# ОБРАБОТКА МАССИВОВ

**Цель работы**: изучить способы определения массивов, правила индексирования и инициализации; приобрести практические навыки передачи массивов функциям; изучить способы создания динамических массивов, правила управления свободной памятью, взаимосвязь между массивами и указателями.

**Задание**

1. Сформировать случайным образом два массива, расположив их, соответственно, в статической и динамической памяти; тип массива: 1) «вектор»; 2) «матрица»; 3) «куб».

2. Реализовать функцию обработки массивов.

**Вариант 4:** Вычислить произведение положительных элементов вектора (матрицы, куба);

**РЕШЕНИЕ**

**Алгоритм:**

1. Выбираем нужный нам тип массивов.
2. В зависимости от выбранного типа массива создаем сначала статическую версию, заполняем её случайными целыми числами, и выводим получившийся массив в консоль.
3. Создаем целую переменную rez равную 1, которая будет хранить результат вычеслений.
4. С помощью оператора for перебираем каждое число и если оно положительное, то умножаем rez на данное число.
5. Выводим результат в консоль.
6. Выполняем действия 2-5, но вместо статического массива создаем динамический.

**Блок-схема**:

Выбор типа массива

3

1

2

Обработка трехмерных массивов

Обработка двухмерных массивов

Обработка одномерных массивов

**Код**:

#include <iostream>

#include <ctime>

#include <Windows.h>

using namespace std;

void vekfill(int a[], int size) {

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

a[i] = rand() % 21 + (-10);

}

}

int obrvek(int a[], int size) {

int rez = 1;

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

if (a[i] > 0) {

rez \*= a[i];

}

}

return rez;

}

void privek(int a[], int size) {

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

cout << a[i] << " ";

}

cout << endl;

}

void matfill(int a[3][3], int size) {

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

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

a[i][j] = rand() % 21 + (-10);

}

}

}

int matobr(int a[3][3], int size) {

int rez = 1;

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

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

if (a[i][j] > 0) {

rez \*= a[i][j];

}

}

}

return rez;

}

void matpri(int a[3][3], int size) {

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

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

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

}

cout << endl;

}

}

void matfill(int \*\*a, int size) {

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

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

a[i][j] = rand() % 21 + (-10);

}

}

}

int matobr(int \*\*a, int size) {

int rez = 1;

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

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

if (a[i][j] > 0) {

rez \*= a[i][j];

}

}

}

return rez;

}

void matpri(int \*\*a, int size) {

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

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

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

}

cout << endl;

}

}

int main()

{

srand(time(0));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Выбирите тип массива:" << endl

<< "1-Вектор" << endl

<< "2-Матрица" << endl

<< "3-Куб" << endl;

int t; cin >> t;

switch(t) {

case 1: {

int a[5];

int\* a1 = new int[5];

int rez;

vekfill(a, 5);

cout << "Статический:" << endl;

privek(a, 5);

rez = obrvek(a, 5);

cout << "Результат=" << rez << endl;

vekfill(a1, 5);

cout << "Динамичeский:" << endl;

privek(a1, 5);

rez = obrvek(a1, 5);

cout << "Результат=" << rez << endl; }

break;

case 2: {

int a[3][3];

int\*\* a1 = new int\* [3];

for (int i = 0; i < 3; i++) { a1[i] = new int[3]; }

int rez;

matfill(a, 3);

cout << "Статический:" << endl;

matpri(a, 3);

rez = matobr(a, 3);

cout << "Результат=" << rez << endl;

matfill(a1, 3);

cout << "Динамичeский:" << endl;

matpri(a1, 3);

rez = matobr(a1, 3);

cout << "Результат=" << rez << endl; }

break;

case 3: {

int a[2][2][2];

int\*\*\* a1 = new int\*\* [2];

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

a1[i] = new int\* [2];

for (int j = 0; j < 2; j++) {

a1[i][j] = new int[2];

}

}

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

for (int j = 0; j < 2; j++) {

for (int c = 0; c < 2; c++) {

a[i][j][c] = rand() % 21 + (-10);

}

}

}

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

for (int j = 0; j < 2; j++) {

for (int c = 0; c < 2; c++) {

a1[i][j][c] = rand() % 21 + (-10);

}

}

}

cout << "Статический:" << endl;

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

for (int j = 0; j < 2; j++) {

for (int c = 0; c < 2; c++) {

cout << a[i][j][c] << " ";

}

cout << endl;

}

cout << endl;

}

int rez = 1;

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

for (int j = 0; j < 2; j++) {

for (int c = 0; c < 2; c++) {

if (a[i][j][c] > 0) {

rez \*= a[i][j][c];

}

}

}

}

cout << "Результат=" << rez << endl;

cout << "Динамичeский:" << endl;

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

for (int j = 0; j < 2; j++) {

for (int c = 0; c < 2; c++) {

cout << a1[i][j][c] << " ";

}

cout << endl;

}

cout << endl;

}

rez = 1;

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

for (int j = 0; j < 2; j++) {

for (int c = 0; c < 2; c++) {

if (a1[i][j][c] > 0) {

rez \*= a1[i][j][c];

}

}

}

}

cout << "Результат=" << rez << endl;}

break;

default:

break;

}

system("pause");

return 0;

}

**Тесты**:





