Информационные технологии и программирование


# Язык Си
# Массивы

Ракова Ирина Константиновна
каф.О7

# Массивы

**Массив** – это набор данных одного типа, имеющих общее имя и расположенных в памяти непосредственно друг за другом.

**Особенности:**


- все элементы имеют один тип
- весь массив имеет одно имя
- у каждого элемента есть свой индекс
- индексация начинается с 0
- все элементы расположены в непрерывной области памяти в порядке возрастания индексов
- выход за пределы массива в СИ не контролируется
- действия выполняются отдельно над каждым элементом

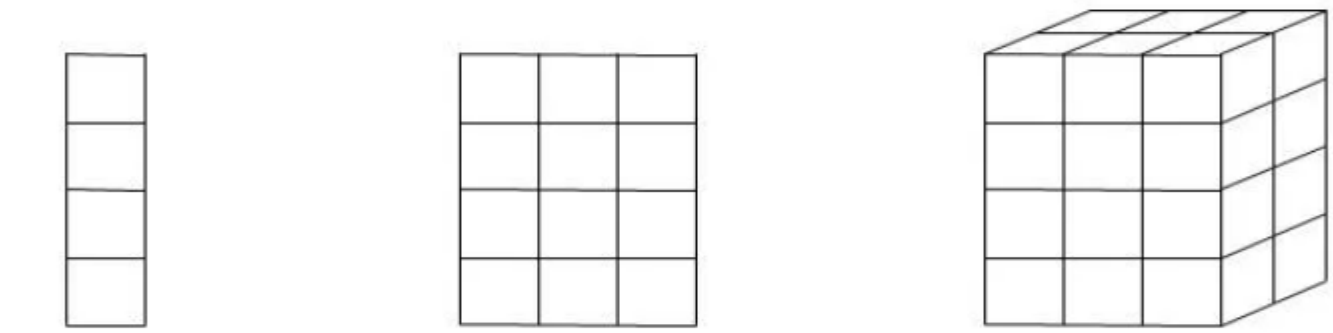
Примеры массива в жизни: дома на улице, книжная полка, строка таблицы, матрица.

# Массивы

Любой массив обладает:

- **именем**
- **типом элементов**
- **размерностью** (количеством измерений)
- **размером** (рангом, количеством элементов)

Различают одномерные массивы, двумерные (матрицы) и многомерные.



# Массивы

**С массивом как единым целым в языке Си ничего сделать нельзя!**

- Ввод, вывод и обработка массива всегда в цикле.
- У каждого элемента есть свой индекс (~ порядковый номер) целого типа.

Доступ к элементу массива осуществляется через имя и индекс в [ ] или через указатель.

- Индекс может быть задан переменной, константой, целочисленным выражением

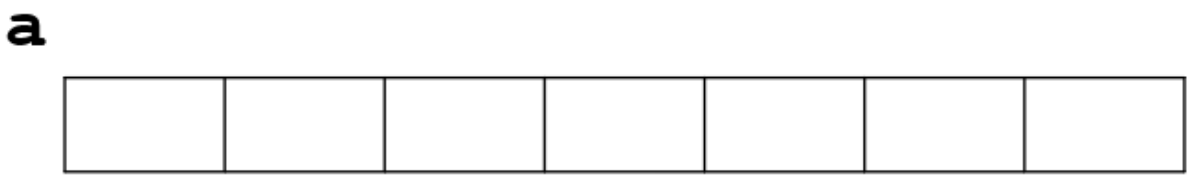
a[0] a[3] a[i] a[i+1]

Индексация элементов массива в Си начинается с **НУЛЯ**!

# Статические и динамические массивы

**По способу выделения памяти** массивы делятся на:

- **статические** – память под массив рассчитывается (выделяется) во время компиляции программы

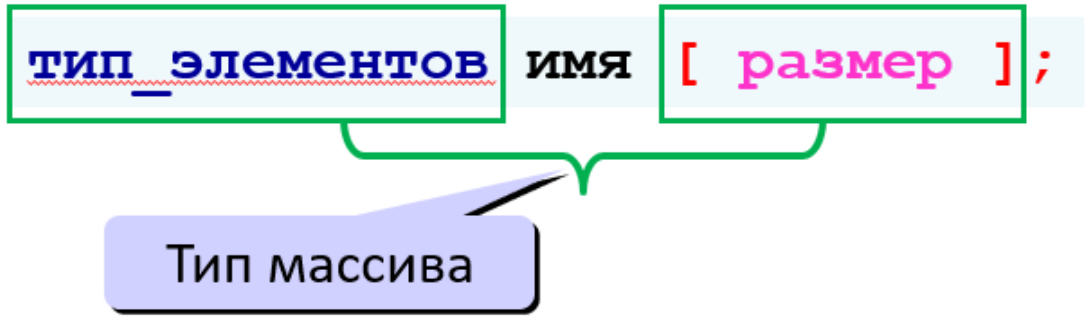



- **динамические** – память под массив выделяется в процессе работы программы


Алгоритмы обработки массивов и способы обращения к элементам массива не зависят от способа выделения памяти

# Статический массив

Общий вид объявления массива:




Примеры объявлений:

*int x[10], y[10];*
*double a[20];*
*char s[80];*

- Размер массива N может быть задан только константой, явной или символической. Используемое количество элементов может изменяться от 0 до N.
- Имя массива - указатель-константа на первый элемент массива.
- Имя массива является адресом первого элемента

  $a \equiv \&a[0]$
- Индекс элемента массива можно рассматривать как смещение элемента относительно начала массива

  $a + 3 \equiv \&a[3]$ $\quad *(a + 3) \equiv a[3]$

# Одномерный массив

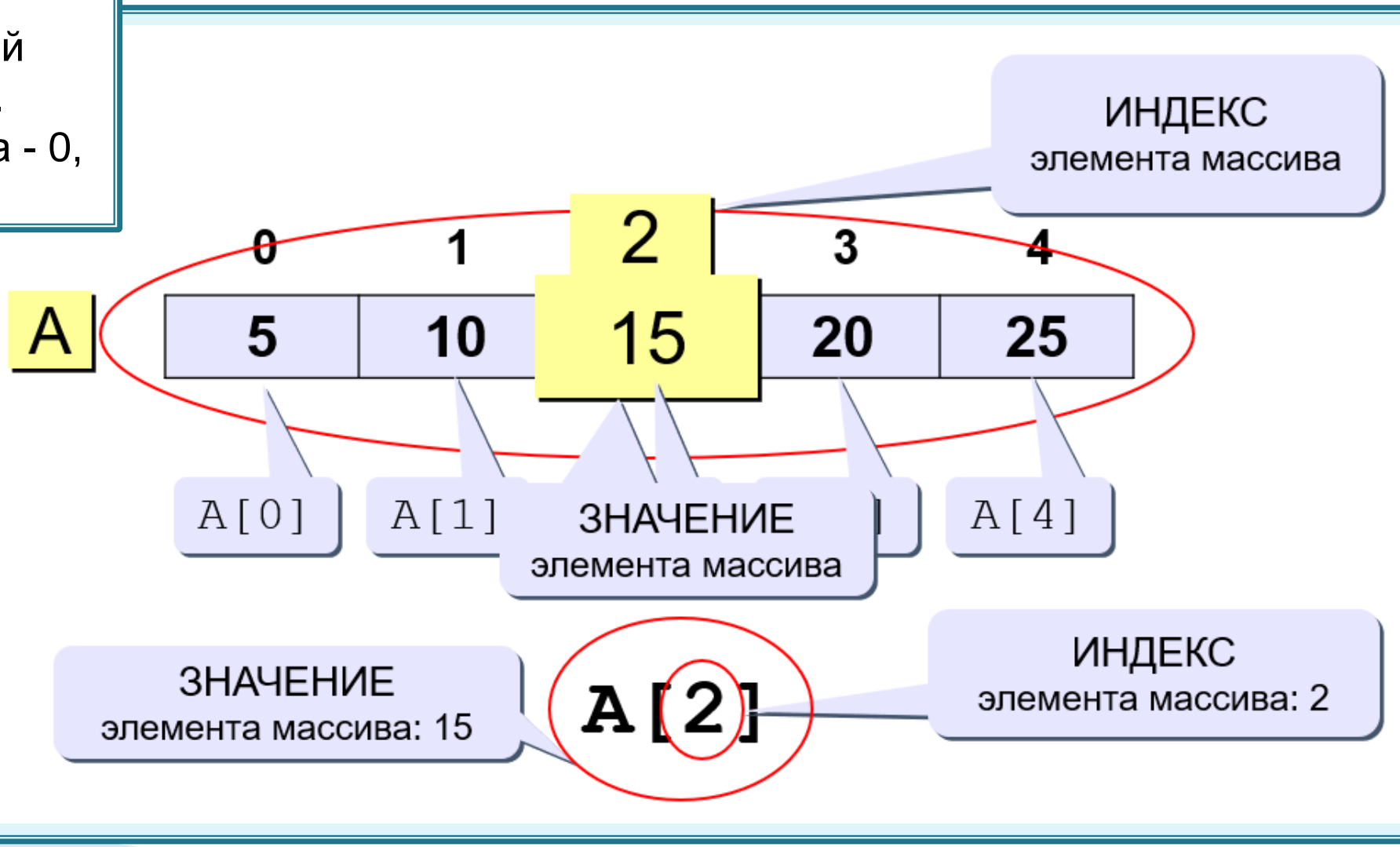
**int A[5];**
Объявлен целочисленный массив A из 5 элементов. Индекс первого элемента - 0, последнего – 4.

Обращение к 3 элементу массива

**A[2] ≡ *(A+2)**

С помощью индексного выражения

С помощью указателя-константы



# Объявление статических массивов

С использованием явной константы:

```
int x[10], y[10];
double a[20];
char s[80];
```

С использованием символической константы:

```
#define N 5
int m [N];
```

С инициализацией (присвоением начальных значений):

```
int a[4] = { 8, -3, 4, 6 };
double b[10] = { 1.1 };
int q[ ] = { 4, 0, -11 };
```

Если список инициализации меньше размера массива, то остальные элементы = 0

Размер массива определяется списком инициализации

Если начальные значения не заданы, в ячейках находится «мусор»!

# Пример

Ввести с клавиатуры массив из 5 элементов, умножить все элементы на 2 и вывести полученный массив на экран.

```
#include <stdio.h>
#define N 5
int main()
{
int a[N], i;
printf("Введите %d элементов:\n“,N);
for( i=0; i < N; i++ )
{
  printf (“a[%d] = ", i );
  scanf ("%d", & a[i] );
}
for( i = 0; i < N; i++ ) a[i] *= 2;
printf("Результат:\n");
for( i=0; i < N; i++ )
  printf("%4d", a[i]);
return 0;
}
```

Ввод элементов массива

Обработка массива

Вывод массива

**Ввод:**
*a[0] = 5*
*a[1] = 12*
*a[2] = 34*
*a[3] = 56*
*a[4] = 13*

**Результат:**
*10 24 68 112 26*

Для обработки массивов всегда используются циклы

# Задание элементов массива

Возможны разные способы задания элементов массива:

- Инициализацией при объявлении массива

```
int a[5] = { 8, -3, 4, 6, 2 };
```

- В результате ввода

```
for( i=0; i < N; i++ )
{   printf ("a[%d] = ", i );
    scanf ("%d", & a[i] );  }
```

- В результате вычислений, присваиванием

```
for( i=0; i < N; i++ ) a[i] = i+1;
```

- Заполнением случайными числами

```
srand (time(NULL));
for( i=0; i < N; i++ )
    a[i] =rand()%21-10;
```

Генерация случайных чисел:

*в библиотеке stdlib*

- *rand() - возвращает целое псевдослучайное число в диапазоне от 0 до RAND_MAX*
- *srand() – задает начальное значение (инициализирует) генератору случайных чисел*

*в библиотеке time*

- *time() – выдает значение текущего времени в сек. (истекшее с 0 часов 1 января 1970 года)*

# Примеры

Найти в массиве элементы, принадлежащие заданному отрезку [c; d], и их индексы.

```
for (i = 0; i < N; ++i )
   if ( a[i] >= c && a[i] <= d)
      printf (“a[%d]=%d\n”, i, a[i] );
```

Найти среднее арифметическое отрицательных элементов массива.

```
s=0; k=0;
for (i = 0; i < N; ++i )
   if ( a[i] < 0)
   { s+=a[i];
     k++; }
if (k) s=s/k;
```

Проверить, что все элементы в массиве положительные.

```
f = 0;
for (i = 0; i < N; ++i )
   if ( a[i] <= 0)
   { f++; break; }
if (f) printf (“No\n”);
else printf (“Yes\n”);
```

# Примеры

Найти значение максимума.

```
max = a[0];
for (i = 1; i < N; ++i )
   if ( a[i] > max)
      max = a[i];
printf (“max = %d\n”, max);
```

Найти индекс максимального элемента.

```
imax = 0;
for (i = 1; i < N; ++i )
   if ( a[i] > a[imax])
      imax = i;
printf (“imax = %d\n”, imax);
```

Определить, сколько максимумов в массиве.

```
max = a[0]; c = 1; // Нужен счетчик
for (i = 1; i < N; ++i )
   if ( a[i] > max)
      { max = a[i]; c = 1; }
   else if (a[i] == max) c++;
if (c == 1) printf (“Один\n”);
else if (c == N) printf (“Все равны\n”);
else printf (“Несколько - %d\n”, c);
```

# Примеры

Найти максимальный среди отрицательных элементов массива.

Поиск первого отрицательного

```
for (i = 0; i < N && a[i] >= 0; i++ );
if (i < N)
{
  max = a[i];
  for (i++; i < N; ++i )
     if (a[i] < 0 && a[i] > max)
        max = a[i];
   printf (“max = %d\n”, max);
}
else printf(“Нет отр.эл.\n”);
```

Если хоть один отрицательный нашелся

Поменять местами элементы с индексами 3 и 5 (метод трёх стаканов).

```
temp = a[3];
a[3] = a[5];
a[5] = temp;
```

Поменять порядок следования элементов в массиве на обратный.

```
for (i = 0, k = N-1; i < k; i++, k-- )
{
   temp = a[i];
   a[i] = a[k];
   a[k] = temp;
}
```

# Примеры

Сформировать новый массив из положительных элементов заданного.

```
// Надо описать 2 массива по N эл.
for (i = 0, k = 0; i < N; i++ )
  if (a[i] > 0)
    b[k++] = a[i];
printf (“Результат:\n”);
// Реально в новом мас. будет k эл.
for (i = 0; i < k; i++ )
   printf (“%d\n”, b[i]);
```

Удалить в массиве один элемент с индексом *k*.

```
for (i = k; i < N-1; i++ )
   a[i] = a[i+1];
```

*При вставке элемента сдвиг надо начинать с конца массива*

Удалить из массива все нулевые и отрицательные числа.

```
// Вместо того, чтобы удалять
// лишнее, оставляем нужное
for (i = 0, k = 0; i < N; i++ )
  if (a[i] > 0)
    a[k++] = a[i];
```

# Примеры

Удалить из массива максимальный элемент, если он есть и он единственный.

```
#include <stdio.h>
#define N 50 // Макс.размер массива
int main()
{
int a[N], i, n, imax, f;
printf("Введите размер массива:\n“);
scanf(“%d”, &n);  // Реальный размер
if (n > N) n = N;
printf("Введите %d элементов:\n“,n);
for( i=0; i < n; i++ )   // Ввод массива
{
  printf (“a[%d] = ", i );
  scanf ("%d", & a[i] );
}
```

```
imax = 0; f = 1;   // f - признак
// Проверка того, что max - один
for (i = 1; i < n; i++ )
  if (a[i] > a[imax])
    {  imax = i; f = 1; }
  else
    if (a[i] == a[imax]) f = 0;

n -= f;  // Изм. кол-ва элементов
if (f)     // Если надо удалять
   for (i = imax; i < n; i++ )
     a[i] = a[i+1];
printf (“Полученный массив:\n”);
for (i = 0; i < n; i++ )   // Вывод
   printf (“%4d”, a[i]);
return 0;
}
```

# Спасибо за внимание