**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

\_\_\_\_\_\_\_\_\_\_\_ ФИО студента

«\_\_\_»\_\_\_\_\_\_\_\_ 2014

Принял

ст.пр.каф.БИС

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

оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО преподавателя

«\_\_\_»\_\_\_\_\_\_\_\_ 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 ……………………………………………………..… 11

Заключение ............................................................................................................... 13

**Постановка задачи**

Текст задания:

# Вариант № 14

Разработать блок-схему алгоритма программы и программу, которая:

а) вводит с клавиатуры или читает из файла двумерный массив вещественных значений размерностью [3 × 6] элементов и выводит его на экран. Вещественные элементы должны содержать не более 5-и знаков после запятой;

б) многократно, в зависимости от запроса, производит подсчет количества отрицательных, или положительных, или нулевых элементов матрицы и выдает результат 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 число отрицательных, положительных и нулевых элементов массива 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. Сумма отрицательных элементов
2. Сумма положительных элементов
3. Сумма нулевых элементов
4. Запись данных в файл
5. Выход

**Блок-схема алгоритма**

Начало

Неправильная команда

while(1)

Меню

Ввод массива вручную

Чтение массива из файла

default

1

2

sw1

while(c1)

Меню

int i, j, sw, sw1, c1=1, s1=0, s2=0, s3=0

float M[3][6]

switch(sw1)

3

def

4

2

1

Конец

Запись данных в файл

Сумма нулевых элементов

Сумма положительных

Сумма отрицательных

**Текст программы**

#include <cstdlib>

#include <iostream>

#include <fstream>

using namespace std;

void keyboard\_enter(float M[3][6])

{

int i, j;

cout<<"\nVvedite elementy massiva: \n";

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

{

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

{

cin >> M[i][j];

}

}

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

{

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

{

printf("%.0f ", M[i][j]);

}

cout << endl;

}

}

void file\_enter(float M[3][6])

{

int i, j;

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

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

{

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

{

file >> M[i][j];

}

}

cout<<"\nMassiv: \n";

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

{

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

{

printf("%.0f ", M[i][j]);

}

cout << endl;

}

}

void otr(float s1, int i, int j, float M[3][6])

{

s1=0;

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

{

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

{

if (M[i][j]<0)

s1=s1+1;

}

}

cout<<"\nChislo otricatelnuh elementov: "<<s1;

}

int pol(int i, int j, float M[3][6])

{

int s2=0;

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

{

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

{

if (M[i][j]>0)

s2=s2+1;

}

}

cout<<"\nchislo polojitelnyh elementov: "<<s2;

return s2;

}

int nul (int i, int j, float M[3][6])

{

int s3=0;

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

{

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

{

if (M[i][j]==0)

s3++;

}

}

cout<<"\nChislo nulevyh elementov: "<<s3;

}

void save(int i, int j, int s1, int s2, int s3, float M[3][6])

{

s1=0;

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

{

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

{

if (M[i][j]<0)

s1=s1+1;

}

}

s2=0;

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

{

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

{

if (M[i][j]>0)

s2=s2+1;

}

}

s3=0;

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

{

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

{

if (M[i][j]==0)

s3++;

}

}

FILE \*f1, \*f2;

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

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

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

{

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

{

fprintf(f1,"%.5f ", M[l][k]);

}

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

}

fprintf(f2,"\n %d %d %d", s1, s2, s3);

fclose(f1);

fclose(f2);

cout<<"\nDannye zapisany v fayl.";

}

int main()

{

int i, j, sw, sw1, c1=1, s1, s2, s3;

float M[3][6];

printf("\n\nMENU:\n1) Vvesti massiv s klaviatury\n2) Chtenie massiva iz fayla\n\n");

while(c1)

{

cout<<"\nVvedite comandu: ";

cin>>sw1;

switch(sw1)

{

case 1:

keyboard\_enter(M);

c1=0;

break;

case 2:

file\_enter(M);

c1=0;

break;

default: printf("\nОOshibka: nepravilny symvol.\n\n");

}

}

printf("\n\n\nМMENU:\n1) Chislo otricatelnyh elementov\n2) Chislo polojitelnyh elementov\n3) Chislo nulevyh elementov\n4) Zapis dannyh v fayl\n5) Vyhod");

while(1)

{

cout<<"\n\n\nVvedite comandu: ";

cin>>sw;

switch(sw)

{

case 1:

otr(s1, i, j, M);

break;

case 2:

pol(i, j, M);

break;

case 3:

nul(i, j, M);

break;

case 4:

save(i, j, s1, s2, s3, M);

break;

default: return 0;

}

}

}

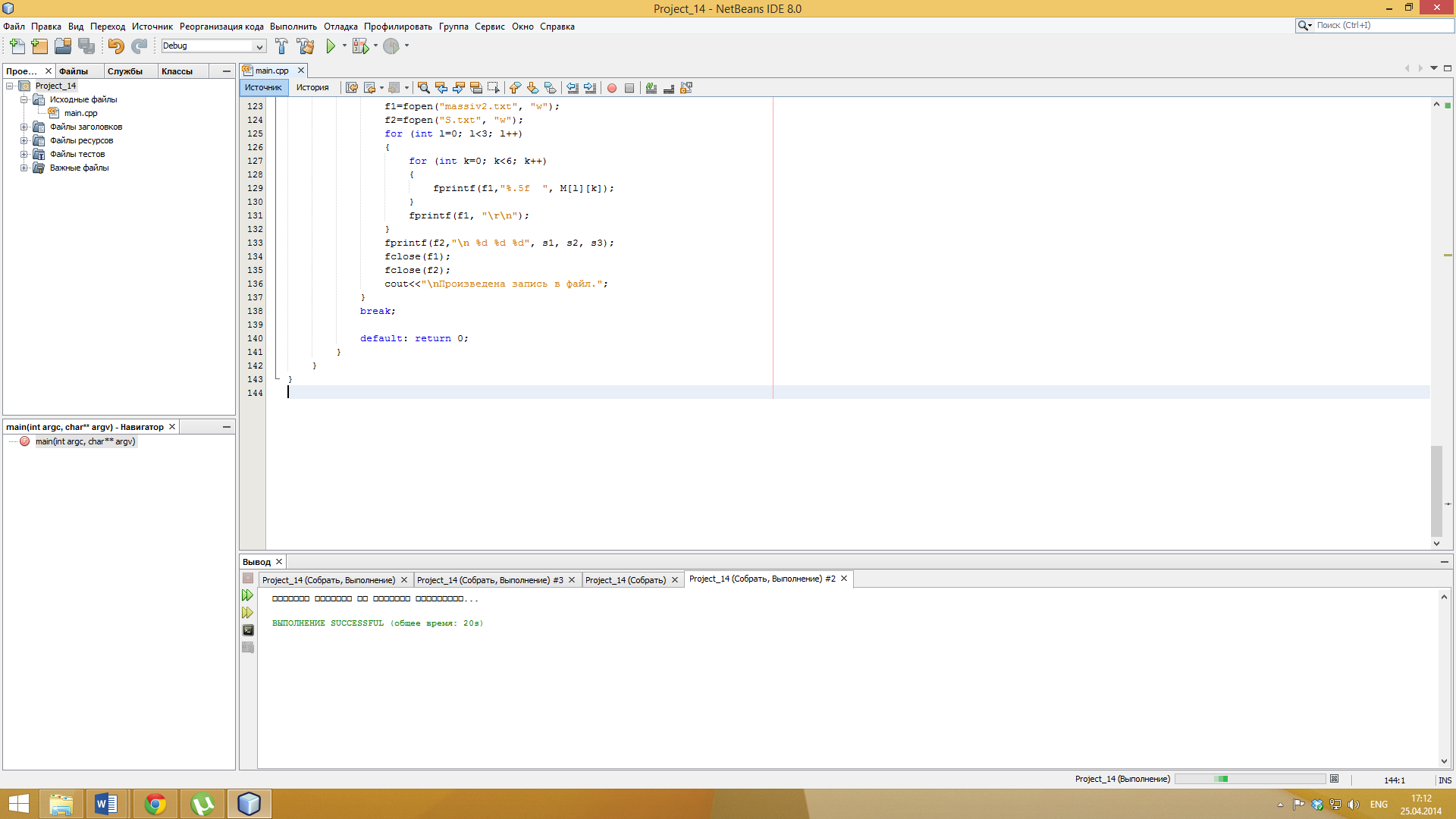
****

Рисунок 1 – Процесс разработки программы в NetBeans

**Руководство пользователя**

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

Назначение программы – обработка 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 5 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. 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. Команда «1» позволяет сосчитать число отрицательных элементов массива, команда «2» - положительных, «3» - нулевых. Наконец, по команде «4» массив и суммы соответствующих элементов записываются в файлы. 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

s1 – сумма отрицательных элементов массива

s2 – сумма положительных элементов массива

s3 – сумма нулевых элементов массива

**Список функций программы:**

keyboard\_enter – ввод массива с клавиатуры

file\_enter – чтение массива из файла

otr – подсчёт количества отрицательных элементов

pol – подсчёт количества положительных элементов

nul – подсчёт количества нулевых элементов

save – запись данных в файл

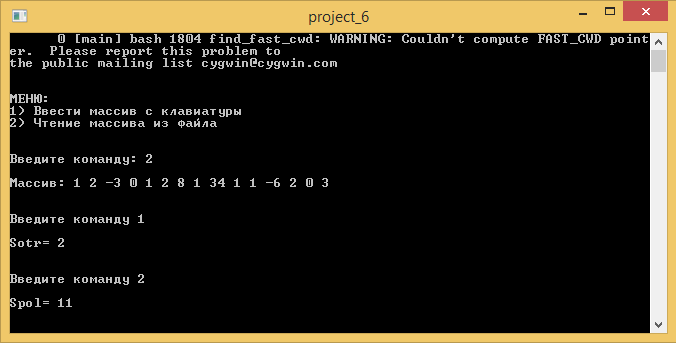


Рисунок 2 – Демонстрация работы программы

**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.