

# Углубленный курс информатики

---

Составление программ с использованием  
одномерных массивов

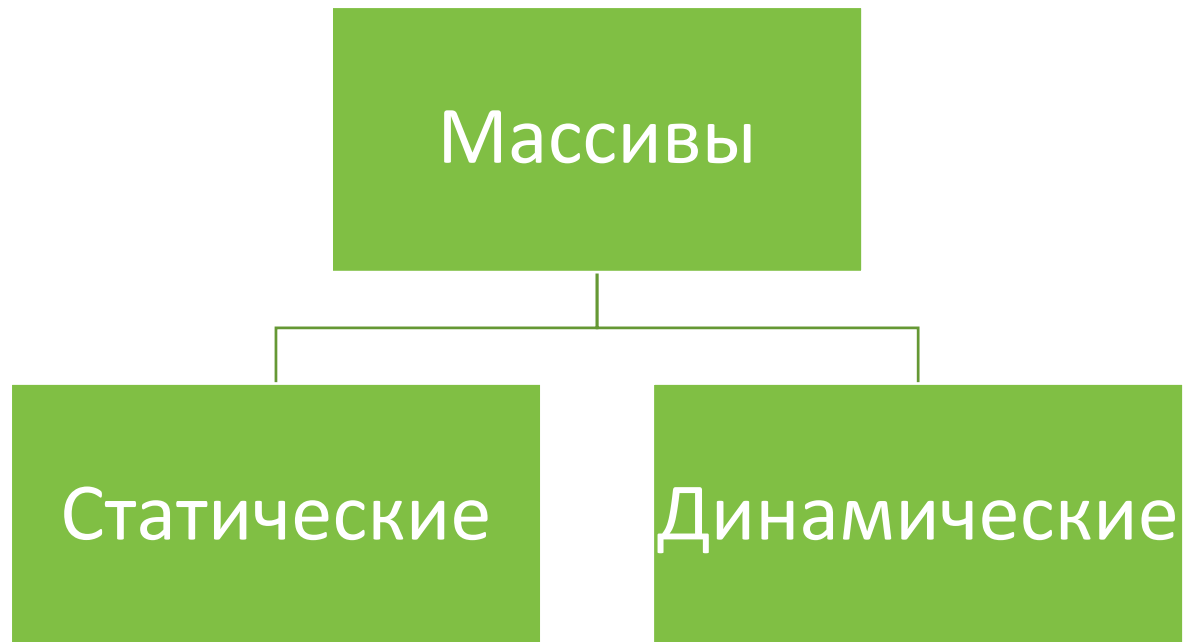


Чузлов Вячеслав Алексеевич

к.т.н., доцент ОХИ ИШПР

# МАССИВЫ

- Массив – это хранимая нумерованная последовательность однотипных элементов с непосредственным доступом к любому элементу по его номеру.
- Нумерация в массивах может начинаться от различных значений, а сами номера называются индексами элементов массива.



# СТАТИЧЕСКИЕ МАССИВЫ

- Статический массив имеет размерный тип и память под него распределяется на этапе компиляции программы.
- Одновременно может быть выполнена инициализация элементов массива.
- Границы индексов статического массива неизменны и должны быть указаны в программе константами или выражениями, содержащими только константы.

## Статический массив описывается в виде

**var** ИмяМассива: **array**[m..n] **of** Тип;

- Конструкция вида m..n описывает минимальное и максимальное значение, которое может принимать индекс массива. Эта конструкция задается константой порядкового типа.
- Количество элементов в массиве можно вычислить по формуле  $n - m + 1$ .
- Объявленный таким образом массив создается компилятором и под него отводится необходимое место в памяти в соответствии с типом элементов, поэтому для статических массивов описание и создание массива не разделяются.

# СТАТИЧЕСКИЕ МАССИВЫ

Описание статического массива можно совместить с инициализацией его элементов:

```
begin
  var a: array[3..6] of integer := (1, 2, 3, 4);
  Println(a);

  var b: array[0..2] of real := (1.2, 5, -3.05); // 5 - целочисленная
  Println(b);

  var c: array [1..4] of real := (10, 11, 12, 13); // все целочисленные
  Println(c)
end.
```

[1, 2, 3, 4]

[1.2, 5, -3.05]

[10, 11, 12, 13]

# ДИНАМИЧЕСКИЕ МАССИВЫ

- Динамический массив создается в том месте программы, где он впервые требуется.
- **Размер массива** – это количество элементов, которое в нем в данный момент содержится.
- Доступ к любому элементу массива осуществляется путем указания его имени, за которым в квадратных скобках следует индекс.
- В динамических массивах индексы начинаются с нуля.

**Динамический массив описывается в виде:**

```
var ИмяМассива : array of Тип;
```

- Переменная ИмяМассива будет являться всего лишь **ссылкой** на некоторое место в памяти, где должны располагаться элементы массива.
- Объявленный таким образом массив еще не создан, память под него не выделена и попытка обратиться к этому массиву вызовет ошибку.
- Поэтому следующим шагом должно быть распределение памяти под элементы массива.

# ДИНАМИЧЕСКИЕ МАССИВЫ

Наиболее просто распределить память под динамический массив, используя операцию **new**:

```
ИмяМассива := new Тип[КоличествоЭлементов]
```

Можно объединить описание массива с его созданием:

```
var a := new integer[15]; // самый компактный способ  
var b : array of real;  
b := new real[7];
```

# ДИНАМИЧЕСКИЕ МАССИВЫ

Распределение и перераспределение памяти под массив можно выполнить, используя процедуру **SetLength**(ИмяМассива, ЧислоЭлементов):

```
begin
  var a: array of integer;

  SetLength(a, 5); // теперь в массиве 5 элементов
  (a[0], a[1], a[2], a[3], a[4]) := (10, 11, 12, 13, 14);
  a.Println;

  SetLength(a, 7); // увеличим число элементов до 7
  a[6] := 16;
  a.Println
end.
```

```
10 11 12 13 14
10 11 12 13 14 0 16
```

*Элементу a[5] значение не было присвоено, поэтому оно осталось значением по умолчанию, которым инициализируются все элементы массива при выделении памяти под них.*

# ДИНАМИЧЕСКИЕ МАССИВЫ

- Актуальное количество элементов в динамическом массиве можно, получить при помощи вызова `ИмяМассива.Count`.
- Такой же результат дает `Имя Массива.Length`.

```
begin
  var a: array of integer;

  SetLength(a, 5);
  (a[0], a[1], a[2], a[3], a[4]) := (10, 11, 12, 13, 14);
  a.Count.Println;

  SetLength(a, 7);
  a[6] := 16;
  a.Length.Println
end.
```

5

7



# ГЕНЕРАТОРЫ ДИНАМИЧЕСКИХ МАССИВОВ

Функция `Arr( )` создает динамический массив из значений, заданных в качестве аргументов, поэтому все описание массива сводится лишь к указанию ключевого слова **var**.

```
begin
  var a := Arr(1, 2, 3, 4, 5, 6, 7);
  a.Println;

  var b := Arr(3.5, -2.7, 0.0, 13.9);
  b.Println
end.
```

```
1 2 3 4 5 6 7
3.5 -2.7 0 13.9
```

# ГЕНЕРАТОРЫ ДИНАМИЧЕСКИХ МАССИВОВ

Для целых значений параметров генерируются массивы элементов типа **integer**:

**ArrFill**(n, a) – n элементов со значением a;

**ArrRandom**(n, a, b) – n случайных чисел на интервале [a;b];

**ArrRandomInteger**(n, a, b) – аналог ArrRandom.

```
begin
  var arr1 := ArrFill(5, 0);
  arr1.Println;

  var arr2 := ArrRandom(10, 0, 10);
  arr2.Println;

  var arr3 := ArrRandomInteger(10, 0, 10);
  arr3.Println
end.
```

```
0 0 0 0 0
1 7 7 6 5 10 9 7 6 9
4 8 6 8 9 3 10 3 8 1
```

# ГЕНЕРАТОРЫ ДИНАМИЧЕСКИХ МАССИВОВ

Генераторы массивов из элементов типа **real**:

**ArrFill**(n, a) – n элементов со значением a типа real;

**ArrRandomReal**(n, a, b) – n случайных чисел на интервале [a; b]; a, b – типа real.

```
begin
  var arr1 := ArrFill(5, 0.5);
  arr1.Println;

  var arr2 := ArrRandomReal(10, 0, 10);
  arr2.Println
end.
```

```
0.5 0.5 0.5 0.5 0.5
8.93325934602565 2.61845979495834 4.6721544557587
6.3362589787395 9.89379467903347 4.56312559291866
9.39167760749891 8.13143739389788 2.12302495358653
4.50410369993379
```

# ВВОД ЭЛЕМЕНТОВ МАССИВА С КЛАВИАТУРЫ

- **ReadArrInteger(n)** – создание массива на основе ввода n элементов типа integer;
- **ReadArrInteger('текст приглашения', n)** – то же, с приглашением ко вводу;
- **ReadArrReal(n)** – создание массива на основе ввода n элементов типа real;
- **ReadArrReal('текст приглашения', n)** – то же, с приглашением ко вводу.

```
begin
  var arr1 := ReadArrInteger('Введите 5 целых чисел:', 5);
  arr1.Println;

  var arr2 := ReadArrReal('Введите 3 вещественных числа', 3);
  arr2.Println
end.
```

```
Введите 5 целых чисел: 1 2 3 4 5
1 2 3 4 5
Введите 3 вещественных числа 0.2 0.5 25.6
0.2 0.5 25.6
```

# ОПЕРАЦИИ СО ВСЕМИ ЭЛЕМЕНТАМИ МАССИВА

Для динамического массива `a` длиной `n` определены следующие функции и методы

- `a.Count`, `a.Length` и `Length(a)` возвращают `n` - длину массива;
- `a.High` и `High(a)` возвращают значение индекса последнего элемента массива длины `n`, равного `n-1`;
- `a.Low` и `Low(a)` возвращают `0` – значение индекса первого элемента массива.

Копию массива `a` можно получить при помощи функции `Copy(a)`. Такую копию можно присвоить другому динамическому массиву.

```
begin
  var a := ArrRandom(5, 0, 10);
  a.Println;
  var b := Copy(a); // копия массива a
  var c := a; // копируется ссылка и c - просто еще одно имя для a

  c[0] := 77; // a[0] тоже изменится
  a.Println;
  b.Println;
  c.Println
end.
```

```
1 10 3 3 5 (a)
77 10 3 3 5 (a)
1 10 3 3 5 (b)
77 10 3 3 5 (c)
```

# ПЕРЕБОР ЭЛЕМЕНТОВ МАССИВА В ЦИКЛЕ

- Для перебора всего массива удобнее всего использовать циклы **for** и **foreach**.
- Цикл **for** используют, если требуется работа с индексом массива, поскольку в нем доступ к элементу массива производится именно по индексу.
- Если нужно только получить значение элемента, проще использовать цикл **foreach**, но в теле этого цикла массив считается доступным лишь на чтение.

```
begin
  var arr := ArrRandom(10, -20, 30);
  arr.Println;

  for var i := 0 to arr.High do
    arr[i] := 2 * arr[i] + 1;
  arr.Println
end.
```

```
-16 -7 11 18 30 -5 -4 15 12 30
-31 -13 23 37 61 -9 -7 31 25 61
```

```
begin
  var arr := ArrRandom(10, -20, 30);
  arr.Println;

  var s := 0;
  foreach var item in arr do
    s += item * item - 5;
  s.Println
end.
```

```
6 9 -3 20 30 -17 5 -3 4 -5
1740
```

# МАКСИМУМ, МИНИМУМ, СУММА, ПРОИЗВЕДЕНИЕ И СРЕДНЕЕ

Максимальное **a.Max** и минимальное **a.Min** значения элементов массива:

```
begin
  var a := ArrRandom(10, -20, 30);
  a.Println;

  a.Max.Println;
  a.Min.Println
end.
```

```
26 30 -14 22 15 10 12 22 11 -5
30
-14
```

# МАКСИМУМ, МИНИМУМ, СУММА, ПРОИЗВЕДЕНИЕ И СРЕДНЕЕ

- Сумма значений всех элементов массива `a` может быть получена при помощи метода `a.Sum`.
- Произведение – при помощи метода `a.Product`,
- Метод `a.Average` вернет среднее арифметическое.

```
begin
  var a := ArrRandom(10, -20, 30);
  a.Println;

  a.Sum.Println;
  a.Product.Println;
  a.Average.Print
end.
```

```
13 -5 -12 8 -7 -1 20 10 -19 21
28
-3485664000
2.8
```



# МАКСИМУМ, МИНИМУМ, СУММА, ПРОИЗВЕДЕНИЕ И СРЕДНЕЕ

- Сумма значений всех элементов массива `a` может быть получена при помощи метода `a.Sum`.
- Произведение – при помощи метода `a.Product`,
- Метод `a.Average` вернет среднее арифметическое.

```
begin
  var a := ArrRandom(10, -20, 30);
  a.Println;

  a.Sum.Println;
  a.Product.Println;
  a.Average.Print
end.
```

```
13 -5 -12 8 -7 -1 20 10 -19 21
28
-3485664000
2.8
```

## Пример 1

Пусть дан массив  $a(5)$ , состоящий из следующих элементов:  $(-1, 2, 3, -5, -2.3)$ .  
Необходимо составить программу для вывода элементов данного массива на экран, при этом отрицательные элементы заменить их квадратами.

```
begin
  var a := Arr(-1.0, 2.0, 3.0, -5.0, -2.3);

  for var i := 0 to a.High do
    if a[i] < 0 then
      a[i] := sqr(a[i]);

  a.Print
end.
```

1 2 3 25 5.29

## Пример 2

Составьте программу для заполнения массива  $f(10)$  случайными числами в интервале  $[0; 9]$ . Определите элементы массива  $a(10)$  по следующей формуле:

$$a[i] = f[i] ** 2 - 0.5 \cdot f[i]$$

Выведите элементы массива  $a$  на экран.

```
begin
  var f := ArrRandom(10, 0, 9);
  var a := ArrFill(f.Length, 0.0);

  for var i := 0 to a.High do
    a[i] := f[i] ** 2 - 0.5 * f[i];

  a.Print
end.
```

```
0.5 0 45.5 45.5 3 45.5 33 3 60 7.5
```

## Пример 3

Пусть дан массив  $a(7)$ , состоящий из следующих элементов:

$$a(7) = 1.35, -1e-2, 0.8, 50.2, -3.8, 4.5, 1.6$$

**Необходимо найти:**

1. Минимальный элемент массива;
2. Максимальный элемент массива;
3. Сумму элементов массива;
4. Произведение элементов массива.

```
begin  
  var a := Arr(1.35, -1e-2, 0.8, 50.2, -3.8, 4.5, 1.6);  
  
  a.Max.Println;  
  a.Min.Println;  
  a.Sum.Println;  
  a.Product.Println  
end.
```

50.2

-3.8

54.64

14.8334976

## Пример 3

*Без использования методов Max, Min, Sum и Product*

```
begin
  var a := Arr(1.35, -1e-2, 0.8, 50.2, -3.8, 4.5, 1.6);
  var max := a[0];
  var min := a[0];
  var s := 0.0;
  var p := 1.0;

  for var i := 0 to a.High do
    begin
      s += a[i];
      p *= a[i];

      if a[i] > max then
        max := a[i];

      if a[i] < min then
        min := a[i];
    end;

    max.Println;
    min.Println;
    s.Println;
    p.Println
  end.
```

```
50.2
-3.8
54.64
14.8334976
```

# Задание 1

*Заполните массив  $a(7)$  случайными числами из интервала  $[0, 10]$  и выведите значения элементов массива на экран.*

**В полученном массиве необходимо найти:**

1. Индекс минимального элемента массива
2. Индекс максимального элемента массива
3. Сумму элементов массива
4. Произведение элементов массива, превышающих 1

## Задание 2

*Составьте программу для вычисления:*

$$y = \sum_{i=1}^4 (a_i + \sqrt{b}) + \prod_{j=1}^4 \sin(c_j)$$

где  $\sum$  - это сумма,  $\prod$  – произведение;

$b = 18$ ;

$a(4) = 2.1, 3.2, 4.8, 5.7$ ;

$c(4) = 2, 4, 6, 8$ .

## Задание 3

Составьте программу для вычисления количества отрицательных ( $K_o$ ) и количества положительных ( $K_p$ ) элементов массива  $B(10)$ . Полученные значения подставьте в формулу для вычисления  $K$ :

$$K = \frac{K_o + K_p}{K_o}$$

$B(10) = 2, 5, -6, 8, -3, 7, 12, -45, 106, 4$



## Задание 4

Составьте программу для заполнения элементов массива  $V(7)$  случайными числами из интервала  $[-3; 10]$  и выведите полученные значения на экран. Если наибольший элемент массива  $V(7)$  превышает значение  $c$ , то все элементы массива нужно разделить на  $a$ , в противном случае – умножить на сумму положительных элементов массива. Результат вывести на экран.

$$a = 7.7; \quad c = 9.5$$




# КОНТАКТНАЯ ИНФОРМАЦИЯ


---

**ЧУЗЛОВ ВЯЧЕСЛАВ АЛЕКСЕЕВИЧ**

к.т.н., доцент ОХИ ИШПР

 Учебный корпус №2, ауд. 136

 +7-962-782-66-15

 [chuva@tpu.ru](mailto:chuva@tpu.ru)