МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

Лабораторная работа №11

**«Создание приложений Windows Forms для**

**формирования и обработки двумерных массивов»**

**по дисциплине**

**«Алгоритмизация и программирование»**

Выполнил: студент гр. БЭИ2202 Бояркин Е. В.

Вариант №4

Проверил: доц. Воробейчиков Л. А.

Москва, 2023 г.

**Задание:**

Сформируйте двумерный массив X(m, n) заданных размеров из случайных  
вещественных чисел в заданном диапазоне. Найдите в нем строку, содержащую  
максимальный элемент массива\*. Создайте новый массив Y(m+1, n), в котором  
найденная строка продублирована.

\* Если наибольших (наименьших) значений несколько, возьмите первый  
встретившийся

**Интерфейс:**

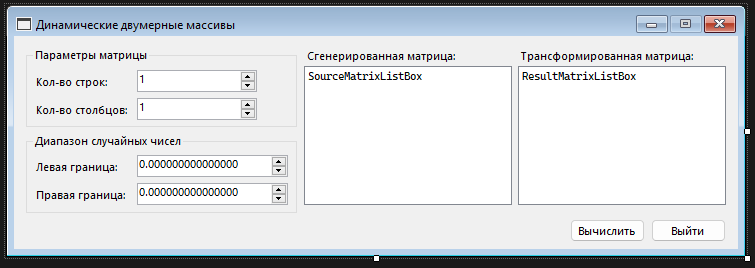


Рисунок 1

**Схема алгоритма:**

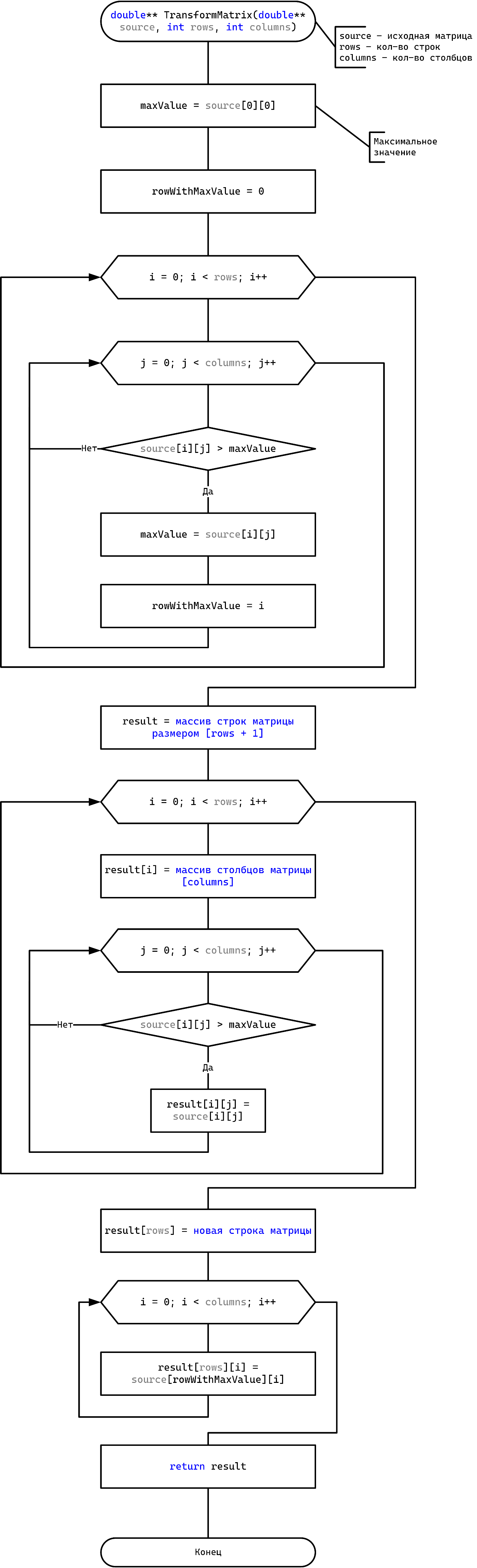


Рисунок 2

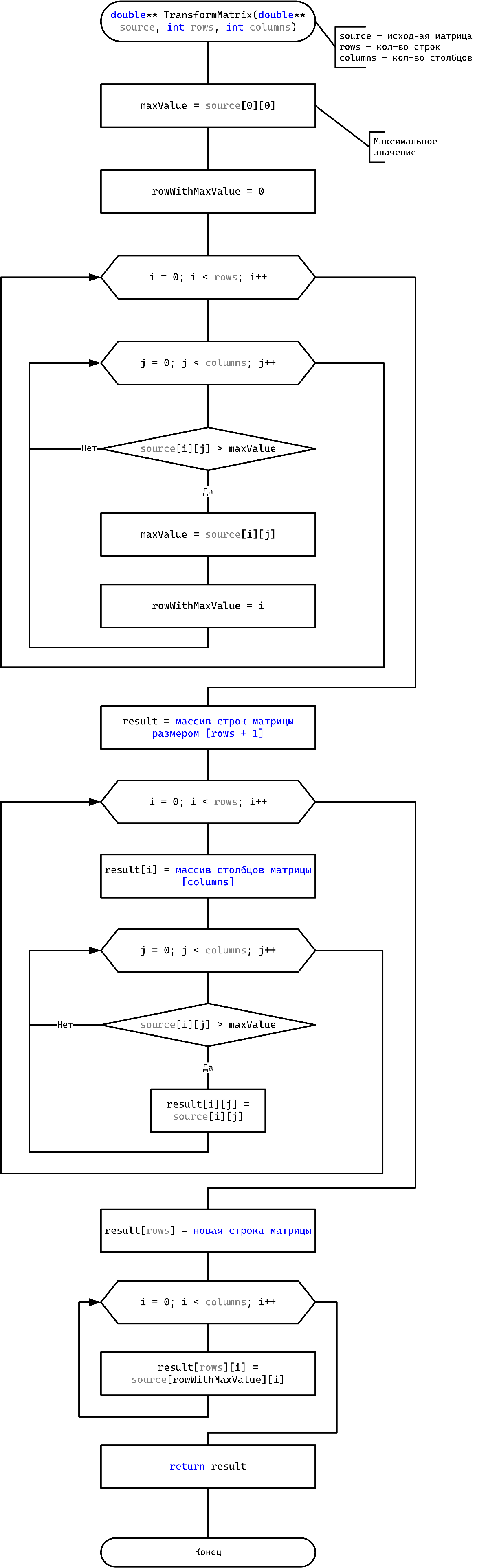


Рисунок 3

**Код:**

|  |
| --- |
| #include "pch.h"  #include "MainForm.h"  using namespace WfApp;  using namespace System;  using namespace System::Windows::Forms;  double\*\* CreateMatrix(int rows, int columns, double rangeStart, double rangeEnd)  {  double\*\* matrix = new double\*[rows];  Random^ random = gcnew Random();  double rangeLength = rangeEnd - rangeStart;  for (int i = 0; i < rows; i++)  {  matrix[i] = new double[columns];  for (int j = 0; j < columns; j++)  matrix[i][j] = random->NextDouble() \* rangeLength + rangeStart;  }  return matrix;  }  double\*\* TransformMatrix(double\*\* source, int rows, int columns)  {  double maxValue = source[0][0];  int rowWithMaxValue = 0;  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < columns; j++)  {  if (source[i][j] > maxValue)  {  maxValue = source[i][j];  rowWithMaxValue = i;  }  }  }  double\*\* result = new double\*[rows + 1];  for (int i = 0; i < rows; i++)  {  result[i] = new double[columns];  for (int j = 0; j < columns; j++)  result[i][j] = source[i][j];  }  result[rows] = new double[columns];  for (int i = 0; i < columns; i++)  result[rows][i] = source[rowWithMaxValue][i];  return result;  }  void OutputMatrix(ListBox^ listBox, double\*\* matrix, int rows, int columns)  {  listBox->Items->Clear();  for (int i = 0; i < rows; i++)  {  String^ s = String::Empty;    for (int j = 0; j < columns; j++)  s += String::Format(L"{0,-25}", matrix[i][j]);  listBox->Items->Add(s);  }  }  void FreeMatrix(double\*\* matrix, int rows)  {  for (int i = 0; i < rows; i++)  delete[] matrix[i];  delete[] matrix;  }  void MainForm::EvaluateButtonClick(Object^ sender, EventArgs^ e)  {  int rows = safe\_cast<int>(RowsCountNumericUD->Value),  columns = safe\_cast<int>(ColumnsCountNumericUD->Value);  double rangeStart = safe\_cast<int>(RangeStartNumericUD->Value),  rangeEnd = safe\_cast<int>(RangeEndNumericUD->Value);  if (rangeStart > rangeEnd)  {  double temp = rangeStart;  rangeStart = rangeEnd;  rangeEnd = temp;  }  double\*\* sourceMatrix = CreateMatrix(rows, columns, rangeStart, rangeEnd);  OutputMatrix(SourceMatrixListBox, sourceMatrix, rows, columns);  double\*\* resultMatrix = TransformMatrix(sourceMatrix, rows, columns);  OutputMatrix(ResultMatrixListBox, resultMatrix, rows + 1, columns);  FreeMatrix(sourceMatrix, rows);  FreeMatrix(resultMatrix, rows + 1);  }  void MainForm::ExitButtonClick(Object^ sender, EventArgs^ e)  {  Application::Exit();  } |

**Результаты выполнения программы:**

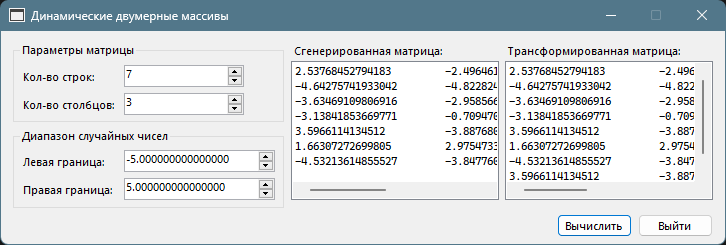


Рисунок 4