

Глава 5 «Массивы»

§5.1 Массивы

Массив в C++

Массив — это упорядоченный набор однотипных элементов, к каждому из которых можно обратиться по индексу. Массивы есть в большинстве языков программирования и нужны для удобного хранения нескольких значений и быстрой работы с ними. Массив имеет постоянную длину, а в качестве индексов обычно выступают целые числа.

| | | | | |
|----|----|-----|----|----|
| A | | | | |
| 0 | 1 | 2 | 3 | 4 |
| 12 | 18 | -10 | 35 | 50 |

ИНДЕКС
элемента массива

ЗНАЧЕНИЕ
элемента массива

Объявление и инициализация массива.

Для того, чтобы объявить массив, необходимо сначала указать **тип переменных** которые будут использованы в массиве, затем его **имя** и в **квадратных скобках []** **длину**. Размер статического массива **обязан быть константой**, явно указанной или инициализированной **раньше**. Инициализируется он часто с помощью **.....** та.

Объявление массива целых чисел arr
длины 10

```
double arr[5];
```

Объявление массива вещественных чисел arr
длины 5

```
char arrch[7];
```

Объявление массива символов arrch длины

7

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

Инициализация массива циклом
for

Объявление и инициализация массива.

Инициализировать массив можно сразу при объявлении. Для этого необходимо после ранее упомянутых действий написать **знак равенства =**, а затем в **фигурных скобках {}** через запятую перечислить все его элементы. Важно, чтобы количество элементов в фигурных скобках **совпадало** с длиной массива.

```
int arr[5] = {6,0,4,10,-15};
```

Объявление массива целых чисел arr длины 5 и инициализация его элементами 6 0 4 10 -15

```
int arr[5] = {6,0,4};
```

В данном случае программа может выдать ошибку, так как длина массива = 5, а количество элементов в списке = 3, соответственно часть элементов будет не инициализирована

Элементы и индексы массивов

У каждого массива есть элементы. Их количество равно длине. У элемента имеется свой **индивидуальный порядковый номер**, который называется **индексом элемента массива**. Нумерация индексов начинается с **0**, то есть первый элемент имеет индекс 0, второй – 1 и т.д. По индексу можно **обратиться** к любому из элементов, достаточно указать его в `[]`.

```
arr[2] = 3;
```

Присвоение элементу массива с индексом 2 значение, равное 3

```
mass[3] = x * 2;
```

Присвоение элементу массива с индексом 3 значение, равное значению переменной x, умноженной на 2.

Массивы. Примеры программ.

```
#include <iostream>
using namespace std;

int main()
{
    int arr[5];
    for (int i = 0; i < 5; i++){
        arr[i] = i*2;
        cout << arr[i] << " ";
    }

    return 0;
}
```

Вывод

0 2 4 6 8

Программа, которая создает массив целых чисел длины 5, заполняет его удвоенным номером индекса элемента массива и выводит на экран результат.

Ввод

8

Вывод

125

Ввод

19

Вывод

235

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int arr[10];
    int x;
    cin >> x;
    int summ = 0;
    arr[0] = x;
    summ += arr[0];
    for (int i = 1; i < 10; i++){
        arr[i] = arr[i - 1] + 1;
        summ += arr[i];
    }
    cout << summ;
    return 0;
}
```

Программа, которая создает массив целых чисел длины 10. Далее пользователь вводит число x, затем массив заполняется числами начиная с x, а каждое следующее число больше на 1 и вычисляется их сумма.

Ошибки при работе с массивами.

Выход за пределы границ массива.

```
int arr[5];  
arr[6] = 3;
```

Это ошибка, так как массив имеет длину 5, у него не существует элемента с индексом 6.

```
int arr[5];  
arr[-2] = 3;
```

Это ошибка, так как отрицательных индексов элементов массива не существует.

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int arr[7];  
    for (int i = 0; i < 10; i++){  
        arr[i] = i + 5;  
    }  
    return 0;  
}
```

Этот код ошибочный, так как при выполнении цикла мы выходим за пределы массива.

Ошибки при работе с массивами.

Не указана длина при объявлении массива (память не выделена).

```
#include <iostream>
using namespace std;

int main()
{
    int arr[];
    for (int i = 0; i < 10; i++){
        arr[i] = i;
    }
    return 0;
}
```

Этот код ошибочный, так как при объявлении массива не указана его длина, соответственно программа не может быть выполнена из-за того, что память не выделена.

Ошибки при работе с массивами.

Не соответствие типов переменных при
объявлении и инициализации.

```
#include <iostream>
using namespace std;

int main()
{
    double arr[10];
    for (int i = 0; i < 10; i++){
        arr[i] = "*";
        cout << arr[i];
    }
    return 0;
}
```

Этот код
ошибочный,
так как
объявлен
массив типа
double, а в
цикле
элементам
этого массива
мы
присваиваем
тип char.

```
#include <iostream>
using namespace std;

int main()
{
    int arr[10];
    double x;
    for (int i = 0; i < 10; i++){
        x = i;
        arr[i] = x / 2.0;
        cout << arr[i];
    }
    return 0;
}
```

Этот код
ошибочный, так
как объявлен
массив типа int, а
в цикле мы делим
два числа типа
double,
соответственно
результат будет
double.