**Санкт-Петербургский государственный университет телекоммуникаций им.проф. М.А.Бонч-Бруевича**

**Факультет информационных систем и технологий**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Курсовая работа**

**по дисциплине «Алгоритмизация и программирование»**

Выполнил

студ.1 курса

направление подготовки 210700

\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_ 2014

Принял

\_\_\_\_\_\_\_\_\_\_\_\_\_

оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_ 2014

Санкт-Петербург, 2014

**Аннотация**

Целью проделанной работы было научиться применять полученные знания для самостоятельного решения на ПК конкретных практических задач.

По полученному заданию была спроектирована и написана программа, выполняющая требующиеся в задании действия. После этого был написан отчёт о проделанной работе.

В процессе работы были получены как навыки в разработке программ, так и опыт по части подготовки курсовых работ и отчётов по ним.

**Оглавление**

Аннотация ………………………………………………………………………….. 2

Оглавление ………………………………………………………………………..... 3

Постановка задачи …………………………………………………………………. 4

Формализация задачи ................................................................................................ 5

Блок-схема алгоритма ……………………………………………………………... 6

Текст программы …………………………………………………………………... 7

Руководство пользователя ……………………………………………………..… 10

Заключение ............................................................................................................... 12

**Постановка задачи**

Текст задания:

# Вариант № 25

Разработать блок-схему алгоритма программы и программу, которая:

а) вводит с клавиатуры или читает из файла двумерный массив вещественных чисел размерностью [2 × 6] элементов и выводит его на экран в виде матрицы. При выводе все вещественные числа ограничены 4-мя знаками после запятой. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит в зависимости от запроса поиск *max* или *min* элемента указанной строки матрицы и результат выдает на экран;

в) по окончании работы сохраняет массив в файле. Данные этого массива сохраняются в виде матрицы, размерностью [2 × 6] элементов с двумя пробелами между ними, а результаты поиска *max* или *min* элемента указанных строк матрицы и их номера – в другом файле. В этом же файле сохраняется результат определения самого максимального и самого минимального из найденных элементов в указанных строках.

Таким образом, требуется составить по заданию блок-схему, иллюстрирующую принцип работы программы, и на её основе написать саму программу в любой удобной среде.

**Формализация задачи**

Для начала нужно подключить необходимые библиотечные модули и объявить глобальные переменные, а также определить параметры массива и дать им характеристику. Только после этого можно начинать проектировать «скелет» программы.

Предполагается что программа будет работать в два этапа, соответственно будет два последовательных меню, реализованных с помощью оператора «switch».

Первое меню будет включать в себя два пункта:

1. Ввод массива с клавиатуры
2. Чтение массива из файла

Второе меню уже четыре:

1. Поиск минимального элемента строки
2. Поиск максимального элемента строки
3. Запись данных в файл
4. Выход

**Блок-схема алгоритма**

Неправильная команда

for(;;)

Меню

Чтение из файла

Ввод массива вручную

default

1

2

Начало

matrix[2][6], answer\_1, answer\_2

Меню

for(;;)

3

4

2

1

Поиск минимального

элемента

Запись в файлы

Поиск максимального

элемента

Конец

**Текст программы**

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

float keybard\_enter(float matrix[2][6])

{

int i, j;

for (i = 0; i < 2; i++) {

for (j = 0; j < 6; j++) {

cout << "\nВведите элемент массива:" << " X[" << i << "][" << j << "] = ";

cin >> matrix[i][j];

}

}

return matrix[2][6];

}

float file\_enter(float matrix[2][6])

{

int i, j;

ifstream F("matrix.txt", ios::in);

for (i = 0; i < 2; i++)

for (j = 0; j < 6; j++)

F >> (matrix[i][j]);

F.close();

cout << endl;

return matrix[2][6];

}

void out\_matrix\_display(float matrix[2][6]) {

int i, j;

cout << "Массив:\n";

for (i = 0; i < 2; i++) {

for (j = 0; j < 6; j++) {

cout << fixed << setprecision(4) << matrix[i][j] << " ";

}

cout << endl;

}

}

void out\_matrix\_file(float matrix[2][6]) {

int i, j;

ofstream F("end\_matrix.txt", ios::app);

F << "Конечная матрица:";

F << "\r";

for (i = 0; i < 2; i++) {

F << "\n";

for (j = 0; j < 6; j++) {

F << fixed << setprecision(4) << matrix[i][j] << " ";

}

F << "\r";

}

F << endl << endl;

F.close();

cout << "\nФайлы успешно созданы!\n\n";

}

void min\_el\_matrix(float matrix[2][6]) {

int i, j;

float min = INT\_MAX;

for (;;) {

int N = 0;

cout << "Введите номер столбца, в которой нужно найти минимальный элемент:\n";

cin>>N;

if (N > 0 && N <= 2) {

for (i = 0; i < 2; i++) {

for (j = 0; j < 6; j++) {

if (matrix[N - 1][j] < min)

min = matrix[N - 1][j];

}

}

ofstream F("min\_max.txt", ios::app);

F << "Минимальный элемент " << fixed << setprecision(4) << N << " строки:\n\n" << min << "\n";

F << "\r\n";

F.close();

cout << "Минимальный элемент " << N << " строки:\n\n";

cout << min << endl;

cout << "\nНайти минимальный элемент в другой строке? (Y/N)\n\n";

char ch;

cin>>ch;

if (ch != 'Y') {

break;

}

} else {

cout << "Ошибка! Такой строки нет. Повторите ввод.\n\n";

}

}

}

void max\_el\_matrix(float matrix[2][6]) {

int i, j;

float max = 0;

for (;;) {

int N = 0;

cout << "\nВведите номер строки, в которой нужно найти максимальный элемент:\n";

cin>>N;

if (N > 0 && N <= 2) {

for (i = 0; i < 2; i++) {

for (j = 0; j < 6; j++) {

if (matrix[N - 1][j] > max)

max = matrix[N - 1][j];

}

}

ofstream F("min\_max.txt", ios::app);

F << "Максимальный элемент " << fixed << setprecision(4) << N << " строки:\n\n" << max << "\n";

F << "\r\n";

F.close();

cout << "Максимальный элемент " << N << " строки:\n\n";

cout << max << endl;

cout << "\nНайти максимальный элемент в другой строке? (Y/N)\n\n";

char ch;

cin>>ch;

if (ch != 'Y') {

break;

}

} else {

cout << "Ошибка! Такой строки нет. Повторите ввод.\n\n";

}

}

}

int main()

{

float matrix[2][6];

char answer\_1, answer\_2;

setlocale(LC\_ALL, "russian");

for (;;) {

cout << "\nПрограмма вычисления максимального или минимального элемента матрицы в заданной строке\n" << endl;

cout << "1. Загрузить массив из файла\n2. Ввести вручную\n\n" << endl;

cin>>answer\_1;

if (answer\_1 == '1') file\_enter(matrix);

if (answer\_1 == '2') keybard\_enter(matrix);

if ((answer\_1 == '1') || (answer\_1 == '2')) break;

}

out\_matrix\_display(matrix);

cout << "\n\n";

for (;;)

{

cout << "\n\n1. Найти минимальный элемент в строке\

\n2. Найти максимальный элемент в стороке\

\n3. Записать конечные файлы\

\n4. Выйти из программы\n" << endl;

cin>>answer\_2;

if (answer\_2 == '4') break;

switch (answer\_2) {

case '1': min\_el\_matrix(matrix);

break;

case '2': max\_el\_matrix(matrix);

break;

case '3': out\_matrix\_file(matrix);

break;

default: cout << "\nВыберите 1, 2, 3 или 4!\n";

break;

}

}

return 0;

}

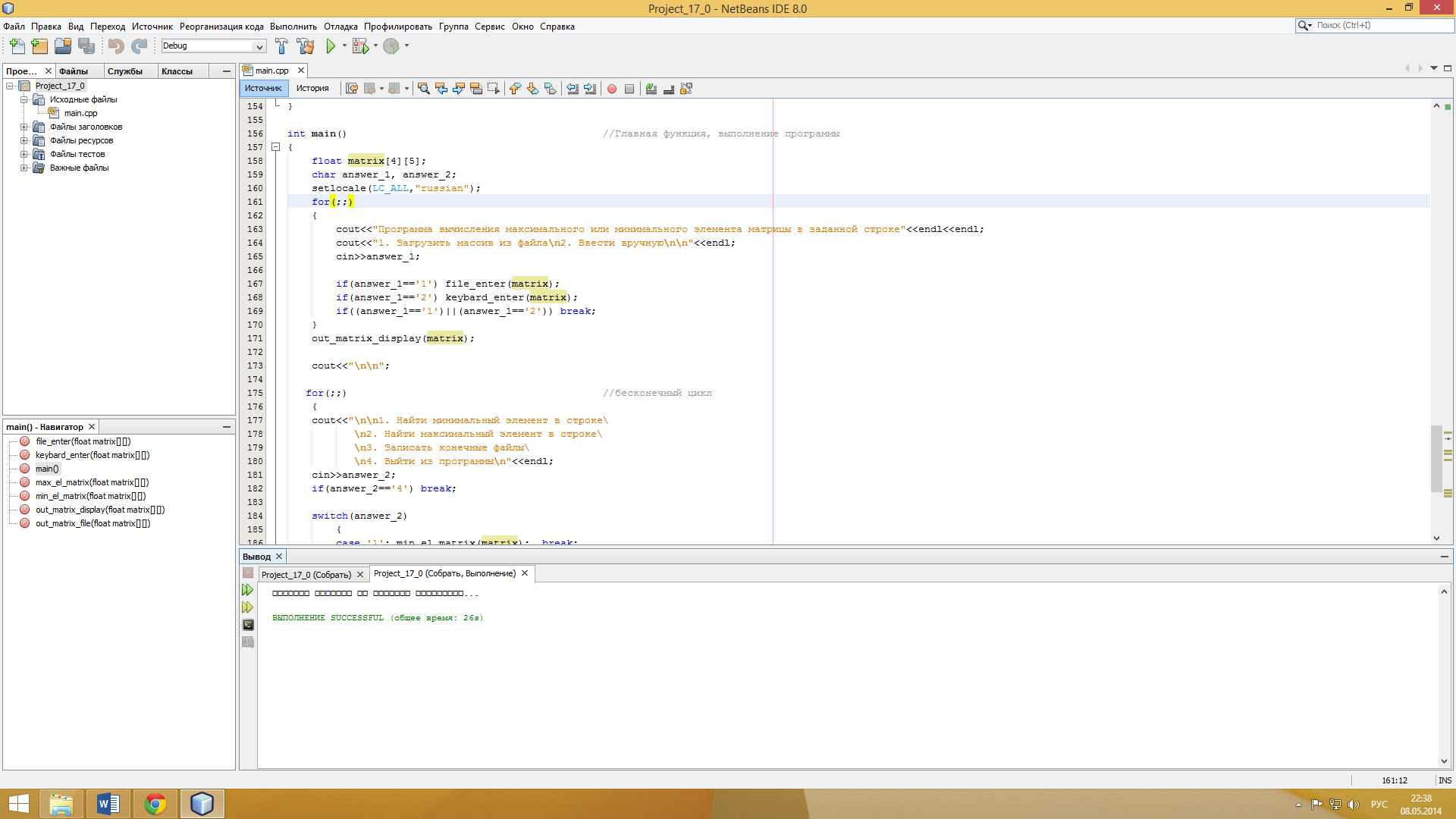
****

Рисунок 1 – Процесс разработки программы в NetBeans

**Руководство пользователя**

Программа разработана и запускается в среде NetBeans. Языком написания программы является C++.

Назначение программы – обработка двумерных вещественных массивов, демонстрация чтения из файла и записи в файл.

В качестве команд в программу можно вводить только целые числа; в качестве элементов массива – как целые, так и вещественные до 4 знаков после запятой. Введение символов может привести к сбою программы.

На первом этапе работы программы пользователю предлагается ввести массив, который программа будет обрабатывать, либо загрузить готовый массив из файла. Команда «1» загружает готовый массив, соответственно команда «2» запускает ввод массива пользователем, и программа переходит ко второму этапу работы. Введение каких-любо других цифр на данном этапе безрезультативно.

На втором этапе можно произвести три действия с массивом. Команда «1» позволяет найти минимальный элемент выбранной строки матрицы, команда «2» - максимальный, «3» - записывает матрицу в файл. Наконец, по команде «4» программа завершает работу. Ввод каких-либо других чисел безрезультативен и выводит сообщение об ошибке.

**Идентификаторы и их назначение:**

matrix[2][6] – исходный массив

i, j, answer\_1, answer\_2 – целочисленные переменные

min – минимальный элемент массива

max – максимальный элемент массива

**Список функций программы:**

keybard\_enter – ввод массива с клавиатуры

file\_enter – чтение массива из файла

out\_matrix\_display – вывод матрицы на экран компьютера

out\_matrix\_file – запись матрицы в файл

min\_el\_matrix – поиск минимального элемента столбца матрицы, запись его в файл

max\_el\_matrix – поиск максимального элемента столбца матрицы, запись его в файл

main – главная функция программы

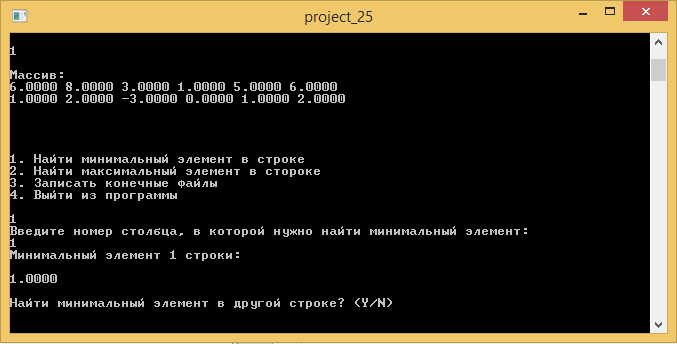


Рисунок 2 – Демонстрация работы программы

**ЗАКЛЮЧЕНИЕ**

Учебная задача решена.

По сути задача является поисковой, так как её конечный результат – демонстрационная программа, для написания которой требовалось знание определенных команд языка C++.

Для упрощения работы программы её интерфейс был написан на русском языке.

Наиболее сложным разделом пояснительной записки стала разработка массива, работающего с вещественными числами, и реализация функции записи этого массива в файл.

**Список использованной литературы**

1. Керниган Б., Ритчи Д. Язык программирования Си. – СПб, 2012. – 352 с.