**Федеральное государственное образовательное бюджетное учреждение  
высшего образования  
«Финансовый университет при Правительстве РФ»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПРАКТИЧЕСКАЯ РАБОТА №5**

Тема «Создание DLL библиотек в средах программирования Microsoft Visual Studio»

Студент: Адещенко К.Р.\_\_\_\_

Группы: 2ПКС – 316\_\_\_\_\_\_\_\_

Дата: 26.02.18\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель: \_\_*Гуриков С.Р.*

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2018

Задание: разработать динамические библиотеки на языках Си, в которых разместить пользовательские функции для последующей работы с массивами.

Условие: в динамической библиотеке создать функцию для вычисления количества положительных элементов матрицы вещественных чисел, который формируется в пользовательской процедуре. Сгенерировать новыую матрицу в отдельной процедуре из элементов исходной матрицы , значения которых по модулю больше найденного числа. Вызов пользовательских функций и процедур осуществляется в событийной процедуре основного проекта, к которому будет подключена динамическая библиотека.

Далее представлен листинг процедуры, которая устанавливает длину массива и заполняет его элементы случайными числами вещественного типа в диапазоне от -10 до 10:

***Win Forms***

Листинг файла (.h):

#ifdef ARRAYDLL\_EXPORTS

#define ARRAYDLL\_API \_\_declspec(dllexport)

#else

#define ARRAYDLL\_API \_\_declspec(dllimport)

#endif

namespace arrayG

{

class MyArray

{

public:

// Генерация исходного массива

static ARRAYDLL\_API void generateMatrix (double \*\*arr, int n, int m);

// Нахождение кол-ва положительных элементов в исходном массиве

static ARRAYDLL\_API int pos(double \*\*arr, int n, int m, int count);

//Генерация результирующего массива из элементов исходного массива

static ARRAYDLL\_API void newMatrix (double \*\*arr, double \* result, int n, int k,

int &j);

};

}

Листинг файла исходного кода(C++):

#include "header.h"

#include "Array.h"

#include <math.h>

#include <iostream>

namespace arrayG

{

void MyArray:: generateMatrix (double \*\*arr, int n, int m)

{

for (int i = 0; i < n; i++){

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

arr[i][j] = (double)(rand() % 1000) / 15,5;

}

}

int MyArray:: pos (double \*\*arr, int n, int m, int count)

{

for (int i = 0; i < n; i++){

for(int j = 0; j < m; j++){

if (arr[i][j] >= 0) count += 1;

}

return count;

}

void MyArray:: newMatrix(double \*\*arr, double \*\*result, int n, int m, int k, int &j, int &p)

{

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

{

for (int l = 0; l < m; l++)

if (abs(arr[i][l]) > k)

{

result [j][p] = arr[i][l];

j++;

p++;

}

}

}

}

После подключения DLL библиотеки мы можем приступить к разработке программы.

Листинг заполнения массива:

…

void HW3::MyForm::outputArray(double \*\*arr, int n, int m, DataGridView^ grid)

{

grid->ColumnCount = n;

grid->RowCount = m;

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

{

for(int j = 0; j < n; i++){

grid->Rows[i]->Cells[j]->Value = arr[i][j];

}

}

}

Листинг вызова функций из DLL библиотеки:

#include "Array.h"

using namespace Microsoft::VisualBasic;

using namespace arrayG;

..

..

void outputArray double \*arr, int n, int m, DataGridView^ grid);

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int k, j = 0, p = 0, count = 0;

String^ g = Interaction::InputBox("Введите количество элементов в массиве", "Ввод", "", -1, -1);

String^ k = Interaction::InputBox("Введите количество элементов в массиве", "Ввод", "", -1, -1);

int n = Convert::ToInt16(g);

int m = Convert::ToInt16(g);

double\*\* arr = new double\*[n];

double\*\* result = new double\*[n];

for(int i = 0; i < m; i++){

\*arr = new new double[m];

\* result = new double[m];

}

MyArray::generateArray (mas, n, m);

newArray (mas, n, m, dataGridView1);

k = MyArray::pos(mas, n, m, count);

MessageBox::Show("Количество положительных элементов: " + Convert::ToString(k), "", MessageBoxButtons::OK, MessageBoxIcon::Information);

Label1->Text = "Количество положительных элементов: " + Convert::ToString(k);

MyArray::generateArray(arr, result, n,m, k, j);

outputArray (rezmas, j, p, dataGridView2);

}

Запустим программу и введем количество элементов исходного массива, как показано на рисунке 3 и нажмем кнопку «ОК».

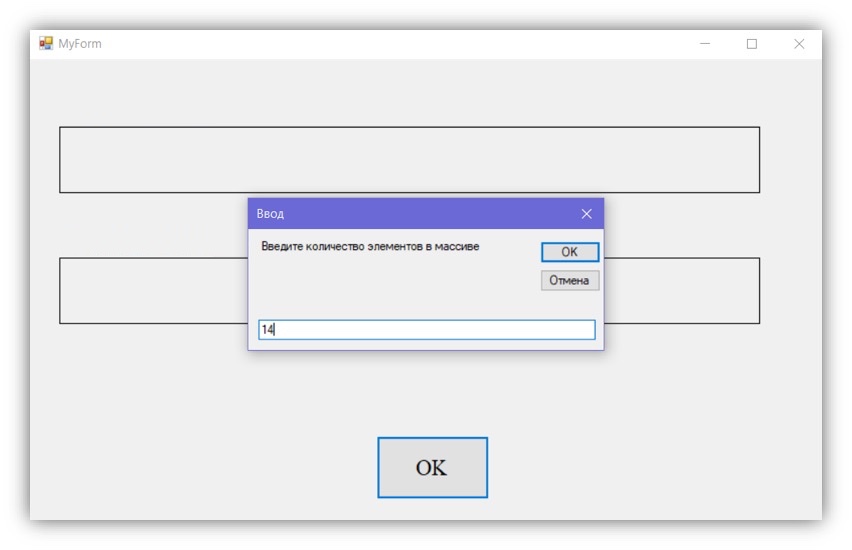


Рисунок 3 - Ввод количества элементов массива

На рисунке 4 представлен результат работы программы, а именно исходный и результирующий массивы вещественных чисел.

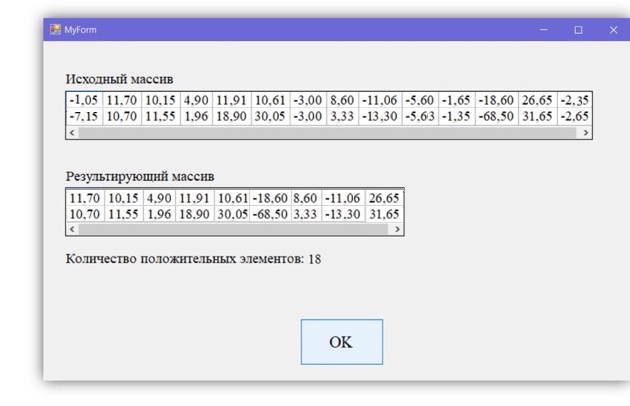


Рисунок 4 - Результат работы программы