

# МАССИВЫ



## *ОДНОМЕРНЫЕ МАССИВЫ*



Презентацию подготовила

Ученица 11 Б

Карапетян Наташа.

# Представление массива

*При обработке набора данных одного типа в языке Паскаль имеется специальная структура, называемая массив.*

**Массив** - это структура, представляющая собой упорядоченную совокупность элементов одного типа, объединенных одним именем.

*Массив в Паскале имеет фиксированное количество элементов.*

*Имя массива, количество элементов и тип элементов массива определяются пользователем при описании массива.*



Для получения доступа к элементу массива используется индекс.

**Индекс массива** - это величина целого типа, характеризующая положение элемента в массиве.

Каждому массиву, используемому в программе, выделяется место в памяти.

Массиву отводится не одна ячейка, а последовательность расположенных друг за другом ячеек, в каждую из которых записывается значение соответствующего элемента.

### Пример:

Представим массив, состоящий из 88 элементов типа real, в котором элементы нумеруются от 11 до 99.

	x [11]	x [12]	x [13]	...	x [98]	x [99]
Индекс	11	12	13	...	98	99
Значение	1.23	1.61	0.01	...	1.34	1.91

# Описание массива



Описание массива включает в себя:

- 1) *Имя массива:***
- 2) *Ключевое слово `array`;***
- 3) *Диапазон изменения индексов, определяемый граничной парой, например: `1..40`. `1..N`. `5..20`.***

*Нижняя граница показывает наименьшее возможное значение индекса, верхняя - наибольшее. Нижняя граница не может превосходить верхнюю.*

*Нижняя граница отделяется от верхней двумя точками;*

- 4) *Тип элементов массива.***

Описание массива строится по следующей схеме:

**<имя массива> : array [граничная пара] of <тип элементов>**

(массив)

(из)

*Если несколько массивов имеют один и тот же тип, и одинаковый диапазон изменения индекса, то их описание можно объединить, разделив имена массивов запятыми:*

**Var**

**< имя 1, имя2.....>: array [граничная пара] of <тип эл-тов>**

**Примеры:**

**Var**

**Vector: array [1 ..60] of integer;**

**xBlock: array [40..50] of real;**

**X1,x2: array [0..60] of integer;**



## Действия над элементами массива

После объявления массива каждый его элемент можно обработать, указав имя массива и индекс элемента в квадратных скобках.

*Например, запись `xBlock [42]`, `Vektor[10]` позволяет обратиться к элементу массива `xBlock` с индексом 42 и элементу массива `Vektor` с индексом 10.*



# **Заполнение массива.**

**Заполнить элементы одномерного массива значениями мы можем:**

- вводя значения с клавиатуры;**
- случайным образом;**
- по формуле.**



**Надо помнить, что во всех трех случаях нам не обойтись без организации цикла.**

**Будем считать, что объявили массив из 10 элементов (тип элементов в данном случае значения не имеет).**

### 1. Заполнение с клавиатуры.

```
FOR i:=1 to 10 do  
begin  
writeln('введите значение элемента');  
readln(a[i])  
end;
```



### 2. Заполнение случайным образом.

Для этого мы должны подключить датчик случайных чисел.

```
FOR i:=1 to 10 do  
a[i]:=random(n); {n должно быть заранее задано}
```

### 3. Заполнение по формуле.

Каждый элемент массива должен быть рассчитан по формуле (например,  $a_i = \sin i - \cos i$ ).

```
FOR i=1 to 10 do  
a[i]:=sin(i)-cos(i);
```



## Вычисление суммы элементов

```
S:=0;  
FOR i:=1 to 10 do  
S := S + a[i];
```

## Вычисление среднего значения элементов массива

```
S:=0;  
FOR i:=1 to 10 do  
S := S + a[i];  
SR:= S/10;
```



## Поиск максимального (минимального) элемента массива.

Пусть мы имеем одномерный массив: 20,-2,4, 10,7,21,-12,0,4, 17.

Алгоритм поиска максимального (минимального) элемента мы построим так, чтобы сравнивать пару чисел, повторяя действие сравнения нужное количество раз.

Введем дополнительную переменную с именем `max`.

Она и будет одним из чисел, второе число — это очередной элемент массива.

Для того, чтобы провести первую операцию сравнения необходимо переменной `max` присвоить некоторое начальное значение.

Здесь могут быть два варианта:

- 1) присвоить переменной `max` первый элемент массива, тогда количество повторений операции сравнения равно  $n - 1$ ;
- 2) присвоить число заведомо меньшее всех элементов массива. В этом случае, количество повторений операции сравнения  $n$ .



# KOHEIL

