Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 3

Варіант 11(1)

з курсу: «Розробка програмного забезпечення мобільних пристроїв»

**Виконала:**  
студентка 4-го курсу,  
групи ТВ-11

Гундяк Валерія Русланівна

Посилання на GitHub репозиторій:

<https://github.com/pojuipolynu/PW_TV-11_Hundiak_Valeriia_Ruslanivna.git>

**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

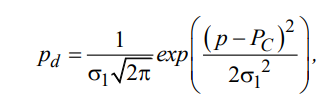
Лабораторна робота № 2

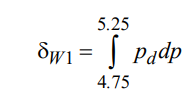
**Завдання:**

Створіть мобільний калькулятор розрахунку прибутку від сонячних електростанцій з встановленою системою прогнозування сонячної потужності:

Сонячна електростанція, під’єднана до ОЕС України, генерує енергію з середньодобовою потужністю РС = 5 МВт. На станції встановлена система прогнозування сонячної потужності з середньоквадратичним відхиленням σ1 = 1 98 МВт. Порахувати який прибуток принесе вдосконалення системи прогнозу, що дозволить зменшити похибку до σ2 = 0.25 МВт. Вартість електроенергії складає В = 7 грн / кВт‧год. Закон розподілу похибки прогнозу вважати нормальним.

**Хід виконання:**

Основні формули для роботи калькулятора були задані у практичному матеріалі  












**Код програми:**

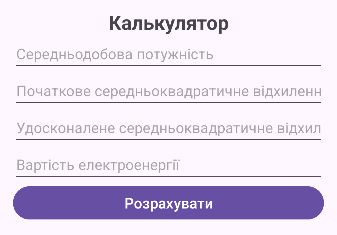
Основний код програми має вигляд:

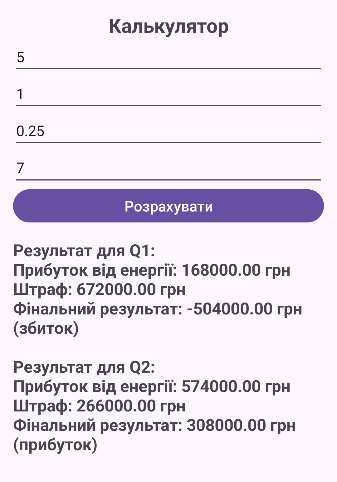
package com.example.myapplication  
  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
import kotlin.math.round  
import kotlin.math.*PI*import kotlin.math.exp  
import kotlin.math.sqrt  
import kotlin.math.pow  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var result: TextView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val calculateButton = findViewById<Button>(R.id.*calculateButton*)  
 val P = findViewById<EditText>(R.id.*P*)  
 val Q1 = findViewById<EditText>(R.id.*Q1*)  
 val Q2 = findViewById<EditText>(R.id.*Q2*)  
 val B = findViewById<EditText>(R.id.*B*)  
 result = findViewById(R.id.*resultTextView*)  
  
 calculateButton.setOnClickListener **{** val pc = P.*text*.toString().*toDoubleOrNull*()  
 val q1 = Q1.*text*.toString().*toDoubleOrNull*()  
 val q2 = Q2.*text*.toString().*toDoubleOrNull*()  
 val b = B.*text*.toString().*toDoubleOrNull*()  
  
 if (pc != null && q1 != null && q2 != null && b != null) {  
 calculateResults(pc, b, q1, q2)  
 } else {  
 result.*text* = "Введіть валідні значення."  
 }  
 **}** }  
  
 private fun calculateEnergyShare(Pc: Double, q: Double): Double {  
 val delta = Pc \* 0.05  
 val lowerBound = Pc - delta  
 val upperBound = Pc + delta  
 val step = 0.001  
  
 var integral = 0.0  
 for (p in *generateSequence*(lowerBound) **{ it** + step **}**.*takeWhile* **{ it** < upperBound **}**) {  
 val pd = (1 / (q \* *sqrt*(2 \* *PI*))) \* *exp*(-((p - Pc).*pow*(2)) / (2 \* q.*pow*(2)))  
 integral += pd \* step  
 }  
 return integral  
 }  
  
 private fun calculateResults(averagePower: Double, cost: Double, q1: Double, q2: Double) {  
 val q1Result = calculateProfitAndPenalty(averagePower, cost, q1)  
 val q2Result = calculateProfitAndPenalty(averagePower, cost, q2)  
  
 result.*text* = """  
 Результат для Q1:  
 Прибуток від енергії: ${"%.2f".*format*(q1Result.first)} грн  
 Штраф: ${"%.2f".*format*(q1Result.second)} грн  
 Фінальний результат: ${"%.2f".*format*(q1Result.third)} грн (${if (q1Result.third >= 0) "прибуток" else "збиток"})  
   
 Результат для Q2:  
 Прибуток від енергії: ${"%.2f".*format*(q2Result.first)} грн  
 Штраф: ${"%.2f".*format*(q2Result.second)} грн  
 Фінальний результат: ${"%.2f".*format*(q2Result.third)} грн (${if (q2Result.third >= 0) "прибуток" else "збиток"})  
 """.*trimIndent*()  
 }  
  
 private fun calculateProfitAndPenalty(Pc: Double, cost: Double, q: Double): Triple<Double, Double, Double> {  
 val energyShare = calculateEnergyShare(Pc, q)  
 val energyWithoutImbalance = *round*(Pc \* 24 \* energyShare)  
 val profit = energyWithoutImbalance \* cost \* 1000  
 val energyWithImbalance = *round*(Pc \* 24 \* (1 - energyShare))  
 val penalty = energyWithImbalance \* cost \* 1000  
 return Triple(profit, penalty, profit - penalty)  
 }  
}

В сій частині коду відбувається отримання даних з рядків вводу інтерфейсу. Після заповнення всіх полів(відбувається перевірка чи всі поля заповнені) відбуваються обрахунки.   
Код, що формує вигляд інтерфейсу має вигляд:  
<androidx.coordinatorlayout.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp"  
 tools:context=".MainActivity">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/mainTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/app\_name"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="16dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <EditText  
 android:id="@+id/P"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/p"  
 android:inputType="numberDecimal"  
 android:layout\_marginBottom="12dp"  
 app:layout\_constraintTop\_toBottomOf="@id/mainTitle"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <EditText  
 android:id="@+id/Q1"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/q1"  
 android:inputType="numberDecimal"  
 android:layout\_marginBottom="12dp"  
 app:layout\_constraintTop\_toBottomOf="@id/P"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <EditText  
 android:id="@+id/Q2"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/q2"  
 android:inputType="numberDecimal"  
 android:layout\_marginBottom="12dp"  
 app:layout\_constraintTop\_toBottomOf="@id/Q1"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <EditText  
 android:id="@+id/B"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/B"  
 android:inputType="numberDecimal"  
 android:layout\_marginBottom="16dp"  
 app:layout\_constraintTop\_toBottomOf="@id/Q2"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <Button  
 android:id="@+id/calculateButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:text="@string/calculate"  
 android:backgroundTint="?attr/colorPrimary"  
 android:textColor="@android:color/white"  
 android:textSize="18sp"  
 android:layout\_marginBottom="16dp"  
 app:layout\_constraintTop\_toBottomOf="@id/B"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 <TextView  
 android:id="@+id/resultTextView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:text=""  
 android:textSize="20sp"  
 android:textStyle="bold"  
 android:paddingTop="16dp"  
 android:layout\_marginBottom="24dp"  
 app:layout\_constraintTop\_toBottomOf="@id/calculateButton"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"/>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

В сьому коді створюються кнопки та комірки для ведення даних.

**Результат роботи програми за контрольним прикладом:**





**Висновок**

В результаті виконання лабораторної роботи №3 було опрацьовано алгоритм розрахунку прибутку від сонячних електростанцій з встановленою системою прогнозування сонячної потужності. Також було продовжено роботу з Kotlin та реалізовано дані розрахунку у вигляді калькулятора Kotlin.