ПРИЛОЖЕНИЕ А

(ОБЯЗАТЕЛЬНОЕ)

Листинг программы

Код программы для Windows

using Excel = Microsoft.Office.Interop.Excel;

using Word = Microsoft.Office.Interop.Word;

namespace WinFormsApp1

{

public partial class Form1 : Form

{

public double resultx, resulty, resultz, resultA, resultB, resultC, resultD, distance;

public string reshenie = string.Empty;

public Form1()

{

InitializeComponent();

}

/// <summary>

/// calculateButton\_Click выполняет действия по нажатию кнопки "Рассчитать"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void calculateButton\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = textBox2.ForeColor = textBox3.ForeColor = textBox4.ForeColor = textBox5.ForeColor = textBox6.ForeColor = textBox7.ForeColor = textBox8.ForeColor = Color.Black;

try

{

// приравнивание textBox к переменной result

resultx = EvaluateExpression(textBox1);

resulty = EvaluateExpression(textBox2);

resultz = EvaluateExpression(textBox3);

resultA = EvaluateExpression(textBox4);

resultB = EvaluateExpression(textBox5);

resultC = EvaluateExpression(textBox6);

resultD = EvaluateExpression(textBox7);

distance = Math.Abs(resultA \* resultx + resultB \* resulty + resultC \* resultz + resultD) / Math.Sqrt(resultA \* resultA + resultB \* resultB + resultC \* resultC); // рассчёт по формуле

textBox8.Text = "Расстояние от точки (" + resultx + ", " + resulty + ", " + resultz + ") до плоскости " + resultA + "x + " + resultB + "y + " + resultC + "z + " + resultD + " = 0; равно " + distance; // вывод результата на форму

reshenie = "Решение: " + resultA + " \* " + resultx + " + " + resultB + " \* " + resulty + " + " + resultC + " \* " + resultz + " + " + resultD + " / " + resultA + " \* " + resultA + " + " + resultB + " \* " + resultB + " + " + resultC + " \* " + resultC + " = " + distance; // решение типа string

if (reshenie.Contains("\u221E"))

{

textBox8.Text = "На ноль делить нельзя"; // вывод сообщения

reshenie = ""; // обнуление переменной

distance = 0.0; // обнуление переменной

}

if (textBox8.Text.Contains("число"))

{

textBox8.Text = "Пример не может быть решен.";

}

MessageBox.Show("Рассчёт выполнен!");

}

catch (Exception ex)

{

textBox8.Text = "Ошибка: " + ex.Message; // вывод ошибки

}

}

/// <summary>

/// clearButton\_Click выполняет действия по нажатию кнопки "Очистить"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void clearButton\_Click(object sender, EventArgs e)

{

// очистка textBox

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

resultx = resulty = resultz = resultA = resultB = resultC = resultD = 0.0; // очистка переменных типа double

reshenie = string.Empty; // очистка переменных типа string

textBox1.ForeColor = textBox2.ForeColor = textBox3.ForeColor = textBox4.ForeColor = textBox5.ForeColor = textBox6.ForeColor = textBox7.ForeColor = textBox8.ForeColor = Color.Black;

}

/// <summary>

/// exitButton\_Click выполняет действия по нажатию кнопки "Выйти"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void exitButton\_Click(object sender, EventArgs e)

{

this.Close(); // закрывает приложение

}

/// <summary>

/// wordButton\_Click выполняет действия по нажатию кнопки "Показать в Word"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void wordButton\_Click(object sender, EventArgs e)

{

calculateButton.PerformClick(); // выполняет действие, аналогичное нажатию кнопки

Word.Application word = new(); // создание объекта Word

Word.Document doc = word.Documents.Add(); // создание нового документа Word

if (textBox8.Text == "На ноль делить нельзя")

{

doc.Paragraphs[1].Range.Text = textBox8.Text; // вывод текста

}

else

{

doc.Paragraphs[1].Range.Text = "x = " + resultx + "\ny = " + resulty + "\nz = " + resultz + "\nA = " + resultA + "\nB = " + resultB + "\nC = " + resultC + "\nD = " + resultD + "\n" + this.textBox8.Text + "\n" + reshenie + "\n" + "Ответ: " + distance; // вывод текста

}

word.Visible = true;

}

/// <summary>

/// excelButton\_Click выполняет действия по нажатию кнопки "Показать в Excel"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void excelButton\_Click(object sender, EventArgs e)

{

calculateButton.PerformClick(); // выполняет действие, аналогичное нажатию кнопки

Excel.Application excel = new(); // создание объекта Excel

Excel.Workbook workbook = excel.Workbooks.Add(); // создание новой книги Excel

Excel.Worksheet sheet = (Excel.Worksheet)workbook.ActiveSheet; // выбор активного листа

// вывод текста на ячейку

if (textBox8.Text == "На ноль делить нельзя")

{

sheet.Cells[1, 1].Value = textBox8.Text;

}

else

{

sheet.Cells[1, 1].Value = "x = " + resultx;

sheet.Cells[2, 1].Value = "y = " + resulty;

sheet.Cells[3, 1].Value = "z = " + resultz;

sheet.Cells[4, 1].Value = "A = " + resultA;

sheet.Cells[5, 1].Value = "B = " + resultB;

sheet.Cells[6, 1].Value = "C = " + resultC;

sheet.Cells[7, 1].Value = "D = " + resultD;

sheet.Cells[1, 2].Value = textBox8.Text;

sheet.Cells[2, 2].Value = reshenie;

sheet.Cells[3, 2].Value = "Ответ: " + distance;

}

sheet.Columns.AutoFit(); // выравнивание столбцов

sheet.Rows.AutoFit(); // выравнивание строк

excel.Visible = true;

}

/// <summary>

/// EvaluateExpression преобразует выражения в числа

/// </summary>

/// <param name="expression"></param>

/// <returns></returns>

private static double EvaluateExpression(TextBox expression)

{

try

{

expression.Text = expression.Text.Replace(" ", ""); // удаление пробелов из выражения

System.Data.DataTable dt = new(); // создание объекта для вычисления математических выражений

var result = dt.Compute(expression.Text, ""); // вычисление математических выражений

return Convert.ToDouble(result); // возврат результата

}

catch (Exception ex)

{

expression.ForeColor = Color.Red;

expression.Text = "0";

MessageBox.Show($"Ошибка в {expression}: " + ex.Message + "Все поля с ошибками были заменены на 0"); // вывод ошибки

return 0.0;

}

}

/// <summary>

/// textBox1\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox1

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

textBox2.Focus(); // переводит фокус на другой TextBox

}

}

/// <summary>

/// textBox2\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox2

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

textBox3.Focus(); // переводит фокус на другой TextBox

}

}

/// <summary>

/// textBox3\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox3

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

textBox4.Focus(); // переводит фокус на другой TextBox

}

}

/// <summary>

/// textBox4\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox4

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

textBox5.Focus(); // переводит фокус на другой TextBox

}

}

/// <summary>

/// textBox5\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox5

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox5\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

textBox6.Focus(); // переводит фокус на другой TextBox

}

}

/// <summary>

/// textBox6\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox6

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox6\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

textBox7.Focus(); // переводит фокус на другой TextBox

}

}

/// <summary>

/// textBox7\_KeyPress выполняет действие по нажатию кнопки во время фокусировки на textBox7

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox7\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

e.Handled = true; // предотвращает дальнейшую обработку нажатия клавиши Enter

calculateButton.PerformClick(); // выполняет действие, аналогичное нажатию кнопки

}

}

}

}

Код программы для Android

using Android.App;

using Android.OS;

using Android.Widget;

using ClosedXML.Excel;

using System;

using System.IO;

namespace PowerCalculator

{

[Activity(Label = "@string/app\_name", Theme = "@style/AppTheme", MainLauncher = true)]

public class MainActivity : Activity

{

private double resultx, resulty, resultz, resultA, resultB, resultC, resultD, distance;

private string reshenie = string.Empty;

private EditText textBox1, textBox2, textBox3, textBox4, textBox5, textBox6, textBox7;

private TextView textBox8;

private string externalPath;

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

SetContentView(Resource.Layout.Main);

textBox1 = FindViewById<EditText>(Resource.Id.textBox1);

textBox1.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox1.SetSingleLine(true);

textBox2 = FindViewById<EditText>(Resource.Id.textBox2);

textBox2.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox2.SetSingleLine(true);

textBox3 = FindViewById<EditText>(Resource.Id.textBox3);

textBox3.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox3.SetSingleLine(true);

textBox4 = FindViewById<EditText>(Resource.Id.textBox4);

textBox4.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox4.SetSingleLine(true);

textBox5 = FindViewById<EditText>(Resource.Id.textBox5);

textBox5.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox5.SetSingleLine(true);

textBox6 = FindViewById<EditText>(Resource.Id.textBox6);

textBox6.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox6.SetSingleLine(true);

textBox7 = FindViewById<EditText>(Resource.Id.textBox7);

textBox7.InputType = Android.Text.InputTypes.ClassText | Android.Text.InputTypes.TextFlagCapSentences | Android.Text.InputTypes.TextFlagNoSuggestions | Android.Text.InputTypes.TextFlagMultiLine;

textBox7.SetSingleLine(true);

textBox8 = FindViewById<TextView>(Resource.Id.textBox8);

var calculateButton = FindViewById<Button>(Resource.Id.calculateButton);

calculateButton.Click += CalculateButton\_Click;

var clearButton = FindViewById<Button>(Resource.Id.clearButton);

clearButton.Click += ClearButton\_Click;

var exitButton = FindViewById<Button>(Resource.Id.exitButton);

exitButton.Click += ExitButton\_Click;

var excelButton = FindViewById<Button>(Resource.Id.excelButton);

excelButton.Click += ExcelButton\_Click;

externalPath = Path.Combine(GetExternalFilesDir(null).AbsolutePath, "название\_файла.xlsx");

}

/// <summary>

/// CalculateButton\_Click выполняет действия по нажатию кнопки "Evaluate"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void CalculateButton\_Click(object sender, System.EventArgs e)

{

// меняет цвет текста на чёрный

textBox1.SetTextColor(color: Android.Graphics.Color.Black);

textBox2.SetTextColor(color: Android.Graphics.Color.Black);

textBox3.SetTextColor(color: Android.Graphics.Color.Black);

textBox4.SetTextColor(color: Android.Graphics.Color.Black);

textBox5.SetTextColor(color: Android.Graphics.Color.Black);

textBox6.SetTextColor(color: Android.Graphics.Color.Black);

textBox7.SetTextColor(color: Android.Graphics.Color.Black);

try

{

// приравнивание textBox к переменной result

resultx = EvaluateExpression(textBox1);

resulty = EvaluateExpression(textBox2);

resultz = EvaluateExpression(textBox3);

resultA = EvaluateExpression(textBox4);

resultB = EvaluateExpression(textBox5);

resultC = EvaluateExpression(textBox6);

resultD = EvaluateExpression(textBox7);

distance = Math.Abs(resultA \* resultx + resultB \* resulty + resultC \* resultz + resultD) / Math.Sqrt(resultA \* resultA + resultB \* resultB + resultC \* resultC); // рассчёт по формуле

textBox8.Text = "Расстояние от точки (" + resultx + ", " + resulty + ", " + resultz + ") до плоскости " + resultA + "x + " + resultB + "y + " + resultC + "z + " + resultD + " = 0; равно " + distance; // вывод результата на форму

reshenie = "Решение: " + resultA + " \* " + resultx + " + " + resultB + " \* " + resulty + " + " + resultC + " \* " + resultz + " + " + resultD + " / " + resultA + " \* " + resultA + " + " + resultB + " \* " + resultB + " + " + resultC + " \* " + resultC + " = " + distance; // решение типа string

if (reshenie.Contains("Infinity"))

{

textBox8.Text = "На ноль делить нельзя"; // вывод сообщения

reshenie = ""; // обнуление переменной

distance = 0.0; // обнуление переменной

}

if (textBox8.Text.Contains("NaN"))

{

textBox8.Text = "Пример не может быть решен."; // вывод ошибки

}

}

catch (Exception ex)

{

textBox8.Text = "Ошибка: " + ex.Message; // вывод ошибки

}

}

/// <summary>

/// ClearButton\_Click выполняет действия по нажатию кнопки "Clear"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void ClearButton\_Click(object sender, EventArgs e)

{

// очистка textBox

textBox1.Text = textBox2.Text = textBox3.Text = textBox4.Text = textBox5.Text = textBox6.Text = textBox7.Text = textBox8.Text = string.Empty;

resultx = resulty = resultz = resultA = resultB = resultC = resultD = 0.0; // очистка переменных типа double

reshenie = string.Empty; // очистка переменных типа string

textBox1.SetTextColor(color: Android.Graphics.Color.Black);

textBox2.SetTextColor(color: Android.Graphics.Color.Black);

textBox3.SetTextColor(color: Android.Graphics.Color.Black);

textBox4.SetTextColor(color: Android.Graphics.Color.Black);

textBox5.SetTextColor(color: Android.Graphics.Color.Black);

textBox6.SetTextColor(color: Android.Graphics.Color.Black);

textBox7.SetTextColor(color: Android.Graphics.Color.Black);

textBox8.SetTextColor(color: Android.Graphics.Color.Black);

}

/// <summary>

/// ExcelButton\_Click выполняет действия по кнопке "Show in Excel"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void ExcelButton\_Click(object sender, EventArgs e)

{

CalculateButton\_Click(sender, e);

var workbook = new XLWorkbook(); // создание нового документа Excel

var worksheet = workbook.Worksheets.Add("Sheet1"); // создание нового документа Excel

worksheet.Column("B").Width = 100;

// добавление данных в ячейки

if (textBox8.Text == "На ноль делить нельзя")

{

worksheet.Cell("A1").Value = textBox8.Text;

}

else if (textBox8.Text == "Пример не может быть решен.")

{

worksheet.Cell("A1").Value = textBox8.Text;

}

else

{

worksheet.Cell("A1").Value = "x = " + resultx;

worksheet.Cell("A2").Value = "y = " + resulty;

worksheet.Cell("A3").Value = "z = " + resultz;

worksheet.Cell("A4").Value = "A = " + resultA;

worksheet.Cell("A5").Value = "B = " + resultB;

worksheet.Cell("A6").Value = "C = " + resultC;

worksheet.Cell("A7").Value = "D = " + resultD;

worksheet.Cell("B1").Value = textBox8.Text;

worksheet.Cell("B2").Value = reshenie;

worksheet.Cell("B3").Value = "Ответ: " + distance;

}

workbook.SaveAs(externalPath); // сохранение файла

}

/// <summary>

/// ExitButton\_Click выполняет действия по нажатию кнопки "Exit"

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void ExitButton\_Click(object sender, EventArgs e)

{

FinishAffinity(); // закрывает приложение

}

/// <summary>

/// EvaluateExpression рассчитывает выражения

/// </summary>

/// <param name="expressionEditText"></param>

/// <returns></returns>

private double EvaluateExpression(EditText expressionEditText)

{

try

{

string expression = expressionEditText.Text.Replace(" ", ""); // удаление пробелов из выражения

NCalc.Expression expr = new NCalc.Expression(expression); // создание объекта для вычисления математических выражений

double result = Convert.ToDouble(expr.Evaluate()); // вычисление математического выражения

return result; // возврат результата

}

catch (Exception ex)

{

expressionEditText.SetTextColor(Android.Graphics.Color.Red); // красит текст в красный

expressionEditText.Text = "0"; // приравнивает текст к нулю

// вывод ошибки

AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.SetTitle("Ошибка");

alert.SetMessage("Ошибка: " + ex.Message + ". Все поля с ошибками были заменены на 0.");

alert.SetPositiveButton("OK", (senderAlert, args) => { });

alert.Show();

return 0.0; // возврат результата

}

}

}

}