**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра «Информатики»

**Отчет по лабораторной работе**

по дисциплине «Информатика»

на тему:

«Разработка программы «Построитель матрицы»»

Выполнил: студент группы

БВТ1804

Морочко Александр Александрович

Руководитель:

Волков Андрей Иванович

Москва 2018

Оглавление

[1 Постановка задачи 3](#_Toc534817829)

[2 Таблица используемых элементов 4](#_Toc534817830)

[3 Схемы алгоритмов 5](#_Toc534817831)

[4 Текст программы 8](#_Toc534817832)

[5 Результаты тестирования 9](#_Toc534817833)

[Заключение 10](#_Toc534817834)

# 1 Постановка задачи

Требуется создать в среде разработки Visual Studio программу на языке программирования C++ с помощью оконного интерфейса, которая будет представлять собой построитель матрицы из массива n, который задается пользователем в текстовом поле, по заданному образцу, указанному в задании. Приложение должно иметь отдельное текстовое поле для ввода пользователем с клавиатуры массива, а также отдельную кнопку, которая будет отвечать за построение самой матрицы и текстовое поле, куда будет выводиться сама матрица, без возможности изменения текста в нем, с множеством строк. Расширение размера окна будет влиять на размер текстового поля, для того чтобы пользователю было комфортнее видеть результат. Полученная матрица должна быть отформатирована, то есть все полученные столбцы должны быть выровнены, в виде корректной записи чисел (разряд под разрядом). Программа должна корректно обрабатывать ошибки введённых данных (выдавая соответствующее сообщение в отдельное окно для ошибок).

# 2 Таблица используемых элементов

Элементы, используемые в программе, представлены в таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Объект** | **Имя объекта** | **Свойство** | **Значение свойства** |
| Форма | MyForm | Text | Индивидуальные задания |
| Метка | lblPor | Text | Размер массива |
| Текстовое поле | txtPor | Text |  |
| Текстовое поле | txtMatrix | Text |  |
| Кнопка | btnBuild | Text | Построить |
| Текстовое поле | nlen | Text |  |
| Текстовое поле | nmin | Text |  |
| Текстовое поле | nmax | Text |  |

Таблица 1 – Используемые элементы

# 3 Схемы алгоритмов

На рисунке 1 представлена схема алгоритма, отвечающего за генерацию массива.



Рисунок 1 – Схема алгоритма "nGen"

На рисунке 2 представлена схема алгоритма, отвечающего за построение матрицы.







Рисунок 2 – Схема алгоритма "btnBuild"

# 4 Текст программы

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

txtmassivn->Text = "";

int a, b;

if ((nlen->Text == "" || Convert::ToInt32(nlen->Text) == 0)) {

MessageBox::Show("Введите размер массива", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

}

else {

if (nlen->Text != "") {

n = Convert::ToInt32(nlen->Text);//берем количество из txtNum

}

if (nmin->Text != "") {

a = Convert::ToInt32(nmin->Text);//мин

}

if (nmax->Text != "") {

b = Convert::ToInt32(nmax->Text);//макс

}

if (nmin->Text == "") a = 0;

if (nmax->Text == "") b = 0;

if (nlen->Text == "") n = 0;

String^ tempn;

Random^ randn = gcnew Random();

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

tempn += Convert::ToString(randn->Next(a, b + 1)) + " ";

//выводим рандомное число

} //в txtNum

txtmassivn->Text = tempn->Remove(tempn->Length - 1);

}

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

int n, n1;

String^ tor = Convert::ToString(txtmassivn->Text);

txtMatrix->Text = "";

if (txtmassivn->Text->Length == 0) {

MessageBox::Show("Введите размер массива", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

txtmassivn->Focus();

return;

}

array<String^>^ stringsn = txtmassivn->Text->Split(' ');

Converter<String^, Int64>^ convertern =

gcnew Converter<String^, Int64>(Convert::ToInt64);

array<String^>^ cleanedn = gcnew array<String^>(stringsn->Length);

int indexn = 0;

for each(String^ sn in stringsn) {

if (!String::IsNullOrEmpty(sn)) {

cleanedn[indexn++] = sn;

}

}

Array::Resize(cleanedn, indexn);

array<Int64>^ massivn;

massivn = Array::ConvertAll(cleanedn, convertern);

txtmassivn->Text = "";

for (int i = 0; i < cleanedn->Length; ++i) {

txtmassivn->Text += Convert::ToString(cleanedn[i]) + " ";

}

int count = 0;

nlen->Text = Convert::ToString(massivn->Length);

Int64 max = 0, min = 100000;

for (int i = 0; i < massivn->Length; ++i) {

if (massivn[i] < min)

min = massivn[i];

if (massivn[i] > max)

max = massivn[i];

}

nmin->Text = Convert::ToString(min);

nmax->Text = Convert::ToString(max);

n = Convert::ToInt32(nlen->Text);

int \*\*matrix = new int\*[n];//объявление двумерного массива

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

matrix[i] = new int[n];//память для массива

//заполнение двумерного массива

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

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

matrix[k][t] = pow(massivn[t], k+1);

}

}

int a;

int maxnn = abs(matrix[0][0]);

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

{

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

{

if (abs(matrix[i][j]) > maxnn) maxnn = abs(matrix[i][j]);

}

}

String^ temp1;

long int Length;

bool zap;

String^ temp2 = "";

String^ formatString;

int Length1 = (Convert::ToString(maxnn)) -> Length;

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

{

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

{

zap = false;

temp1 = Convert::ToString(matrix[i][j]);

Length = temp1->Length;

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

{

if (temp1[i] == '-') Length--;

}

temp2 = Convert::ToString(((2 \* (Convert::ToString(maxnn)->Length) + 1) - Length));

txtMatrix->Text += String::Format("{0," + temp2 + ":0}", matrix[i][j]);

}

txtMatrix->Text += Environment::NewLine;

}

txtmassivn->Text = tor;

}

}

# 5 Результаты тестирования

На таблице 2 представлены результаты тестирования программы.

Таблица 2 – Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Длина массива** | **Мин** | **Макс** | **Линейный массив n** | **Матрица** |
| 5 | 2 | 7 | 7 3 3 6 2 | 7 3 3 6 2  49 9 9 36 4  343 27 27 216 8  2401 81 81 1296 16  16807 243 243 7776 32 |

На рисунке 2 показана корректная работа программы.

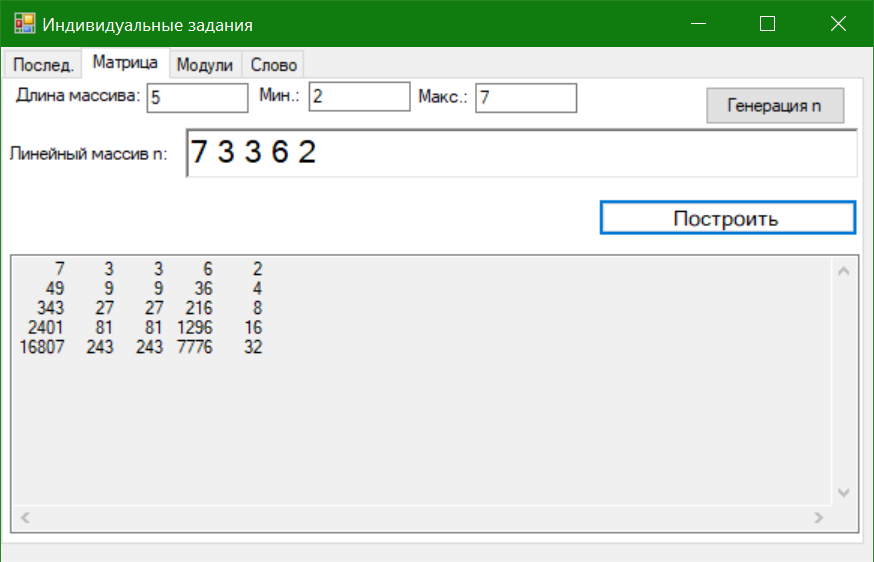


Рисунок 2 – Работа программы

# Заключение

Итак, опираясь на результаты тестирования, можно сделать вывод, что я научился создавать в среде разработки Visual Studio на языке программирования C++ программу с помощью оконного интерфейса, которая представляет собой построитель квадратной матрицы, корректно обрабатывающее введенные пользователем данные, и выполняющее все поставленные задачи.