**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, гр.ИСТ311

направление подготовки 230400

\_\_\_\_\_\_\_\_\_\_\_\_\_ Иванов А.А.

«\_\_\_»\_\_\_\_\_\_\_\_ 2014

Принял

ст.пр.каф.БИС

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

оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_ Евстигнеев В.А.

«\_\_\_»\_\_\_\_\_\_\_\_ 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

Блок-схема алгоритма ……………………………………………………………... 5

Текст программы …………………………………………………………………... 6

Руководство пользователя ………………………………………………………… 9

Заключение ............................................................................................................... 10

**Постановка задачи**

Текст задания:

# Вариант № 9

Разработать блок-схему алгоритма программы и программу, которая:

а) вводит с клавиатуры или читает из файла двумерный массив целых чисел размерностью [3 × 6] элементов и выводит его на экран в виде матрицы. 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 [3 × 6] элементов с двумя пробелами между ними, а результаты поиска записываются в другой файл.

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

**Блок-схема алгоритма**

Начало

Неправильная команда

while(1)

Меню

Ввод массива вручную

Чтение массива из файла

default

1

2

sw1

while(c1)

Меню

i, j, sw, sw1, c1=1, a1, a2,

min, max, M[3][6]

switch(sw1)

3

def

2

1

Запись в файл

Конец

Минимальный элемент

Максимальный элемент

**Текст программы**

#include <cstdlib>

#include <iostream>

#include <fstream>

#include <locale.h>

using namespace std;

void keyboard(int M[3][6]) {

int i, j;

cout << "\nВведите элементы массива: \n";

for (i = 0; i < 3; i++) {

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

cin >> M[i][j];

}

}

cout << "\nМассив: \n";

for (i = 0; i < 3; i++) {

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

cout << M[i][j] << " ";

}

cout << endl;

}

}

void file\_enter(int M[3][6]) {

int i, j;

std::ifstream file("massiv1.txt");

for (i = 0; i < 15; i++) {

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

file >> M[i][j];

}

}

cout << "\nМассив: \n";

for (i = 0; i < 3; i++) {

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

cout << M[i][j] << " ";

}

cout << endl;

}

}

void find\_max(int M[3][6]) {

int max, a1, j;

cout << "\nВыберите строку: \n";

cin>>a1;

max = M[a1][0];

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

if (M[a1][j] > max) {

max = M[a1][j];

}

}

cout << "\nMax: " << max;

FILE \*f2;

f2 = fopen("max.txt", "w");

fprintf(f2, "\n %d %d", max, a1);

fclose(f2);

}

void find\_min(int M[3][6]) {

int a2, j, min;

cout << "\nВыберите строку: \n";

cin>>a2;

min = M[a2][0];

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

if (M[a2][j] < min) {

min = M[a2][j];

}

}

cout << "\nMin: " << min;

FILE \*f3;

f3 = fopen("min.txt", "w");

fprintf(f3, "\n %d %d", min, a2);

fclose(f3);

}

void file\_save(int M[3][6]) {

int max, min, a1, a2;

FILE \*f1;

f1 = fopen("massiv2.txt", "w");

for (int l = 0; l < 3; l++) {

for (int k = 0; k < 6; k++) {

fprintf(f1, " %3d", M[l][k]);

}

fprintf(f1, "\r\n");

}

fclose(f1);

cout << "\nПроизведена запись в файл.";

}

int main(int argc, char\*\* argv) {

setlocale(LC\_CTYPE, "Russian");

int i, j, sw, sw1, c1 = 1, M[3][6];

printf("\n\nМЕНЮ:\n1) Ввести массив с клавиатуры\n2) Чтение массива из файла\n\n");

while (c1) {

cout << "\nВведите команду: ";

cin>>sw1;

switch (sw1) {

case 1:

keyboard(M);

c1 = 0;

break;

case 2:

file\_enter(M);

c1 = 0;

break;

default: printf("\nОшибка: неправильный символ.\n\n");

}

}

printf("\n\n\nМЕНЮ:\n1) Поиск максимального\n2) Поиск минимального\n3) Запись даннных в файл\n4) Выход из программы");

while (1) {

cout << "\n\n\nВведите команду ";

cin>>sw;

switch (sw) {

case 1:

find\_max(M);

break;

case 2:

find\_min(M);

break;

case 3:

file\_save(M);

break;

default: return 0;

}

}

}

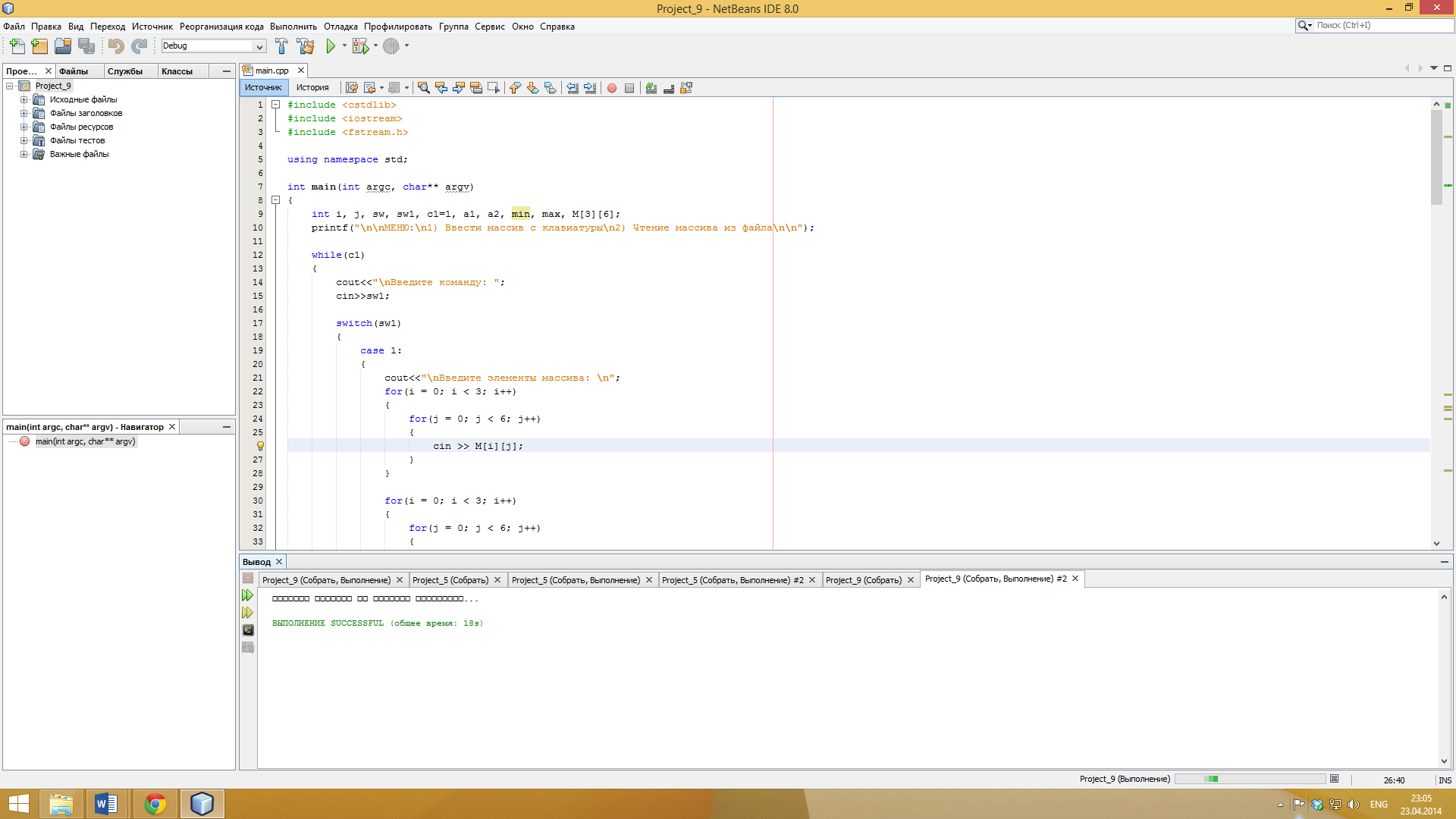
****

Рисунок 1 – Процесс разработки программы в NetBeans

**Руководство пользователя**

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

Назначение программы – обработка одномерных целочисленных массивов, демонстрация чтения из файла и записи в файл.

В программу можно водить только целые числа, в том числе и в качестве управляющих команд. 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. Соответственно, команда «1» запускает ввод массива пользователем, команда «2» загружает готовый массив, 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. По команде «1» программа ищет минимальный элемент массива, после её ввода нужно выбрать строку, в которой будет осуществляться поиск. Команда «2» действует аналогично, но ищет уже минимальный элемент. Наконец, команда «3» сохраняет полученный массив в файл. Ввод каких-либо других чисел останавливает работу программы. Иными словами, для прекращения работы программы достаточно ввести какое-либо число кроме номеров команд.

**Идентификаторы и их назначение:**

M[3][6] – исходный массив

i, j, sw, sw1, c1 – целочисленные переменные

a1, a2 – переменные-идентификаторы выбранных строк

max – максимальный элемент строки

min – минимальный элемент строки

**Список функций программы:**

Keyboard – ввод массива с клавиатуры

file\_enter – чтение массива из файла

find\_max – поиск максимального элемента в выбранной строке

find\_min – поиск минимального элемента в выбранной строке

file\_save – запись данных в файл

main – главная часть программы

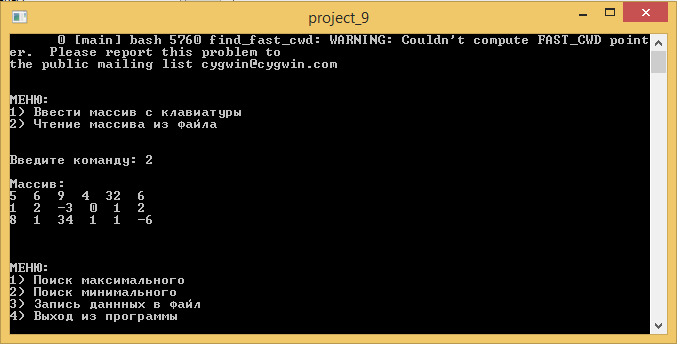


Рисунок 2 – Демонстрация работы программы

**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 стало составление блок-схемы программы, для чего потребовалось определённое усердие и аккуратность.