



Алгоритмы и алгоритмические языки

Лекция 11

Процедуры в языке Паскаль.



Назначение процедур.

Задача 1.

Вычислить скалярное произведение векторов $x = (x_1, \dots, x_n)$ и $y = (y_1, \dots, y_n)$.

Задача 2.

Вычислить значение дроби

$$\frac{r}{s} = \left(\frac{1}{10} + \frac{1}{11} + \dots + \frac{1}{80} \right) \cdot \left(\frac{1}{m} + \frac{1}{m+1} + \dots + \frac{1}{2m} \right) \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{m} \right)$$

Язык Паскаль. Описание процедуры

$$\frac{p}{q} := \frac{0}{1}; \quad \forall i: \quad \frac{p}{q} := \frac{p}{q} + \frac{1}{i} = \frac{pi + q}{qi}$$

Язык Паскаль. Описание процедуры

$$\frac{p}{q} := \frac{0}{1}; \quad \forall i: \quad \frac{p}{q} := \frac{p}{q} + \frac{1}{i} = \frac{pi + q}{qi}$$

```
procedure SUM(H,K:integer; var p,q:integer);
```

```
  var i: integer;
```

```
begin
```

```
  p:=0; q:=1;
```

```
  for i:=H to K do begin p:=p*i+q; q:=q*i end
```

```
end;
```



Язык Паскаль. Оператор процедуры

read(a)

write(a,x)

SUM(10,80,a,b);

SUM(m,2*m,c,d);

SUM(1,m,e,f);

r = a * c * e;

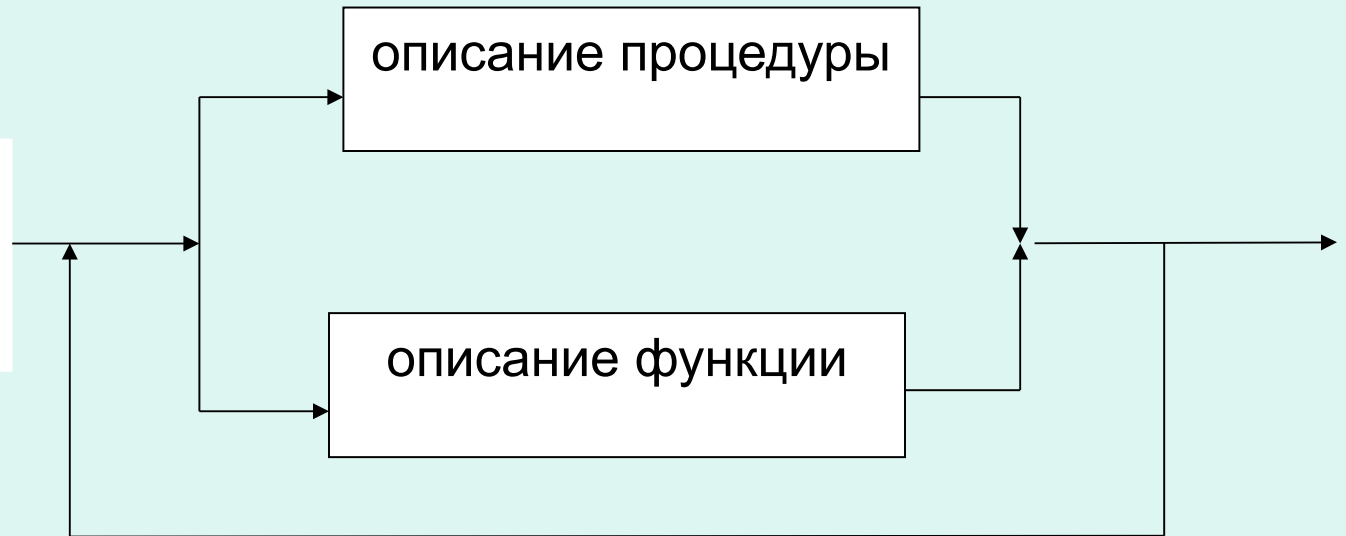
s = b * d * f

Требование: число, порядок и типы *формальных* и *фактических* параметров должны совпадать.

Язык Паскаль. Описание процедуры

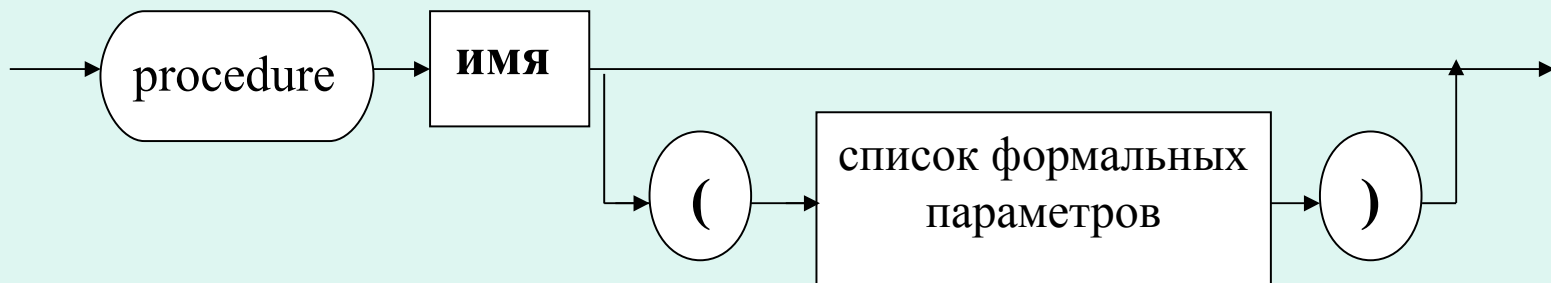
Синтаксис

раздел
процедур
и функций



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

заголовок
процедуры





Параметры процедур

Процедуры без параметров

```
procedure correct;
```

```
begin
```

```
  writeln('Could you, please, correct the input data')
```

```
end;
```

```
.....correct;.....correct;
```

Процедуры с параметрами

Сколько параметров может быть у процедуры?

В каком порядке перечислять параметры?

Что нужно указывать при описании формального параметра?

Формальные параметры

список формальных параметров

секция формальных параметров

;

var

ИМЯ

:

ИМЯ ТИПА

,

заголовок процедуры

заголовок функции




Параметры процедур

Пример 1

```
procedure exch1 (a, b : integer);  
  var c:integer;  
begin  
  c:=a; a:=b; b:=c  
end;
```

Пример 2

```
procedure exch2 (var a, b : integer);  
  var c:integer;  
begin  
  c:=a; a:=b; b:=c  
end;
```



Вспомогательные объекты процедур

Пример

```
var a,b ; integer;  
procedure example (a: char);  
  const b=true;  
begin  
  {тело процедуры}  
end;
```



Правило локализации

1. Имя, описанное в блоке, существует только в этом блоке и не существует вне его.
2. В каждом блоке каждое имя может быть описано только один раз.
3. В каждой точке программы действует описание имени из ближайшего объемлющего блока

Областью видимости имени является весь блок, в котором это имя описано, за исключением тех вложенных блоков, где описано такое же имя.



Глобальные и локальные имена процедур

Локальные имена - описанные в самой процедуре: имена формальных параметров и вспомогательных объектов.

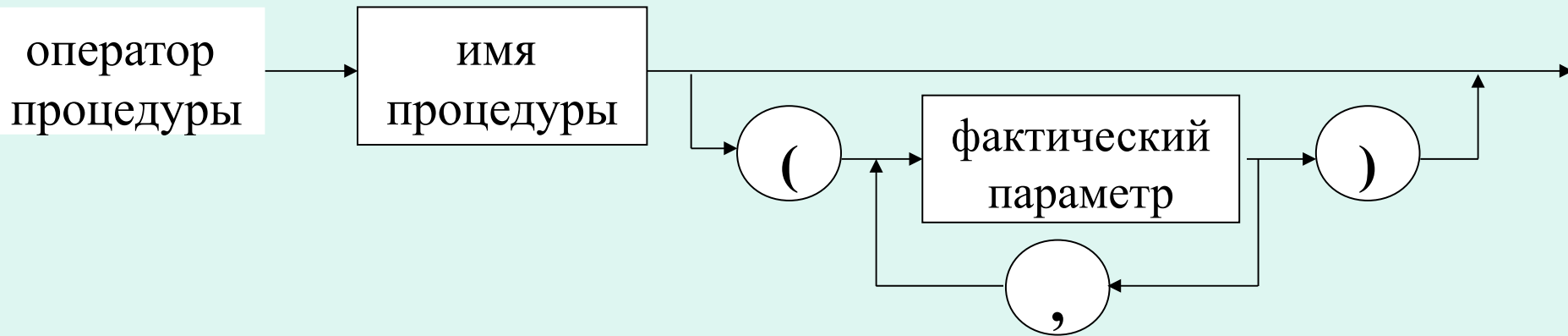
Глобальные имена — стандартные имена и имена из программы

```
const c='*';  
  procedure P (var x : real);  
    var k: integer;  
  begin  
    k:=ord(c); x:=k/2  
  end;
```

Правило:

Все вспомогательные объекты, нужные процедуре, следует описывать внутри нее

Оператор процедуры



<фактический параметр> ::= <переменная> | <выражение> |
<имя процедуры> | <имя функции>



Пример для самостоятельного рассмотрения

```
program P (output);  
  const t = true;  
        true = false;  
begin  
  writeln(t=true)  
end.
```





Семантика оператора процедуры

```
var a, b: integer;
procedure P(x: integer; var y: integer);
  var z: integer;
begin
  z:=2; x:=x+z; y:=y+z
end;
  ...
a:=1; b:=1;
P(a,b); writeln(a, ' ', b);
```