Программирование «сверху вниз»

Процедуры и функции пользователя в Pascal

Цели урока:

- познакомить учащихся с методикой программирования "сверху вниз",
- научить создавать собственные подпрограммы,
- воспитать культуру конструктивного мышления.

Процедуры и функции

Все процедуры и функции делятся на

- **стандартные**
- **встроенные**
- **определенные пользователем**.

Встроенные и стандартные вызываются без предварительного описания.

Стандартные процедуры и функции

Арифметические :

abs(x); cos(x); sqrt(x); и др.

Скалярные:

dec(x,n); inc(x,n);

pred(s); suss(s); odd(x);

Функции и процедуры преобразования типов:

chr(i); round(x); trunc(x);

val(s,x,code); str(n,s);

Встроенные процедуры и функции

Встроенные процедуры и функции выделены в группы, которые размещаются в отдельных **модулях**.

Модуль имеет имя.

Подключается через слово **Uses**.

По умолчанию подключается модуль **System**.

Числа Армстронга

- 1, 2, …, 9
- $153=1^3+5^3+3^3=1+125+27$
- $370=3^3+7^3+0^3=27+343$
- 371?

Фрагмент программы нахождения **n**-ой (n>0) степени натурального числа **a** (a^n)

```
st:=1;
 for i:=1 to n do st:=st*a;
```

Фрагмент программы нахождения количества цифр **k** натурального числа **n (n>0)**

```
k:=0;
 while n>0 do     begin
    inc(k);
    n:=n div10; {число n без последней цифры}
                end;
```

Фрагмент программы нахождения суммы цифр **s** натурального числа **n**

```
s:=0;
while n>0 do   begin
  s:=s+n mod10; {n mod 10 – младшая цифра числа}
  n:=n div10;
                    end;
```

Методика программирования "**сверху вниз**"

- Задача разбивается на несколько более простых, которые оформляются в виде подпрограмм.

Пользовательские процедуры и функции

- Подпрограммы – это процедуры и функции для выполнения часто повторяющихся операций с данными, которые могут меняться.
- В Pascal имеется два вида подпрограмм — **процедуры** и **функции**.

Описание подпрограммы

procedure Имя_процедуры(параметры);
begin
 Код процедуры;
end;

function Имя_функции(параметры):тип_результата;
begin
 Код функции;
 Имя_функции:=*результат*;
end;

Параметры

- **Формальные (параметры - переменные) и фактические.**

Количество, типы и порядок формальных и фактических параметров должны совпадать.

- **Локальные и глобальные**

Процедура вычисления степени натурального числа a^n, где 0<=a<=9, 1<=n<=9

```
procedure stepen(a,n:byte; var st:longint);
  var i:byte;
  begin
    st:=1;
     for i:=1 to n do st:=st*a;
  end;
```

Функция вычисления степени натурального числа a^n, где 0<=a<=9, 1<=n<=9

```
function stepen (a,n:byte):longint;
  var i,st:byte;
  begin
    st:=1;
    for i:=1 to n do st:=st*a;
     stepen:=st;
  end;
```

Процедура вычисления количества цифр **k** натурального числа **n** ($0<n<10^4$)

```
procedure kol(n:integer; var k:byte);
  var i:byte;
  begin
    k:=0;
   while n>0 do begin
     inc(k);    n:=n div10;
                              end;
   end;
```

Функция вычисления количества цифр **k** натурального числа **n** ($0<n<10^4$)

```
function kol (n:integer):byte;
  var k:byte;
  begin     k:=0;
        while n>0 do begin
         inc(k);        n:=n div10;
                        end;
         kol:=k;
  end;
```

Процедура вычисления суммы цифр натурального числа **n** ($0<n<10^4$)

```
procedure sum(n:integer; var s:byte);
  var x,k:byte;
  begin  s:=0;
       while n>0 do begin
           s:=s+ n mod 10;
              n:=n div10;end;
   end;
```

Функция вычисления суммы цифр натурального числа **n** ($0<n<10^4$)

```
function sum(n:integer):byte;
   var k:byte;
   begin    s:=0;
        while n>0 do begin
                s:=s+ n mod 10;
                n:=n div10;
        end;
    sum:=s;  end;
```

Вызов подпрограммы

Процедура и функция вызываются по имени, в круглых скобках записываются фактические параметры.

Имя (Список фактических параметров*)*

Количество, типы и порядок формальных и фактических параметров должны совпадать.

Определение числа Армстронга

(ПРОЦЕДУРЫ)

```
program p1;
var n,n1:integer; k:byte;x,s:longint;
procedure stepen(a,n:byte; var st: longint);
    var i:byte;
    begin
      st:=1;   for i:=1 to n do st:=st*a;
    end;
procedure kol(n:integer; var k:byte);
    var i:byte;
    begin
     k:=0;
     while n>0 do begin
          inc(k);
          n:=n div 10;  end;
    end;
```

```
begin readln(n);      {вводим натуральное число}
  s:=0; k:=0;    {s-для хранения суммы степеней цифр числа,
    k-для хранения количества цифр числа}
 n1:=n;       {n1 для хранения введенного числа}
 kol(n,k);
 n:=n1;      {в n восстановим введенное число}
 s:=0;
 while n>0 do begin
  stepen(n mod 10,k,x); {x- для хранения степени младшей
   цифры числа}
   s:=s+ x;   n:=n div 10;
                  end;
  if s=n1 then writeln(‘число ’, n1, ' явл. числ. Армстронга')
            else writeln(‘число ’, n1, ' не явл. числ. Армстронга');
end.
```

Определение числа Армстронга

(ФУНКЦИИ)

```
program p2;
var n,n1:integer; k:byte; s:longint;
function stepen (a,n:byte):longint;
  var i:byte; st:longint;
  begin     st:=1;
       for i:=1 to n do st:=st*a;
       stepen:=st;  end;
function kol (n:integer):byte;
     var k:byte;
     begin
        k:=0;
        while n>0 do begin
           inc(k);    n:=n div 10;    end;
       kol:=k;    end;
```

```
begin
    readln(n);
    n1:=n;
    k:=kol(n);
    s:=0;
    n:=n1;
    while n>0 do begin
     s:=s+stepen(n mod 10,k);
     n:=n div 10;
          end;
    if s=n1 then writeln(‘число ’,n1, ' явл. числ.
    Армстронга')
         else writeln(‘число ’,n1, ' не явл. числ. Армстронга')
end.
```

Вывод чисел Армстронга из диапазона от **a** до **b**($0<=a<b<10^4$).

(ФУНКЦИИ)

```
program p4;
    var a,b,i:integer; k:byte;s:longint;
    function stepen (a,n:byte):longint;
     var i:byte; st:longint;
     begin   st:=1;
     for i:=1 to n do st:=st*a;
      stepen:=st;  end;
    function kol (n:integer):byte;
     var k:byte;
     begin  k:=0;
          while n>0 do begin
          inc(k);  n:=n div 10;
                    end;
          kol:=k;     end;
```

```
function sum(n:integer; k:byte):longint;
    var s:longint;
    begin
     s:=0;
     while n>0 do begin
          s:=s+ stepen(n mod 10,k);
          n:=n div 10;
               end;
     sum:=s;
    end;
```

```
begin
    readln(a,b);
    for i:=a to b do begin
     n:=i;
     k:=kol(n);
     n:=i;
     s:=sum(n,k);
     if s=i then writeln(i)
                end;
end.
```

Домашнее задание

а) Найти пять наибольших чисел Армстронга из диапазона от **a** до **b**($0<a<b<10^4$).

б) Найдите пять наибольших чисел Армстронга, состоящих не более чем из **n** цифр. Вывести их по возрастанию.

Пример input.txt output.txt

4 **371 407 1634 8208 9474**

1 **5 6 7 8 9**

Хорошая программа должна содержать главным образом **обращения к процедурам и функциям**, и для этого рекомендую:

- выделять в процедуру (функцию) небольшой логически завершенный фрагмент алгоритма;
- называть процедуры (функции) мнемоническими именами;
- использовать функцию, если алгоритм должен вычислить одно скалярное значение.

- Подпрограммы – это процедуры и функции для выполнения часто повторяющихся операций с данными, которые могут меняться. Описание подпрограммы состоит из ключевого слова **procedure** или **function**, за которым следует имя подпрограммы со списком формальных параметров, заключённых в скобки.
- Вызов подпрограммы происходит в основной программе по имени с указанием фактических параметров.
- Количество, типы и порядок формальных и фактических параметров должны совпадать.