Лабораторная работа №2

**«Одномерные массивы»**

**1) Индивидуальное задание на разработку проекта**

Разработать программный код решения следующей задачи:

Дан динамический вещественный массив положительных и отрицательных чисел. Найдите количество элементов, находящихся в диапазоне от 2 до 10, разработав соответствующий метод. Разработайте метод, который формирует новый массив, из тех элементов исходного массива, значения которых меньше найденного количества элементов, находящихся в диапазоне от 2 до 10.

**2) Техническое задание**

1. Разработать алгоритмы решения заданной задачи (количество алгоритмов равно количеству используемых пользовательских функций + алгоритм событийной). Кроме того, должна быть выполнена схема иерархии подпрограмм.

2. Добавить в существующую DLL-библиотеку функции, решающие предписанные задачи. Обмен данными между подпрограммами должен осуществляться через параметры, без использования глобальных переменных. Событийная процедура должна содержать только операторы вызова пользовательских подпрограмм.

3. В ходе решения задачи своего варианта предусмотреть написание функций для генерации массива, поиска максимального положительного элемента в массиве, создания нового массива и вывода массива. Для решения задачи своего варианта использовать многочисленные методматериалы, которые перечислены в пояснениях (ЭОИС).

4. Подготовить тесты для контрольного решения задачи. Показать их использование, отобразив в отчете по заданию.

5. Выполнить проект и получить результаты, которые отобразить в виде рисунков при написании отчета по заданию.

6. Выполнить отчет в соответствии с ГОСТ на схемы алгоритмов и согласно ГОСТ оформления отчетов по НИР.

**3) Схемы алгоритмов**

Алгоритмы ввода/вывода представлены на рисунках 1, 2. Для реализации данных функций использовались встроенные методы преобразования чисел: Convert::ToDounle() для преобразования строки в вещественное число, а также Convert::ToString() для преобразования числа в строку.

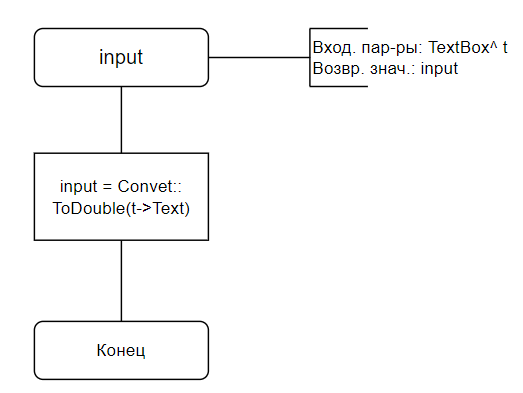


Рисунок 1 – Функция ввода чисел из TextBox

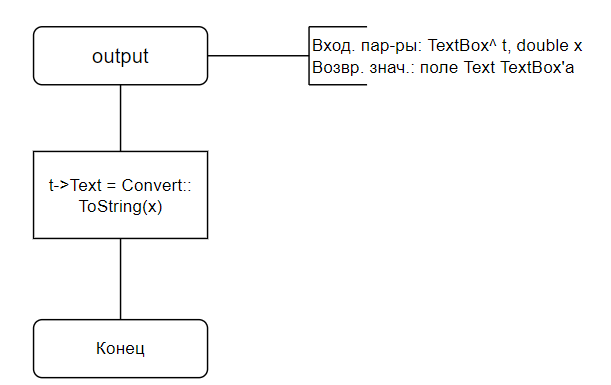


Рисунок 2 – Функция вывода чисел в TextBox

Для функции, решающей поставленную задачу, были написаны вспомогательные функции: функция генерации случайного массива gen\_mas(), алгоритм которой представлен на рисунке 3; функция подсчёта числа чисел в нужном диапазоне count\_in\_bounds(), алгоритм которой представлен на рисунке 4. Математические функции были взяты из математического пакета Math.h.

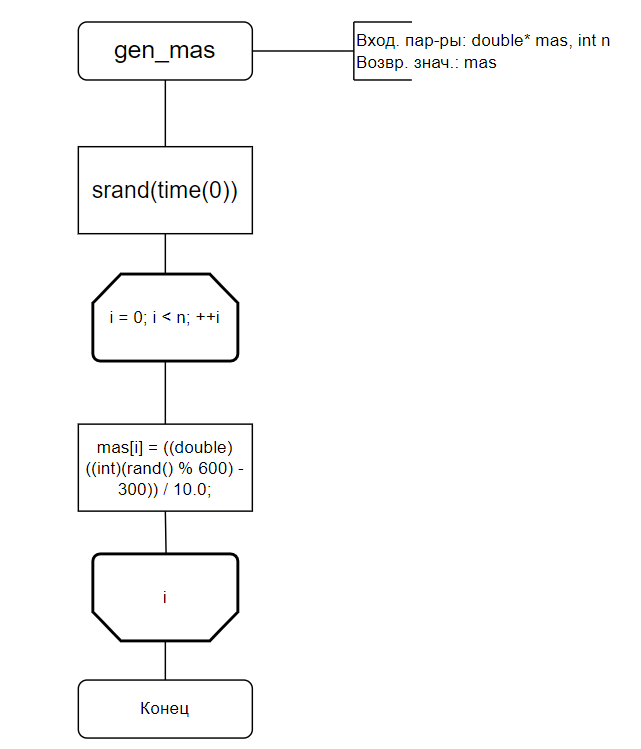


Рисунок 3 – Функция генерации случайного массива

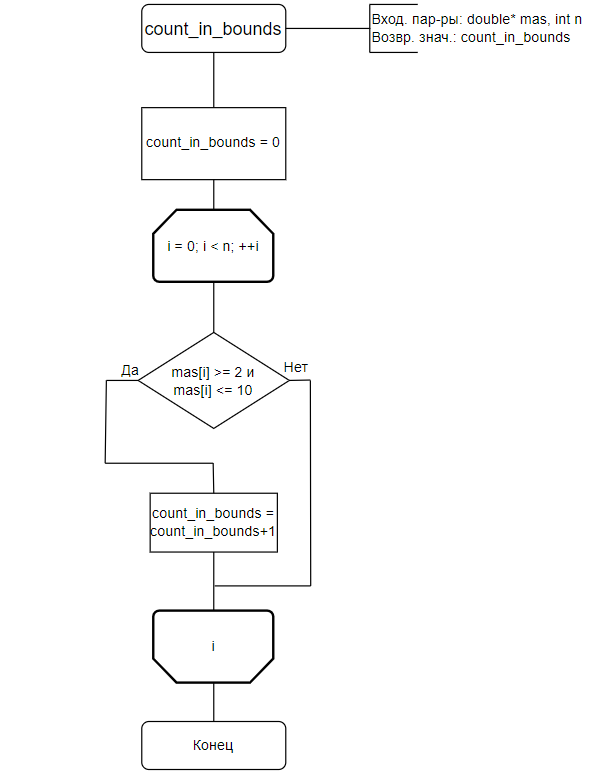


Рисунок 4 – Функция подсчёта количества элементов массива в диапазоне

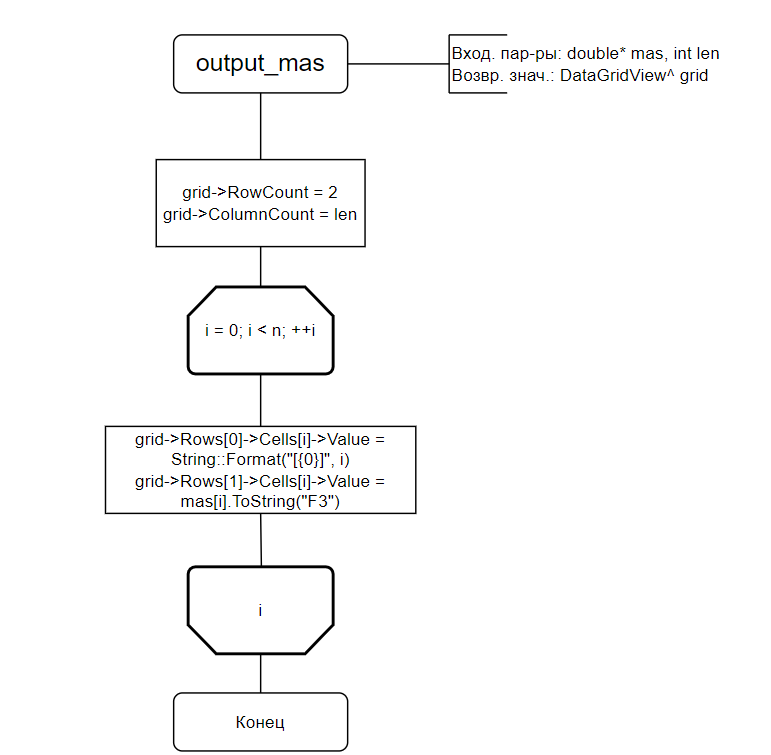


Рисунок 5 – Функция вывода массива в DataGridView

Алгоритм функции, решающей поставленную задачу, представлен на рисунке 6.

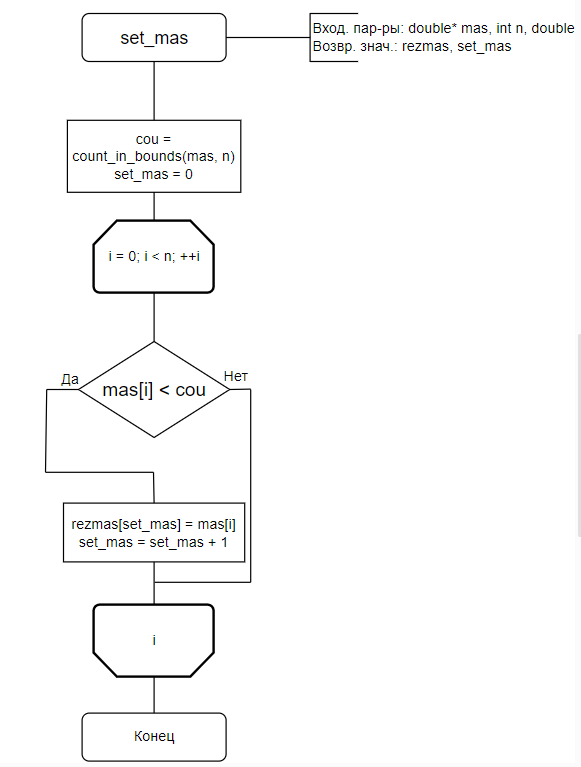


Рисунок 6 – Функция создания нового массива

Алгоритм событийной процедуры представлен на рисунке 6. В алгоритме показаны обработка ошибки ввода пользователем значения неподходящего для вычислений.

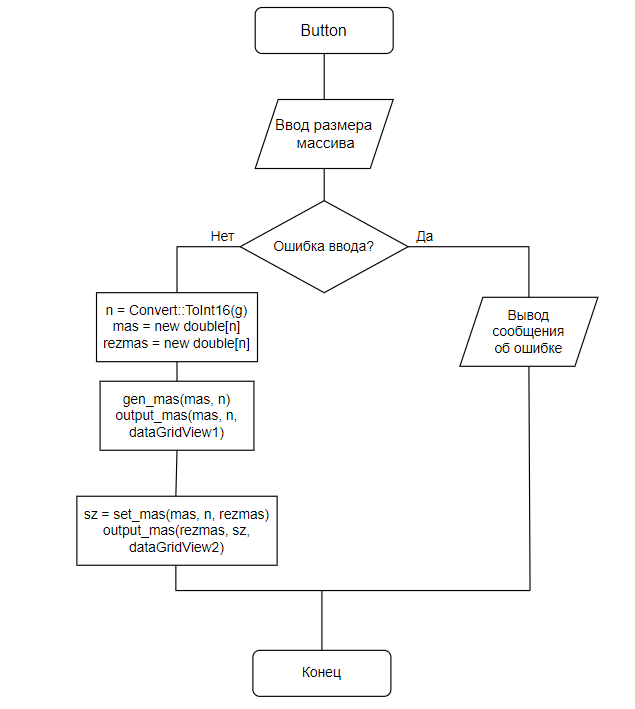


Рисунок 7 – Алгоритм событийной процедуры

**4) Программный код DLL-библиотеки**

В ходе создания DLL-библиотеки необходимо было написать код в двух файлах: заголовочном (.h) и исходном (.cpp).

Программный код заголовочного файла представлен ниже.

#pragma once

using namespace System;

using namespace System::Windows::Forms;

namespace dll\_kuleshov\_techprog\_sem3\_bei2202 {

public ref class Functions\_Class

{

public:

// ЛР #1

static double input(TextBox^ t);

static void output(double x, TextBox^ t);

static double get\_Radius(double L);

static double get\_Area(double L);

static void get\_Area\_mult(double L1, double& S1, double L2, double& S2, double L3, double& S3);

// ЛР #2

static void gen\_mas(double\* mas, int n);

static void output\_mas(double\* mas, int len, DataGridView^ grid);

static int count\_in\_bounds(double\* mas, int n);

static int set\_mas(double\* mas, int n, double\* rezmas);

};

}

Программный код исходного файла со всеми функциями представлен ниже.

#include "pch.h"

#include "dll\_kuleshov\_techprog\_sem3\_bei2202.h"

#include "Math.h"

#include <cstdlib>

#include <time.h>

using namespace System::Windows::Forms;

using namespace System;

using namespace System::IO;

using namespace System::Data::OleDb;

using namespace System::Runtime::InteropServices;

namespace dll\_kuleshov\_techprog\_sem3\_bei2202 {

double Functions\_Class::input(TextBox^ t) {

return Convert::ToDouble(t->Text);

}

void Functions\_Class::output(double x, TextBox^ t) {

t->Text = Convert::ToString(x);

}

double Functions\_Class::get\_Radius(double L) {

return L / (2 \* Math::PI);

}

double Functions\_Class::get\_Area(double L) {

double S = Math::PI \* Math::Pow(get\_Radius(L), 2);

return S;

}

void Functions\_Class::get\_Area\_mult(double L1, double& S1, double L2, double& S2, double L3, double& S3) {

S1 = Functions\_Class::get\_Area(L1);

S2 = Functions\_Class::get\_Area(L2);

S3 = Functions\_Class::get\_Area(L3);

}

// ЛР №2

void Functions\_Class::gen\_mas(double\* mas, int n) {

srand(time(0));

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

mas[i] = ((double)((int)(rand() % 600) - 300)) / 10.0;

}

}

int Functions\_Class::count\_in\_bounds(double\* mas, int n) {

int count = 0;

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

if (mas[i] >= 2 && mas[i] <= 10) {

count++;

}

}

return count;

}

void Functions\_Class::output\_mas(double\* mas, int len, DataGridView^ grid) {

grid->RowCount = 2;

grid->ColumnCount = len;

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

{

grid->Rows[0]->Cells[i]->Value = String::Format("[{0}]", i);

grid->Rows[1]->Cells[i]->Value = mas[i].ToString("F3");

}

}

int Functions\_Class::set\_mas(double\* mas, int n, double\* rezmas) {

int cou = count\_in\_bounds(mas, n);

int j = 0;

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

if (mas[i] < cou) {

rezmas[j] = mas[i];

j++;

}

}

return j;

}

}

**5) Программный код событийной процедуры**

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

String^ g = Microsoft::VisualBasic::Interaction::InputBox("Введите количество элементов в массиве",

"Ввод", "", -1, -1);

int n;

double\* mas;

double\* rezmas;

try {

n = Convert::ToInt16(g);

mas = new double[n];

rezmas = new double[n];

}

catch (System::FormatException^ e) {

MessageBox::Show("Ошибка при вводе данных", "Сообщение", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

Functions\_Class::gen\_mas(mas, n);

Functions\_Class::output\_mas(mas, n, dataGridView1);

int sz = Functions\_Class::set\_mas(mas, n, rezmas);

Functions\_Class::output\_mas(rezmas, sz, dataGridView2);

label3->Text = "Количество чисел в диапазоне[2; 10]: " + Functions\_Class::count\_in\_bounds(mas, n);

delete[] mas;

delete[] rezmas;

}

**6) Результаты работы программы**

На рисунке 8 представлен результат работы программы при выдуманном размере массива. Как видно, программа работает корректно.

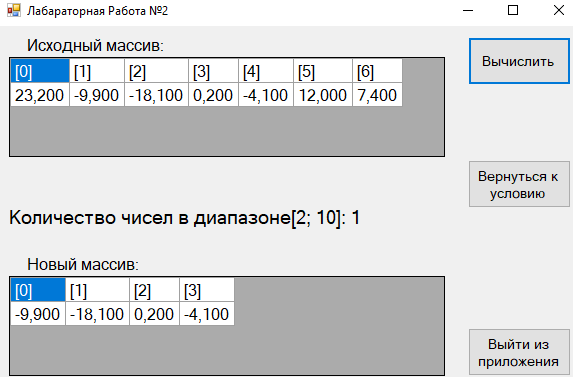


Рисунок 8 – Результаты работы при тестовых данных

Попробуем ввести неподходящие данные, например, оставлю пустое поле. См рис. 9.

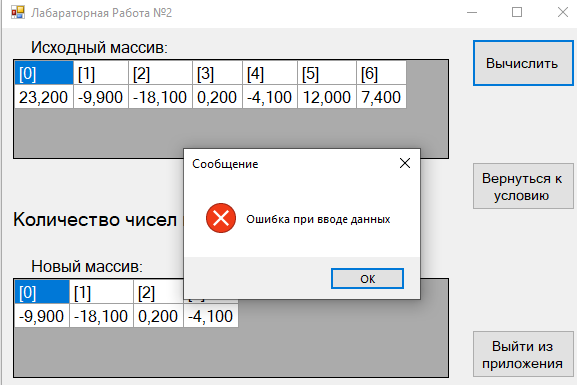


Рисунок 9 – Результат при некорректно введенных данных

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Гуриков С. Р. Основы алгоритмизации и программирования на Visual C++ : учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2022. — 515 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1039154. – ISBN 978-5-16-015500-5. – Текст: электронный. – URL: https://znanium.com/catalog/product/1039154 – Режим доступа: по подписке.