

# Программирование на C++



| Минцифры  
РОССИИ

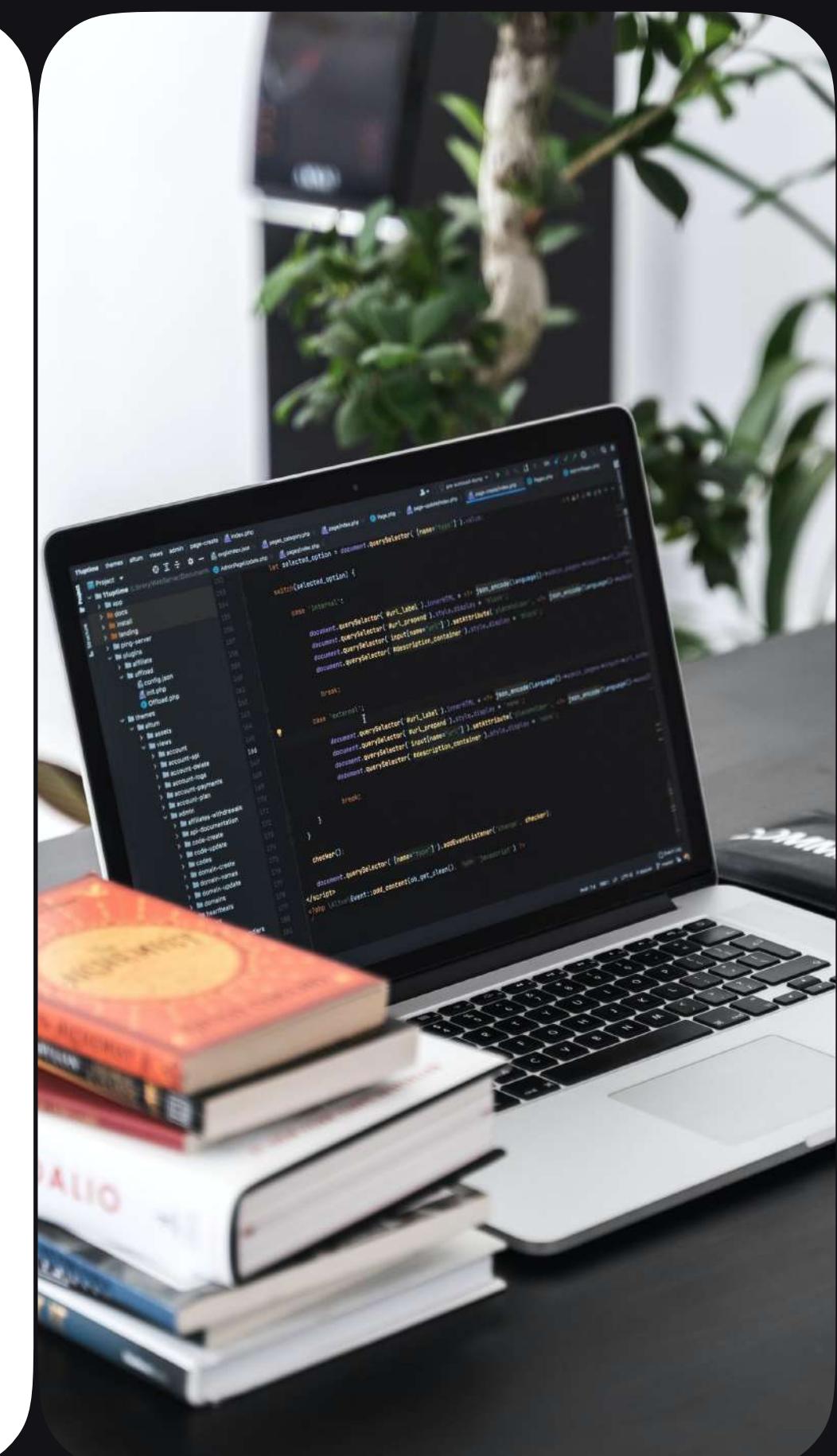
UCHI DOMA

**20.35**  
УНИВЕРСИТЕТ

Урок 10 Модуль 1

# Массив

Полезные материалы



## Задача



int a, b, c

Переменные используются  
для хранения единичных значений.

Как хранить и обрабатывать  
большие объемы данных?

?

# Цели урока

- ★ изучить массивы
- ★ отработать на практике составление алгоритмов с хранением данных в массиве на Си



# Массив

**Массив** — последовательность объектов одинакового типа, обозначаемая одним именем.



Каждый элемент массива имеет номер — индекс



Нумерация элементов массива начинается с 0



Все элементы массива должны быть одного типа

$$m = [m_0, m_1, \dots, m_{n-1}]$$

# Объявление и инициализация массивов

тип имя[размерность] = {инициализация};

Объявление массива с именем «а» содержащего  
10 целочисленных элементов

`int a[10];`

**Инициализация массива** – присвоение элементам массива  
начальных значений

`int a[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}; // массив а из 10 целых чисел`

# Пример инициализации массива

Если количество значений для инициализации меньше, чем количество элементов, то всем остальным элементам массива присваивается значение 0

```
int b[10] = {0}; // массив b из 10 элементов, инициализированных 0
```

Если массив инициализируется при объявлении, то количество элементов можно не указывать

```
int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

# Элемент массива

При обращении к элементам массива индекс требуемого элемента указывается в квадратных скобках [].

## Пример

Вывести на экран второй элемент массива

```
1 #include <stdio.h>
2 int main ( )
3 {
4     int m[5]={1,2,3,4,5};
5     printf("%d",m[1]);
6     return 0;
7 }
```

Результат работы программы:

2

# Пример ввода и вывода значений элементов массива

```
1 #include <stdio.h>
2 int main ( )
3 {
4     int a[5]; // объявлен массив a из 5 элементов
5     // Ввод элементов массива
6     for (int i=0; i<5; i++)
7     {
8         printf("a[%d]=", i);
9         scanf("%d", &a[i]); // &a[i] - адрес i-го элемента массива
10    }
11    // Вывод элементов массива
12    for (int i=0; i<5; i++)
13        printf("%d", a[i]); // пробел в формате печати обязателен
14    return 0;
15 }
```

# Пример ввода и вывода значений элементов массива

Результат работы программы:

```
a [0] = 1  
a [1] = 2  
a [2] = 3  
a [3] = 4  
a [4] = 5  
1 2 3 4 5
```

# Заполнение массива случайными значениями

```
1 #include <stdio.h>
2 #include <cstdlib>
3 #include <time> // содержит time()
4 int main()
5 {
6     int m[3] = {};
7
8     srand(time(NULL));
9     for (int i = 0; i < 3; i++)
10    {
11        m[i] = rand();
12        printf("%d", m[i]);
13    }
14    return 0;
15 }
```

Результат работы программы:

```
26125 17091 4767
```

# Заполнение массива случайными значениями

```
1 #include <stdio.h>
2 #include <cstdlib> ← содержит rand() и srand()
3 #include <time> ← содержит time()
4 int main()
5 {
6     int m[3] = {};
7
8     srand(time(NULL)); ← инициализацию генератора случайных
9     for (int i = 0; i < 3; i++) ← чисел rand
10    {
11        m[i] = rand(); ← генерация случайного целого значения
12        printf("%d", m[i]);
13    }
14    return 0;
15 }
```

time используется для того, чтобы инициализировать функцию rand каждый раз с разным параметром, иначе будут генерировать псевдослучайные числа из одной последовательности.