

ТЕМА: Ввод и отладка программ с переменными множественного типа Операции над множествами.

Множество в языке Паскаль представляет собой набор различных элементов одного (базового) типа. Отличие термина "множество" в Turbo Pascal от его математического понимания в том, что множества в Turbo Pascal могут включать только элементы порядковых типов.

Базовый тип – это совокупность всех возможных элементов множества. Всего в базовом типе должно быть не более 256 различных элементов. Значение переменной множественного типа может содержать любое количество различных элементов базового типа – от нуля элементов (пустое множество) до всех возможных значений базового типа, то есть возможными значениями переменных множественного типа являются все подмножества базового типа.

Множества могут быть описаны следующим образом:

```
Type <имя типа> = Set of <тип элементов>;
Var <имя множества>: <имя типа>;
```

либо:

```
var <имя множества>: Set of <тип элементов>;
```

Например:

```
Type mnog = set of char;
Var mn1: mnog;
    mn2: set of char;
    mn3: set of 'A'..'Z';
    a: set of byte;
    b: set of 100..200;
```

Здесь **mn1** и **mn2** – это множества символов (различных символов всего 256); **mn3** – множество больших латинских букв;

a – множество целых чисел (так как тип *byte* содержит целые числа от 0 до 255, его можно использовать в качестве базового типа); **b** – множество чисел в диапазоне от 100 до 200.

Формирование (конструирование) множеств: В программе элементы множества задаются в квадратных скобках, через запятую. Если элементы идут подряд друг за другом, то можно использовать диапазон.

Например:

```
Type mnog = set of 1..200;
Var a, b, c, d: mnog;
Begin
    a:=[3, 9, 67, 5];
    b:=[7..60];
    c:=[1..10, 80..90];
    d:=[ ];
```

Операции над множествами.

Объединением двух множеств называется множество элементов, принадлежащих хотя бы одному из этих множеств. Знак операции объединения множеств в математике – “ $\cup$ ”, в Паскале “+”.

Примеры:

- 1) ['A', 'F'] + ['B', 'D'] = ['A', 'F', 'B', 'D'];
- 2) [1,2,7,4,6,8] + [3,4,5,6] = [1..8]
- 3) [1..3,5,7,11] + [3..8,10,12,15..20] = [1..8,10..12,15..20]

Пересечением двух множеств называется множество элементов, принадлежащих одновременно к первому и второму множеству. Знак операции пересечения множеств в математике – “ $\cap$ ”, в Pascal – “ * “.

Примеры:

1) ['A' , 'F'] * ['B' , 'D'] = [], так как общих элементов нет.

2) [1,2,3,4] * [3,4,5,6] = [3,4]

3) [1..3,5,7,11] * [3..8,10,12,15..20] = [3,5,7]

Разностью двух множеств называется множество, состоящее из тех элементов первого множества, которые являются элементами второго. Знак операции вычитания множеств в математике “ \ ”, в Pascal “ - “.

Примеры:

1) [' A ' , ' F '] - [' B ' , ' D '] = [' A ' , ' F ']

2) [1, 2, 3, 4] - [3, 4, 5, 6] = [1, 2]

3) [1..3, 5, 7, 11] - [3..8, 10, 12, 15..20] = [1, 2, 11]

Операция *определения принадлежности элемента* множеству. Математическое обозначение операции – “ $\in$ “, в Pascale – “ in “. Результат операции имеет значение true, если элемент входит в множество, и false в противном случае.

Примеры

4 in [3, 4, 6, 8] = true

' a ' in [' A ' .. ' Z '] = false

Сравнение множеств.

Для сравнения множеств используются операции отношения:

= проверка на равенство (совпадение) двух множеств;

<> проверка на неравенство двух множеств;

<=, < проверка на вхождение первого множества во второе;

>=, > проверка на вхождение второго множества в первое.

Множества представляют собой гибкий и наглядный механизм для решения многих задач. В частности, их удобно использовать при обработке строк и текстов, а также при работе с упорядоченными совокупностями чисел.

Пример. " Решето Эратосфена ". Составить программу поиска простых чисел в промежутке [1..n]. Число n вводится с клавиатуры.

Решение.

Простым называется число, которое не имеет других делителей, кроме единицы и самого этого числа. Идея этого метода заключается в следующем: формируется множество М, в котором содержатся все числа заданного промежутка. Затем из множества последовательно удаляем элементы, кратные 2, 3, 4 и т.д. до $n \div 2$. После этого в множестве М останутся только простые числа.

```
Program Simple;
Var m:set of byte;           {Объявление множества М}
    i,k,n:integer;
Begin
    Write('n='); readln(n);  {Ввод числа N с клавиатуры}
    M:=[2..n];               {Формирование элементов множества М}
    For k:=2 to n div 2 do    {Внешний цикл по параметру k – делитель}
        For i:=2 to n do     {Внутренний цикл по параметру i перебирающий все элементы
                               множества от 2 до N}
            If (i mod k=0) and (i<>k) then
                m:=m-[i];    {Тело цикла. Проверяем если каждый элемент i имеет делитель k,
                               т.е. без остатка делится на k (функция mod возвращает остаток от
                               деления), т.е. если остаток равен 0 и делимое не совпадает с
```

делителем, тогда этот элемент удаляем из множества М. В множестве остаются только те элементы которые не имеют никаких делителей кроме самих себя т.е. простые числа }

```

For i:=1 to n do {Для всех чисел i в диапазоне от 1 до N}
  If i in m then write(i:3); {Если число i принадлежит множеству М, то выводим этот элемент}
Readln;
End.

```

Пример2. Заданы два множества. Вывести на экран результат объединения этих множеств, проверить на вхождение элементов множества А в множество В.

Теперь продемонстрируем, как ввести множество, как записываются операции над множествами и как организовать вывод элементов множества. При выводе элементов множества приходится проверять наличие каждого возможного элемента на предмет вхождения данного элемента в выводимое множество. Если множество определено как **c: set of 1..10**, то это означает, что элементами множества могут быть числа **1, 2, 3, 4, 5, 6, 7, 8, 9, 10**. В цикле для всех **i** от **1** до **10** необходимо провести проверку вхождения числа **i** в выводимое множество **c**.

```
for i := 1 to 10 do if i in c then write(i:3);
```

Программа:

```

program operation;
type
menge = set of 1..10; {Задан новый тип - множество}
var a,b,c : menge; {Описаны переменные множественного типа menge}
i, n : integer;
begin
writein ('Введите элементы множества a. Конец ввода - 0');
a := [];
repeat {Цикл для ввода множества a}
readln(i);
if i in [1..10] then a := a + [i];
until i = 0;
writeln ('Введите элементы множества b. Конец ввода - 0');
b := [];
repeat {Цикл для ввода множества b}
readln(i)
if i in [1..10] then b := b + [i];
until i = 0;
c := a + b; {Операция - объединение множеств}
writeln ('Объединение множеств');
for i := 1 to 10 do if i in c then write(i:3);
writeln;
writeln; {Проверка на подмножество}
if a <= b then writeln(' a содержится в b')
else writeln (' a не содержится в b')
end.

```

ЗАДАНИЯ

Для всех обязательно:

1. Дано натуральное число N. Составить программу, печатающую в порядке возрастания все не четные числа кратные 7 в промежутке [1..n]. Число n вводится с клавиатуры.

2. Дано натуральное число X . Составить программу, печатающую в порядке убывания все четные числа кратные 6 в промежутке $[1..x]$. Число x вводится с клавиатуры.
3. Дано натуральное число k . Составить программу, печатающую в порядке возрастания все не четные числа в промежутке $[1..k]$. Число k вводится с клавиатуры.
4. Дано натуральное число N . Составить программу, печатающую в порядке возрастания все числа кратные 3 в промежутке $[1..n]$. Число n вводится с клавиатуры.
5. Дано натуральное число N . Составить программу, печатающую в порядке убывания все не четные числа в промежутке $[1..n]$. Число n вводится с клавиатуры.
6. Дано натуральное число N . Составить программу, печатающую в порядке возрастания все числа кратные 11 в промежутке $[1..n]$. Число n вводится с клавиатуры.
7. Дано натуральное число k . Составить программу, печатающую в порядке возрастания все не четные числа кратные трем в промежутке $[1..k]$. Число k вводится с клавиатуры.
8. Даны два множества. Из множества A удалить четные элементы входящие в множество B .
9. Выдать элементы множества A кратные 13 и не входящие в множество B .
10. Выдать элементы множества A кратные 3 и не входящие в множество B .

На оценку «4»

11. Даны три множества выдать на экран пересечение первых двух множеств и объединение получившегося множества с третьим.
12. Даны три множества выдать на экран пересечение суммы первых двух множеств с третьим множеством.
13. Даны три множества выдать на экран пересечение разности первых двух множеств с третьим множеством.
14. Даны три множества выдать на экран объединение элементов пересечения множеств A и B с C .
15. Даны три множества выдать на экран пересечение первых двух множеств и разность получившегося множества с третьим.
16. Даны три множества выдать на экран разность элементов пересечения множеств A и B с C .
17. Даны три множества. Из третьего множества удалить элементы, одновременно принадлежащие первым двум множествам.
18. Даны три множества. Из третьего множества удалить элементы, первого множества не входящие во второе множество.
19. Даны три множества. Из третьего множества удалить элементы второго множества не входящие в первое множество.
20. Даны три множества. Найти объединение третьего множества с элементами, одновременно принадлежащими первым двум множествам.
21. Даны три множества. Найти пересечение третьего множества с элементами, одновременно принадлежащими первым двум множествам.
22. Даны три множества. Найти разность элементов одновременно принадлежащих первым двум множествам с третьим множеством.
23. Подсчитать сколько одинаковых элементов имеют два множества A и B .
24. Даны два множества. Найти произведение элементов одновременно принадлежащим обоим множествам.

25. Из данного множества A выдать элементы являющиеся делителями суммы элементов этого множества.

На оценку «5»

26. Даны три множества. Удалить из третьего множества элементы не принадлежащие первым двум множествам.
27. Даны два множества. Найти произведение элементов кратных 3 и одновременно принадлежащим обоим множествам.
28. Из данного множества A удалить элементы являющиеся делителями суммы элементов этого множества.
29. Даны два множества. Из множества A удалить элементы являющиеся делителями суммы элементов множества B .
30. Даны два множества. Выдать элементы множества A являющиеся делителями суммы элементов множества B .
31. Даны два множества. Выдать произведение элементов множества A на сумму элементов множества B .
32. Даны два множества. Выдать разность суммы элементов множества B с элементами множества A .
33. Подсчитать сколько разных элементов имеют два множества A и B .
34. Из данного множества удалить элементы кратные минимальному элементу.
35. Из данного множества удалить элементы кратные максимальному элементу.
36. Даны три множества. Из третьего множества удалить четные элементы входящие в множество A и нечетные элементы входящие в множество B .
37. Дано натуральное число N . Составить программу, печатающую в порядке возрастания все цифры, не входящие в десятичную запись данного натурального числа.
38. Составить программу для подсчёта количества гласных и согласных букв в заданном тексте и определения, каких букв больше (гласных или согласных); учесть, что в строке могут быть и другие символы, кроме букв.
39. Составить программу печати в алфавитном порядке (по одному разу) всех строчных гласных букв, входящих в заданный текст.
40. Составить программу печати всех символов заданного текста, входящих в него по одному разу.