



# МНОЖЕСТВО В ЯЗЫКЕ PASCAL

# ПОНЯТИЕ

**Множество** — это структурированный тип данных, представляющий собой набор взаимосвязанных по какому-либо признаку или группе признаков объектов, которые можно рассматривать как единое целое.

**Каждый объект в множестве называется элементом множества.**



Все элементы множества должны принадлежать одному из порядковых типов, содержащему не более 256 значений.

Этот тип называется **базовым типом** множества. Базовый тип задается диапазоном или перечислением.

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

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




Множество в памяти хранится как **массив битов**, в котором каждый бит указывает является ли элемент принадлежащим объявленному множеству или нет.

**Максимальное число** элементов множества **256**, а данные типа множество могут занимать не более **32 байт**.

Число байтов, выделяемых для данных типа множество, вычисляется по формуле:

$$\text{ByteSize} = (\text{max div } 8) - (\text{min div } 8) + 1,$$

где max и min — верхняя и нижняя границы базового типа данного множества.



Номер байта для конкретного элемента E вычисляется по формуле:

$$\text{ByteNumber} = (E \text{ div } 8) - (\text{min div } 8),$$

номер бита внутри этого байта по формуле:

$$\text{BitNumber} = E \text{ mod } 8$$

Не имеет значения порядок записи элементов множества внутри конструктора.

Например, [1, 2, 3] и [3, 2, 1] — это эквивалентные множества.

Каждый элемент в множестве учитывается только один раз. Поэтому множество [1, 2, 3, 4, 2, 3, 4, 5] эквивалентно [1..5].

Переменные множественного типа описываются так:

**Var <идентификатор> : set of <базовый тип>;**



НАПРИМЕР:

**Var**

**A, D : Set Of Byte;**

**B : Set Of 'a'..'z';**

**C : Set Of Boolean;**



Нельзя вводить значения во множественную переменную процедурой ввода и выводить процедурой вывода.

Множественная переменная может получить конкретное значение только в результате выполнения оператора присваивания:

**<множественная переменная> := <множественное выражение>;**

Например:

**A := [50, 100, 150, 200];**

**B := ['m', 'n', 'k'];**

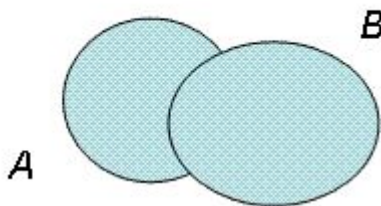
**C := [True, False];**

**D := A;**



# ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

**Объединением** двух множеств  $A$  и  $B$  называется множество, состоящее из элементов, входящих хотя бы в одно из множеств  $A$  или  $B$ . Знак операции объединения в Паскале «+».



Примеры:

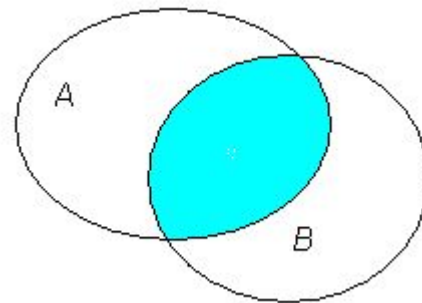
- 1)  $[1, 2, 3, 4] + [3, 4, 5, 6] \Rightarrow [1, 2, 3, 4, 5, 6]$
- 2)  $[] + ['a'..'z'] + ['A'..'E', 'k'] \Rightarrow ['A'..'E', 'a'..'z']$
- 3)  $[5 < 4, \text{true and false}] + [\text{true}] \Rightarrow [\text{false}, \text{true}]$



# ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

**Пересечением** двух множеств  $A$  и  $B$  называется множество, состоящее из элементов, одновременно входящих во множество  $A$  и во множество  $B$ .

Знак операции пересечения в Паскале « $*$ »

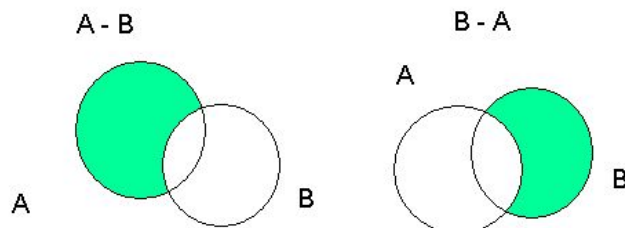


Примеры:

- 1)  $[1, 2, 3, 4] * [3, 4, 5, 6] \Rightarrow [3, 4]$
- 2)  $['a'..'z'] * ['A'..'E', 'k'] \Rightarrow ['k']$
- 3)  $[5 < 4, \text{true and false}] * [\text{true}] \Rightarrow []$

# ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

**Разностью** двух множеств  $A$  и  $B$  называется множество, состоящее из элементов множества  $A$ , не входящих во множество  $B$ .



Примеры:

1a)  $[1, 2, 3, 4] - [3, 4, 5, 6] \Rightarrow [1, 2]$

1b)  $[3, 4, 5, 6] - [1, 2, 3, 4] \Rightarrow [5, 6]$

2a)  $['a'..'z'] - ['A'..'E', 'k'] \Rightarrow ['a'..'j'], ['i'..'z']$

2b)  $['A'..'E', 'k'] - ['a'..'z'] \Rightarrow ['A'..'E']$

3a)  $[5 < 4, \text{true and false}] - [\text{true}] \Rightarrow [\text{false}]$

3b)  $[\text{true}] - [5 < 4, \text{true and false}] \Rightarrow [\text{true}]$


# ОПЕРАЦИЯ ВХОЖДЕНИЯ

Это операция, устанавливающая связь между множеством и скалярной величиной, тип которой совпадает с базовым типом множества.

Если  $x$  — такая скалярная величина, а  $M$  — множество, то операция вхождения записывается так:  **$x \text{ in } M$** .

Результат — логическая величина `true`, если значение  $x$  входит в множество  $M$ , и `false` — в противном случае.

Например,  $4 \text{ in } [3, 4, 7, 9]$  — `true`,  $5 \text{ in } [3, 4, 7, 9]$  — `false`.



# ЗАДАЧА 1.

Дана строка. Сохранить в ней только первые вхождения символов, удалив все остальные.

```
program ex_set_3;
var m : set of char;
s : string;
i : byte;
begin
write('Введите строку: ');
readln(s);
m := [];
i := 1;
while i <= length(s) do
if s[i] in m then
delete(s, i, 1)
else begin
m:=m+[s[i]];
i := i + 1
end
end while;
```

## ЗАДАЧА 2. (САМОСТОЯТЕЛЬНАЯ РАБОТА)

В городе имеется  $n$  высших учебных заведений, которые производят закупку компьютерной техники. Есть шесть компьютерных фирм: «Диалог», «Avicom», «Нэта», «Сервер», «Декада», «Dega.ru». Ответить на следующие вопросы:

- 1) в каких фирмах закупка производилась каждым из вузов?
- 2) в каких фирмах закупка производилась хотя бы одним из вузов?
- 3) в каких фирмах ни один из вузов не закупал компьютеры?

Решим задачу с использованием множеств. Для удобства дальнейших манипуляций в порядке следования занумеруем компьютерные фирмы, начиная с единицы. Занесём информации о месте закупок компьютеров каждым из вузов в отдельное множество.

Ответ на первый вопрос можно получить, выполнив пересечение всех таких множеств.

Ответ на второй вопрос – результат объединения множеств.

И, наконец, на последний – разность множества всех фирм и множества фирм, где хотя бы один вуз делал покупки.



### ЗАДАЧА 3. (САМОСТОЯТЕЛЬНАЯ РАБОТА)

Сгенерировать  $n$  множеств  
(нумерацию начать с 1). Вывести  
элементы, которые входят во все  
множества с номерами, кратными  
трём, но не входят в первое  
множество.

