Московский технический университет связи и информатики

Факультет СиСС

Кафедра инфокоммуникационных технологий и систем связи

Лабораторная работа №4

«**Создание приложений Windows Forms для формирования и обработки двумерных массивов»**

По информатике

17 вариант

Выполнил работу

Столяревский Илья,

студент группы БИН 2009

Москва, 2020

**Общее задание**

1) Изучите вопросы использования указателей и программирования алгоритмов формирования и обработки двумерных динамических массивов.

2) Выберите вариант задания из таблицы 1.

3) Разработайте проект графического интерфейса пользователя.

4) Решите поставленную задачу формирования и обработки динамических массивов. Для этого разработайте схемы алгоритмов и программные коды следующих функций:

· функции ввода исходных данных, предусмотрев контроль значений исходных данных;

· функции формирования динамического двумерного массива из случайных чисел в заданном диапазоне;

· функции обработки двумерного динамического массива согласно вашему индивидуальному заданию. При необходимости разбейте эту функцию на ряд вспомогательных функций. Если по заданию требуется

создать новый массив, то предусмотрите выделение памяти под него по необходимому минимуму;

· функции вывода динамического двумерного массива в элемент управления ListBox.

5) Разработайте программный код событийной процедуры, которая вызывает описанные выше функции для решения поставленной задачи.

6) Создайте проект приложения Windows Form, содержащий:

· графический интерфейс;

· файлы с раздельно откомпилированными программными кодами функций, необходимых для решения задачи:

§ функций ввода исходных данных, формирования исходного массива и вывода результатов;

§ функций, решающих поставленную задачу.

· заголовочный файл с прототипами разработанных функций;

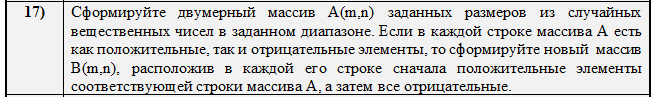
· событийную процедуру, которая должна содержать только операторы вызова пользовательских функций.

7) Подготовьте тесты для контрольного решения задачи, предусмотрев различные значения исходного массива данных для решения. Для проверки работы функции создания нового массива при различных значениях исходного массива рекомендуется в событийной процедуре проинициализировать исходный массив списком подготовленных значений и выполнить проект для этих тестовых значений, а затем удалить (закомментировать) инициализацию массива списком.

8) Выполните проект и получите результаты.

9) Докажите правильность полученных результатов на заранее разработанных тестах.

**Вариант индивидуального задания**

****

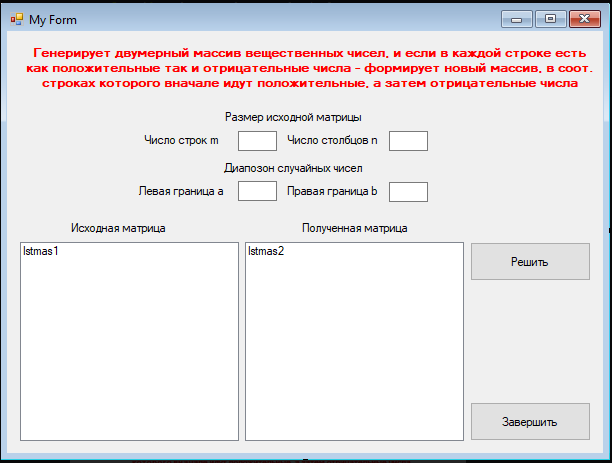
****

Рисунок 1 – Графический интерфейс

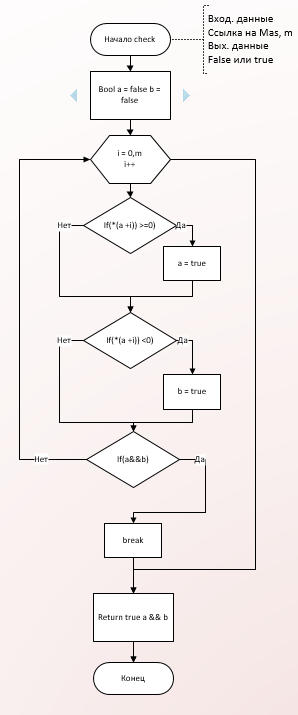


Рисунок 2 – схема алгоритма функции проверки

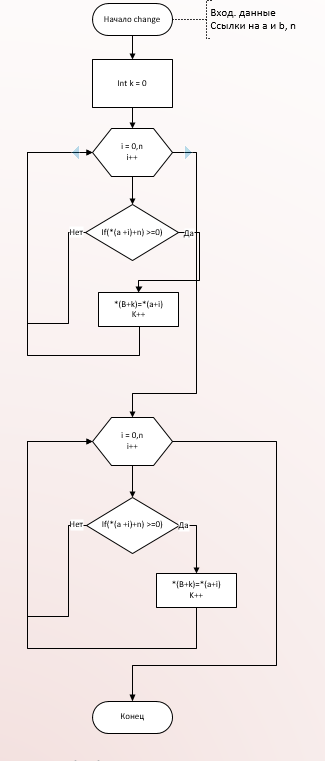


Рисунок 3 – схема алгоритма функции, преобразовывающей строки

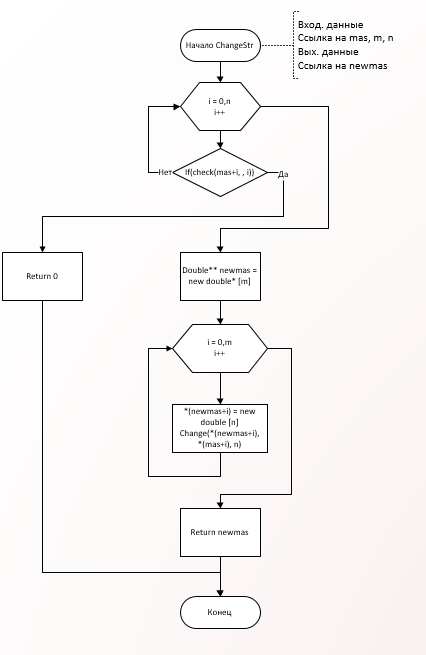


Рисунок 4 – схема алгоритма функции, выполняющей задание

**Программный код:**

**Task.cpp**

#include "task.h"

//Проверка строки на положительные и отрицательные элементы

bool check(double\* mas, int m) {

bool a = false, b = false;

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

if (\*(mas + i) >= 0) a = true;

if (\*(mas + i) < 0) b = true;

if (a && b) break;

}

return a && b;

}

//Преобразование строки

void change(double\* b, double\* a, int n) {

int k = 0;

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

if(\*(a + i) >= 0){

\*(b + k) = \*(a + i);

k++;

}

}

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

if (\*(a + i) < 0) {

\*(b + k) = \*(a + i);

k++;

}

}

}

//Функция решения задачи

double\*\* task(double\*\* mas, int m, int n) {

//Проверка всей матрицы

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

if (!check(\*(mas+i), n)) return 0;

}

//Создание новой матрицы

double\*\* newmas = new double\* [m];

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

\*(newmas + i) = new double[n];

change(\*(newmas + i), \*(mas + i), n);

}

return newmas;

}

**GetPut.cpp**

#include "task.h"

//Генерация матрицы

double\*\* input(int m, int n, int a, int b) {

double\*\* matr = new double\* [m];

if (a > b) { int t = a; a = b; b = t; }

Random^ rnd = gcnew Random;

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

\*(matr + i) = new double[n];

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

//Генерация вещественного числа

\*(\*(matr + i) + j) = int((a + (b - a) \* rnd->NextDouble()) \* 10) / 10.0;

}

}

return matr;

}

//Вывод матрицы

void output(double\*\* mas, int m, int n, ListBox^ LB) {

LB->Items->Clear();

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

String^ s = "";

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

double tmp = \*(\*(mas + i) + j);

s += tmp.ToString() + " ";

}

LB->Items->Add(s);

}

}

//Считывание целых значений

bool GetI(int& x, TextBox^ TB, String^ S) {

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

MessageBox::Show(S, "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

TB->Focus();

return false;

}

x = Convert::ToInt32(TB->Text);

return true;

}

//Отчистка памяти от матрицы

void del(double\*\* a, int m) {

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

delete[] \* (a + i);

}

delete[] a;

}

**Код событийных процедур**

#pragma endregion

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

this->Close();

}

//Событийная функция нажатия кнопки

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

lstmas1->Items->Clear();

lstmas2->Items->Clear();

int m, n, a, b;

if (!GetI(m, txtm, "Введите число строк матрицы")) return;

if (!GetI(n, txtn, "Введите число столбцов матрицы")) return;

if (!GetI(a, txta, "Введите левую границу диапазона")) return;

if (!GetI(b, txtb, "Введите правую границу диапазона")) return;

double\*\* mas1 = input(m, n, a, b);

output(mas1, m, n, lstmas1);

double\*\* mas2 = task(mas1, m, n);

if (mas2) {

output(mas2, m, n, lstmas2);

del(mas2, m);

}

else MessageBox::Show("Новый массив не создан");

del(mas1, m);

}

**Результат выполнения**

