

Глава 5 «Массивы»

§5.3 Двумерные массивы. Массивы с вложенными циклами.

Двумерный массив

Двумерный массив - это такой массив, у каждого элемента которого два индекса (часто их называют матрицами). В них удобно хранить и обрабатывать различные табличные данные.

Двумерный массив можно визуально представить в виде таблицы, состоящей из строк и столбцов.

a[0][0]	a[0][1]	a[0][2]	a[0][3]	...	a[0][n]
a[1][0]	a[1][1]	a[1][2]	a[1][3]	...	a[1][n]
a[2][0]	a[2][1]	a[2][2]	a[2][3]	...	a[2][n]
...
a[m][0]	a[m][1]	a[m][2]	a[m][3]	...	a[m][n]

Двумерный массив представление в виде матрицы с m строками и n столбцами

Двумерный массив. Объявление и инициализация.

В C++ не предусмотрен специальный тип представления двумерных массивов. Вместо этого создается такой массив, каждый элемент которого является массивом. После имени переменной в своих квадратных скобках [] указывается 2 числа: сначала количество строк, а затем количество столбцов.

```
int dvarr [5][5];
```

Объявление двумерного массива целых чисел dvarr, 5 строк, 5 столбцов.

```
char dvarr2 [7][5];
```

Объявление двумерного массива символов dvarr2, 7 строк, 5 столбцов

```
double dvarr3 [5][10];
```

Объявление двумерного массива вещественных чисел dvarr3, 5 строк, 10 столбцов

Двумерный массив. Объявление и инициализация.

Инициализировать его нужно через два цикла `for`, вложенных друг в друга. Внешний отвечает за итерирование по строкам (переменная `row`), а внутренний по столбцам (переменная `col`). Также, как и с одномерным массивом можно инициализировать при объявлении, указав в фигурных скобках элементы строк.

```
double dvarr [5][10];  
for (int row = 0; row < 5; row++){  
    for (int col = 0; col < 10; col++){  
        dvarr [row][col] = 5;  
    }  
}
```

Инициализация двумерного массива `dvarr` элементами 5.

```
int dvarr [2][3] {3, 6}, {2, 5}, {1, 4};
```

Инициализация при объявлении. Двумерный массив `dvarr` заполняется элементами: первая строка 3 и 6, вторая строка 2 и 5, третья строка 1 и 4

Динамический двумерный массив.

Двумерные массивы также могут быть и динамическими, то есть можно изменить количество строк и столбцов в процессе выполнения программы.

```
int** dvarr = new int* [m];  
for(int i = 0; i < m; i++){  
    dvarr[i] = new int[n];  
}
```

Объявление двумерного динамического массива размера m на n.

Обработка массивов.

- Обход массива (итерирование):

```
int arr[5];  
for (int i = 0; i < 5; i++){  
    arr[i] = i;  
}
```

Итерирование (доступ к каждому элементу массива) с помощью цикла for

- Вывод элементов массива:

```
for(int i = 0; i < n; i++){  
    cout << arr[i] << " ";  
}
```

Вывод элементов массива длины n (динамический)

```
for(int i = 0; i < 5; i++){  
    cout << arr[i] << " ";  
}
```

Вывод элементов массива длины 5 (статический)

```
double dvarr [4][3];  
for (int row = 0; row < 4; row++){  
    for (int col = 0; col < 3; col++){  
        cout << dvarr [row][col] << " ";  
    }  
    cout << endl;  
}
```

Вывод элементов двумерного массива 4 на 3 (статический)

Обработка массивов.

- Ввод элементов массива с консоли:

```
for (int i = 0; i < n; i++){  
    cin >> arr[i];  
}
```

Ввод элементов
массива длины n
(динамический)

```
for (int i = 0; i < 5; i++){  
    cin >> arr[i];  
}
```

Ввод элементов массива
длины 5 (статический)

```
double dvarr [4][3];  
for (int row = 0; row < 4; row++){  
    for (int col = 0; col < 3; col++){  
        cin >> dvarr [row][col];  
    }  
}
```

Ввод элементов
двумерного массива 4 на 3
(статический)