**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 6**

**«работА с шаблонами функций»**

Специальность 09.02.03«Программирование в компьютерных системах»

Дисциплина «Прикладное программирование»

|  |  |
| --- | --- |
| Преподаватель:  Сорокин Д.Ю.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2335  Котлицкий С.А. |

Санкт-Петербург

2018/2019

**Цель работы:** продемонстрировать работу с шаблонами функций

**Практическое задание:** Выполнить третью лабораторную работу, разделив программу на шаблоны функций:

1. Первая функция не возвращает значений (void), выполняет динамическое выделение памяти для элементов массива и инициализируются их с данными, введёнными с клавиатуры;
2. Вторая функция выполняет задание по варианту;
3. Третья функция печатает результат. Освобождение памяти происходит в главной функции.

**Индивидуальное задание:**

Даны вещественные массивы A[5][6], B[5]. Определить значения вещественного массива C[5] по правилу:

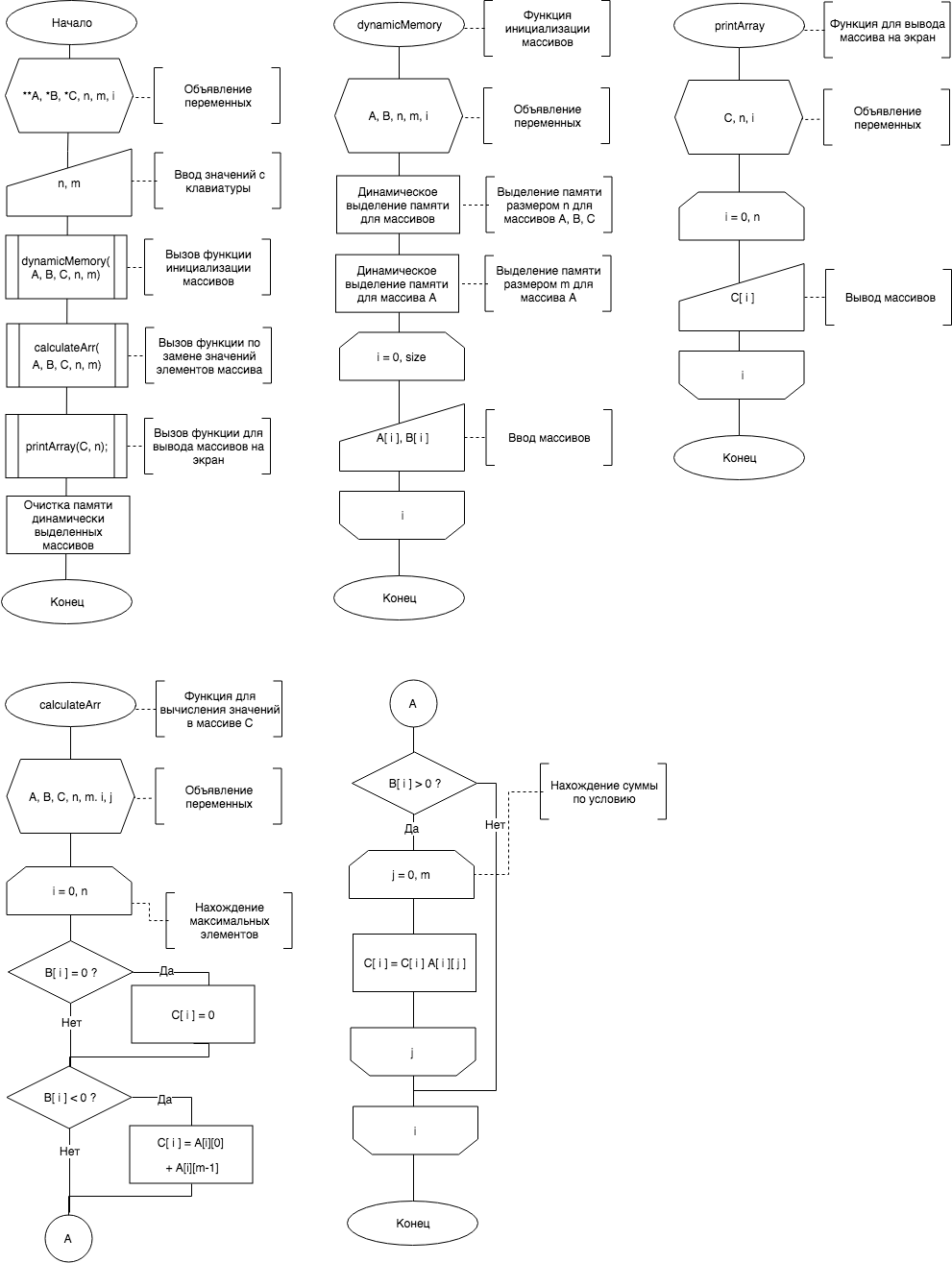
C[i] = сумме i-й строки массива A при B[i]>0

C[i]=0 при B[i]=0

C[i]=A[i][0]+A[i][5] при B[i]<0

Значение переменной i изменяется от 0 до 4.

Блок-схема алгоритма работы программы представлена ниже (Рисунок 1):



*Рисунок 1 Блок-схема алгоритма программы*

Код программы:

#include <iostream> //Подключаем библиотеки

#include <cmath> //Математическая библеотека

using namespace std; //Указываем пространство имен

template <typename T>

void dynamicMemory(T \*\*& A, T \*& B, T \*& C, int n, int m){

int i, j;

A = new T\*[n]; //Динамические выделение памяти для массивов

B = new T[n];

C = new T[n];

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

A[i] = new T[m]; //Динамическое выделение памяти для массива А[i]

}

cout << "Введите массив А - разрядность " << n << "x" << m << endl;

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

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

cin >> A[i][j]; //Ввод массива А

}

}

cout << "Введите массив B - разрядность " << n << endl;

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

cin >> B[i]; //Ввод массива B

}

}

template <typename H>

void calculateArr(H \*\*A, H \*B, H \*C, int n, int m){

int i,j;

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

if(B[i] == 0) {

C[i] = 0; //Условие если B[i]=0

}

else if(B[i] < 0) {

C[i] = A[i][0] + A[i][m-1]; //Условие если B[i]<0

} else {

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

C[i] += A[i][j]; //Условие если B[i]>0

}

}

}

}

template <typename Y>

void printArray(Y\*C, int n) {

int i;

cout << "массив C" << endl;

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

cout << C[i] << " ";

}

cout << endl;

}

int main() {

float \*\*A, \*B, \*C;// Объявление указателей типа float

int n, m, i; //Объявляем переменные типа int

cout << "Введите размерность массивов ( n, m )\n";

cin >> n >> m; //Ввод размера массивов

dynamicMemory(A, B, C, n, m);

calculateArr(A, B, C, n, m);

printArray(C, n);

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

delete[]A[i];

}

delete[]A;

delete[]B;

delete[]C;

return 0;

}

Протокол программы:

Введите размерность массивов ( n, m )

4 4

Введите массив А - разрядность 4x4

1 2 3 4

5 6 7 8

1 1 1 1

2 2 2 2

Введите массив B - разрядность 4

0 2 -6 0

массив C

0 26 2 0

Program ended with exit code: 0