

Двумерные массивы

Язык программирования C++

Геометрическое представление

ФИО	ОП	Истор.	Мат-ка	Ф-ра
Иванов И.И.	7	12	9	7
Петров П.П.	5	4	2	6
Сидоров С.С.	7	7	6	8
Захаров З.З.	9	10	8	11
Антонов А.А.	7	8	6	6
Карпов К.К.	11	8	10	9
Игорев И.И.	8	6	8	5
Макаров М.М.	9	10	8	11



Матрица — это двумерный массив, каждый элемент которого имеет два индекса: номер строки и номер столбца, поэтому для работы с элементами матрицы необходимо использовать два цикла. Если значениями параметра первого цикла будут номера строк матрицы, то значениями параметрами второго — столбцы (или наоборот). Обработка матрицы заключается в том, что вначале поочередно рассматриваются элементы первой строки (столбца), затем второй и т.д. до последней.



В C++ определены и двумерные массивы (матрицы), например матрица A, состоящая из 4 строк и 5 столбцов..

$$A = \begin{pmatrix} 6 & -9 & 7 & 13 & 19 \\ 5 & 8 & 3 & 8 & 22 \\ 3 & 7 & 88 & 33 & 71 \\ 55 & 77 & 88 & 37 & 61 \end{pmatrix}$$

Двумерный массив (матрицу) можно объявить так:
тип имя_переменной [n] [m];

где **n** – количество строк в матрице(строки нумеруются от 0 до **n-1**),
m – количество столбцов (столбцы нумеруются от 0 до **m-1**).

Например,

```
int x[10][15];
```

Описана матрица x, состоящая из 10 строк и 15 столбцов (строки нумеруются от 0 до 9, столбцы от 0 до 14).

Для обращения к элементу матрицы необходимо указать ее имя, и в квадратных скобках номер строки, а затем в квадратных скобках – номер столбца. Например, x[2][4] – элемент матрицы x, находящийся в третьей строке и пятом столбце.

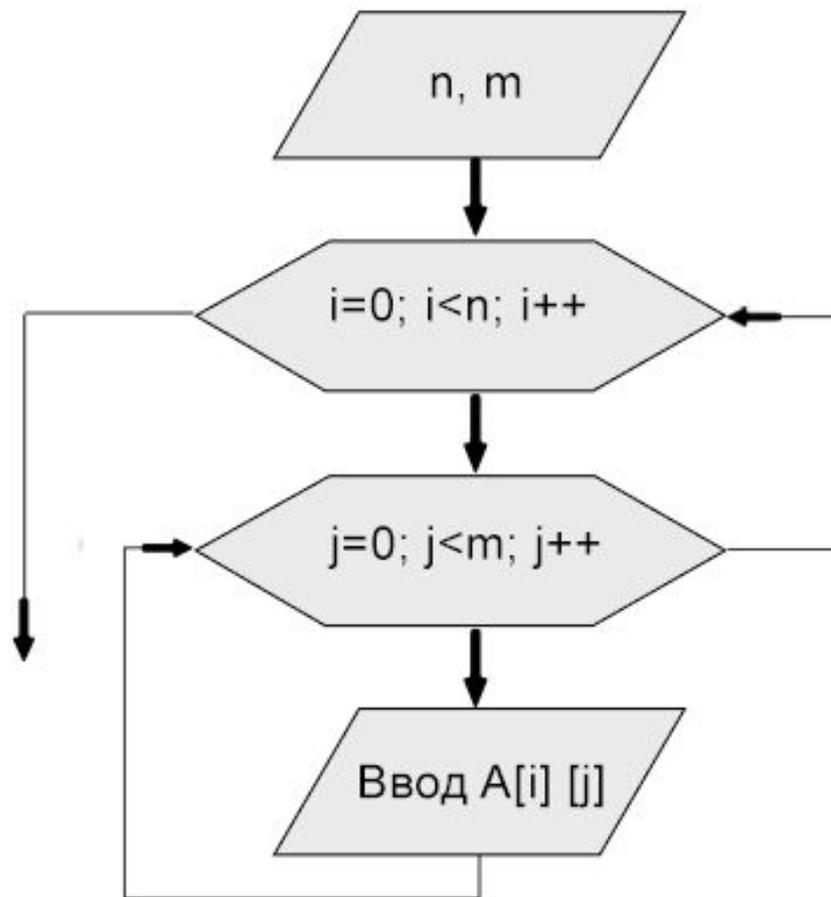
В C++ можно описать многомерные массивы, которые можно объявить с помощью оператора следующей структуры:

```
тип имя_переменной [n1] [n2]...[nk];
```

Ввод-вывод матриц

Матрицы, как и одномерные массивы, нужно вводить (выводить) поэлементно.

Блок-схема ввода элементов матрицы $A[n][m]$ изображена ниже:





```
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
setlocale (LC_ALL, "RUS");
int i,j,N,M,a[20][20];
cout<<"N="; //ввод количества строк
cin>>N;
cout<<"M="; //ввод количества столбцов
cin>>M;
cout<<"Input matrix A \n";
//цикл по переменной i, в которой перебираем строки матрицы
for (i=0; i<N; i++)
//цикл по переменной j, в котором перебираем элементы внутри строки
for (j=0; j<M; j++)
cin>>a[i][j]; //ввод очередного элемента матрицы
cout<<"matrix A \n";
for (i=0; i<N; i++)
{
//цикл по переменной i, в котором перебираем строки матрицы
for (j=0; j<M; j++)
cout<<a[i][j]<<"\t"; //вывод очередного элемента матрицы
cout<<endl; //переход на новую строку после вывода всех элементов строк
}
system("pause");
return 0;
}
```



```
N=3
```

```
M=4
```

```
Input matrix A
```

```
32
```

```
65
```

```
5
```

```
-5
```

```
45
```

```
76
```

```
4
```

```
3
```

```
2
```

```
6
```

```
7
```

```
8
```

```
matrix A
```

```
32
```

```
65
```

```
5
```

```
-5
```

```
45
```

```
76
```

```
4
```

```
3
```

```
2
```

```
6
```

```
7
```

```
8
```

```
Для продолжения нажмите любую клавишу . . .
```

Kvodo.ru

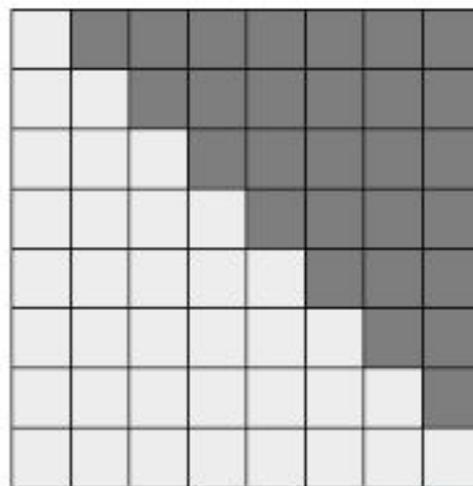


Теперь, давайте рассмотрим некоторые свойства матриц:

- если номер строки элемента совпадает с номером столбца ($i = j$), это означает, что элемент лежит на главной диагонали матрицы;
- если номер строки превышает номер столбца ($i > j$), то элемент находится ниже главной диагонали;
- если номер столбца больше номера строки ($i < j$), то элемент находится выше главной диагонали;
- элемент лежит на побочной диагонали, если его индексы удовлетворяют равенству $i+j+1=n$;
- неравенство $i+j+1 < n$ характерно для элемента, находящегося выше побочной диагонали;
- соответственно, элементу, лежащему ниже побочной диагонали, соответствует выражение $i+j+1 > n$.

Задача 1

Найти сумму элементов матрицы, лежащих выше главной диагонали (рис. ниже)



Kvodo.ru

Алгоритм решения данной задачи можно построить следующим образом: обнулить ячейку для накопления суммы (переменная S). Затем с помощью двух циклов просмотреть каждый элемент матрицы, суммировать элементы при условии, что элемент находится выше главной диагонали. Текст решения

Задача 2

Проверить, является ли заданная квадратная матрица единичной. Единичной называют матрицу, у которой элементы главной диагонали — единицы, а все остальные — нули. Например:

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

Решать задачу будем так. Предположим, что матрица единичная и попытаемся доказать обратное. Если окажется, что хотя бы один диагональный элемент не равен единице или любой из элементов вне диагонали не равен нулю, то матрица единичной не является. Воспользовавшись логическими операциями языка C++, все эти условия можно соединить в одно и составить программу:



Задача 3

Преобразовать исходную матрицу так, чтобы нулевой элемент каждой строки был заменен средним арифметическим элементом этой строки.

Для решения задачи необходимо найти в каждой строке сумму элементов, которую разделить на их количество. Полученный результат записать в нулевой элемент соответствующей строки.

Описание двумерного массива

7	12	9	7
5	4	2	6
7	7	6	8

ТипЭлементов ИмяМассива [колич.строк] [колич.столбцов];

```
int a [ 3 ] [ 4 ];
```

3 строки

4 столбца

Доступ к элементам массива

7	12	9	7
5	4	2	6
7	7	6	8

ИмяМассива [ном.строки] [ном.столбца]

$a[0][0] = 7;$

$a[1][2] = 2;$

$a[2][3] = 8;$

Ввод массива по строкам

7	12	9	7
---	----	---	---

5	4	2	6
---	---	---	---

7	7	6	8
---	---	---	---

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

```
for ( int j = 0; j < 4; j + + )
```

```
    cin >> a [ i ] [ j ];
```

Ввод массива по столбикам

7
5
7

12
4
7

9
2
6

7
6
8

```
for ( int j = 0; j < 4; j + + )
```

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

```
    cin >> a [ i ] [ j ];
```

Вывод элементов массива таблицей

7	12	9	7
5	4	2	6
7	7	6	8

```
for ( int i = 0; i < 3; i + + )  
{  
    for ( int j = 0; j < 4; j + + )  
        cout << a [ i ] [ j ] << "\t ";  
    cout << "\n ";  
}
```

Примеры работы с двумерными массивами

1	0	1
1	0	1

```
for ( int i = 0; i < 2; i + + )  
for ( int j = 0; j < 3; j + + )  
if ( j == 1 ) a [ i ] [ j ] = 0;  
    else a [ i ] [ j ] = 1;
```

Примеры работы с двумерными массивами

1	1	0
1	1	0

```
for ( int i = 0; i < 2; i + + )  
{ for ( int j = 0; j < 2; j + + )  
    a [ i ] [ j ] = 1;  
  for ( int j = 2; j < 3; j + + )  
    a [ i ] [ j ] = 0; }
```

Задача 1 варианта

```
int a[10][20];
```

```
...
```

```
x=0;
```

```
for (int i=0; i<10; i++)
```

```
    for (int j=0; j<20; j++)
```

```
        x=x+a[i][j];
```

```
y = x/200.0;
```

Задача 2 варианта

```
for (int i=0; i<5; i++)  
    for (int j=0; j<6; j++)  
        a [ i ][ j ]=rand% ()21-10;  
  
...  
  
x=1; y=0;  
for (int i=0; i<5; i++)  
    for (int j=0; j<6; j++)  
        if (a[i][j]>0)  
            { x*=a[i][j]; y++;}
```

Задача 3 варианта

```
for (int i=0; i<20; i++)
```

```
    for (int j=0; j<30; j++)
```

```
        a[i][j] =rand%() 61-80;
```

...

```
int x = - 80;
```

```
for (int i=0; i<20; i++)
```

```
    for (int j=0; j<30; j++)
```

```
        if (a[i][j]>x && a[i, j] % 2 == 0)
```

```
            { x=a[i][j]; y=i; }
```

Закрепление материала

Что означают эти операторы в программе

```
for (int i=0; i<10; i++)  
cout<<a[i];
```

Что означают эти операторы в программе

```
for (int i=0; i<10; i++)  
a[i] = rand()%41-20;
```

Что означают эти операторы в программе

```
int k=0;
```

```
for (int i=0; i<10; i++)
```

```
if (a[i]==0) k++;
```

Что означают эти операторы в программе

```
int p=1;  
for (int i=0; i<10; i++)  
if (a[i]>0) p*=a[i];
```

Что означают эти операторы в программе

```
int S=0;
```

```
for (int i=0; i<10; i++)
```

```
S+=a[i];
```

```
cout<<(float) S/k;
```

Что означают эти операторы в программе

```
int k=0;  
for (int i=0; i<10; i++)  
if (a[i]>5 && a[i]<30)  
k++;
```

Что означают эти операторы в программе

```
int k=0;  
for (int i=0; i<10; i++)  
if (a[0] ==a[i])  
k++;
```

Что означают эти операторы в программе

```
int S=0;
```

```
for (int i=10/2; i<10; i++)
```

```
k+=a[i];
```

Что означают эти операторы в программе

```
for (int i=9; i>=0; i-- )  
cout<<a[i];
```

Что означают эти операторы в программе

```
for (int i=0; i<10/2; i++)  
cout<<a[i]<< "\n";
```

Что означают эти операторы в программе

```
for (int i=0; i<10; i++ )  
if (a[i] % 3 ==0)  
cout<<a[i];
```

Что означают эти операторы в программе

```
for (int i=0; i<10; i++ )  
if (i % 2 ==0)  
cout<<a[i];
```

Что означают эти операторы в программе

```
int b=a[0];
```

```
a[0]=a[5];
```

```
a[5]=b;
```

Что означают эти операторы в программе

```
int b=max;
```

```
max=min;
```

```
min=b;
```

Что означают эти операторы в программе

```
int b=a[n_max];
```

```
a[n_max]=a[n_min];
```

```
a[n_min]=b;
```

Что выведется на экран?

1 1 -1 0

0 -1 1 1

0 -1 0 -1

```
S=0;
```

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

```
    for (int j=0; j<4; j++)
```

```
        S+=a[i][j];
```

```
cout<< "S="<< S;
```

Что выведется на экран?

2 4 -2 0 1

4 -2 6 4 0

1 0 0 2 2

```
int k=0;
```

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

```
    for (int j=0; j<5; j++)
```

```
        if (a[i][j] == 0)
```

```
            k++;
```

```
cout<< "k="<< k;
```

Что выведется на экран?

```
int k=0, S=0;
```

```
for (int j=0; i<4; j++)
```

```
4   2   3  -4   for (int i=0; i<3; i++)
```

```
4  -3   5   4   if (a[i][j] % 2 != 0)
```

```
2   3   2  -6   {   S+= a[i, j]
```

```
    k++;
```

```
    }
```

```
cout<< (float)S/k;
```

Что выведется на экран?

23 -42 2 64 1

41 2 77 1 85

53 -46 88 2 -2

```
int k=0;
```

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

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

```
        if (a[i][j] > -9 && a[i][j] < 9) k++;
```

```
cout<< "k="<< k;
```

Что выведется на экран?

2	4	-2	0	1	m=a[0][3];
4	-2	6	4	0	for (int i=0; i<3; i++)
1	0	0	2	2	for (int j=3; j<5; j++)
-4	2	4	3	5	if (a[i][j] >m)
2	4	-5	3	1	m=a[i][j];
					cout<< "m="<< m;