ?: "",

onValueChange = **{** amountMap["T-110 kV"]?.value = **it }**,

label = **{** Text("T-110 kV") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["T-35 kV"]?.value ?: "",

onValueChange = **{** amountMap["T-35 kV"]?.value = **it }**,

label = **{** Text("T-35 kV") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["T-10 kV (кабельна мережа 10 кВ)"]?.value ?: "",

onValueChange = **{** amountMap["T-10 kV (кабельна мережа 10 кВ)"]?.value = **it }**,

label = **{** Text("T-10 kV (кабельна мережа 10 кВ)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["T-10 kV (повітряна мережа 10 кВ)"]?.value ?: "",

onValueChange = **{** amountMap["T-10 kV (повітряна мережа 10 кВ)"]?.value = **it }**,

label = **{** Text("T-10 kV (повітряна мережа 10 кВ)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["B-110 kV (елегазовий)"]?.value ?: "",

onValueChange = **{** amountMap["B-110 kV (елегазовий)"]?.value = **it }**,

label = **{** Text("B-110 kV (елегазовий)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["B-10 kV (малолойний)"]?.value ?: "",

onValueChange = **{** amountMap["B-10 kV (малолойний)"]?.value = **it }**,

label = **{** Text("B-10 kV (малолойний)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["B-10 kV (вакуумний)"]?.value ?: "",

onValueChange = **{** amountMap["B-10 kV (вакуумний)"]?.value = **it }**,

label = **{** Text("B-10 kV (вакуумний)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["Збірні шини 10 кВ на 1 приєднання"]?.value ?: "",

onValueChange = **{** amountMap["Збірні шини 10 кВ на 1 приєднання"]?.value = **it }**,

label = **{** Text("Збірні шини 10 кВ на 1 приєднання") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["АВ-0,38 кВ"]?.value ?: "",

onValueChange = **{** amountMap["АВ-0,38 кВ"]?.value = **it }**,

label = **{** Text("АВ-0,38 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["ЕД 6,10 кВ"]?.value ?: "",

onValueChange = **{** amountMap["ЕД 6,10 кВ"]?.value = **it }**,

label = **{** Text("ЕД 6,10 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["ЕД 0,38 кВ"]?.value ?: "",

onValueChange = **{** amountMap["ЕД 0,38 кВ"]?.value = **it }**,

label = **{** Text("ЕД 0,38 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["ПЛ-110 кВ"]?.value ?: "",

onValueChange = **{** amountMap["ПЛ-110 кВ"]?.value = **it }**,

label = **{** Text("ПЛ-110 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["ПЛ-35 кВ"]?.value ?: "",

onValueChange = **{** amountMap["ПЛ-35 кВ"]?.value = **it }**,

label = **{** Text("ПЛ-35 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["ПЛ-10 кВ"]?.value ?: "",

onValueChange = **{** amountMap["ПЛ-10 кВ"]?.value = **it }**,

label = **{** Text("ПЛ-10 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["КЛ-10 кВ (траншея)"]?.value ?: "",

onValueChange = **{** amountMap["КЛ-10 кВ (траншея)"]?.value = **it }**,

label = **{** Text("КЛ-10 кВ (траншея)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(8.*dp*))

OutlinedTextField(

value = amountMap["КЛ-10 кВ (кабельний канал)"]?.value ?: "",

onValueChange = **{** amountMap["КЛ-10 кВ (кабельний канал)"]?.value = **it }**,

label = **{** Text("КЛ-10 кВ (кабельний канал)") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

// кнопка для виконання обчислень

Button(

onClick = **{**

var w\_0\_c = 0.0

var tvoc = 0.0

var kaoc = 0.0

var kpos = 0.0

var wdk = 0.0

var wds = 0.0

amountMap.*forEach* **{** (key, value) **->**

val amount = value.value.*toIntOrNull*() ?: 0

if (amount > 0) {

w\_0\_c += amount \* *data*[key]?.failureRate!!

*println*(key)

*println*(amount \* *data*[key]?.failureRate!!)

*println*(*data*[key]?.averageRepairTime!!)

tvoc += amount \* *data*[key]?.averageRepairTime!! \* *data*[key]?.failureRate!!

}

**}**

tvoc = tvoc / w\_0\_c

kaoc = (tvoc \* w\_0\_c) / 8760

*println*(kaoc)

kpos = 1.2 \* 43 / 8760

wdk = 2 \* w\_0\_c \* (kaoc + kpos)

wds = wdk + 0.02

Woc = *roundToTwoDecimalString*(w\_0\_c)

Tvoc = *roundToTwoDecimalString*(tvoc)

Kaoc = *roundToTwoDecimalString*(kaoc)

Kpoc = *roundToTwoDecimalString*(kpos)

Wdk = *roundToTwoDecimalString*(wdk)

Wds = *roundToTwoDecimalString*(wds)

**}**,

modifier = Modifier.*fillMaxWidth*()

) **{**

Text("Обчислити")

**}**

// вивід результатів

Text(text = "Частота відмов одноколової системи: " + Woc,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Середня тривалість відновлення: " + Tvoc,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Коефікієнт аварійного простою одноколової системи: " + Kaoc,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Коефікієнт планового простою одноколової системи: " + Kpoc,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Частота відмов одночасно двох кіл: " + Wdk,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Частота відмов двоколової системи з урахуванням секційного вимикача: " + Wds,

style = MaterialTheme.typography.bodyLarge)

**}**

}

fun roundToTwoDecimalString(value: Double): String {

// return (Math.round(value \* 1000) / 1000.0).toString()

return value.toString()

}

**Завдання 2**

CalculatorTwo.kt

package com.example.kpikotlinlab5

import androidx.compose.foundation.clickable

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.Spacer

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.height

import androidx.compose.foundation.layout.padding

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.verticalScroll

import androidx.compose.material3.Button

import androidx.compose.material3.ExperimentalMaterial3Api

import androidx.compose.material3.MaterialTheme

import androidx.compose.material3.OutlinedTextField

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.setValue

import androidx.compose.ui.platform.*LocalFocusManager*

import androidx.compose.ui.unit.dp

import androidx.compose.ui.Modifier

@OptIn(ExperimentalMaterial3Api::class)

@Composable

fun CalculatorTwo(modifier: Modifier = Modifier) {

val focusManager = *LocalFocusManager*.current

var omega by remember **{** *mutableStateOf*("0.01") **}**

var tS by remember **{** *mutableStateOf*("0.045") **}**

var pM by remember **{** *mutableStateOf*("5120") **}**

var tM by remember **{** *mutableStateOf*("6451") **}**

var zAvar by remember **{** *mutableStateOf*("23.6") **}**

var zPlan by remember **{** *mutableStateOf*("17.6") **}**

var kP by remember **{** *mutableStateOf*("0.004") **}**

var mWnedA by remember **{** *mutableStateOf*("") **}**

var mWnedP by remember **{** *mutableStateOf*("") **}**

var mZ by remember **{** *mutableStateOf*("") **}**

// розмітка застосунку

Column(

modifier = Modifier

.*fillMaxSize*()

.*padding*(16.*dp*)

.*verticalScroll*(rememberScrollState())

) **{**

Spacer(modifier = Modifier.*height*(56.*dp*))

OutlinedTextField(

value = omega,

onValueChange = **{** omega = **it }**,

label = **{** Text("частота відмов") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(16.*dp*))

OutlinedTextField(

value = tS,

onValueChange = **{** tS = **it }**,

label = **{** Text("середній час відновлення трансформатора напругою 35 кВ") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(16.*dp*))

OutlinedTextField(

value = pM,

onValueChange = **{** pM = **it }**,

label = **{** Text("потужність") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(16.*dp*))

OutlinedTextField(

value = tM,

onValueChange = **{** tM = **it }**,

label = **{** Text("очікуваний час простою") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(16.*dp*))

OutlinedTextField(

value = zAvar,

onValueChange = **{** zAvar = **it }**,

label = **{** Text("збитки у разі аварійного переривання") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(16.*dp*))

OutlinedTextField(

value = zPlan,

onValueChange = **{** zPlan = **it }**,

label = **{** Text("збитки у разі запланованого переривання") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(16.*dp*))

OutlinedTextField(

value = kP,

onValueChange = **{** kP = **it }**,

label = **{** Text("сер час планового простою") **}**,

modifier = Modifier

.*fillMaxWidth*()

.*clickable* **{** focusManager.clearFocus() **}**

)

Spacer(modifier = Modifier.*height*(24.*dp*))

// кнопка для виконання обчислень

Button(

onClick = **{**

val omegaValue = omega.*toDoubleOrNull*() ?: 0.0

val tSValue = tS.*toDoubleOrNull*() ?: 0.0

val pMValue = pM.*toDoubleOrNull*() ?: 0.0

val tMValue = tM.*toDoubleOrNull*() ?: 0.0

val zAvarValue = zAvar.*toDoubleOrNull*() ?: 0.0

val zPlanValue = zPlan.*toDoubleOrNull*() ?: 0.0

val kPValue = kP.*toDoubleOrNull*() ?: 0.0

val mWnedAValue = omegaValue \* tSValue \* pMValue \* tMValue

val mWnedPValue = kPValue \* pMValue \* tMValue

val mZValue = zAvarValue \* mWnedAValue + zPlanValue \* mWnedPValue

mWnedA = mWnedAValue.toString()

mWnedP = mWnedPValue.toString()

mZ = mZValue.toString()

**}**,

modifier = Modifier.*fillMaxWidth*()

) **{**

Text("Обчислити")

**}**

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Очікувана відсутність енергопостачання в надзвичайних ситуаціях: " + mWnedA,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Очікуваний дефіцит енергії для запланованих: " + mWnedP,

style = MaterialTheme.typography.bodyLarge)

Spacer(modifier = Modifier.*height*(8.*dp*))

Text(text = "Загальна очікувана вартість перерв у роботі: " + mZ,

style = MaterialTheme.typography.bodyLarge)

**}**

}

А також додамо стилів:

styles.xml:  
<resources>

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

<item name="colorPrimary">#6200EE</item>

<item name="colorPrimaryDark">#3700B3</item>

<item name="colorAccent">#03DAC5</item>

</style>

</resources>

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

<resources>

<style name="Theme.Kpikotlinlab5" parent="Theme.AppCompat.Light.NoActionBar">

<!-- Основні кольори теми -->

<item name="colorPrimary">#6200EE</item>

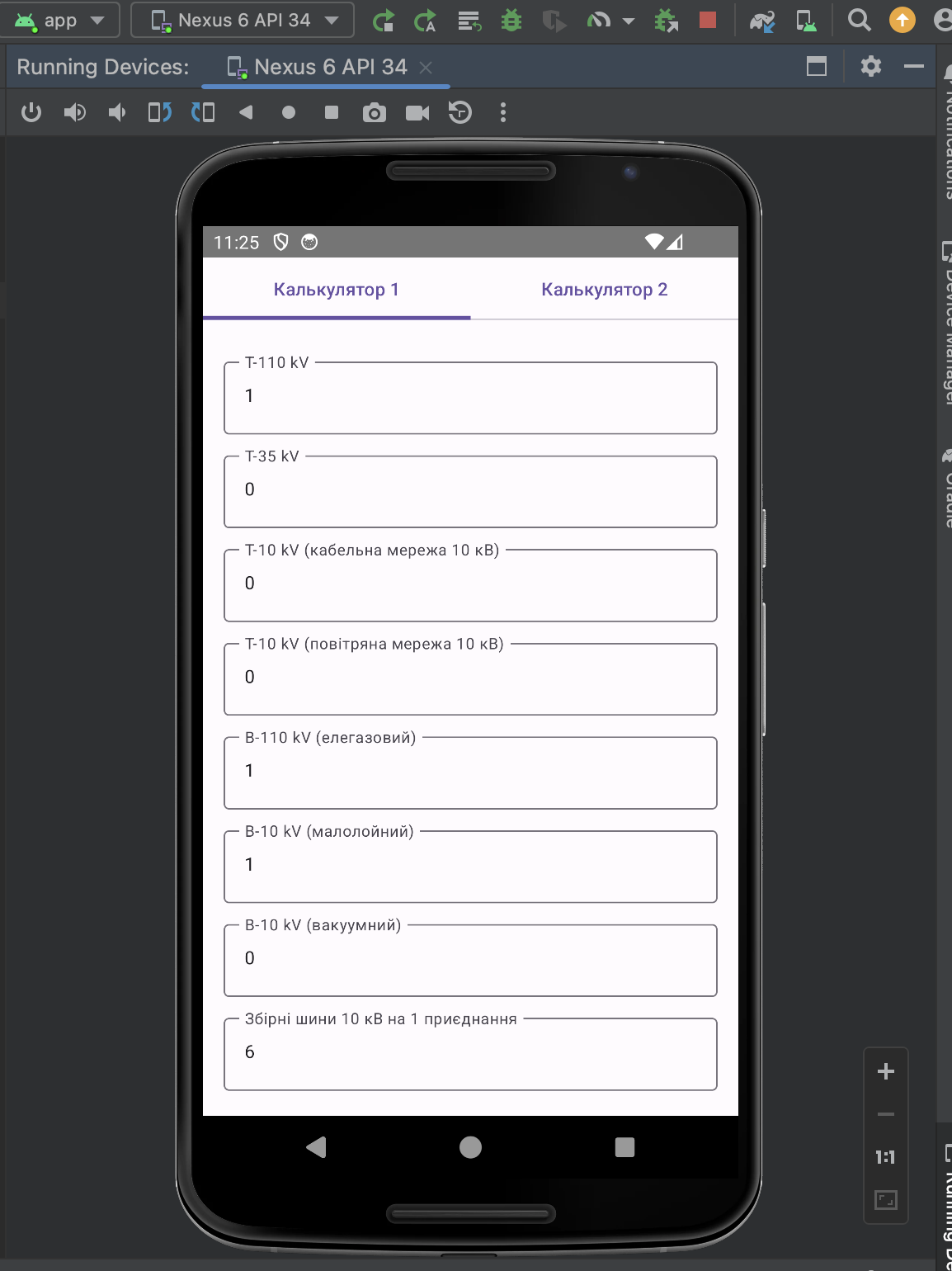
<item name="colorPrimaryDark">#3700B3</item>

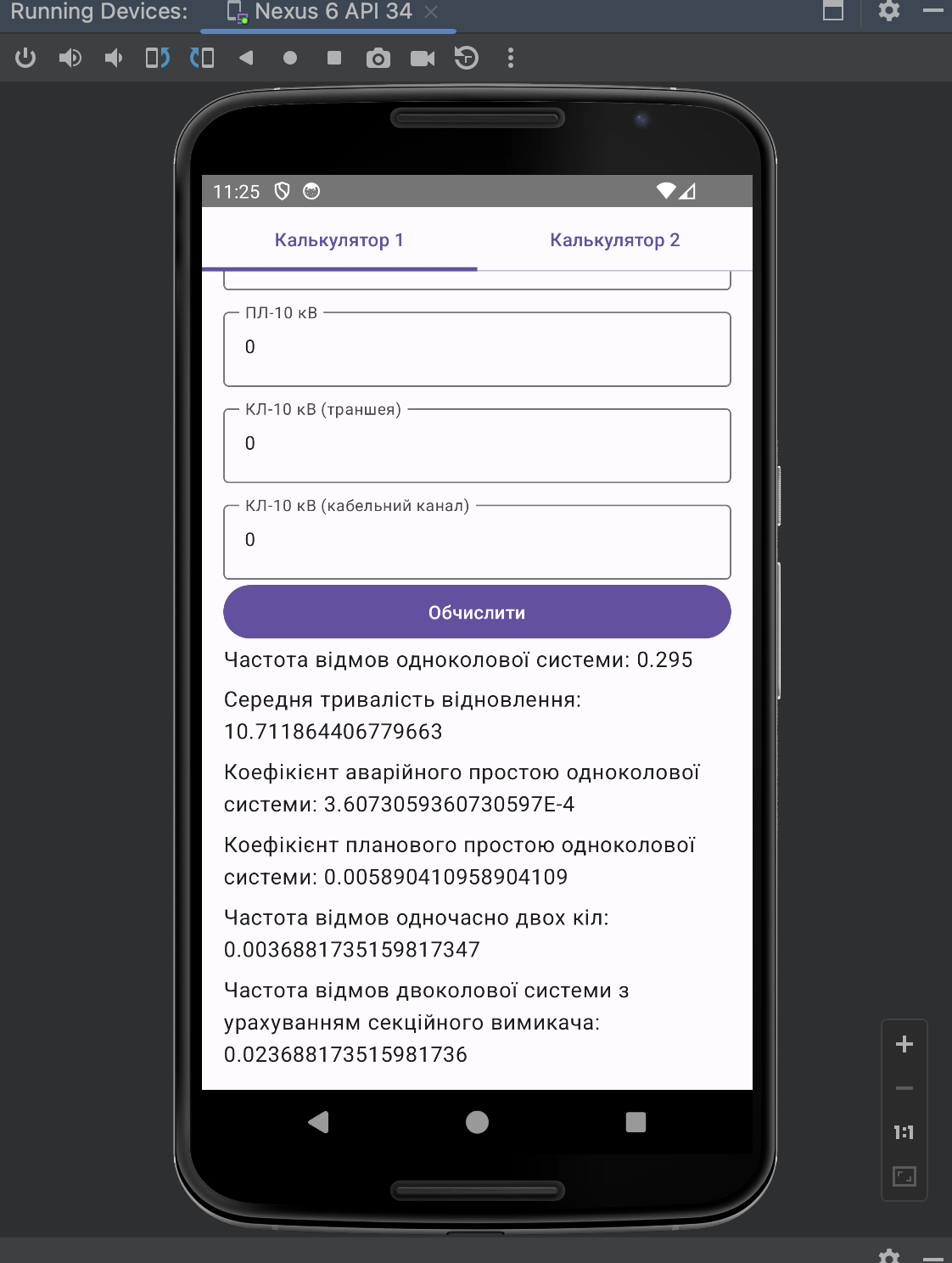
<item name="colorAccent">#03DAC5</item>

</style>

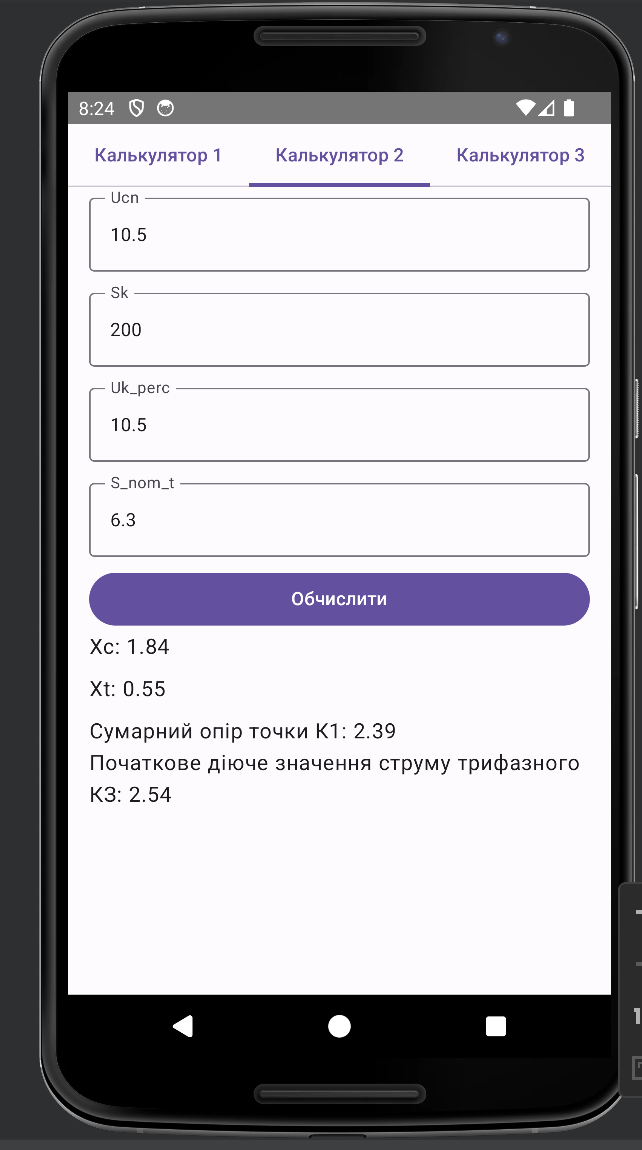
</resources>

Результат виконання (**Завдання 1**):

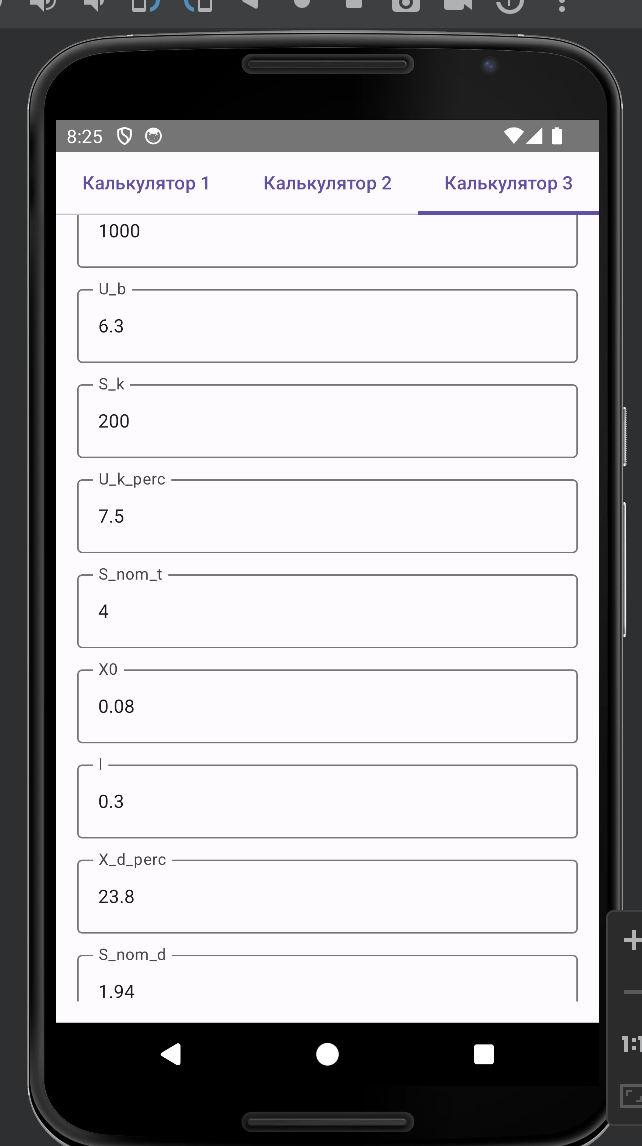


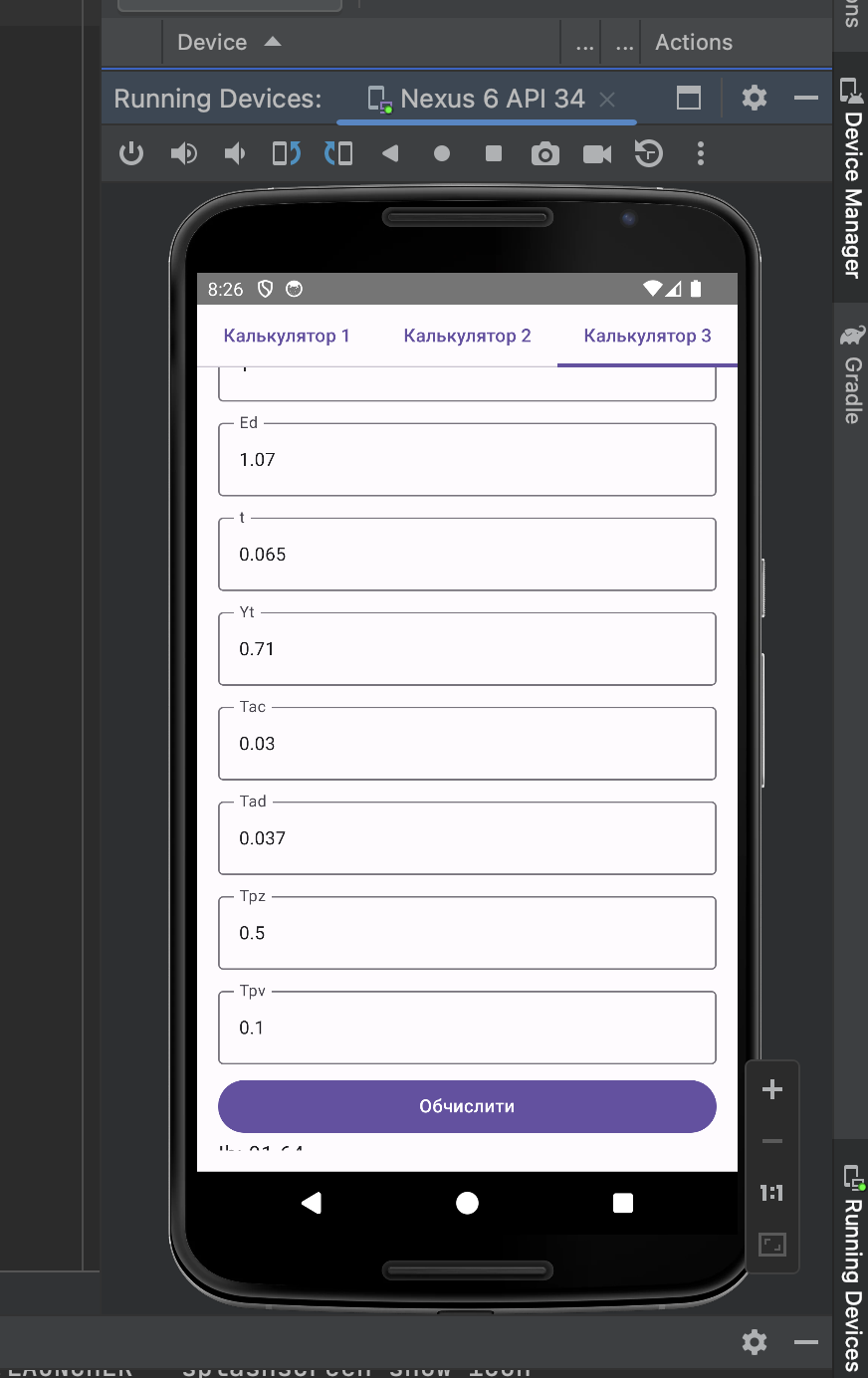


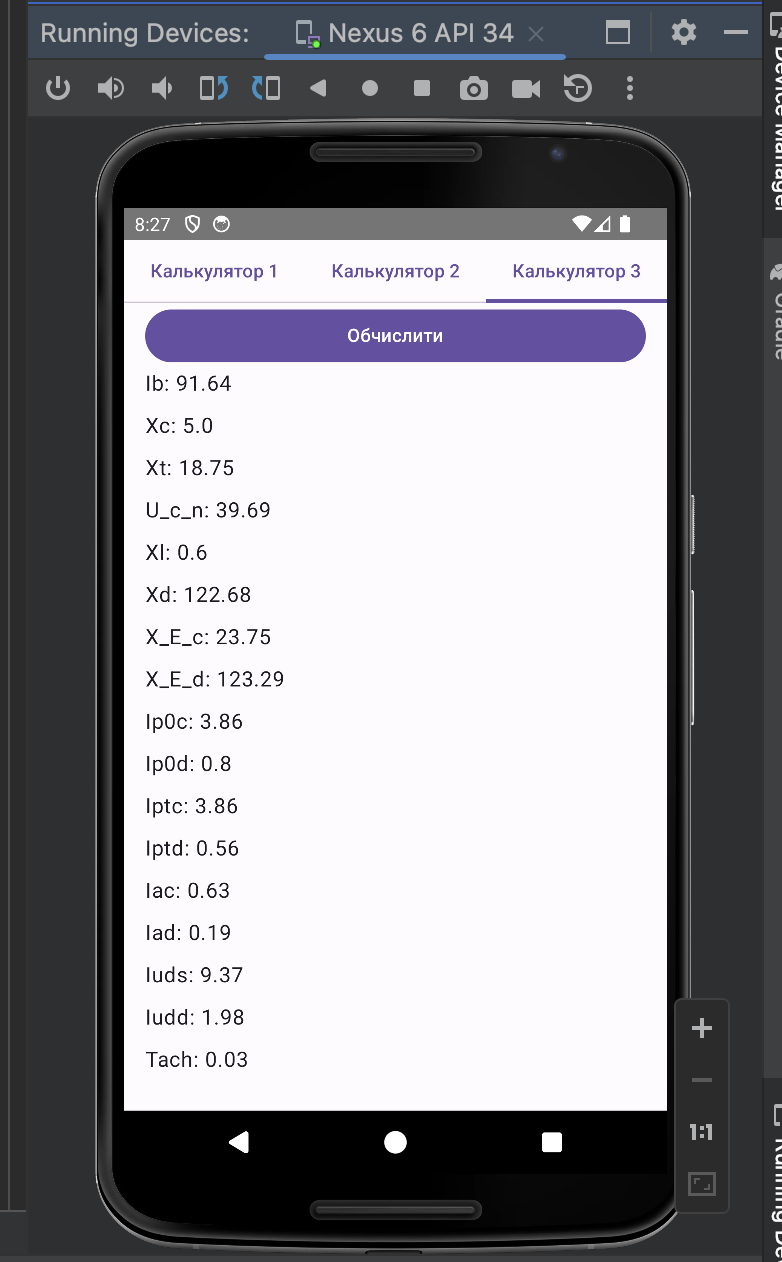
Результат виконання (**Завдання 2**):



Результат виконання (**Завдання 3**):







**Висновок**

У процесі виконання лабораторної роботи було розроблено мобільний додаток для платформи Android, який дозволяє виконувати розрахунки струмів трифазного та однофазного короткого замикання, а також перевіряти кабелі на термічну та динамічну стійкість. Основною метою було підібрати відповідні кабелі для двотрансформаторної підстанції, розрахувати струми короткого замикання на шинах 10 кВ головної понижувальної підстанції (ГПП) та для підстанції Хмельницьких північних електричних мереж (ХПнЕМ) у різних режимах роботи: нормальному, мінімальному та аварійному.

Для створення інтерфейсу користувача було використано мову програмування Kotlin із застосуванням таких компонентів, як DropdownMenu, таби для поділу функціоналу калькуляторів, а також структур listOf та mapOf для організації даних. Для виконання обчислень використовувалася бібліотека math, а також інші елементи Jetpack Compose. Виконані розрахунки підтвердили правильність використаних формул та методик, відповідно до прикладів, наведених у завданні.

Ця лабораторна робота дозволила отримати практичні навички у створенні Android-додатків для електротехнічних розрахунків, а також у використанні Kotlin для розробки інтерактивного інтерфейсу та реалізації математичних обчислень.