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

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПРАКТИЧЕСКАЯ РАБОТА №3**

Тема «Создание DLL библиотек в средах программирования Lazarus и 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 Free Pascal и Си, в которых разместить пользовательские функции для последующей работы с массивами.

Условие: в динамической библиотеке создать функцию для вычисления количества положительных элементов массива вещественных чисел, который формируется в пользовательской процедуре. Сгенерировать новый массив в отдельной процедуре из элементов исходного массива, 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:

**procedure** generateArray (**var** a:mas; n :Integer); StdCall;

**var** i :Integer;

**begin**

SetLength(a, n);

randomize;

**for** i := Low(a) **to** High(a) **do**

a[i] := random(20) – 10;

**end**;

Exports generateArray;

Поиск положительных элементов в исходном массиве:

**function** pos(a :mas) :Integer; StdCall;

**var** i, count :Integer;

**begin**

count := 0;

**for** i := Low(a) **to** High(a) **do**

**if**(a[i] >= 0) **then**

count += 1;

result := count;

**end**;

Exports pos;

Создание нового массива по условию:

**procedure** newArray(a :mas; **var** b :mas; count :Integer); StdCall;

**var** i :Integer;

**begin**

SetLength(b, 0);

**for** i := Low(a) **to** High(a) **do**

**begin**

**if**(abs(a[i]) > count) **then**

**begin**

Setlength(b, length(b) + 1);

b[High(b)] := a[i];

**end**;

**end**;

**end**;

Exports newArray;

Вывод в массив:

**procedure** outputArray(arr :mas; StringGrid :TStringGrid); StdCall;

**var** i :integer;

**begin**

StringGrid.ColCount := 0;

**for** i := Low(arr) **to** High(arr) **do**

**begin**

StringGrid.Cells[i, 0] := IntToStr(i);

StringGrid.Cells[i, 1] := floatToStr(arr[i]);

**end**;

**end**;

Exports outputArray;b

После компиляции, динамическая библиотека была подключена к основному проекту, в котором находится событийная процедура для вызова созданных ранее подпрограмм. Листинг основного проекта представлен ниже:

**Type** mas = **array of** Double;

**implementation**

**procedure** generateArray (**var** a:mas; n :Integer);

**External** 'arraydll.dll' name 'generateArray';

**function** pos (a :mas) :Integer;

**External** 'arraydll.dll' name ' pos ';

**procedure** newArray (a :mas; **var** b :mas; count :Integer);

**External** 'arraydll.dll' name 'newArray';

**procedure** outputArray (arr :mas; StringGrid :TStringGrid);

**External** 'arraydll.dll' name 'outputArray';

**procedure TForm1.Button1Click(Sender: TObject);**

**var** lenArray, counter :Integer;

a, b :mas;

**begin**

Val(InputBox('Введите длину массива, 'Ввод', ''), lenArray);

generateArray (a, lenArray);

counter := pos(a);

Label1.Caption := 'Кол-во положительных элементов: ' + IntToStr(counter);

newArray (a, b, counter);

outputArray (a, StringGrid1);

outputArray (b, StringGrid2);

**end**;

Введем количество элементов массива

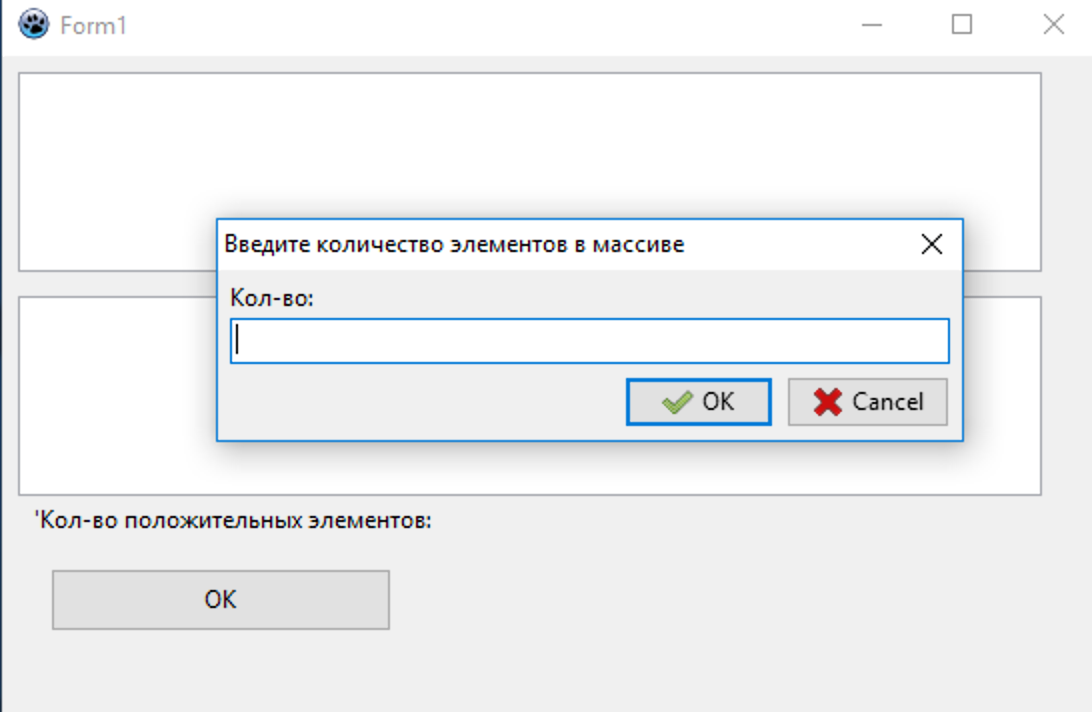


Рисунок 1 - Ввод количества элементов массива

На рисунке 2 представлен результат работы программы

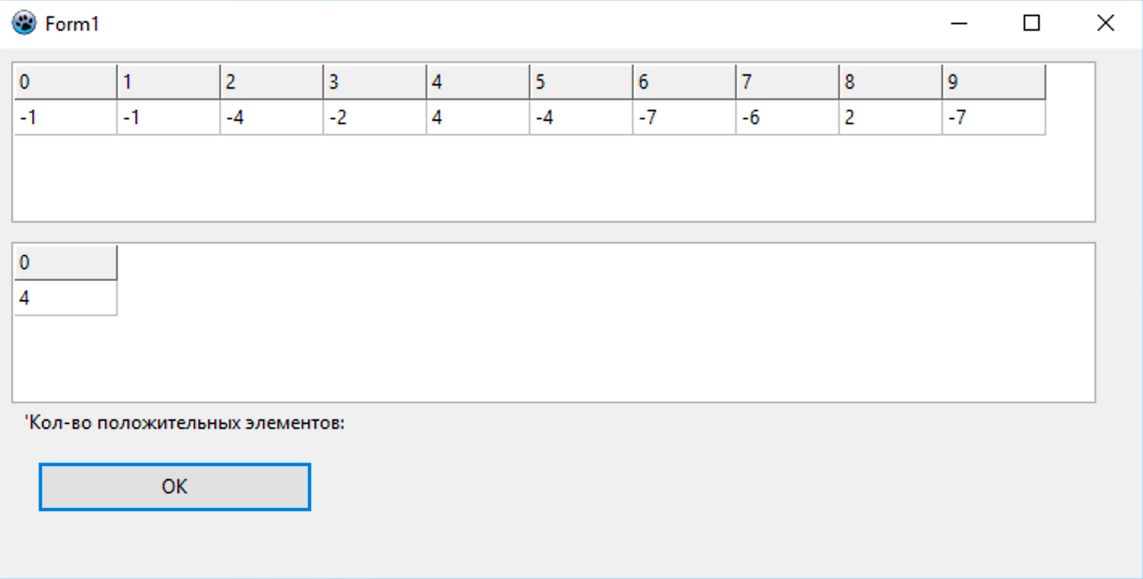


Рисунок 2 - Результат работы программы

***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 generateArray (double \*arr, int n);

// Нахождение кол-ва положительных элементов в исходном массиве

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

//Генерация результирующего массива из элементов исходного массива

static ARRAYDLL\_API void newArray (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:: generateArray (double \*arr, int n)

{

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

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

}

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

{

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

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

return count;

}

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

{

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

{

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

{

result [j] = arr[i];

j++;

}

}

}

}

После подключения DLL библиотеки мы можем приступить к разработке программы.

Листинг заполнения массива:

…

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

{

grid->ColumnCount = len;

grid->RowCount = 2;

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

{

grid->Rows[0]->Cells[i]->Value = i;

grid->Rows[1]->Cells[i]->Value = arr[i];

}

int sum = 0;

for (int s = 0; s < grid->ColumnCount; ++s)

sum += grid->Columns[s]->Width;

if (sum > 730) grid->Width = 730;

else grid->Width = sum;

}

Листинг вызова функций из DLL библиотеки:

#include "Array.h"

using namespace Microsoft::VisualBasic;

using namespace arrayG;

..

..

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

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

int k, j = 0, count = 0;

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

int n = Convert::ToInt16(g);

double\* arr = new double[n];

double\* result = new double[n];

MyArray::generateArray (mas, n);

newArray (mas, n, dataGridView1);

k = MyArray::pos(mas, n, 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, k, j);

outputArray (rezmas, j, dataGridView2);

}

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

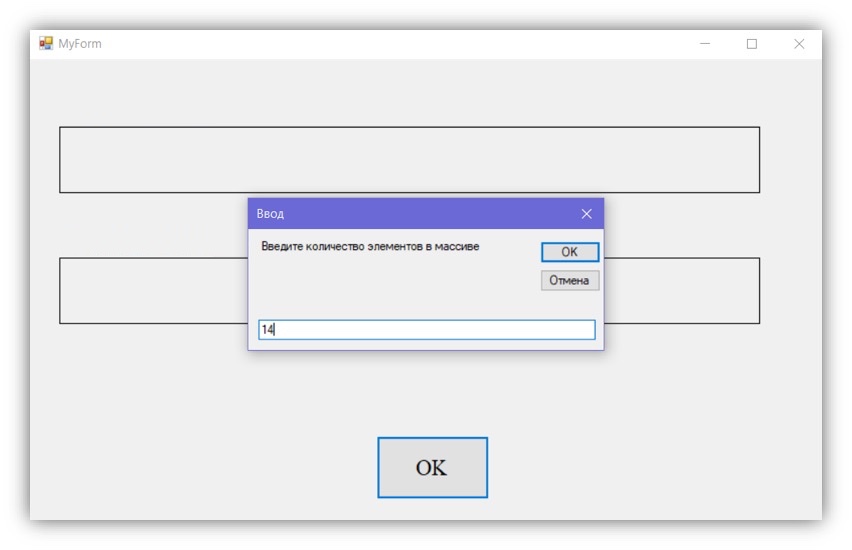


Рисунок 3 - Ввод количества элементов массива

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

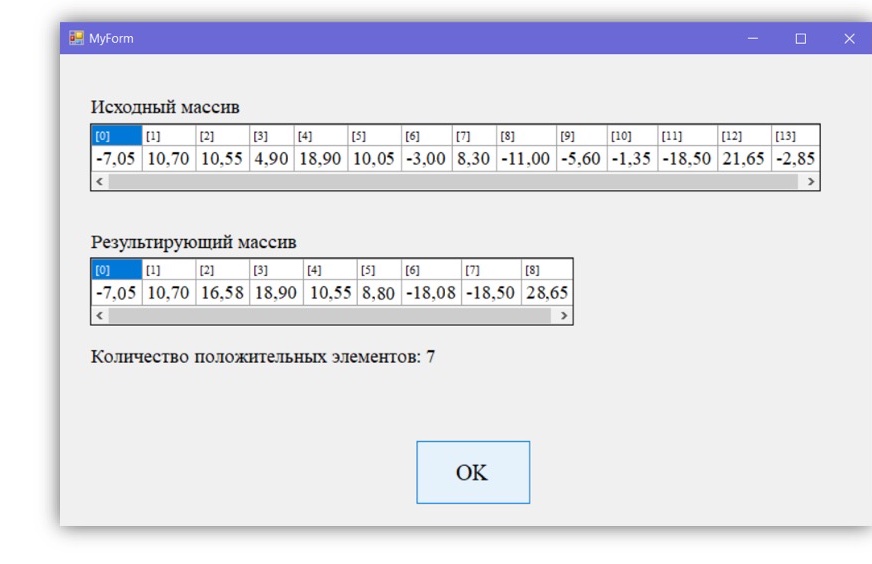


Рисунок 4 - Результат работы программы