

# ***МАССИВЫ***



**Автор: Юдахина Татьяна Михайловна,  
Учитель информатики МОУ  
Октябрьская СОШ  
Радищевского района Ульяновской  
области**

# Понятие массива

- **Массив** – это упорядоченная совокупность однородных величин, обозначенных одним и тем же именем с различными целочисленными индексами, изменяющимся по порядку;
- Это пронумерованная последовательность однотипных данных

# Для чего?

- Позволяет упорядочить элементы и тем самым облегчить их массовую обработку. При этом упрощается распознавание элементов массива, т. к. достаточно указать его порядковый номер.

# Модели массива

- Стеллажи с данными о книгах (картотека)
- Комод с ящиками
- Места в кинотеатре и т. д.



# Характеристики массива

- **Имя** массива задается так же, как и имя переменной, с учетом типа переменных.
- **Размерность** - количество измерений (одномерный или двумерный)
- **Размер** - количество элементов массива.
- **Тип** - это тип используемых элементов (числовой или символьный)

# Специальный оператор

- **Оператор DIM.** Он необходим для того, чтобы транслятор мог выделить необходимую область памяти для хранения всех элементов массива. В описании массива должно быть указано **ИМЯ**, **ЧИСЛО ЭЛЕМЕНТОВ**, содержащихся в массиве, или **ДОПУСТИМЫЙ ДИАПАЗОН ИНДЕКСОВ** и **ТИП** каждого элемента. Одним оператором DIM можно описать сразу несколько массивов. Например, **DIM A (8), B\$ (11), 0% (5,5)**
  - Чтобы обратиться к элементу массива, надо указать его **ИМЯ И ИНДЕКС**. Например, следующий оператор присваивает элементу массива с индексом 5 значение 1: **A(5) = 1**
- Индекс массива записывается в круглых скобках.

# Обработка массива

- С точки зрения программиста, массив представляет собой список данных, который часто приходится обрабатывать как одно целое. Последовательно применить одни и те же действия ко всем элементам массива позволяет цикл со счетчиком. В этом случае переменную цикла используют как индекс при обращении к элементу массива или для вычисления такого индекса.

- **DIM F (10)**  
**FOR I=1 TO 10**  
**.....**  
**NEXT I**

# Заполнение массива

- Датчиком случайных чисел,
- вводить с клавиатуры (INPUT)
- использовать операторы DATA, READ
- по формулам
- По условию



# Случайными числами

- **Пример 1.**

Программа	Пояснения
<pre>DIM A (10) FOR I=1 TO 10 <b>A(I) = INT(RND (I)*100)</b> PRINT A (I)  NEXT I</pre>	<p>Задается массив</p> <p>Открывается цикл</p> <p>Заполняется массив датчиком чисел</p> <p>Выводятся элементы массива на экран</p> <p>Закрывается цикл</p>

- Если в массиве должны быть числа с разными знаками, то используется запись:

$A(I) = \text{INT}(\text{RND}(I) * 100) - \text{INT}(\text{RND}(I) * 100).$

# С помощью INPUT

## Пример 2.

### Программа для создания одномерного массива

Программа	Пояснения
<pre>INPUT N DIM F (N) FOR I = 1 TO N INPUT F (I) NEXT I FOR I = 1 TO N PRINT F (I); NEXT I</pre>	<p>Вводится количество элементов массива</p> <p>Задается массив</p> <p>Цикл для ввода с клавиатуры элементов массива</p> <p>Цикл для вывода элементов массива на экран</p>

# С помощью DATA READ

- **Пример 3.**

**Определить количество положительных элементов массива C (10), элементами которого являются числа: 8, -15, -5, 0, 1, 4.5, -1.2, 61, 59, -3.**

- **K = 0**
- **DIM C (10)**
- **DATA 8, -15, -5, 0, 1, 4.5, -1.2, 61, 59, -3**
- **FOR I=1 TO 10**
- **READ C (I)**
- **IF C (I) > 0 THEN K = K + 1**
- **NEXT I**
- **PRINT «Количество положительных элементов =»; K**
- **END**

- **Пояснение: Оператор DATA подразумевает перечисление заранее известных данных через запятую, а оператор READ предполагает использование этих данных при решении одно за другим. Эти два оператора используются всегда вместе. (Работает не во всех версиях Basic)**

- **При подсчете суммы элементов начальное значение должно быть равным 0!**

# С помощью формулы

- **Пример 4.**
- **Распечатать двумерный массив, который заполняется с помощью формулы  $T_{IJ} = I * J$**
- **DIM T%( 9, 9)**
- **FOR I = 1 TO 9**
- **FOR J = 1 TO 9**
- **T%( I, J) = I\*J**
- **PRINT T%( I, J);**
- **NEXT J**
- **PRINT**
- **NEXT I**
- **END**
- **Данная программа печатает произведения чисел из таблицы умножения.**

# С помощью условия

- **Пример 5.**
- **Заполнить массив  $10*10$  так, чтобы на главной диагонали стояли 1, остальные - 0**
- **DIM F (10, 10)**
- **FOR I = 1 TO 10**
- **FOR J = 1 TO 10**
- **IF I=J THEN F (I,J)=1 ELSE F (I,J)=0**
- **PRINT F (I, J);**
- **NEXT J**
- **PRINT**
- **NEXT I**
- **END**

Без комментариев...

# Основные алгоритмы работы с массивами

- Нахождение суммы элементов массива
- Нахождение произведения элементов массива
- Нахождение минимального (максимального) элемента
- Менять местами элементы (до половины) в одномерном массиве

# Нахождение суммы элементов массива

- есть одномерный массив A (10)-(уже создан) см выше
- Входим в созданный массив
- $S=0$  – обозначаем сумму
- FOR I=1 TO 10
- $S= S+A(I)$
- NEXT I
- PRINT «Сумма элементов массива равна»; S
- END

# Нахождение произведения элементов массива

- есть одномерный массив  $A$  (10)-(уже создан) см выше
- Входим в созданный массив
- $P=1$  – обозначаем произведение
- FOR I=1 TO 10
- $P = P * A(I)$
- NEXT I
- PRINT «Произведение элементов массива равно»; P
- END



# Нахождение max(min) элемента массива

*есть одномерный массив A(10)-(уже создан) см выше*

**Входим в созданный массив**

- **M=A(1)** – **обозначаем максимальный элемент**
- **FOR I=2 TO 10**
- **If A(I)>M THEN M=A(I) ...A(I)<M**
- **NEXT I**
- **PRINT «Наибольший элемент равен»; M...**  
**наименьший**
- **END**

# Перевернутый массив

```
10 INPUT "Введите размер";N
20 DIM A(N)
30 FOR I=1 TO N
40 INPUT A(I)
50 NEXT I
70 M=INT(N/2)
80 FOR I=1 TO M
90 P=A(I)
100 A(I)=A(N-I+1)
110 A(N-I+1)=P
120 NEXT I
130 ? "Перевернутый массив"
140 FOR I=1 TO N
150 PRINT A(I);
160 NEXT I
170 END
```

# Практическая работа

- Создать и вывести на экран массив, элементами которого являются результаты умножения на «5» чисел от 1 до 10.

