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 5

з курсу: «Кросплатформна розробка мобільних застосунків»

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

Барабаш Маріна Володимирівна

Посилання на GitHub репозиторій: https://github.com/Aylosteraa/PW\_TV11\_Barabash\_Marina\_Volodymyrivna

**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

**Лабораторна робота № 3**

**Завдання:** Кросплатформний мобільний програмний калькулятор на мові DART для розрахунку прибутку від сонячних електростанцій з встановленою системою прогнозування сонячної потужності.

**Код програми:**

Основний код програми має вигляд:

import 'package:flutter/material.dart';

import 'dart:math';

void main() {

  runApp(CalculatorApp());

}

class CalculatorApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Fuel Emission Calculator',

      theme: ThemeData(primarySwatch: Colors.blue),

      home: CalculatorScreen(),

    );

  }

}

class CalculatorScreen extends StatefulWidget {

  @override

  \_CalculatorScreenState createState() => \_CalculatorScreenState();

}

class \_CalculatorScreenState extends State<CalculatorScreen> {

  final TextEditingController powerController = TextEditingController();

  final TextEditingController sigma1Controller = TextEditingController();

  final TextEditingController sigma2Controller = TextEditingController();

  final TextEditingController costController = TextEditingController();

  String Profit1 = "", Penalty1 = "", Loss1 = "", Profit2 = "", Penalty2 = "", Loss2 = "";

  bool showResult = false;

  List<dynamic> calculate(double power, double cost, double sigma) {

    double delta = power \* 0.05;

    double b1 = power - delta;

    double b2 = power + delta;

    double step = 0.001;

    double energyShare = 0.0;

    for (double p = b1; p < b2; p += step) {

      double pd = (1 / (sigma \* sqrt(2 \* pi))) \* exp(-((p - power) \* (p - power)) / (2 \* sigma \* sigma));

      energyShare += pd \* step;

    }

    double energyWithoutImbalance = (power \* 24 \* energyShare).roundToDouble();

    double profit = energyWithoutImbalance \* cost \* 1000;

    double energyWithImbalance = (power \* 24 \* (1 - energyShare)).roundToDouble();

    double penalty = energyWithImbalance \* cost \* 1000;

    double loss = profit - penalty;

    String choice = " (збиток)";

    if (loss > 0) {

      choice = " (прибуток)";

    }

    return [profit, penalty, loss, choice];

  }

  void showResults() {

    double power = double.tryParse(powerController.text) ?? 0.0;

    double cost = double.tryParse(costController.text) ?? 0.0;

    double sigma1 = double.tryParse(sigma1Controller.text) ?? 0.0;

    double sigma2 = double.tryParse(sigma2Controller.text) ?? 0.0;

    List<dynamic> results1 = calculate(power, cost, sigma1);

    List<dynamic> results2 = calculate(power, cost, sigma2);

    setState(() {

      Profit1 = "${results1[0].toStringAsFixed(2)} грн";

      Penalty1 = "${results1[1].toStringAsFixed(2)} грн";

      Loss1 = "${results1[2].toStringAsFixed(2)} грн ${results1[3]}";

      Profit2 = "${results2[0].toStringAsFixed(2)} грн";

      Penalty2 = "${results2[1].toStringAsFixed(2)} грн";

      Loss2 = "${results2[2].toStringAsFixed(2)} грн ${results2[3]}";

      showResult = true;

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text('Калькулятор')),

      body: Padding(

        padding: EdgeInsets.all(16.0),

        child: Column(

          children: [

          TextField(

          controller: powerController,

          keyboardType: TextInputType.number,

          decoration: InputDecoration(labelText: 'Середньодобова потужність'),

        ),

        TextField(

          controller: sigma1Controller,

          keyboardType: TextInputType.number,

          decoration: InputDecoration(labelText: 'Початкове середньоквадратичне відхилення'),

        ),

        TextField(

          controller: sigma2Controller,

          keyboardType: TextInputType.number,

          decoration: InputDecoration(labelText: 'Середньоквадратичне відхилення після вдосконалення'),

        ),

        TextField(

          controller: costController,

          keyboardType: TextInputType.number,

          decoration: InputDecoration(labelText: 'Вартість електроенергії'),

        ),

        SizedBox(height: 20),

        ElevatedButton(

          onPressed: showResults,

          child: Text('Розрахувати'),

        ),

        SizedBox(height: 20),

            if (showResult) ...[

              Text(

                'Початкове середньоквадратичне відхилення',

                style: TextStyle(

                    fontWeight: FontWeight.bold,

                  ),

                  textAlign: TextAlign.left

              ),

              Text('Прибуток: $Profit1', textAlign: TextAlign.left),

              Text('Штраф: $Penalty1', textAlign: TextAlign.left),

              Text(Loss1, textAlign: TextAlign.left),

              Text(

                'Середньоквадратичне відхилення після вдосконалення',

                style: TextStyle(

                    fontWeight: FontWeight.bold,

                  ),

                  textAlign: TextAlign.left

              ),

              Text('Прибуток: $Profit2', textAlign: TextAlign.left),

              Text('Штраф: $Penalty2', textAlign: TextAlign.left),

              Text(Loss2, textAlign: TextAlign.left),

            ]

        ]

        )

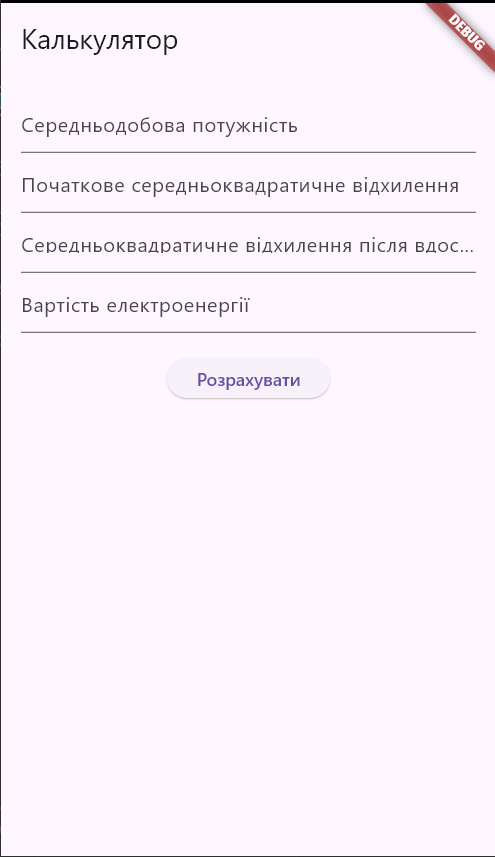
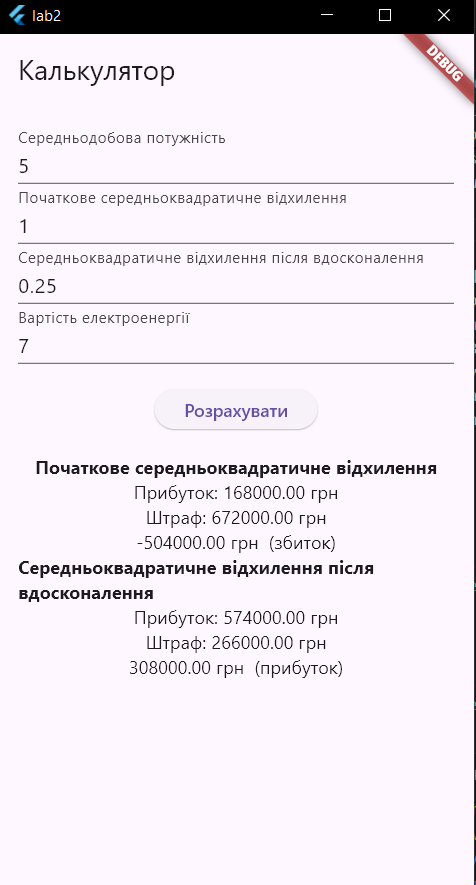
        ),

    );

  }

}

**Результат виконання програми:**

**Висновок:**

У лабораторній роботі 3 було розроблено програму для мобільного застосунку у вигляді калькулятору для обрахунку прибутків та збитків від сонячних електростанцій. Для створення програми використовувалась мова програмування Dart та Flutter.