Алгоритмы и алгоритмические языки

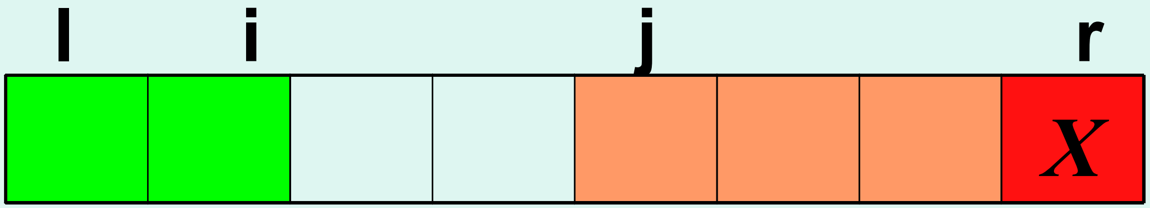
Лекция 13

Функции в языке Паскаль.

Быстрая сортировка массива (quicksort)

```
Quicksort (A,l,r)
    if l < r then
        Partition(A,l,r,q)
        Quicksort(A,l,q-1)
        Quicksort(A, q+1, r)
```

Partition – процедура разбиения массива без использования дополнительной памяти



A[l] .. A[i] <= X

A[i+1] .. A[j-1] >X

A[j] .. A[r-1] – любые (еще не просмотрены)

Быстрая сортировка массива (quicksort)

i	l,j							r
	2	8	7	1	3	5	6	4

l,i	j						r
2	8	7	1	3	5	6	4

l,i		j					r
2	8	7	1	3	5	6	4

l,i			j				r
2	8	7	1	3	5	6	4

l	i			j			r
2	1	7	8	3	5	6	4

l		i			j		r
2	1	3	8	7	5	6	4

l		i				j	r
2	1	3	8	7	5	6	4

l		i					j,r
2	1	3	8	7	5	6	4

2	1	3	4	7	5	6	8

Далее применим метод к левой и правой частям массива.

Быстрая сортировка массива (quicksort)

```
procedure swap (var a,b:integer); {swap variable values}
  var c:integer;
begin c:=a;a:=b;b:=c end;

procedure partition(var A:vector; l,r:integer; var q:integer);
 var i,j,x:integer;
begin x:=A[r];     i:=l-1;
      for j:=l to r-1 do
        if A[j]<=x then  begin  i:=i+1;  swap(A[j],A[i]) end;
      swap(A[i+1],A[r]);
      q:=i+1 {q – index of the element that stays on it's required position}
 end;

procedure qsort(var A:vector; l,r:integer);
  var q:integer;
begin
  if l<r then begin partition(A,l,r,q); qsort(A,l,q-1); qsort(A,q+1,r) end
end;
```

Язык Паскаль. Функции

Стандартные функции:

```
sin, cos, ln,exp, trunc, round, pred, succ……
```

Определение новых функций

Пример:

$$ch\,x = \frac{e^{x} + e^{-x}}{2}$$

```
function ch(x: real): real;
  var p: real;
begin
  p:=exp(x); ch:=(p+1/p)/2
end;
```

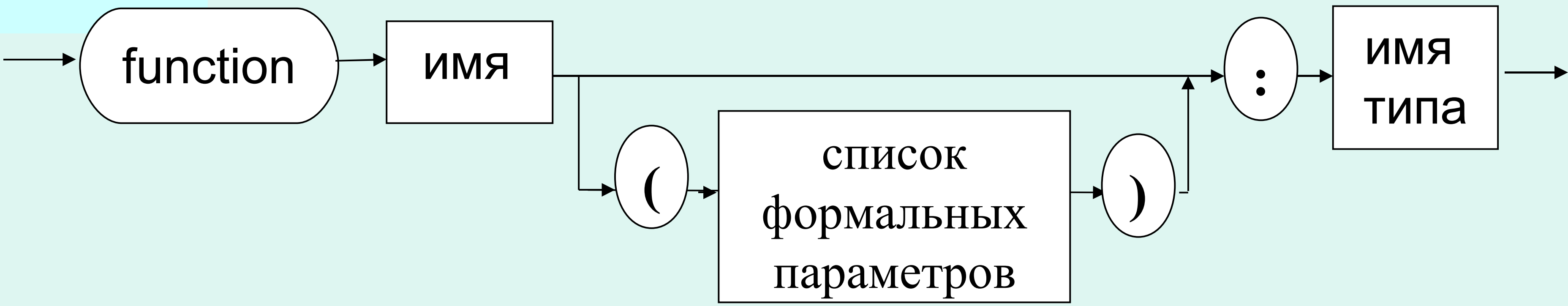
Обращение к функции (указатель функции):

t:=(ch(q)+ch(4))/ch(q+2)

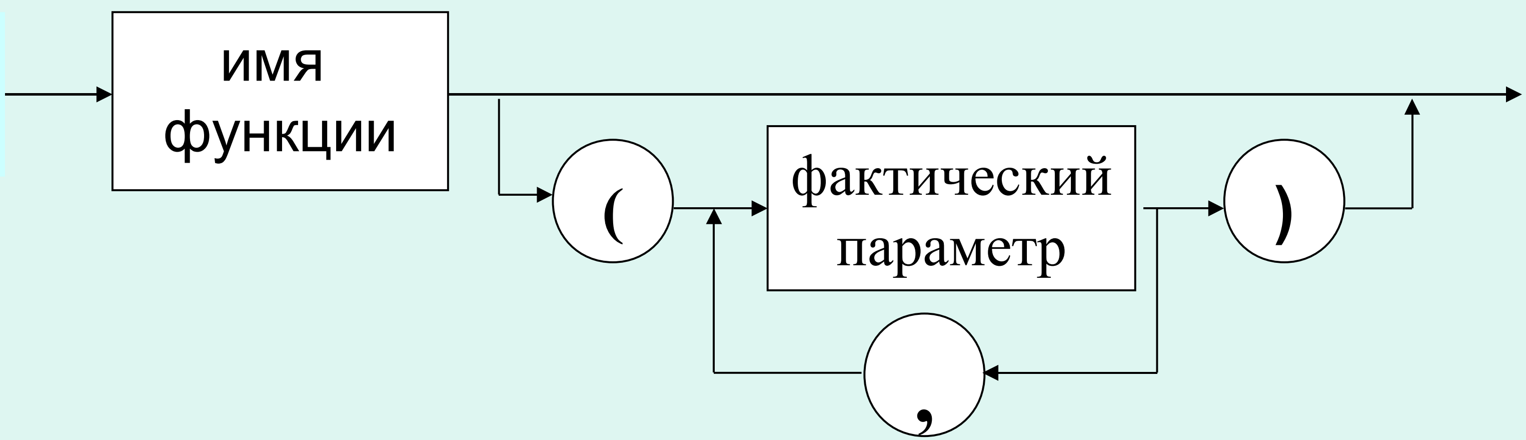
Синтаксис описания и указателя функции

<описание функции> ::= <заголовок функции> ; <блок> ;

заголовок функции



указатель функции



Побочные эффекты функций

Побочный эффект – произведенное функцией изменение в состоянии программы.

Пример 1:

```
var a: integer;  b: boolean;
function f: integer;  begin f:=a; a:=a+1 end;
... a:=2; b:=f=f ...
```

Пример 2:

Определить, есть ли в строке пробелы. Если да – заменить их на ‘-’

```
type str=packed array[1..n] of char; ...
function Spaces(var S: str): boolean;
  var k: integer;
begin  Spaces:=false;
  for k:=1 to n do
    if S[k]=‘ ‘ then
        begin Spaces:=true; S[k] :=‘-’ end;
end;
```

Рекурсивные функции

Пример 1:

Факториал целого неотрицательного числа n

```
function fact(n:integer): integer;
begin
  if n=0 then fact:=1
   else fact:=n*fact(n-1);
end;
```

Рекурсивные функции

Пример 2: **Числа Фибоначчи**

Fibonacci (Leonardo of Pisa),
ок. 1175–1250

1 ;1 ; 2 ; 3 ; 5 ; 8 ; 13 ; 21 ; 34 ; 55; 89 ; 144 ; …

Рекурсивные функции

Функция вычисления n-го числа Фибоначчи

```
function fib(n:integer): integer;
begin
  if n<3 then fib:=1
   else fib:= fib(n-1)+fib(n-2);
end;
```

Примеры рекурсивных функций

Пример 3. Функция Маккарти

```
function f(n: integer): integer;
begin
   if n>100 then f:=n-10
    else f:=f(f(n+11))
end;
```

Джон Маккарти (1927-2011)

Примеры рекурсивных функций

Пример 4. Вычисление наибольшего общего делителя

```
function gcd(a,b:integer):integer;
 begin
   if b=0 then gcd := a
     else gcd := gcd (b,a mod b)
 end;
```

Косвенная рекурсия

f → g →…→f

$$f(n)=\begin{cases}1 \text{ при } n=1\\ g(n-1) \text{ при } n>1\end{cases}\qquad g(m)=\begin{cases}1 \text{ при } m=1\\ f(m-1)+g(m-1) \text{ при } m>1\end{cases}$$

```
function f(n:integer):integer; forward;

function g(m: integer): integer;
      begin
        if m=1 then g:=1 else g:=f(m-1)+g(m-1)
      end;

function f;
      begin
        if n=1 then f:=1 else f:=g(n-1)
      end;
```

Параметры-функции и параметры-процедуры

Задача: вычислить

$$t = (\mathrm{tg}\,1 + \mathrm{tg}\,2 + \ldots + \mathrm{tg}\,50)*(\sin 1 + \sin 2 + \ldots + \sin(n+20))$$

Подзадача: вычислить
$f(1)+f(2)+ \ldots +f(k)$ для произвольной функции f и произвольного числа k

Определим функцию: `SUM(k,f)=f(1)+f(2)+ ... +f(k)`

```
function SUM(k:integer;
             function f(x:real):real): real;
 var i: integer; S: real;
 begin
    S:=0;
    for i:=1 to k do S:=S+f(i);
       SUM:=S
 end;
```

Параметры-функции и параметры-процедуры

(продолжение)

Дополнительные определения функций:

```
function tg (x:real):real;
  begin tg:=sin(x)/cos(x) end;

function sin1 (x:real):real;
  begin sin1:=sin(x) end;
:

  t:=SUM(50, tg) * SUM(n+20,sin1)
```

Замечание: **SUM (n+20,sin) – ошибка, в качестве параметра нельзя указывать имя стандартной функции**