

Структурированные типы в Pascal И+ПРГ

Pascal

Pascal

Структурированные типы данных в Pascal характеризуются множественностью образующих тип элементов. К этим типам относятся массивы, строки, множества, записи, файлы.

Множества

Этот тип соответствует математическому представлению о множествах.

Типе `<имя_типа> = set of <баз_тип>;`

где `имя_типа` – идентификатор типа множества,

- `set of` – ключевые слова описания типа: **Множество** из набора однотипных логически связанных объектов (не более 256, нумерация объектов начинается с 0 до 255),
- `баз_тип` – базовый тип элементов множества (любой порядковый тип, кроме `word`, `integer`, `longint`).

Можно задать множество прямо в разделе описания переменных:

Var `<имя_переменной> : set of <баз_тип>;`

Для задания констант множественного типа используется **Конструктор множеств**: заключенный в квадратные скобки список спецификаций элементов множества разделенных запятой. Спецификации элементов могут быть константы или выражения базового типа, а также тип-диапазон того же базового типа (например, `1..9`). Конструкция вида `[]` означает пустое множество. Порядок перечисления элементов базового типа в константах не имеет значения.

Структурированные типы в Pascal

Pascal

Множества

Pascal

Примеры определения и задания множественных типов:

```
type Caps = set of 'A' .. 'Z';
Colors = set of (RED, GREEN, BLUE);
Numbers = set of byte;
var oct : set of 0 .. 5;
```

```
type
  digitchar = set of '0' .. '9';
  digit = set of 0 .. 9;
var
  s1, s2, s3 : digitchar;
  s4, s5, s6 : digit;
begin
  .....
  s1 := ['1', '2', '3'];
  s2 := ['3', '2', '1'];
  s3 := ['2', '3'];
  s4 := [0 .. 3, 6];
  s5 := [4, 5];
  s6 := [3 .. 9];
  .....
end.
```

Тип "множество" задает набор всех возможных подмножеств входящих в него элементов (включая пустое). Если базовый тип, на котором строится множество имеет N элементов, то число подмножеств, входящих в это множество, равно 2^N .

А переменная типа "множество" содержит одно конкретное подмножество значений множества.

Например:

var s : 1 .. 3; - переменная интервального типа, принимает значения: 1, 2 и 3,

а множество

var m : set of 1 .. 3; может принимать

значения:

[] [1] [2] [3] [1, 2] [1, 3] [2, 3] [1, 2, 3]

Инициализация переменной-множества:

var index set of 1 .. 9 = [0, 2, 4, 6, 8,];

В Конструкторе в качестве элемента может быть выражение или диапазон: [R, a mod 15, 14] или [chr(65) .. chr(90)]

Pascal

Множества

Pascal

Операции над множествами.

Результат операции – множество

:= - Присваивание

+ - Объединение – элементы первого множества, дополненные элементами второго

***** - Пересечение – элементы, общие для обоих множеств

- - Разность – элементы первого множества, которые не принадлежат второму

Результат операции – логический

= - Тожественность – true, когда все элементы обоих множества одинаковы
(множества эквивалентны)

<> - Нетожественность – true, когда множества неэквивалентны

<= - Содержится в – true, если первое множество включено во второе

>= - Содержит – true, если второе множество включено во первое

in - Принадлежность – true, если первый (левый) операнд операции –выражение- принадлежит второму (правому) операнду -множеству (включен в него)

Процедуры

Include (s, i) - включает элемент **i** во множество **S**

Exclude (s, i) - исключает элемент **i** из множества **S**

Структурированные типы в Pascal

Pascal

Множества

Pascal

Пример: Программа – простой калькулятор на 4-е арифметических действия.
Завершение работы – ввод символа ';'.

```
Program CalcSet;
var  x,y,r : real;  op : char;
begin
  repeat
    write('Ввести op ->');      readln(op);
    write('Ввести x, y -> ');  readln(x,y);

    if op in ['+', '-', '*', '/', ';']
    then
      begin  case op of
        '+' : r:=x+y;
        '-' : r:=x-y;
        '*' : r:=x*y;
        '/' : r:=x/y;
        ';' :          end;
        writeln (x:4:2,' ',op,' ',y:4:2,' =', ' ',r:4:2)
      end
    else writeln('Ошибка ввода операции');
  until op=';';
end.
```

Pascal

Множества

Pascal

Практическое занятие – ПЗ-1

Дана последовательность русских слов (строчными буквами) разделенных пробелами (не менее 5-ти слов). Вывести на экран слова, в которых первая буква встречается ещё раз.

```
Program mnog_twolet;
uses crt;
Const text : string = 'рефракция много заноза факт фанфары';    world : string[10] = '';
Var    partworld : set of char ;    letter : char;    i : integer;

begin
  clrscr;
  letter := text[1];
  for i := 1 to length(text)+1 do
  begin
    if (text[i] <> ' ') and (i <= length(text))
    then
      begin world := world+text[i];    include (partworld, text[i+1]);    end
    else
      begin if (letter in partworld)
        then
          writeln('Слово с повторяющейся первой буквой: ', world);
          letter := text[i+1];    partworld := [];    world := '';
        end;
      end;
    end;
  end;
end.
```

Структурированные типы в Pascal

Pascal

Множества

Pascal

ПЗ-2

Посчитать количество гласных и согласных букв в предложении.

```
program Glasn_Sogl;
Type mnoj= set of 'A'..'я';  (* Задаем тип данных - множество букв русского алфавита *)
var glasn, sogl: mnoj;
    sr: string;              (* строковая переменная sr хранит вводимый текст *)
    i: byte;                (* параметр цикла *)
    g, s: byte;            (* переменные накапливающие количество гласных и согласных букв
                           соответственно *)

begin
  (* Задаем множества гласных букв перечислением, а множество согласных вычитанием из
     всего алфавита гласных букв, мягкого и твердого знаков *)
  glasn:=['A','a','O','o','E','e','Ё','ё','И','и','У','у','Ы','ы','Э','э','Я','я','Ю','ю'];
  sogl:=['A'..'я'] - glasn - ['Ь','ь','Б','б'];
  write('Введите предложение: ');
  readln(sr);
  (* Обнуляем счетчики количества *)
  g:=0;
  s:=0;
  (* Просматриваем все элементы предложения и ищем их в множествах glasn и sogl *)
  For i:=1 to Length(sr) do begin
    if sr[i] in glasn then g:=g+1;
    if sr[i] in sogl then s:=s+1;
  end;
  writeln('В данном предложении ',g,' гласных и ',s,' согласных букв');
End.
```

Структурированные типы в Pascal

И+ПРГ

Pascal

Множества

Pascal

ПЗ-3

Дана последовательность русских слов (строчными буквами) разделенных пробелами (не менее 10-и слов). Вывести на экран все звонкие согласные буквы (б, в, г, д, ж, з, л, м, н, р), которые входят в нечётные слова и не входят ни в одно четное слово.

Контрольная строка: ура ребята здесь много дел где можно браво наломать дров

Результат: Звонкие согласные входящие в нечетные слова и не входящие в четные: **Ж З Л**

```
Program mnog_zvon;
Const   mas : string[10] = 'бвгджзлмнр';
Var     letters, index1, index2, index : set of char; text: string;  ns, i: integer;
Begin
  letters := ['б','в','г','д','ж','з','л','м','н','р'];
  index := []; index1 := []; index2 := [];
  writeln('Введите 10 русских слов строчными буквами. ');
  writeln('Введённые слова разделяйте пробелами. По окончании нажмите Enter. ');
  readln(text);
  ns := 1;
  for i := 1 to length(text) do
  begin
    if (text[i] in letters) then
      if (ns mod 2) <> 0 then include (index1, text[i]) else include (index2, text[i]);
      if (text[i] = ' ') then inc(ns);
    end;
  index := index1 - index2;
  writeln('Звонкие согласные входящие в нечетные слова и не входящие в четные: ');
  for i:=1 to length(mas) do
    if (mas[i] in index) then
      write(mas[i], ' ');
  end.
```