

# Множество

в языке Pascal

**Множеством** называется неупорядоченный ограниченный набор однотипных данных.

Все элементы множества должны принадлежать одному из скалярных (простых) типов, кроме вещественных. Этот тип называется **базовым** типом множества. Базовый тип задается перечислением или диапазоном значений.

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

Количество элементов множества называется его **мощностью** (максимальное количество 255).

В множестве не может быть повторяющихся элементов.

**Область значений типа множество** – набор всевозможных подмножеств, составленных из элементов базового типа.

# Способы описания множеств

1. **Var** ИМЯ МНОЖЕСТВА : **set of** ТИП ЭЛЕМЕНТОВ;

2. **Type** ИМЯ ТИПА = **set of** ТИП ЭЛЕМЕНТОВ;  
**Var** ИМЯ МНОЖЕСТВА : **ИМЯ ТИПА**;

где,

*set of* - служебные слова ( множество из);

*тип элементов* – любой порядковый.

*Например:*

**Type** mn = **set of** char;

**Var** a,b: mn;

**c: set of** 2..6;

**Begin**

    a:=['A'..'Z'];

    c:=[2, 4..6];

    b:=[];

    . . .

**End.**

# Операции над множествами

1. **Операция IN** - используется для проверки принадлежности какого-либо значения указанному множеству.

Обычно применяется в операторах условного перехода . Результат выполнения операции логического типа ( True или False).

## Пример:

```
Var    s:char;
        m1, m2: set of char;

Begin
    m1 := [ '0' .. '9' ];
    m2 := [ 'А' .. 'я' ];
    readln(s);
    If  s in m1 then writeln
('цифра')
    else if s in m2 then writeln
('буква');
    . . .
end.
```

# Операции над множествами

2. Объединение множеств (+) . Объединением двух множеств является третье множество, содержащее элементы обоих множеств.


$$A := [3, 5, 7..10, 15..20]$$

$$B := [1..4, 6, 10..12]$$

$$C := A + B$$

$$\{ C := [1..12, 15..20] \}$$



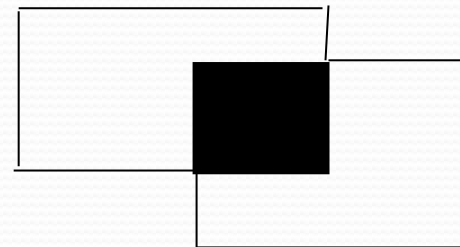
# Операции над множествами

3. Пересечение множеств (\*). Пересечением двух множеств является третье множество, которое содержит элементы, входящие одновременно в оба множества.

$A := [3, 5, 7..10, 15..20]$

$B := [1..4, 6, 10..12]$

$C := A * B \quad \{ C := [3, 10] \}$



# Операции над множествами

4 **Разность множеств (-)**. Разностью двух множеств является третье множество, которое содержит элементы первого множества, не входящие во второе множество.

$$A := [3, 5, 7..10, 15..20]$$

$$B := [1..4, 6, 10..12]$$

$$C := A - B \quad \{ C := [5, 7..9, 15..20] \}$$



# Операции над множествами

5. **Операция «равно» (=).** Два множества  $A$  и  $B$  считаются равными, если они состоят из одних и тех же элементов. Порядок следования элементов в сравниваемых множествах значения не имеет.

$A := [1, 2, 3]$

$B := [3, 2, 1]$

$A = B$     {true}

# *Операции над множествами*

6. **Операция «не равно» ( $\neq$ ).** Два множества  $A$  и  $B$  считаются не равными, если они отличаются по мощности или по значению хотя бы одного элемента.
7. **Операция «больше или равно» ( $\supseteq$ ).** Результат операции  $A \supseteq B$  равен `True`, если все элементы множества  $B$  содержатся в множестве  $A$ .
8. **Операция «меньше или равно» ( $\subseteq$ ).** Если все элементы множества  $A$  содержатся в множестве  $B$ , то результат равен `True`.

# Пример:

Операция	Значение A	Значение B	Выражение	Результат
не равно	[1,2,3]	[3,1,2,4]	$A \neq B$	True
Больше или равно	['e', 'r', 't']	['t', 'e']	$A \geq B$	True

# Ввод элементов множества

Элементом множества задаются значения оператором присваивания.

Пример:

```
Var zn: set of char;
```

```
Begin
```

```
    zn:=[';', '.', '?', ',', '!', ':', '-']; {знаки  
препинания}
```

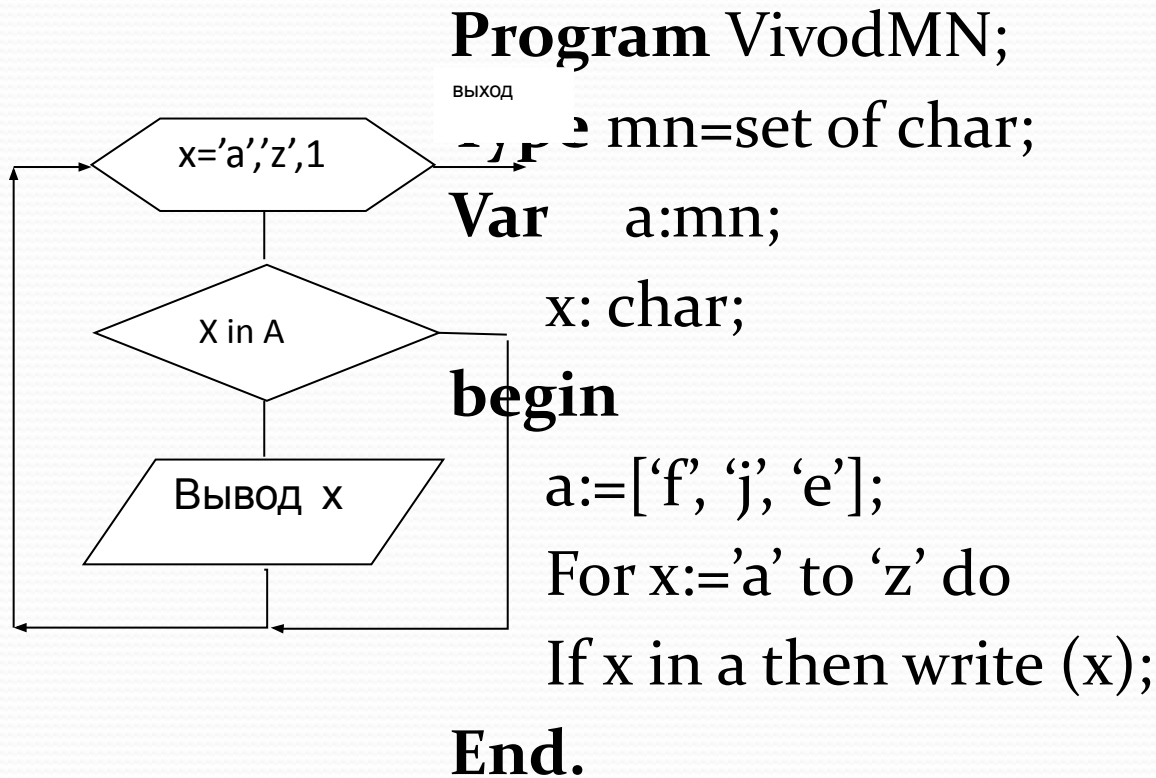
```
    ...
```

```
End.
```

# Стандартные процедуры работы с множествами

Имя и параметры	Типы параметров	Действие
<b>Include (s , x)</b>	s - множество, x - элемент, тип которого совместим с базовым типом множества	включает элемент x во множество s
<b>Exclude (s , x)</b>	s - множество, x - элемент, тип которого совместим с базовым типом множества	исключает элемент x из множества s

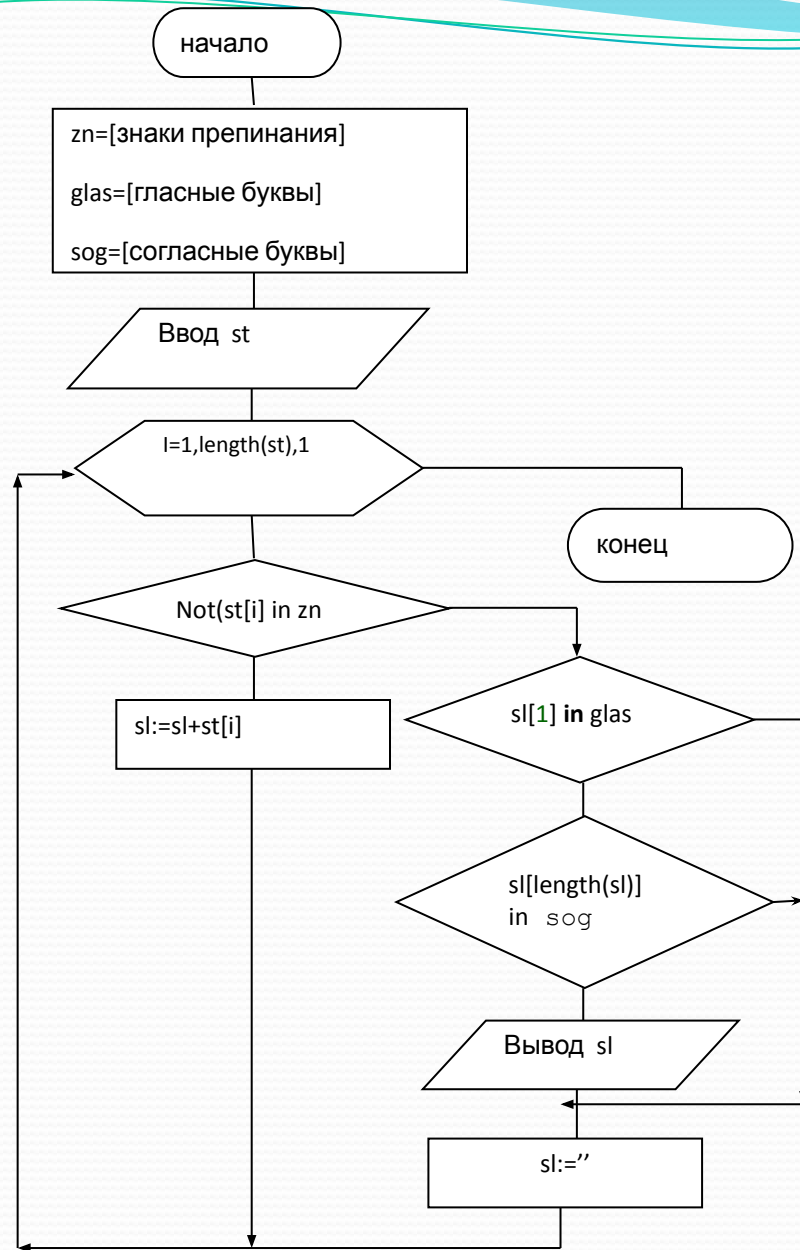
# Вывод элементов множества





## *Пример 1:*

Дана строка символов из строчных русских букв , слова разделены любым знаком препинания . Вывести слова начинающиеся с гласной буквы , а оканчивающиеся - согласной.



```
program Mnog;
```

```
const
```

```
zn=['.',':',',','?', '!', ''];
```

```
glas=['a','e','и','o','y','ю','я','э','ы','ё'];
```

```
sog=['б','в','г','д','ж','з','й','к','л','м','н','п','р','с','т','ф','х','ц','ч','ш','щ'];
```

```
var
```

```
St,sl:string;
```

```
i:integer;
```

```
begin
```

```
Writeln('Введите строку');
```

```
readln(st);
```

```
For i:=1 to length(st) do
```

```
If not(st[i] in zn) then
```

```
sl:=sl+st[i] else
```

```
begin
```

```
if (sl[1] in glas) and
```

```
(sl[length(sl)] in sog) then Writeln(sl);
```

```
sl:='';
```

```
end;
```

```
end.
```

## Результат решения задачи

Введите строку  
мама,арбуз помой!  
арбуз