

Примеры программ на ЯП Паскаль. Часть 2

Перечисляемый и ограниченный типы данных

```
type Days = (Mon, Tue, Wed, Thu, Fri, Sat, Sun);
```

```
...
```

```
var day: Days;
```

```
...
```

```
day:=Wed;
```

```
...
```

```
if day>Fri then writeln('Сегодня выходной');
```

```
...
```

```
if day=Mon then writeln('Началась рабочая неделя');
```

Пример использования перечисляемых типов:

Пусть корабль может двигаться только по четырем направлениям.

- Текущее направление движения определяется переменной типа `Directions = (North, West, South, East)`;
- Этому кораблю можно подавать приказы следующих типов: вперёд, назад, влево, вправо, то есть любой приказ задать переменной типа `Orders = (Ahead, Back, Left, Right)`.
- Пусть корабль двигался по направлению `d1`, после чего ему дали приказ `p`. Программа должна определить, в каком направлении корабль будет двигаться после выполнения приказа.

Program Ship;

type Courses = (North, West, South, East);

Orders = (Ahead, Back, Left, Right);

var d1, d2: Courses;

p: Orders;

i: integer;

s: string;

begin

{Запрашиваем у пользователя информацию о курсе и приказе}

writeln('Введите первоначальный курс корабля ');

write('(0-север, 1-запад, 2-юг, 3-восток) ');

readln(i);

d1:=Courses(i);

writeln('Введите приказ, данный кораблю ');

write('(0-прямо, 1-назад, 2-влево, 3-вправо) ');

readln(i);

p:=Orders(i);

{Определяем новый курс}

```
case p of      {если d1=1, то ((ord(d1)+2) mod 4)=3}
  Ahead : d2:=d1;  {если d1=2, то ((ord(d1)+2) mod 4)=0}
  Back  : d2:=Courses((ord(d1)+2) mod 4);
  Left  : d2:=Courses((ord(d1)+1) mod 4);
  Right : d2:=Courses((ord(d1)+3) mod 4);
          {-1 - нельзя, так как (Ord(North)-1) mod 4 = -1}
end;
{Выводим результат на экран}
case d2 of
  North : s:='север';
  West  : s:='запад';
  South : s:='юг';
  East  : s:='восток';
end;
writeln('Теперь корабль движется на '+s);
readln;
end.
```

Ограниченный тип данных

- В качестве **базового** типа могут использоваться почти все **порядковые** типы.
- Порядковые типы: все целые типы (byte, integer, и т. п.), char, boolean, перечисляемые типы и ограниченные типы.

Допустимы такие объявления:

```
type tSmallLatin = 'a'..'z';    {малые латинские буквы}  
      tHolidays  = 6..7; {выходные}
```

```
Var Hol: tHolidays;
```

...

```
Hol:= 7; {Sunday}
```

Множественный тип данных

Синтаксис

Type <имя типа> = Set Of <базовый тип>;

S1:= ['A', 'B', 'C']

S2:= [0, 1, 5, 2, 8]

S3:= [0..23]

S4:= [1, 2, 3, 10..20]

К множествам можно применять
логические выражения!

- тождественность/нетождественность
- принадлежность
- объединение
- пересечение
- разность

Множественный тип данных

Тождественность:

$$[1, 2, 3, 4, 5] = [2, 1, 5, 4, 3] \Rightarrow \text{True}$$

Принадлежность:

$$5 \text{ In } [1, 2, 3, 4, 5] \Rightarrow \text{True}$$

Объединение:

$$[1, 2, 3, 4] + [3, 4, 5] = [1, 2, 3, 4, 5]$$

Пересечение:

$$[1, 2, 3, 4] * [3, 4, 5] = [3, 4]$$

Разность:

$$[1, 2, 3, 4] - [3, 4, 5] = [1, 2]$$

Множественный тип данных

Пример:

Составить программу, которая вырабатывает и выводит на экран дисплея наборы случайных чисел для игры в "Спортлото 5 из 36".

Для заполнения каждой карточки спортлото необходимо получить набор из пяти псевдослучайных чисел. К этим числам предъявляются два требования:

- числа должны находиться в диапазоне 1..36;
- числа не должны повторяться.

Program Lotto;

var

nb, k: Set of 1..36;

kol, l, i, n: Integer;

begin

Randomize;

WriteLn('ВВЕДИ kol'); ReadLn(kol);

nb:=[1..36];

for i:=1 to kol do

begin

k:=[];

for l:=1 to 5 do

begin

repeat

n:=Random(36)

until (n in nb) and not (n in k);

k:=k+[n];

Write(n:4)

end;

WriteLn

end

end.

Структурированный тип данных: запись

```
const N = 30;
type tPerson = record
    Name,Surname,SecondName: string[30];
    Male: boolean;           {поле}
    Address: string;        {поле}
    Phone: longint;
end;
tStaff = array [1..N] of tPerson; {массив записей}

var p: tPerson;           {p – запись типа tPerson}
...
begin
...
    p.Surname:='Иванов';   {доступ к полям записи p}
    p.Name:='Иван';
    p.SecondName:='Иванович';
...
    if (p.Phone<0) or (p.Phone>999999)
        then writeln('Ошибка');
...
end.
```

Оператор присоединения :

with p do begin

Surname:= 'Иванов';

Name:= 'Иван';

...

end;

Type

```
Str25 = String[25];
```

```
TBookRec = Record
```

```
    Title, Author, ISBN : Str25;
```

```
    Price : Real;
```

```
    End;
```

```
Var myBookRec : TBookRec;
```

```
Begin myBookRec.Title := 'The Code';
```

```
myBookRec.Author := 'Charles Petzold';
```

```
myBookRec.ISBN := '0-7356-1131-9';
```

```
myBookRec.Price := '500';
```

```
Writeln('Here are the book details:'); Writeln;
```

```
Writeln('Title: ', myBookRec.Title);
```

```
Writeln('Author: ', myBookRec.Author);
```

```
Writeln('ISBN: ', myBookRec.ISBN);
```

```
Writeln('Price: ', myBookRec.Price);
```

```
Readln;
```

```
End.
```

Функции числовых параметров

| <i>Название</i> | <i>Значение</i> |
|-----------------|--|
| abs(x) | модуль x |
| cos(x) | косинус x |
| frac(x) | дробная часть x |
| int(x) | целая часть x (т.е. ближайшее целое, не превосходящее x) |
| pi | число π |
| round(x) | x, округлённое до целого |
| sin(x) | синус x |
| sqr(x) | квадрат x |
| sqrt(x) | квадратный корень из x |
| trunc(x) | число, полученное из x отбрасыванием дробной части |

Примеры

...

`Write(frac(3.14159):8:3);` `{0.142}`

`Write(int(3.14159):8:3);` `{3.000}`

`Write(round(3.14159):8);` `{3}`

`Write(round(3.74159):8);` `{4}`

`Write(trunc(3.74159):8);` `{3}`

Подпрограммы в ЯП Паскаль

```
Program Primer8_1; {Демонстрационная программа}
Uses Crt;
Var a, b, sum: real;
Begin
  ClrScr;
  sum:= 0;
  Writeln('Программа рассчитывает сумму двух чисел');
  Writeln('Нажмите ENTER');
  Readln;
  Write('Введите число a: '); Readln(a);
  Write('Введите число b: '); Readln(b);
  sum:= a + b;
  Writeln;
  Writeln('Сумма этих чисел равна: ', sum:8:0);
  Readkey
End.
```


Program Primer8_2; {Демонстрация процедуры без параметров}

Uses Crt;

Var a, b, sum: real;

Procedure summa; {без параметров!!!}

Begin

sum:=0;

sum:= a + b;

End;

Begin

ClrScr;

Writeln('Программа рассчитывает сумму двух чисел');

Writeln('Нажмите ENTER');

Readln;

Write('Введите число a: '); Readln(a);

Write('Введите число b: '); Readln(b);

Writeln;

summa;

Writeln('Сумма этих чисел равна: ', sum:8:0);

Readkey

End.

```
Program Primer8_3; {Демонстрация процедуры с параметрами}  
Uses Crt;  
Var c, d, cplusd: real;
```

```
Procedure summa(a,b: real; var sum: real); {с параметрами!!!}  
    {попробуйте убрать VAR. Что при этом изменится?}
```

```
Begin  
    sum:=0;  
    sum:= a + b;
```

```
End;
```

```
Begin  
    ClrScr;  
    Writeln('Программа рассчитывает сумму двух чисел');  
    Writeln('Нажмите ENTER');  
    Readln;  
    ClrScr;  
    Write('Введите число c: ');  
    Readln(c);  
    Write('Введите число d: ');  
    Readln(d);  
    Writeln;  
    summa(c,d,cplusd);  
    Writeln('Сумма этих чисел равна: ', cplusd:8:0);  
    Readkey  
End.
```

Program Primer8_4; {Демонстрация двух процедур с параметрами}

Uses Crt;

Var c, d, cplusd: real;

Procedure input(var a,b: real); {попробуйте убрать VAR. Что при этом изменится?}

Begin

Write('Введите число a: ');

Readln(a);

Write('Введите число b: ');

Readln(b);

End;

Procedure summa(a,b: real; var sum: real);

Begin

sum:= a + b;

End;

Begin

ClrScr;

Writeln('Программа рассчитывает сумму двух чисел');

Writeln('Нажмите ENTER'); Readln;

input(c,d);

Writeln;

summa(c,d,cplusd);

Writeln('Сумма этих чисел равна: ', cplusd:8:0);

Readkey

End.

Program Primer8_5; {Демонстрация процедуры и функции}

Uses Crt;

Var c, d, cplUSD: real;

Procedure summa(a,b: real; var sum: real);

Begin

sum:= a + b;

End;

Function summ(a,b: real): real;

Begin

summ:= a + b;

End;

Begin

ClrScr;

Writeln('Программа рассчитывает сумму двух чисел'); Writeln('Нажмите ENTER');

Readln;

ClrScr;

Write('Введите число c: '); Readln(c);

Write('Введите число d: '); Readln(d);

Writeln;

summa(c,d,cplUSD);

Writeln('Ответ рассчитан процедурой: ', cplUSD:8:0);

Writeln('Ответ рассчитан функцией: ', **summ(c,d):8:0**);

Readkey

End.

Пример. Сортировка массива.

```
program summ; {Пример: Сортировка массива по возрастанию}  
uses crt;  
var a:array[1..100] of integer;  
    s,j,n,i:integer;  
begin  
    clrscr;  
    write('vvedite chislo elementov: ');  
    read(n);  
    for i:=1 to n do  
    begin  
        write('a[i]: ');  
        read(a[i]);  
    end;
```

```

for i:=1 to n-1 do
  begin
    for j:=i+1 to n do
      begin
        if a[i]>a[j] then
          begin
            s:=a[i];
            a[i]:=a[j];
            a[j]:=s;
          end;
        end;
      end;
    end;
    for i:=1 to n do
      write(a[i]);
    end.
  
```



Procedure Sortirovka;

```

begin
  for i:=1 to n-1 do
    begin
      for j:=i+1 to n do
        begin
          if a[i]>a[j] then
            begin
              s:=a[i];
              a[i]:=a[j];
              a[j]:=s;
            end;
          end;
        end;
      end;
    end;
  end;
  ...
Sortirovka;

```

Type arr = array [1..100] of integer;

Var a :arr; s, j, n, i : integer;

Procedure Sortirovka(var aL: arr; nL: integer);

Var sL, jL, iL : integer;

begin

...

end;

Begin

...

Sortirovka(a,n);

for i:=1 to n **do**

 write(a[i]);

End.

for iL:=1 to nL-1 **do**
 begin

for jL:=iL+1 to nL
 do

begin

if aL[iL]>aL[jL]
 then

begin

 sL:=aL[iL];

 aL[iL]:=aL[jL];

 aL[jL]:=sL;

end;

end;

end;



Пример. Перевод чисел из 10-ой системы в любую другую.

```
Program Dec2Other; {30.03.2009}
Uses
  Crt;
Const
  ALLdigits: string[16]='0123456789ABCDEF';
Var
  num1,num_tmp,SS: longint;
  num2:string;

Begin
  num1:=43;
  SS:=2;
  num2:="";
  num_tmp:=num1;
```



```
While num_tmp<>0 do
  begin
    num2:=ALLdigits[(num_tmp mod SS)+1]+num2;
    num_tmp:=num_tmp div SS;
  end;
```

```
writeln('Num1= ',num1,' to 2 ', 'Num2= ',num2);
{writeln(1 mod 2);
write(1 div 2)}
```

End.

Пример. Перевод чисел из любой системы в 10-ую.

Program Other2Dec; {02.04.2011}

Uses Crt;

Var SS: longint; num1:string;

Function O2Dec(Lnum:string;LSS:longint):real;

Var gotit, k,i:longint; Sum:real;

Const digit: string[2]='01';

begin

 Sum:=0; k:=0;

for i:= (length(Lnum)) **downto** 1 **do**

begin

 gotit:=Pos(Lnum[i],digit);

If gotit <> 1 **then**

begin

 Sum:=Sum + exp(ln(LSS)*k);

end;

 k:=k+1

end;

 O2Dec:=Sum;

end;

Begin

 SS:=2;

 writeln('Введите число: ');

 readln(num1);

 writeln(O2Dec(num1,SS):10:0);

End.

Задание 1. Сортировка пузырьком.

Вход:

массив A , состоящий из элементов $A[1], A[2], \dots, A[n-1], A[n]$

$t := \text{истина}$

цикл пока t :

$t := \text{ложь}$

цикл для $j = 1, 2, \dots, n - 1$:

если $A[j] > A[j+1]$, то:

поменять местами элементы $A[j]$ и $A[j+1]$

$t := \text{истина}$

Задание 2. Гипотеза Гольдбаха.

Дано четное число n ($n > 2$). Проверить гипотезу Гольдбаха:
четное число представляется в виде двух простых чисел.
Создать процедуру (или функцию) распознавания этих чисел.

Простое число – это натуральное число, которое имеет ровно два различных

натуральных делителя: единицу и самого себя.

Все остальные числа, кроме единицы, называются составными.

Таким образом, все натуральные числа больше единицы разбиваются на простые и составные.

Последовательность простых чисел начинается так:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, ...