Федеральное государственное образовательное бюджетное учреждение

высшего образования «Финансовый университет при Правительстве Российской Федерации»

**Колледж информатики и программирования**

**Специальность 09.02.03 Программирование в компьютерных системах**

**ПМ.03 Участие в интеграции программных модулей**

**МДК.03.01. Технология разработки программного обеспечения**

ПРАКТИЧЕСКАЯ РАБОТА №2

Выполнил:

студент группы: 3ПКС-115

Черников А. В.

Проверила:

Морозова М. В.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2018

**Практическая работа №20**

**Условие задачи:**

Задание:

Написать программу с графическим интерфейсом которая будет находить произведение 2 матриц.

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace MatrixMultiplication

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

double[,] matrix1, matrix2, result;

public MainWindow()

{

InitializeComponent();

}

/// <summary>

/// Event handler for all three combo boxes:

/// Creates matrices of requested size filled with zeroes

/// Displays matrices on the form

/// </summary>

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

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

private void matrixdimensions\_changed(object sender, SelectionChangedEventArgs e)

{

int m1rows = 1;

int m1columns\_m2rows = 1;

int m2columns = 1;

// Set the matrix dimensions - note that matrix2 must have the same number of rows as matrix1 has columns

if (matrix1width != null) m1columns\_m2rows = matrix1width.SelectedIndex + 1;

if (matrix1height != null) m1rows = matrix1height.SelectedIndex + 1;

if (matrix2width != null) m2columns = matrix2width.SelectedIndex + 1;

// matrices will be initialized with all zeros

matrix1 = new double[m1columns\_m2rows, m1rows];

matrix2 = new double[m2columns, m1columns\_m2rows];

result = new double[m2columns, m1rows];

// Display the matrices on the form

initializeGrid(grid1, matrix1);

initializeGrid(grid2, matrix2);

initializeGrid(grid3, result);

}

/// <summary>

/// Create a grid on the form displaying the matrix values in editable textboxes

/// </summary>

/// <param name="grid">A WPF grid control</param>

/// <param name="matrix">The matrix to display</param>

private void initializeGrid(Grid grid, double[,] matrix)

{

if (grid != null)

{

// Reset the grid before doing anything

grid.Children.Clear();

grid.ColumnDefinitions.Clear();

grid.RowDefinitions.Clear();

// Get the dimensions

int columns = matrix.GetLength(0);

int rows = matrix.GetLength(1);

// Add the correct number of coumns to the grid

for (int x = 0; x < columns; x++)

{

// GridUnitType.Star - The value is expressed as a weighted proportion of available space

grid.ColumnDefinitions.Add(new ColumnDefinition() { Width = new GridLength(1, GridUnitType.Star), });

}

for (int y = 0; y < rows; y++)

{

// GridUnitType.Star - The value is expressed as a weighted proportion of available space

grid.RowDefinitions.Add(new RowDefinition() { Height = new GridLength(1, GridUnitType.Star), });

}

// Fill each cell of the grid with an editable TextBox containing the value from the matrix - centered in the cell

for (int x = 0; x < columns; x++)

{

for (int y = 0; y < rows; y++)

{

double cell = (double)matrix[x, y];

TextBox t = new TextBox();

t.Text = cell.ToString();

t.VerticalAlignment = System.Windows.VerticalAlignment.Center;

t.HorizontalAlignment = System.Windows.HorizontalAlignment.Center;

t.SetValue(Grid.RowProperty, y);

t.SetValue(Grid.ColumnProperty, x);

grid.Children.Add(t);

}

}

}

}

/// <summary>

/// Do the matrix multiplication

/// </summary>

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

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

private void buttonCalculate\_Click(object sender, RoutedEventArgs e)

{

getValuesFromGrid(grid1,matrix1);

getValuesFromGrid(grid2, matrix2);

int m1stolb\_m2strok, m1strok, m2stolb;

m1stolb\_m2strok =matrix1.GetLength(0) ;

m1strok = matrix1.GetLength(1);

m2stolb = matrix2.GetLength(0);

for (int row=0;row < m1strok; row++)

{

for (int column = 0; column < m2stolb ; column++)

{

double accumulator = 0;

for (int cell = 0; cell < m1stolb\_m2strok; cell++)

{

accumulator =accumulator+( matrix1[cell, row] \* matrix2[column, cell]);

}

result[column, row] = accumulator;

}

}

initializeGrid(grid3, result);

}

/// <summary>

/// Retrieve data from the grid and put it in the array representing the matrix

/// </summary>

/// <param name="grid">WPF Grid control holding matrix data</param>

/// <param name="matrix">Matrix array to store the values</param>

private void getValuesFromGrid(Grid grid, double[,] matrix)

{

// Set up counters

int columns = grid.ColumnDefinitions.Count;

int rows = grid.RowDefinitions.Count;

// We should check that the grid has the same dimensions as the array

// (See Module 4)

// Iterate over cells in Grid, copying to matrix array

for (int c = 0; c < grid.Children.Count; c++)

{

TextBox t = (TextBox)grid.Children[c];

int row = Grid.GetRow(t);

int column = Grid.GetColumn(t);

if (!double.TryParse(t.Text, out matrix[column, row]))

{

MessageBox.Show("Проверьте корректность введенных данных");

return;

}

}}}}

Результаты:

Результат задачи №1 представлен на рисунке 1, 2, 3, 4

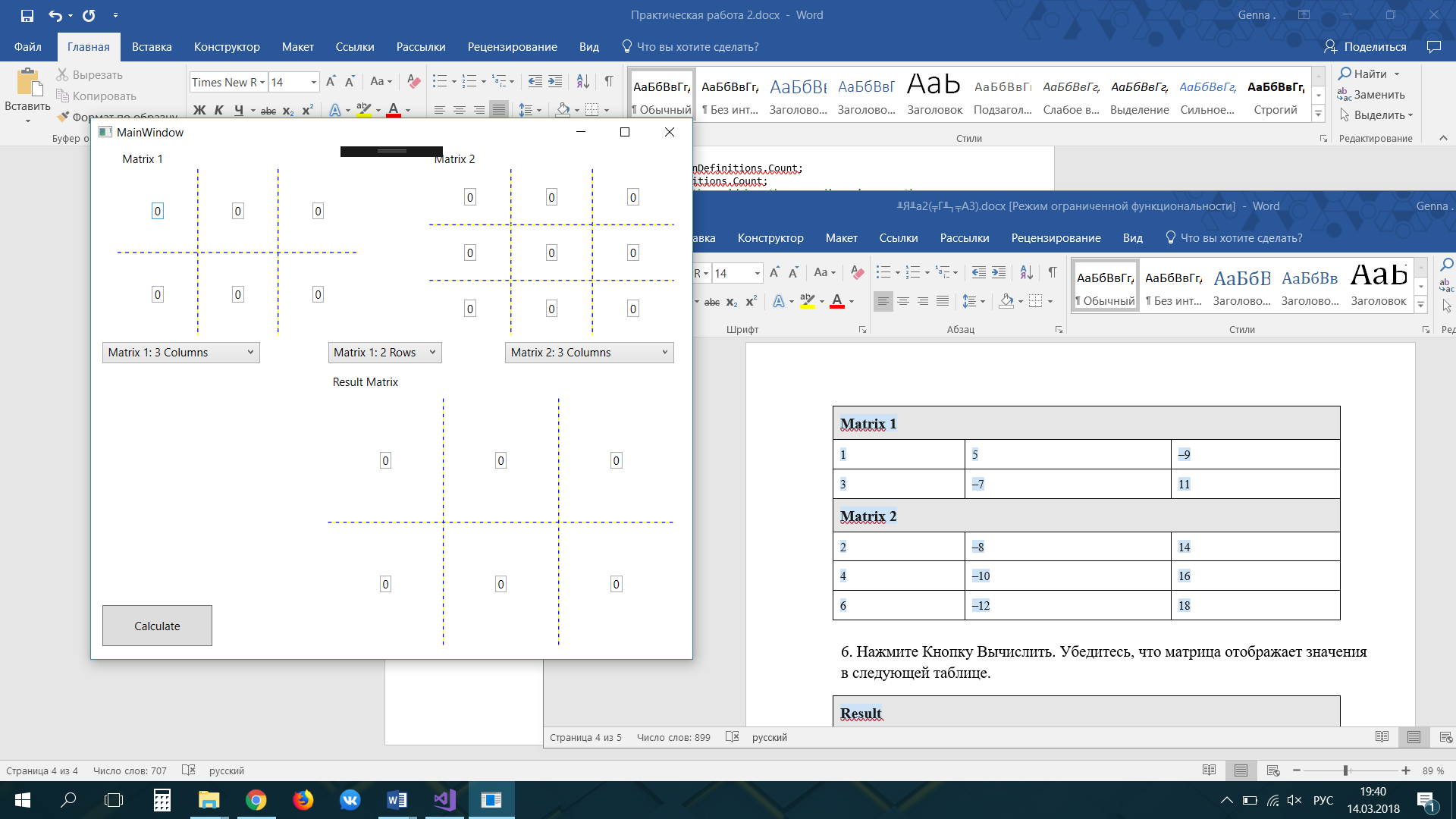


Рисунок 1

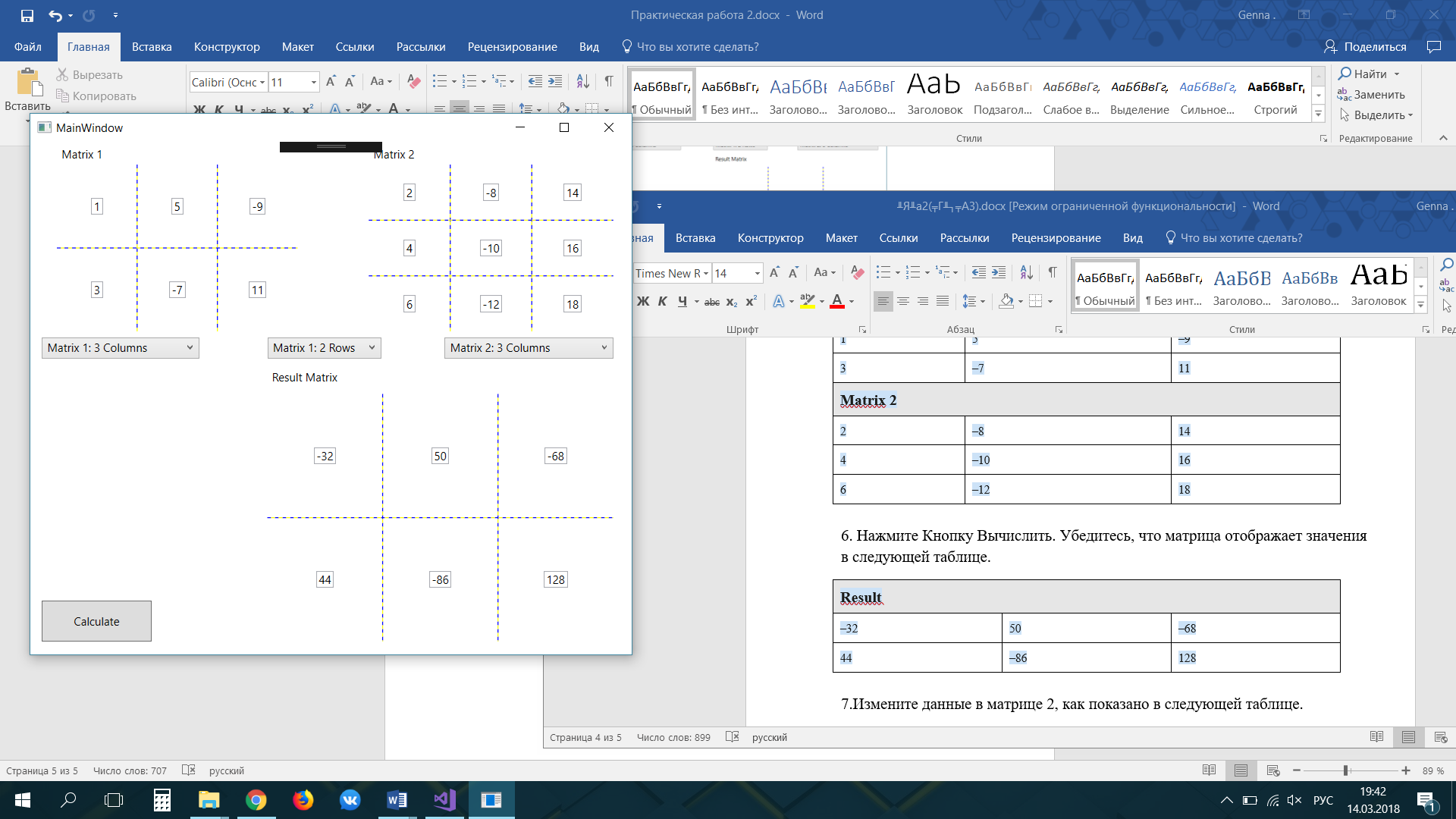


Рисунок 2

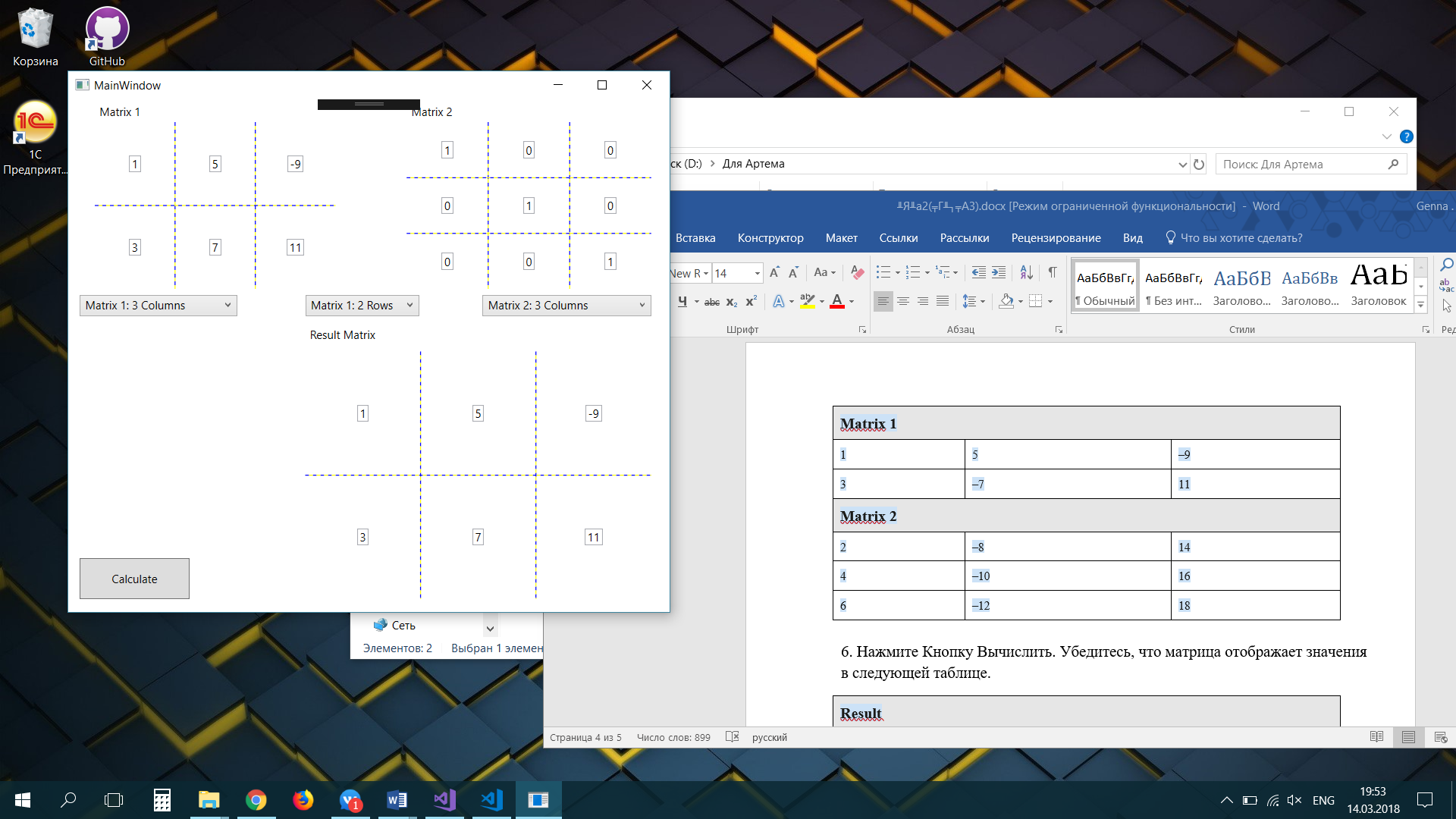


Рисунок 3

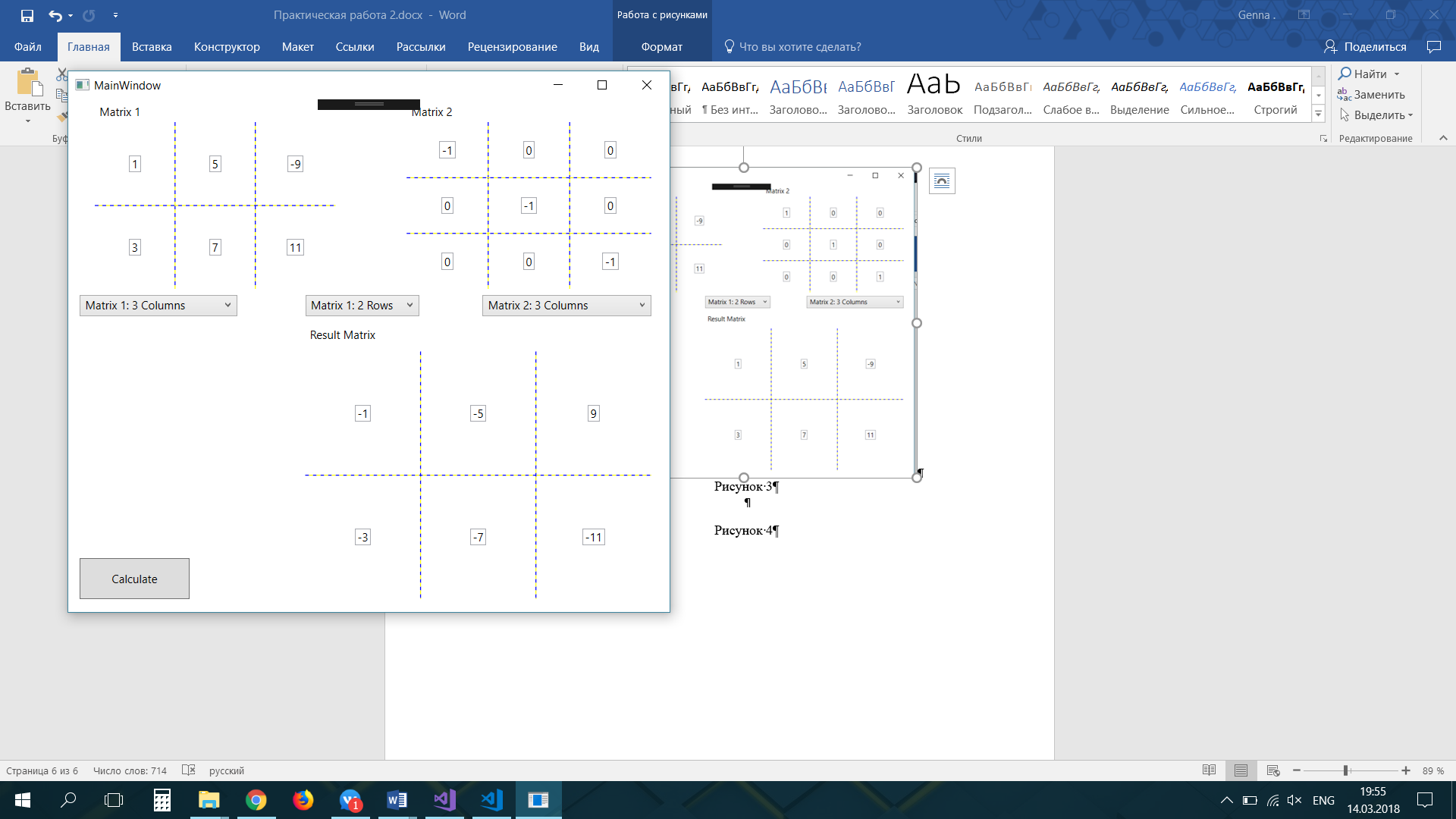


Рисунок 4