ПРИЛОЖЕНИЕ А

(информационное)

Листинг программы

using iText.IO.Font;

using iText.Kernel.Font;

using iText.Kernel.Pdf;

using iText.Layout;

using iText.Layout.Element;

using System.Diagnostics;

using Excel = Microsoft.Office.Interop.Excel;

using Word = Microsoft.Office.Interop.Word;

namespace Mixed\_Vectors

{

public partial class Form1 : Form

{

public double[] vectorA, vectorB, vectorC;

public double result;

// Form1() вызывается при запуске программы

public Form1()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

}

// ButtonCalculate\_Click() вызывается при нажатии на кнопку "Рассчитать"

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

{

Calculate();

MessageBox.Show("Рассчёт выполнен!");

}

// ButtonInstruction\_Click() вызывается при нажатии на кнопку "Инструкция"

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

{

Instruction();

}

// ButtonWord\_Click() вызывается при нажатии на кнопку "Экспорт в ворд"

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

{

ExportToWord();

}

// ButtonExcel\_Click() вызывается при нажатии на кнопку "Экспорт в Excel"

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

{

ExportToExcel();

}

// ButtonPDF\_Click() вызывается при нажатии на кнопку "Экспорт в PDF"

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

{

ExportToPDF();

}

// ButtonClear\_Click() вызывается при нажатии на кнопку "Очистить"

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

{

Clear();

MessageBox.Show("Все поля очищены!");

}

// ButtonExit\_Click() вызывается при нажатаии на кнопку "Выход"

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

{

Application.Exit();

}

// Calculate() вызывается при рассчёте

private void Calculate()

{

try

{

vectorA = vectorB = vectorC = null;

result = 0;

if (textBoxFirstVectorX.Text != string.Empty && textBoxFirstVectorY.Text != string.Empty && textBoxFirstVectorZ.Text != string.Empty

&& textBoxSecondVectorX.Text != string.Empty && textBoxSecondVectorY.Text != string.Empty && textBoxSecondVectorZ.Text !=

string.Empty && textBoxThirdVectorX.Text != string.Empty && textBoxThirdVectorY.Text != string.Empty && textBoxThirdVectorZ.Text

!= string.Empty)

{

vectorA = [double.Parse(textBoxFirstVectorX.Text), double.Parse(textBoxFirstVectorY.Text), double.Parse(textBoxFirstVectorZ.Text)];

vectorB = [double.Parse(textBoxSecondVectorX.Text),

double.Parse(textBoxSecondVectorY.Text),

double.Parse(textBoxSecondVectorZ.Text)];

vectorC = [double.Parse(textBoxThirdVectorX.Text), double.Parse(textBoxThirdVectorY.Text), double.Parse(textBoxThirdVectorZ.Text)];

result = vectorA[0] \* (vectorB[1] \* vectorC[2] - vectorB[2] \* vectorC[1]) + vectorA[1] \* (vectorB[2] \* vectorC[0] - vectorB[0] \*

vectorC[2]) + vectorA[2] \* (vectorB[0] \* vectorC[1] - vectorB[1] \* vectorC[0]);

labelResult.Text = $"Ответ: {result:F2}";

}

else

{

Clear();

MessageBox.Show("Введите числа!");

}

}

catch

{

Clear();

MessageBox.Show("Можно вводить только числа!");

}

}

// Instruction() вызывается при открытии инструкции

private void Instruction()

{

try

{

MessageBox.Show("Для того чтобы найти смешанное произведение векторов:\r\nвведите значение векторов;\r\nнажмите кнопку " +

"\"Рассчитать\" и вы получите детальное решение задачи;\r\nнажмите кнопку \"Экспорт в Word\" и вы получите данные о" +

" рассчёте в программе \"Word\";\r\nнажмите кнопку \"Экспорт в Excel\" и вы получите данные о рассчёте в программе " +

"\"Excel\";\r\nнажмите кнопку \"Экспорт в PDF\" и вы получите данные о рассчёте в формате .pdf;\r\nнажмите кнопку " +

"\"Очистить\" и все поля будут очищены;\r\nнажмите кнопку \"Выйти\" и программа закроется.");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// ExportToWord() вызывается при экспорте в Word

private void ExportToWord()

{

try

{

Calculate();

Word.Application wordApp = new();

wordApp.Visible = true;

Word.Document doc = wordApp.Documents.Add();

doc.Content.Text = $"Результат смешанного произведения векторов\n" +

$"Первый вектор:\n{textBoxFirstVectorName.Text} = " + "{" + $"{textBoxFirstVectorX.Text} ; {textBoxFirstVectorY.Text} ; " +

$"{textBoxFirstVectorZ.Text}" + "}" + $"\nВторой вектор:\n{textBoxSecondVectorName.Text} = " + "{" +

$"{textBoxSecondVectorX.Text} ; {textBoxSecondVectorY.Text} ; {textBoxSecondVectorZ.Text}" + "}" + $"\nТретий вектор:\n" +

$"{textBoxThirdVectorName.Text} = " + "{" + $"{textBoxThirdVectorX.Text} ; {textBoxThirdVectorY.Text} ; " +

$"{textBoxThirdVectorZ.Text}" + "}" + $"\nРешение: {textBoxFirstVectorName.Text}x \* {textBoxSecondVectorName.Text}y \* " +

$"{textBoxThirdVectorName.Text}z + {textBoxFirstVectorName.Text}y \* {textBoxSecondVectorName.Text}z \* " +

$"{textBoxThirdVectorName.Text}x + {textBoxFirstVectorName.Text}z \* {textBoxSecondVectorName.Text}x \* " +

$"{textBoxThirdVectorName.Text}y - {textBoxFirstVectorName.Text}z \* {textBoxSecondVectorName.Text}y \* " +

$"{textBoxThirdVectorName.Text}x - {textBoxFirstVectorName.Text}y \* {textBoxSecondVectorName.Text}x \* " +

$"{textBoxThirdVectorName.Text}z - {textBoxFirstVectorName.Text}x \* {textBoxSecondVectorName.Text}z \* " +

$"{textBoxThirdVectorName.Text}y = {textBoxFirstVectorX.Text} \* {textBoxSecondVectorY.Text} \* {textBoxThirdVectorZ.Text} + " +

$"{textBoxFirstVectorY.Text} \* {textBoxSecondVectorZ.Text} \* {textBoxThirdVectorX.Text} + {textBoxFirstVectorZ.Text} \* " +

$"{textBoxSecondVectorX.Text} \* {textBoxThirdVectorY.Text} - {textBoxFirstVectorZ.Text} \* {textBoxSecondVectorY.Text} \* " +

$"{textBoxThirdVectorX.Text} - {textBoxFirstVectorY.Text} \* {textBoxSecondVectorX.Text} \* {textBoxThirdVectorZ.Text} - " +

$"{textBoxFirstVectorX.Text} \* {textBoxSecondVectorZ.Text} \* {textBoxThirdVectorY.Text} = {result:F2}";

MessageBox.Show("Экспорт в Word завершён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// ExportToExcel() вызывается при экспорте в Excel

private void ExportToExcel()

{

try

{

Calculate();

Excel.Application excelApp = new();

excelApp.Visible = true;

Excel.Workbook workbook = excelApp.Workbooks.Add();

Excel.Worksheet worksheet = workbook.Worksheets[1];

worksheet.Cells[1, 1] = $"Результат смешанного произведения векторов\n" +

$"Первый вектор:\n{textBoxFirstVectorName.Text} = " + "{" + $"{textBoxFirstVectorX.Text} ; {textBoxFirstVectorY.Text} ; " +

$"{textBoxFirstVectorZ.Text}" + "}" + $"\nВторой вектор:\n{textBoxSecondVectorName.Text} = " + "{" +

$"{textBoxSecondVectorX.Text} ; {textBoxSecondVectorY.Text} ; {textBoxSecondVectorZ.Text}" + "}" + $"\nТретий вектор:\n" +

$"{textBoxThirdVectorName.Text} = " + "{" + $"{textBoxThirdVectorX.Text} ; {textBoxThirdVectorY.Text} ; " +

$"{textBoxThirdVectorZ.Text}" + "}" + $"\nРешение: {textBoxFirstVectorName.Text}x \* {textBoxSecondVectorName.Text}y \* " +

$"{textBoxThirdVectorName.Text}z + {textBoxFirstVectorName.Text}y \* {textBoxSecondVectorName.Text}z \* " +

$"{textBoxThirdVectorName.Text}x + {textBoxFirstVectorName.Text}z \* {textBoxSecondVectorName.Text}x \* " +

$"{textBoxThirdVectorName.Text}y - {textBoxFirstVectorName.Text}z \* {textBoxSecondVectorName.Text}y \* " +

$"{textBoxThirdVectorName.Text}x - {textBoxFirstVectorName.Text}y \* {textBoxSecondVectorName.Text}x \* " +

$"{textBoxThirdVectorName.Text}z - {textBoxFirstVectorName.Text}x \* {textBoxSecondVectorName.Text}z \* " +

$"{textBoxThirdVectorName.Text}y = {textBoxFirstVectorX.Text} \* {textBoxSecondVectorY.Text} \* {textBoxThirdVectorZ.Text} + " +

$"{textBoxFirstVectorY.Text} \* {textBoxSecondVectorZ.Text} \* {textBoxThirdVectorX.Text} + {textBoxFirstVectorZ.Text} \* " +

$"{textBoxSecondVectorX.Text} \* {textBoxThirdVectorY.Text} - {textBoxFirstVectorZ.Text} \* {textBoxSecondVectorY.Text} \* " +

$"{textBoxThirdVectorX.Text} - {textBoxFirstVectorY.Text} \* {textBoxSecondVectorX.Text} \* {textBoxThirdVectorZ.Text} - " +

$"{textBoxFirstVectorX.Text} \* {textBoxSecondVectorZ.Text} \* {textBoxThirdVectorY.Text} = {result:F2}";

worksheet.Columns.AutoFit();

worksheet.Rows.AutoFit();

Excel.Range rangeToMerge = worksheet.Range["A1:E1"];

rangeToMerge.Merge();

MessageBox.Show("Экспорт в Excel завершён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// ExportToPDF() вызывается при экспорте в PDF

private void ExportToPDF()

{

try

{

Calculate();

string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "output.pdf");

using var pdfWriter = new PdfWriter(filePath);

using var pdfDocument = new PdfDocument(pdfWriter);

using var pdfDoc = new Document(pdfDocument);

PdfFont timesFont = PdfFontFactory.CreateFont("c:/windows/fonts/times.ttf", PdfEncodings.IDENTITY\_H, true);

pdfDoc.Add(new Paragraph($"Результат смешанного произведения векторов\n" +

$"Первый вектор:\n{textBoxFirstVectorName.Text} = " + "{" + $"{textBoxFirstVectorX.Text} ; {textBoxFirstVectorY.Text} ; " +

$"{textBoxFirstVectorZ.Text}" + "}" + $"\nВторой вектор:\n{textBoxSecondVectorName.Text} = " + "{" +

$"{textBoxSecondVectorX.Text} ; {textBoxSecondVectorY.Text} ; {textBoxSecondVectorZ.Text}" + "}" + $"\nТретий вектор:\n" +

$"{textBoxThirdVectorName.Text} = " + "{" + $"{textBoxThirdVectorX.Text} ; {textBoxThirdVectorY.Text} ; " +

$"{textBoxThirdVectorZ.Text}" + "}" + $"\nРешение: {textBoxFirstVectorName.Text}x \* {textBoxSecondVectorName.Text}y \* " +

$"{textBoxThirdVectorName.Text}z + {textBoxFirstVectorName.Text}y \* {textBoxSecondVectorName.Text}z \* " +

$"{textBoxThirdVectorName.Text}x + {textBoxFirstVectorName.Text}z \* {textBoxSecondVectorName.Text}x \* " +

$"{textBoxThirdVectorName.Text}y - {textBoxFirstVectorName.Text}z \* {textBoxSecondVectorName.Text}y \* " +

$"{textBoxThirdVectorName.Text}x - {textBoxFirstVectorName.Text}y \* {textBoxSecondVectorName.Text}x \* " +

$"{textBoxThirdVectorName.Text}z - {textBoxFirstVectorName.Text}x \* {textBoxSecondVectorName.Text}z \* " +

$"{textBoxThirdVectorName.Text}y = {textBoxFirstVectorX.Text} \* {textBoxSecondVectorY.Text} \* {textBoxThirdVectorZ.Text} + " +

$"{textBoxFirstVectorY.Text} \* {textBoxSecondVectorZ.Text} \* {textBoxThirdVectorX.Text} + {textBoxFirstVectorZ.Text} \* " +

$"{textBoxSecondVectorX.Text} \* {textBoxThirdVectorY.Text} - {textBoxFirstVectorZ.Text} \* {textBoxSecondVectorY.Text} \* " +

$"{textBoxThirdVectorX.Text} - {textBoxFirstVectorY.Text} \* {textBoxSecondVectorX.Text} \* {textBoxThirdVectorZ.Text} - " +

$"{textBoxFirstVectorX.Text} \* {textBoxSecondVectorZ.Text} \* {textBoxThirdVectorY.Text} = {result:F2}").SetFont(timesFont));

pdfDocument.Close();

MessageBox.Show("Экспорт в PDF завершён!");

ProcessStartInfo psi = new()

{

FileName = "cmd",

RedirectStandardInput = true,

UseShellExecute = false,

RedirectStandardOutput = true,

CreateNoWindow = true

};

Process process = new() { StartInfo = psi };

process.Start();

process.StandardInput.WriteLine($"start {filePath}");

process.StandardInput.Flush();

process.StandardInput.Close();

process.WaitForExit();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// Clear() вызывается при очистке всех полей

private void Clear()

{

try

{

textBoxFirstVectorX.Text = string.Empty;

textBoxFirstVectorY.Text = string.Empty;

textBoxFirstVectorZ.Text = string.Empty;

textBoxSecondVectorX.Text = string.Empty;

textBoxSecondVectorY.Text = string.Empty;

textBoxSecondVectorZ.Text = string.Empty;

textBoxThirdVectorX.Text = string.Empty;

textBoxThirdVectorY.Text = string.Empty;

textBoxThirdVectorZ.Text = string.Empty;

labelResult.Text = $"Ответ: ";

vectorA = vectorB = vectorC = null;

result = 0;

textBoxFirstVectorName.Text = "a";

textBoxSecondVectorName.Text = "b";

textBoxThirdVectorName.Text = "c";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}