

Двумерные массивы

Частным случаем многомерного массива является двумерный массив, или матрица. Двумерный массив представляет собой совокупность строк и столбцов, на пересечении которых находится конкретное значение. Для объявления двумерного массива необходимо указать количество строк и столбцов. При этом действуют те же правила, что и при объявлении одномерного массива:

```
тип_данных                                имя_массива  
[число_строк] [число_столбцов];
```

Несмотря на то, что мы представляем двумерный массив в виде матрицы, в памяти любой двумерный массив располагается построчно: сначала нулевая строка, затем первая и так далее. Об этом следует помнить, т.к. выход за пределы массива может повлечь за собой некорректную работу программы, при этом компилятор не сообщает об ошибке.

Обращение к конкретному элементу массива осуществляется по номеру строки и номеру столбца, например: `array [2][1]`. Многомерный массив в C++ по своей сути одномерен. Операции `new` и `delete` позволяют создавать и удалять динамические массивы, поддерживая при этом иллюзию произвольной размерности. Деятельность по организации динамического массива требует дополнительного внимания, однако характеристики массива (операнды операции `new`) могут не быть константными выражениями. Это позволяет создавать многомерные динамические массивы произвольной конфигурации.

Организация двумерного динамического массива производится в два этапа: сначала создаётся одномерный массив указателей, а затем каждому элементу этого массива присваивается адрес одномерного массива:

```

int size_row = 5,
size_col = 5;
int **pArr = new int*[ size_row]; for (int i = 0; i
< size_row; i++)
pArr[i] = new int[size_col];

```

Уничтожение двумерного массива происходит в обратной последовательности:

```

for (int i = 0; i < size_row; i++)
delete[] pArr[i];
delete[] pArr;

```

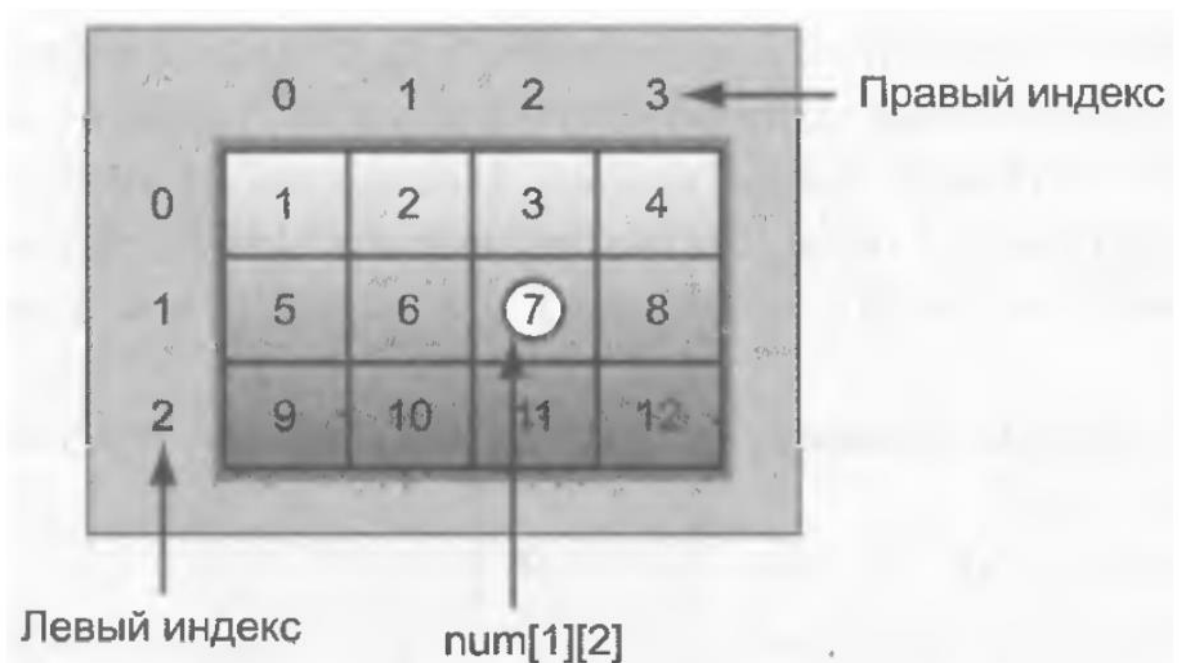


Рис. 5.1. Схематическое представление массива `num`

Примеры реализации сортировок:

```
#include <iostream>
#include <iomanip>
#include <ctime>

using namespace std;

// функция создания двумерного массива
int** formMatrix(int n, int m);
// печатать двумерного массива на экран
void printMatrix(int** matrix, int n, int m);
// заполнение двумерного массива случайными числами
void fillMatrix(int** matrix, int n, int m);
// функция освобождения выделенной под массив памяти
void deleteMatrix(int** matrix, int n, int m);
// меняем местами указанный столбец и ряд матрицы
void changeRowColumn(int** matrix, int n, int m, int k);
// находим номер строки, в которой находится самое больше среднее среди всех строк
int getRowNumWithMaxAvg(int** matrix, int n, int m);

int main() {
    srand(time(NULL));
    // задание №11 - поменять местами указанные столбец и строку
    int size1 = 5;
    int index = 2; // номер строки/столбца для замены
    // создали матрицу-двумерный массив
    int** matrix1 = formMatrix(size1, size1);
    // заполнили массив случайными числами
    fillMatrix(matrix1, size1, size1);
    cout << "Initial matrix: " << endl;
    // печать массива, заполненного массива в консоль
    printMatrix(matrix1, size1, size1);
    // меняем в массиву местами строку и столбец с номером index
    changeRowColumn(matrix1, size1, size1, index);
    // печатаем результат на экран
    cout << "Matrix after change " << index + 1
         << "row and column" << endl;
    printMatrix(matrix1, size1, size1);
    // освобождаем память после окончания работы с массивом
    deleteMatrix(matrix1, size1, size1);

    return 0;
}

int** formMatrix(int n, int m)
{
    // создаем двумерный массив
    // это как бы массив, каждый элемент которого тоже является массивом
    int** arr = new int* [n];
    for (int i = 0; i < n; i++) {
        // каждый элемент массива arr - тоже массив
        arr[i] = new int[m];
    }
    return arr;
}

void printMatrix(int** matrix, int n, int m)
{
    // закрепляем строку (индекс i) и в этой строке идём по каждому из столбцов
    // (меняем j)
    // тем самым проходим по всей строке
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
```

```

        // setw(5) - отступ в пять пробелов между элементами массива
        cout << setw(5) << matrix[i][j];
    }
    cout << endl;
}

// заполнение массива случайными числами
void fillMatrix(int** matrix, int n, int m)
{
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            matrix[i][j] = rand() % 101;
        }
    }
}

// освобождение выделенной оперативной памяти
void deleteMatrix(int** matrix, int n, int m)
{
    for (int i = 0; i < n; i++) {
        // берём и освобождаем память, выделенную под первый элемент массива
        // два варианта синтаксиса
        delete matrix[i]; // delete[] matrix[i]
    }
    delete matrix; // delete[] matrix;
}

void changeRowColumn(int** matrix, int n, int m, int k)
{
    // меняем элементы местами при помощи временной переменной
    for (int i = 0; i < n; i++) {
        int tmp = matrix[k][i];
        matrix[k][i] = matrix[i][k];
        matrix[i][k] = tmp;
    }
    /*
    for (int i = 0; i < n; i++) {
        if (i != k) {
            int tmp = matrix[k][i];
            matrix[k][i] = matrix[i][k];
            matrix[i][k] = tmp;
        }
    }
    */
}

```

Задания для тренировки







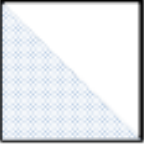
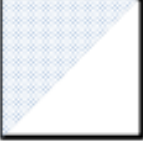

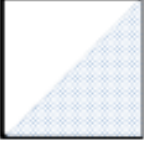





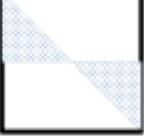
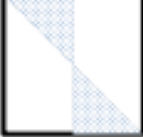
1	Дана действительная матрица размером 7×8 . Найти максимальный элемент матрицы. Поменять строку, которая содержит наибольший элемент с первой строкой матрицы.
2	Дана действительная матрица размером 7×8 . Найти самый большой элемент матрицы. Поменять столбик, который содержит наибольший элемент с первым столбцом матрицы.
3	Составить программу замены всех отрицательных элементов матрицы $A(9, 9)$ на 0, если сумма минимального и максимального элементов этой матрицы окажется меньше P , где P вводится с клавиатуры.
4	Составить программу нахождения максимального элемента в каждом столбике матрицы $A(15, 15)$.
5	Составить программу нахождения минимального положительного элемента в каждом столбце матрицы $A(11, 11)$.
6	Составить программу нахождения количества строк матрицы $A(10, 10)$, сумма элементов которых отрицательная.
7	Составить программу нахождения количества строк матрицы $A(8, 8)$, у которых количество отрицательных элементов больше чем P , где P вводится с клавиатуры.
8	Составить программу формирования вектора $B(16)$, если b_i – сумма минимального и максимального элементов i -й строки матрицы $A(16, 5)$.
9	Составить программу замены всех отрицательных элементов матрицы $A(12, 12)$ на элемент этой матрицы, который имеет максимальное значение.

10	Дан двумерный массив целых чисел размерности 6×15 . Найти номер строки, для которой среднеарифметическое значение элементов максимально.
11	В двумерном массиве целых чисел размерности 9×9 поменять местами строку и столбик, номер которого вводится пользователем.
12	Дан массив $C(14, 14)$. Определить количество "особых" элементов массива, считая элемент "особым", если он больше суммы других элементов своего столбика. Напечатать индексы "особых" элементов.
13	Дан массив $C(11, 11)$. Определить количество "особых" элементов массива, считая элемент "особым", если в строке слева от него находятся элементы меньшие, чем он, а по правую сторону большие.
14	Дан массив целых чисел $C(11, 11)$. Найти минимальный элемент среди максимальных элементов строк этого массива. Определить номер строки и столбика такого элемента.
15	Дан массив целых чисел $C(11, 11)$. Удалить столбец двумерного массива, в котором находится максимальный элемент.
16	Дан массив целых чисел $C(10, 10)$. Найти все неповторяющиеся элементы двумерного массива.
17	Дан массив $C(7, 10)$. Удалить столбики, расположенные между столбиками с минимальным и максимальным элементами.
18	Дан массив $C(9, 11)$. Четные строки массива сдвинуть циклически на K элементов вправо, где K вводится с клавиатуры.
19	Дан массив $C(9, 12)$. Нечетные строки массива сдвинуть циклически на K элементов влево, где K вводится с клавиатуры.
20	Создайте двумерный массив целых чисел $C(11, 9)$. Удалите из него строку и столбец, на пересечении которых расположен минимальный элемент.

1	Вставить строку с указанным номером K
2	Вставить строку в начало матрицы
3	Вставить столбик в начало матрицы
4	Вставить K строк в начало матрицы
5	Вставить K столбцов в начало матрицы
6	Удалить строку с номером K
7	Удалить столбец с номером K
8	Удалить строки, начиная со строки K_1 и до строки K_2
9	Удалить столбцы, начиная со столбца K_1 и до столбца K_2
10	Удалить все четные строки
11	Удалить все четные столбцы
12	Удалить все строки, в которых есть хотя бы один нулевой элемент
13	Удалить все столбцы, в которых есть хотя бы один нулевой элемент
14	Удалить строку, в которой находится самый большой элемент матрицы
15	Добавить строки после каждой четной строки матрицы
16	Добавить столбцы после каждого четного столбца матрицы
17	Добавить K строк, начиная со строки с номером N
18	Добавить K столбцов, начиная со столбца с номером N
19	Добавить строку после строки, которая содержит самый большой

Задание 3.4. Составить программу, которая заполняет двумерный массив следующим образом: элементы, которые принадлежат заштрихованной области, генерируются случайным образом; остальные равняются N, где N – номер варианта. Упорядочить все случайные элементы массива, которые принадлежат заштрихованной области. Четные варианты сортируют по возрастанию, нечетные – по убыванию. При написании программы пользоваться принципами структурного программирования.

Таблица 3.4 – Варианты задания 3.4

№	Рисунок	№	Рисунок	№	Рисунок
1		2		3	
4		5		6	
7		8		9	
10		11		12	
13		14		15	
16		17		18	