

# СТРУКТУРИРОВАННЫЕ ДАнные.

Строки. Множества

# **Программирование обработки строк**

Строка – последовательность символов (длина от 1 до 255), которая при использовании в выражениях, заключается в апострофы

Тип данных :

**VAR**

**S: STRING[50]; {длина до 50 символов}**

**S1: STRING; {длина до 255 символов}**

**Обращение к элементу строки – по индексу, как в массиве:**

**S:='qwertyuio';**

**S[6]:='z'; → S:='qwertzuiio';**

**Распределение памяти строковой переменной:**

0	1	2	3	4	5	6	7	8	9	10	11	12	13
9	q	w	e	r	t	z	u	i	o				

# Операции со строками

**Операция присваивания:**

```
Var S1:string; s2:string[3]; s:string;
```

```
begin
```

```
...
```

```
S1:='qwertyuiop';
```

```
S2:=S1; {S1='qwe'}
```

**Операция сцепления, конкатенации:**

```
S:= 'TURBO'+ 'PASCAL';
```

```
{='TURBOPASCAL'}
```

# Операции со строками

Операции отношения:

Сравнение слева-направо до несовпадения :

Выражение	Результат
<b>‘Hello’ = ‘Hello, ‘</b>	<b>False</b>
<b>‘Hello’ &lt; ‘Hello, ‘</b>	<b>True</b>
<b>‘hello’ &gt; ‘HELLO’</b>	<b>True</b>
<b>‘Hello’ &gt; ‘Hell’</b>	<b>True</b>

# Функции и процедуры работы с символами и строками

**CHR(S)** – возвращает значение символа по коду:

**S:=CHR(60) → S='<'**

**ORD(C)** – возвращает код символа:

**X:=ORD('+') → X=43**

**LENGTH(S)** – возвращает длину строки:

**L:=LENGTH('QQQ') → 3**

**COPY(S,P,L)** – возвращает **строку** длиной **L**, начиная с позиции **P**

**S1:=COPY('asdfghjkl;',4,3) → 'fgh'**

**INSERT(S,S1,P)** – вставляет в строку **S** строку **S1**, начиная с позиции **P**

**INSERT('asdfgh','A13',3) → 'asdA13fgt'**

**DELETE(S,P,N)** – удаляет из строки **N** символов, начиная с позиции **P**

**DELETE('QWERTYUIOP', 5, 2) → 'QWERTIOP'**

**POS(S1, S) – возвращает номер позиции строки S, в которой обнаружен первый символ S1:**

**X:= POS('67','123456789') → X=6**

**STR(X[:m:n], S) – преобразует число X в строку S:**

**STR(12.564567:0:3, S) → '12.564'**

**VAL(S, X, ERR) – преобразует строку S в число X, если ошибка – в переменную ERR (Int) возвращается номер позиции ошибки**

**VAL('12.5678',X, O) → X=12.5678**

**VAL('12.567D8',X, O) → O=7**



Написать программу, сортирующую строки массива строк по алфавиту

```
FOR I:=1 TO N-1 DO  
FOR J:=I+1 TO N DO  
IF M[I,1]>M[J,1] THEN  
  BEGIN  
    S:=M[I];  
    M[I]:=M[J];  
    M[J]:=S;  
  END;
```

Написать программу, удаляющую пробелы из введенной строки:

### 1. Начальные

```
WHILE S[1]=' ' DO  
DELETE(S,1,1);
```

### 2. Конечные

```
WHILE S[LENGTH(S)]=' ' DO  
DELETE(S,LENGTH(S),1);
```

### 3. Сдвоенные

```
WHILE POS('__',S)>0 DO  
DELETE (S,POS('__',S),1);
```

# СТРУКТУРИРОВАННЫЕ ДАННЫЕ.

## Множества

**Множества** – набор элементов одинакового типа, которые рассматриваются как единое целое

Количество элементов - **МОЩНОСТЬ**

Тип элементов множества – **базовый тип**

### Может быть:

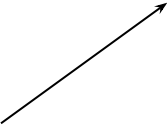
- Стандартным :char, byte, boolean
- Перечисляемым
- Диапазоном

}] Не более  
256  
элементов

ОПИСАНИЕ:

**<ИМЯ>: SET OF БАЗОВЫЙ ТИП;**

Ключевые  
слова



Пример:

**VAR**

**DAY : SET OF (SUN, MON, TUE, WED, THU, FRI, SAT);**

**CHARSET : SET OF CHAR;**

**DIGITS : SET OF 0..9;**

**В МНОЖЕСТВЕ НЕ МОЖЕТ БЫТЬ ОДИНАКОВЫХ  
ЭЛЕМЕНТОВ**

# Задание элементов множества

При задании значений элементов множества используются квадратные скобки:

$[1,2,3,4]$ ;  $['a','b','c']$ ,  $['a'..'z']$

Если множество не имеет элементов, оно называется **пустым** и обозначается  $[\ ]$

ТИП «МНОЖЕСТВО» ЗАДАЕТ НАБОР ВСЕХ ВОЗМОЖНЫХ ПОДМНОЖЕСТВ ЭЛЕМЕНТОВ (ВКЛЮЧАЯ ПУСТОЕ).

ЗНАЧЕНИЕ ПЕРЕМЕННОЙ СОДЕРЖИТ **ОДНО КОНКРЕТНОЕ ЗНАЧЕНИЕ** ПОДМНОЖЕСТВА

ПРИМЕР:

**VAR**

**X : 1..3; { X – МОЖЕТ БЫТЬ 1 ИЛИ 2 ИЛИ 3 }**

**X\_SET : SET OF 1..3;**

**{X\_SET – МОЖЕТ БЫТЬ**

**[],[1],[2],[3],[1,2],[1,3],[2,3],[1,2,3] }**

# Операции над множествами

Знак	Название	Результат
$:=$	Присваивание	Множество
$+$	Объединение	Множество
$*$	Пересечение	Множество
$-$	Дополнение	Множество
$=$	Тождественность (совпадение)	Логический
$\langle \rangle$	Нетожественность (не совпадение)	Логический
$\leq$	Содержится в	Логический
$\geq$	Содержит	Логический
$\text{in}$	Принадлежность	Логический

Сформировать массив **N** случайных неповторяющихся чисел

**A: SET OF BYTE** ; K,N,X:BYTE; **M:ARRAY[1..15] OF BYTE;**

BEGIN RANDOMIZE; **A:= [ ]**; K:=1;

    READLN(N);

    WHILE **K<= N** DO

BEGIN

        X:=RANDOM(N)+1;

        IF **NOT(X IN A)** THEN

**BEGIN**

**A:=A+[X];**

**M[K]:=X; K:=K+1;**

END;

END;

END.