Задание 1

namespace Lab6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

// Функция для парсинга матрицы из текста

static double[,] ParseMatrix(string text, int rows, int cols)

{

double[,] matrix = new double[rows, cols];

string[] lines = text.Trim().Split('\n');

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

{

string[] numbers = lines[i].Trim().Split(' ');

for (int j = 0; j < cols; j++)

{

matrix[i, j] = double.Parse(numbers[j]);

}

}

return matrix;

}

// Функция для перемещения максимального элемента в верхний левый угол

static void MoveMaxToTopLeft(ref double[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

// Находим индексы максимального элемента

int maxRow = 0, maxCol = 0;

double maxValue = matrix[0, 0];

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

{

for (int j = 0; j < cols; j++)

{

if (matrix[i, j] > maxValue)

{

maxValue = matrix[i, j];

maxRow = i;

maxCol = j;

}

}

}

// Переставляем строки

if (maxRow != 0)

{

for (int j = 0; j < cols; j++)

{

double temp = matrix[0, j];

matrix[0, j] = matrix[maxRow, j];

matrix[maxRow, j] = temp;

}

}

// Переставляем столбцы

if (maxCol != 0)

{

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

{

double temp = matrix[i, 0];

matrix[i, 0] = matrix[i, maxCol];

matrix[i, maxCol] = temp;

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

string input = textBox1.Text;

double[,] matrix = ParseMatrix(input, 4, 5);

MoveMaxToTopLeft(ref matrix);

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

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

{

for (int j = 0; j < cols; j++)

{

textBox2.Text += matrix[i, j] + " ";

}

textBox2.Text += Environment.NewLine;

}

}

}

}

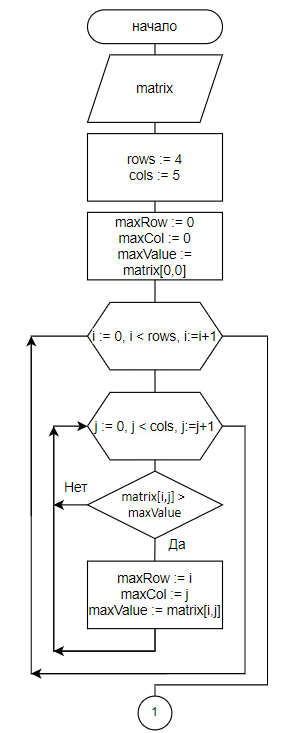


Схема алгоритма

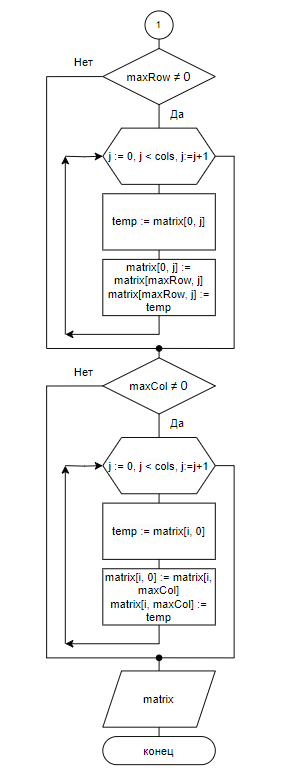
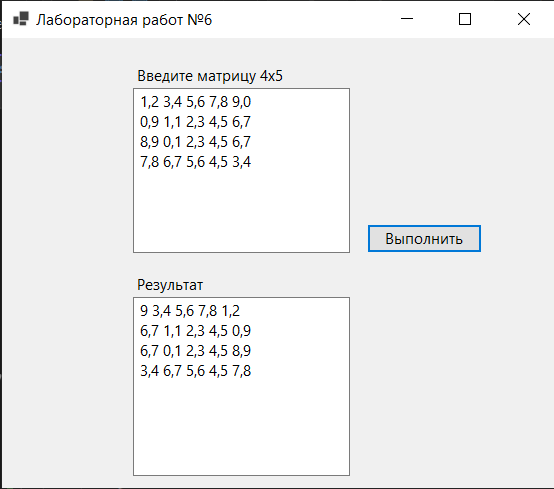


Схема алгоритма - **продолжение**



Задание 2

namespace Lab6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

// Функция для парсинга матрицы из текста

static double[,] ParseMatrix(string text, int rows, int cols)

{

double[,] matrix = new double[rows, cols];

string[] lines = text.Trim().Split('\n');

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

{

string[] numbers = lines[i].Trim().Split(' ');

for (int j = 0; j < cols; j++)

{

matrix[i, j] = double.Parse(numbers[j]);

}

}

return matrix;

}

// Функция для сортировки строк по первому элементу

static void SortRowsByFirstElement(ref double[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

// Преобразуем матрицу в массив строк для удобства сортировки

double[][] rowsArray = new double[rows][];

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

{

rowsArray[i] = new double[cols];

for (int j = 0; j < cols; j++)

{

rowsArray[i][j] = matrix[i, j];

}

}

// Сортируем строки по первому элементу

Array.Sort(rowsArray, (a, b) => a[0].CompareTo(b[0]));

// Возвращаем отсортированные строки в матрицу

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

{

for (int j = 0; j < cols; j++)

{

matrix[i, j] = rowsArray[i][j];

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

string input = textBox1.Text;

int n = int.Parse(textBox3.Text);

int m = int.Parse(textBox4.Text);

double[,] matrix = ParseMatrix(input, n, m);

SortRowsByFirstElement(ref matrix);

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

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

{

for (int j = 0; j < cols; j++)

{

textBox2.Text += matrix[i, j] + " ";

}

textBox2.Text += Environment.NewLine;

}

}

}

}

