

2020

Глава 2 Управляющие конструкции

МГТУ им. Н.Э. Баумана

Факультет Информатика и системы
управления

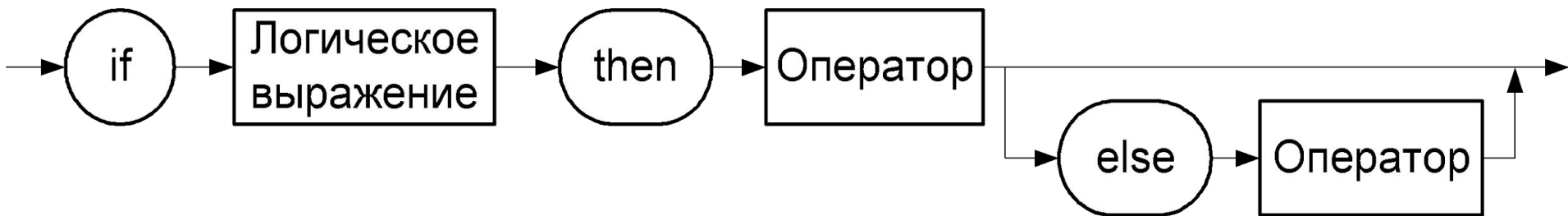
Кафедра Компьютерные системы и сети

Лектор: д.т.н., проф.

Иванова Галина Сергеевна

2.1 Оператор условной передачи управления

Оператор условной передачи управления используется при обработке вариантов вычислений и реализует конструкцию ветвления.



Оператор – простой или составной оператор языка.

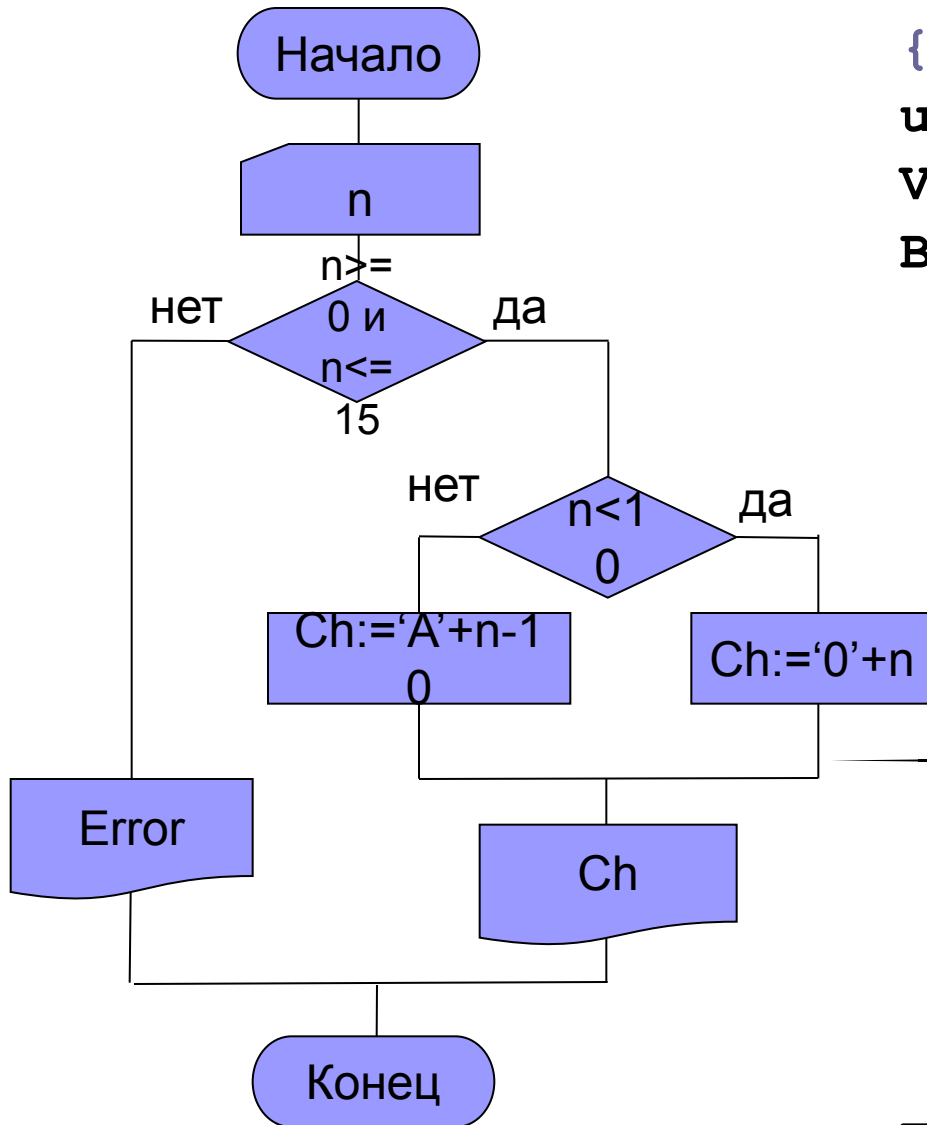
Составной оператор – блок операторов в операторных скобках
begin ...end.

Пример. Разработать программу, которая вводит число от 0 до 15 и выводит его шестнадцатеричный эквивалент, например:

8 ⇒ '8'

12 ⇒ 'C'

Программа отображения 16-тиричных эквивалентов



Program Ex2_1;

```
{ $APPTYPE CONSOLE }
```

```
uses SysUtils;
```

```
Var Ch:char; n:Integer;
```

```
Begin
```

```
Write('Input n:');
```

```
ReadLn(n);
```

```
if (n >= 0) and (n <= 15) then
```

```
begin
```

```
if n < 10 then
```

```
Ch := chr(ord('0') + n)
```

```
else
```

```
Ch := chr(ord('A') + n - 10);
```

```
Write('n=', Ch);
```

```
end
```

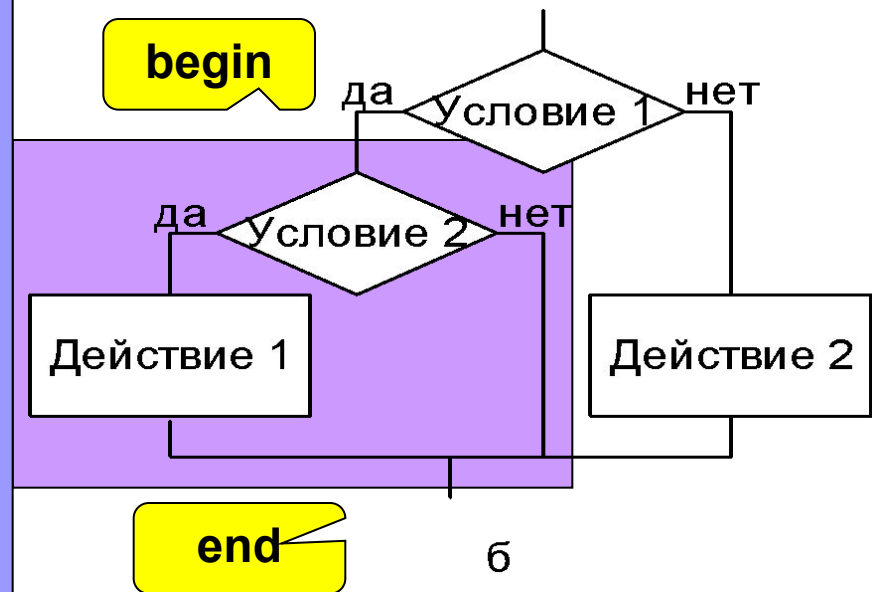
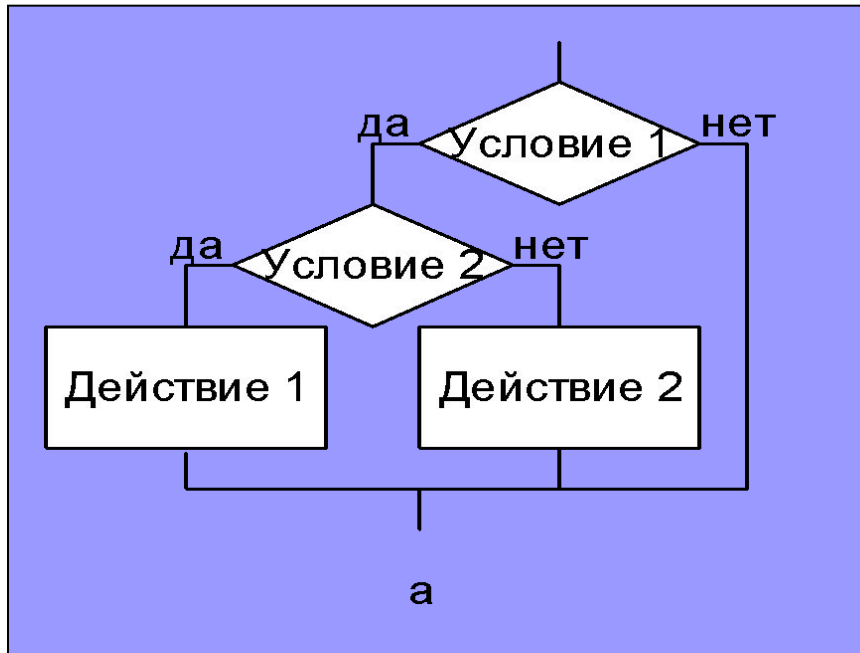
```
else WriteLn('Error');
```

```
ReadLn;
```

```
End.
```

Правило вложения

```
if <Условие1> then  
    if <Условие2> then <Действие1>  
    else <Действие 2>
```



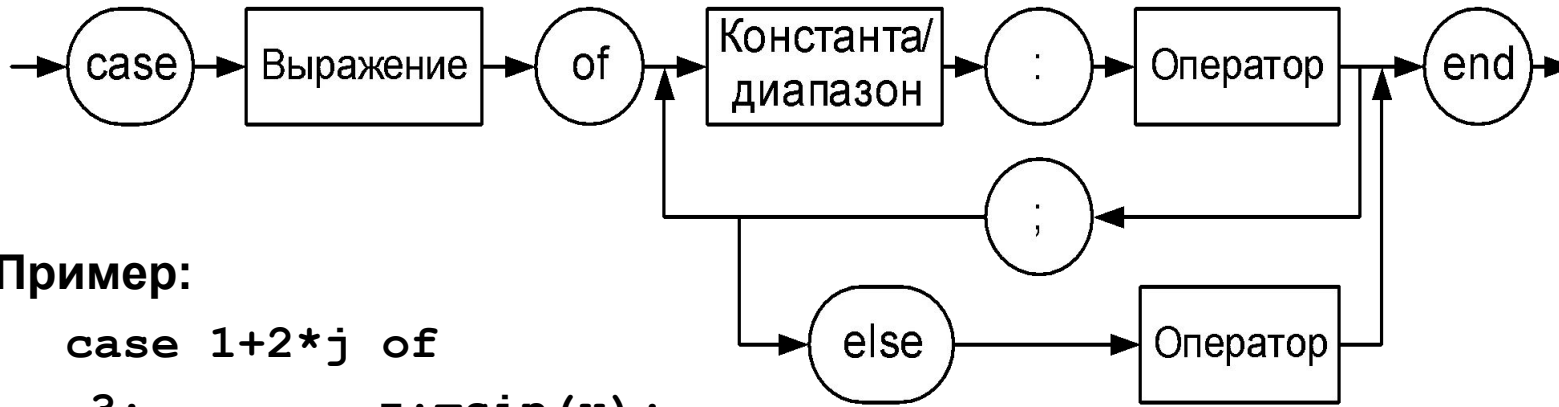
Ветвь else относится к ближайшему if.

Для реализации варианта б используют begin ...end:

```
if <Условие1> then  
    begin if <Условие2> then <Действие1> end  
else <Действие 2>
```

2.2 Оператор выбора

Оператор позволяет программировать несколько вариантов решения.



Пример:

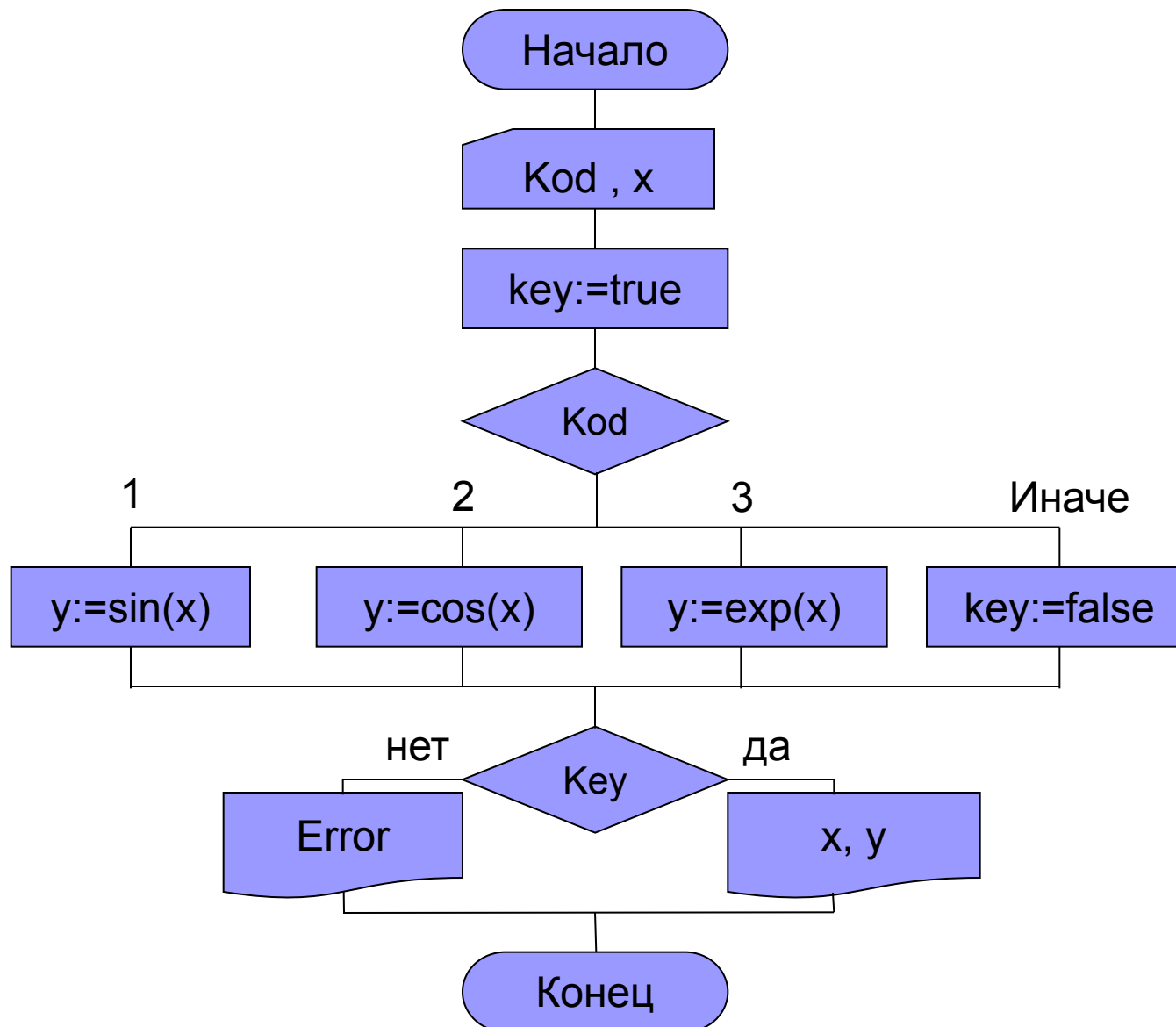
```
case 1+2*j of
  3:          z:=sin(x);
  -1..1,10:  z:=cos(x);
  else       z:=0;
end;
```

Пример. Разработать программу, вычисляющую значения функции. Функция выбирается пользователем из нескольких заданных.

Enter cod:

- 1 - $y = \sin x$
- 2 - $y = \cos x$
- 3 - $y = \exp x$

Схема алгоритма



Программа вычисления значения функции

```
Program Ex2_2;  
{$APPTYPE CONSOLE}  
uses SysUtils;  
Var Kod:Integer;  
    y,x:Single;  
    Key:boolean;  
Begin  
    WriteLn('Enter Kod:');  
    WriteLn('1 - y=sin x');  
    WriteLn('2 - y=cos x');  
    WriteLn('3 - y=exp x');  
    ReadLn(Kod);  
    WriteLn('Enter x:');  
    ReadLn(x);
```

Программа вычисления значения функции (2)

```
Key:=true;
```

```
Case Kod of
```

```
  1: y:=sin(x);
```

```
  2: y:=cos(x);
```

```
  3: y:=exp(x);
```

```
  else Key:=false;
```

```
end;
```

```
if Key then
```

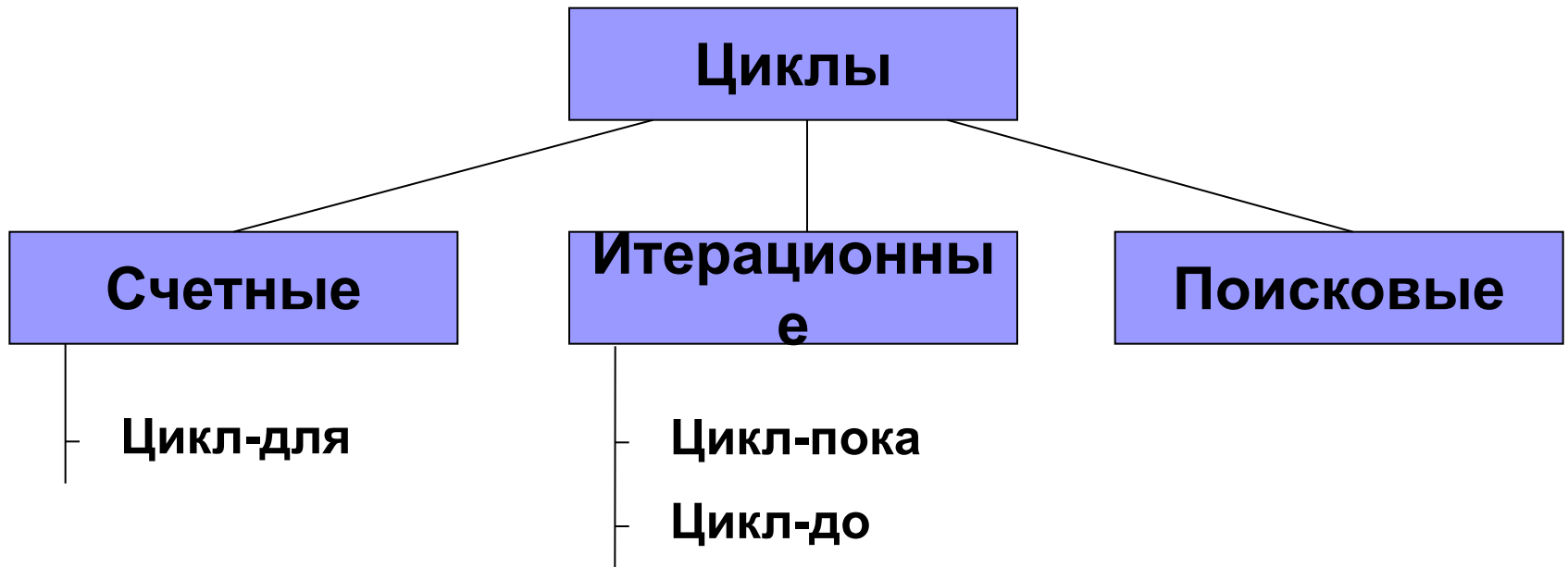
```
  WriteLn(' x =',x:10:6,' y =',y:10:6)
```

```
else WriteLn('Error');
```

```
ReadLn;
```

```
End.
```


2.3 Операторы организации циклов

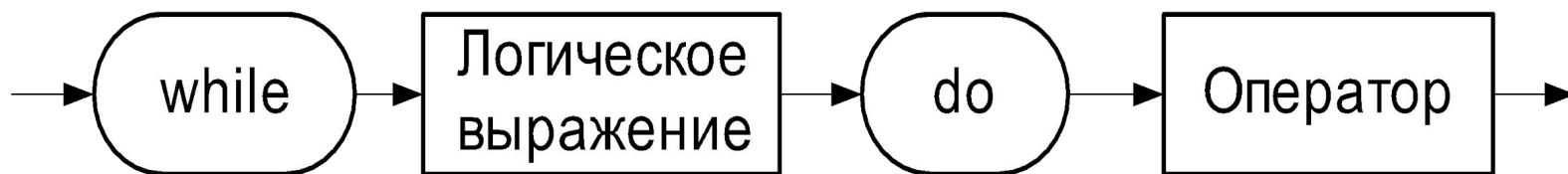
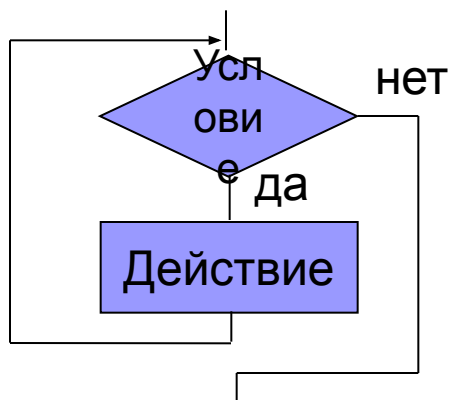


Счетный цикл – цикл, количество повторений которого известно или можно посчитать. Выход из такого цикла программируется по счетчику.

Итерационный цикл – цикл, количество повторений которого неизвестно или считается неизвестным при построении цикла. Выход из цикла программируется по выполнению или нарушению условия.

Поисковый цикл имеет два выхода – нашли и перебрали все и не нашли.

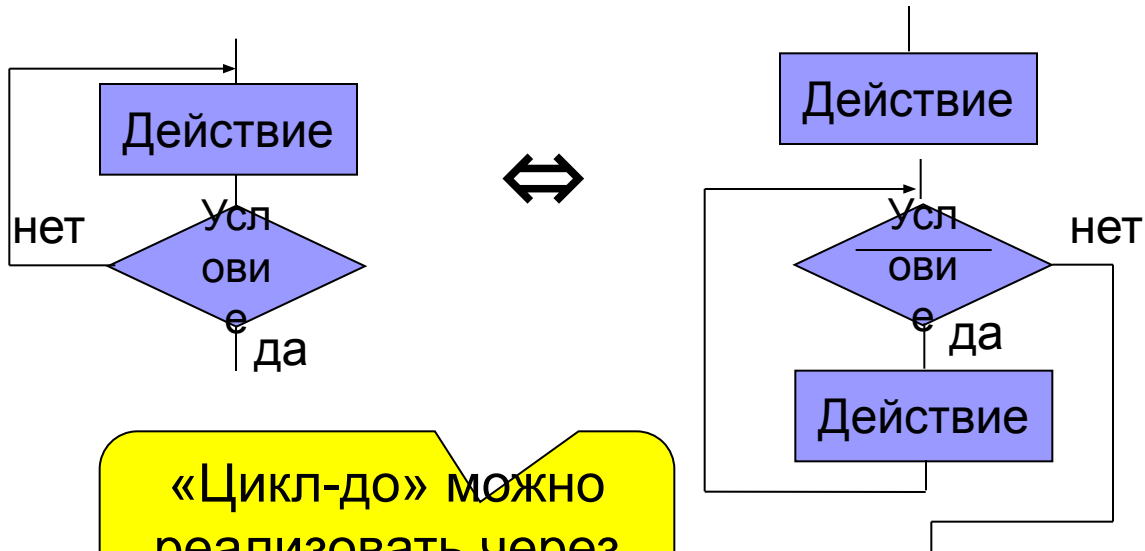
Цикл-пока



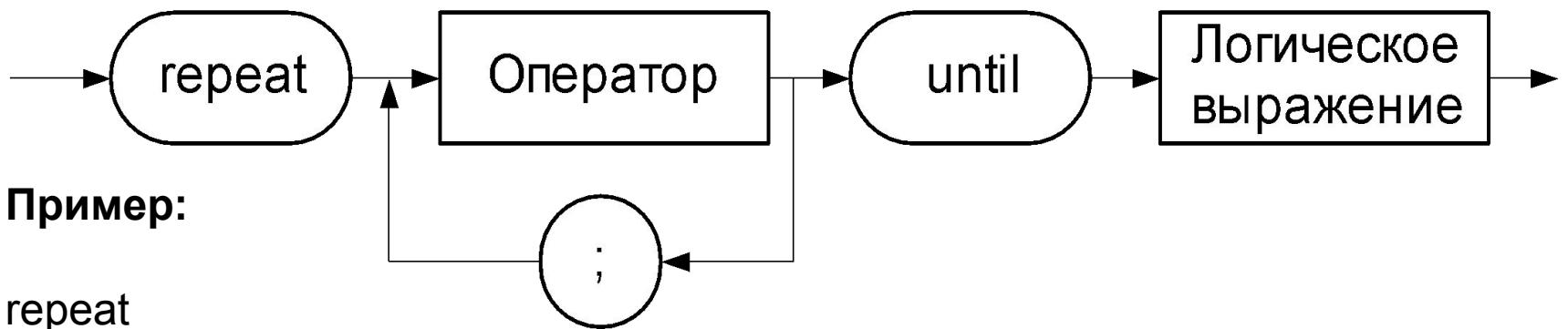
Пример:

```
while abs(e)>=1e-5 do  
  begin  
    x:=x+1;  
    e:=e/10;  
  end;
```

Цикл-до



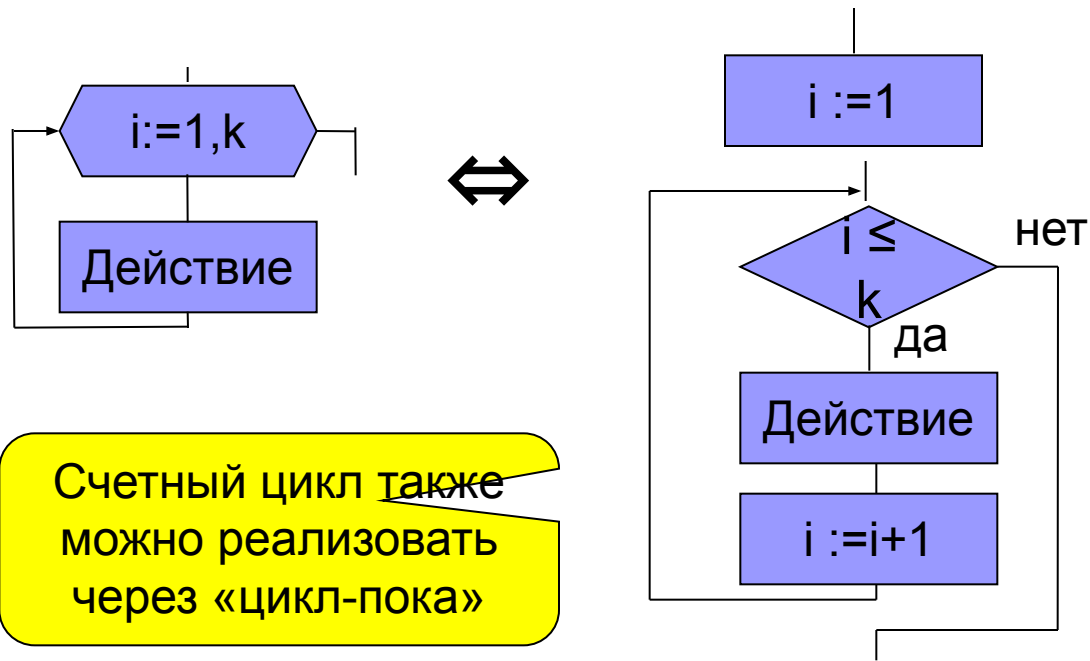
«Цикл-до» можно реализовать через «цикл-пока»



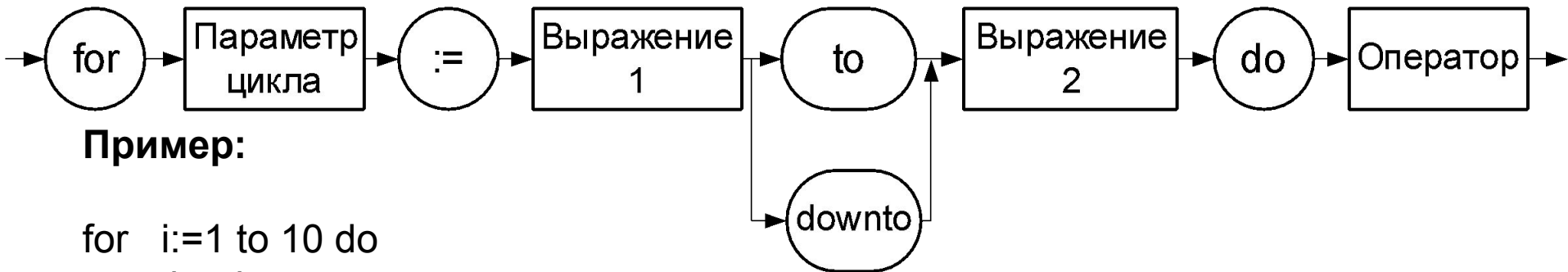
Пример:

```
repeat  
  x:=x+1;  
  e:=e/10;  
until abs(e)<1e-5;
```

Счетный цикл



Счетный цикл также можно реализовать через «цикл-пока»

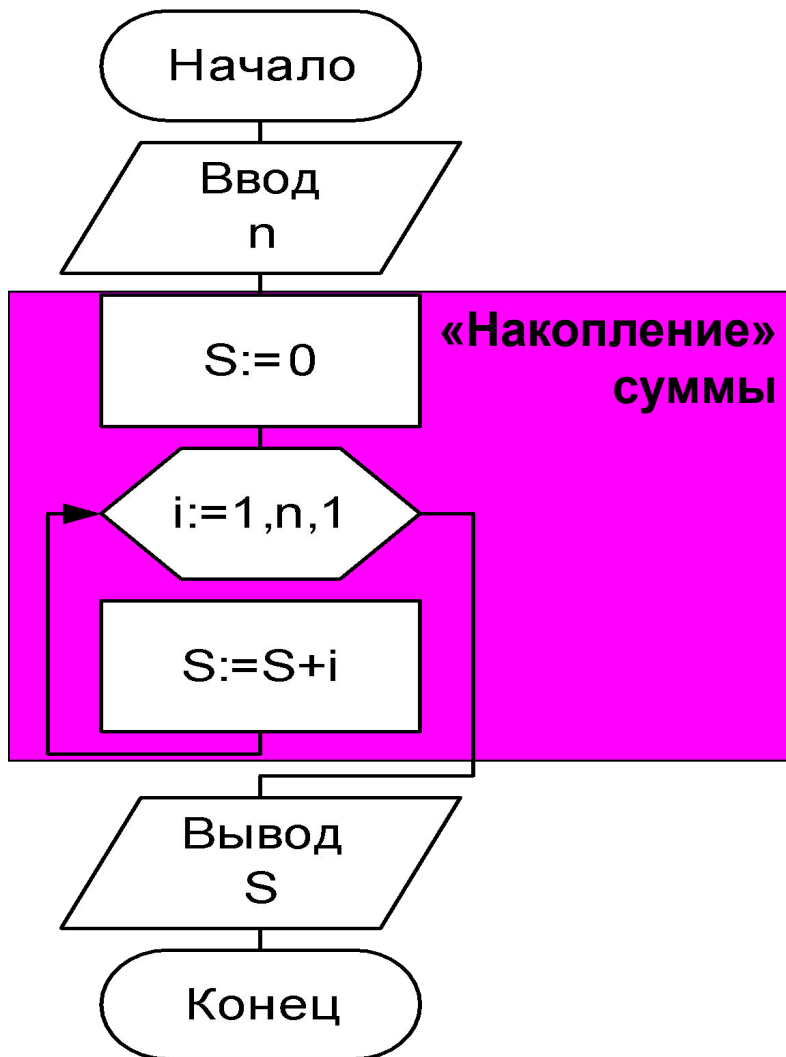


Пример:

```
for i:=1 to 10 do
begin
  x:=x+1;
  e:=e/10;
end;
```

Суммирование натуральных чисел

Найти сумму N натуральных чисел.

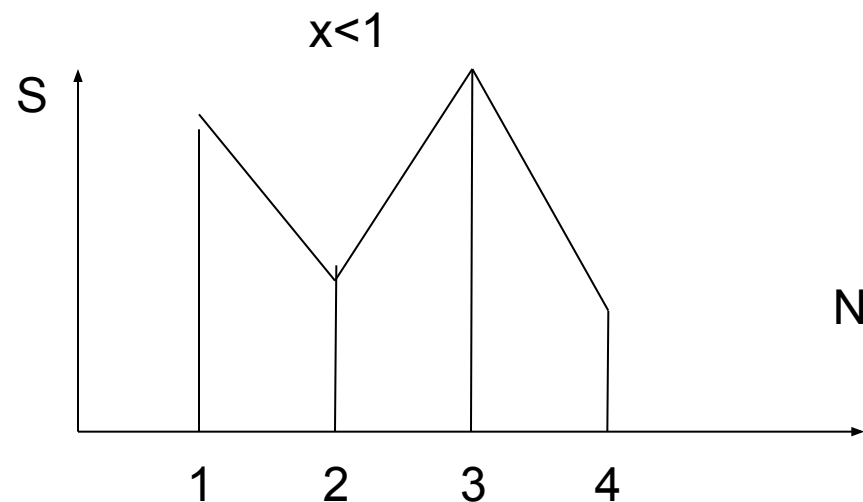
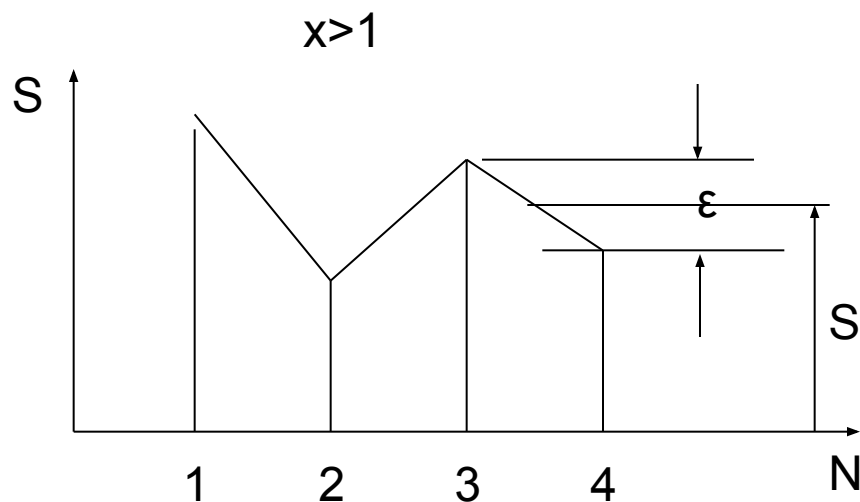


```
Program Ex2_3;  
{ $APPTYPE CONSOLE }  
uses SysUtils;  
Var i, N, S: Integer;  
Begin  
    Write('Input N: ');  
    ReadLn(N);  
    S:=0;  
    For i:=1 to N do  
        S:=S+i;  
    WriteLn('Summa=', S:6);  
    ReadLn;  
End.
```

Суммирование ряда

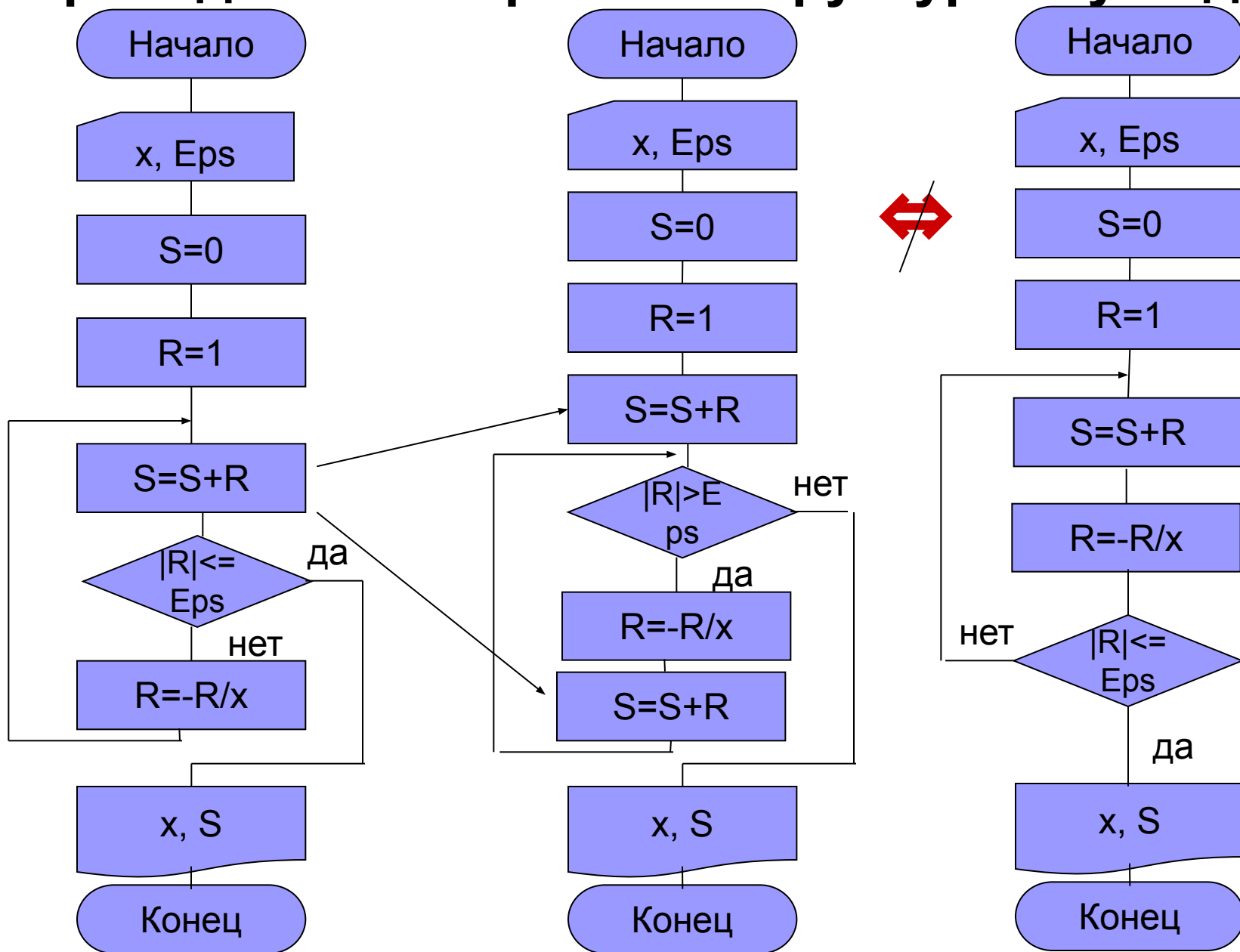
Определить сумму ряда

$S = 1 - 1/x + 1/x^2 - 1/x^3 + \dots$ с заданной точностью ϵ .

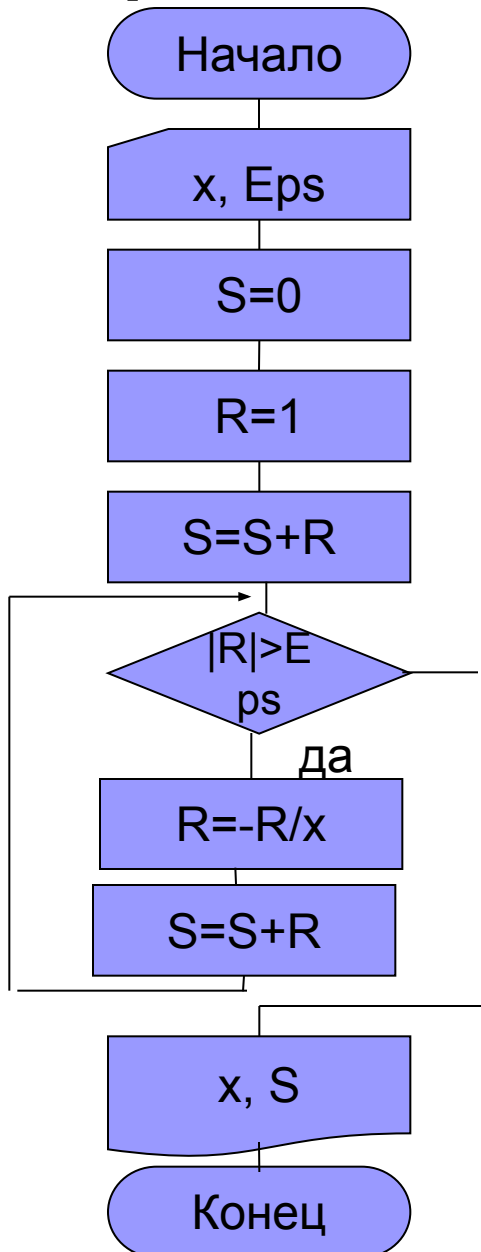


$$R_n = -R_{n-1}/x$$

Приведение алгоритма к структурному виду

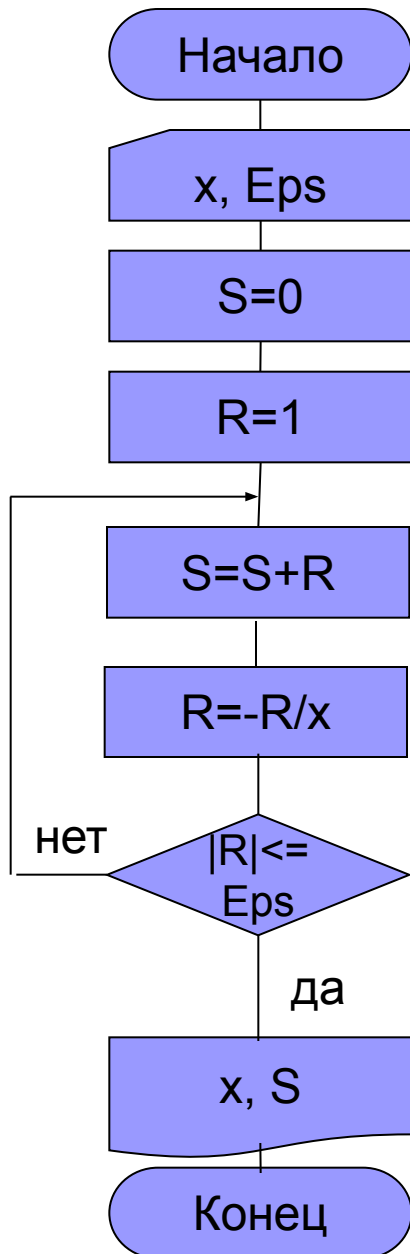


Вариант а



```
Program Ex2_4a;  
{ $APPTYPE CONSOLE }  
uses SysUtils;  
var S,R,X,eps:Single;  
Begin  
    WriteLn('Input x and epsilon:');  
    ReadLn(X,eps);  
    S:=1; {S:=0; & S:=S+R;}  
    R:=1;  
    while abs(R)>eps do  
        begin  
            R:=-R/X;  
            S:=S+R;  
        end;  
    WriteLn('x=', x:6:2,  
            ' S=', S:8:2, ' R=', R:8:6);  
    ReadLn;  
End.
```

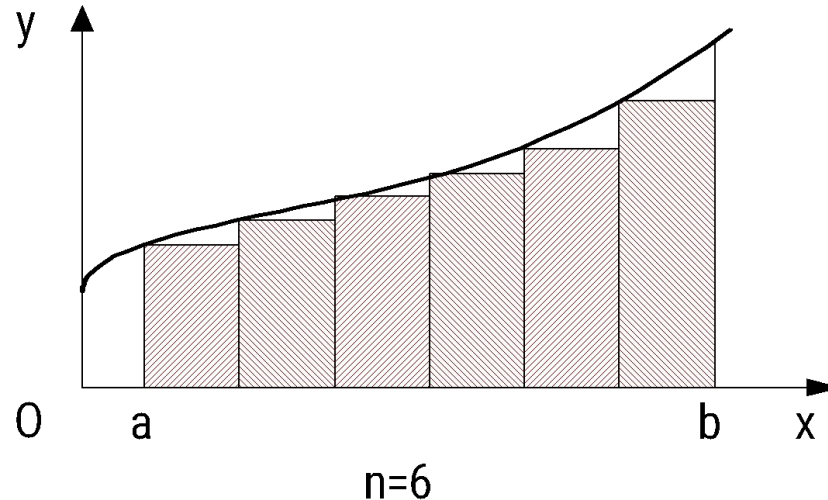

Вариант 6



```
Program Ex2_4b;  
{ $APPTYPE CONSOLE }  
uses SysUtils;  
var S,R,X,eps:Single;  
Begin  
    WriteLn('Input x and epsilon:');  
    ReadLn(X,eps);  
    S:=0;  
    R:=1;  
    repeat  
        S:=S+R;  
        R:=-R/X;  
    until abs(R) <= eps;  
    WriteLn('x=', x:6:2,  
    ' S=', S:8:2, ' R=', R:8:6);  
    ReadLn;  
End.
```

Решение задач вычислительной математики

Задача. Вычислить определенный интеграл функции $f(x)$ на интервале $[a,b]$ методом прямоугольников с точностью δ .



Итак

$$S = f(x_1) \times d + f(x_2) \times d + f(x_3) \times d + \dots + f(x_n) \times d = d \times \sum_{i=1}^n f(x_i), \text{ где } d = (b-a)/n.$$

Увеличивая n , получаем приближения площади: $S_1, S_2, S_3 \dots$

Останавливаемся, когда $|S_k - S_{k+1}| < \delta$

Неформальное описание алгоритма

Алгоритм:

Шаг 1. Ввести a , b , δ .

Шаг 2. Задать число прямоугольников $n:=10$.

Шаг 3. Определить шаг $d:=(b-a)/n$.

Шаг 4. Определить площадь фигуры $S1$.

Шаг 5. Увеличить число прямоугольников вдвое $n:=n*2$.

Шаг 6. Уменьшить шаг вдвое $d:=d/2$.

Шаг 7. Определить площадь фигуры $S2$.

Шаг 8. Если Разность площадей меньше δ , то перейти к шагу 11

Шаг 9. Запомнить новое значение площади $S1:=S2$.

Шаг 10. Перейти к шагу 5.

Шаг 11. Вывести $S1$.

Конец.

Схема алгоритма (неструктурная и неэффективная)

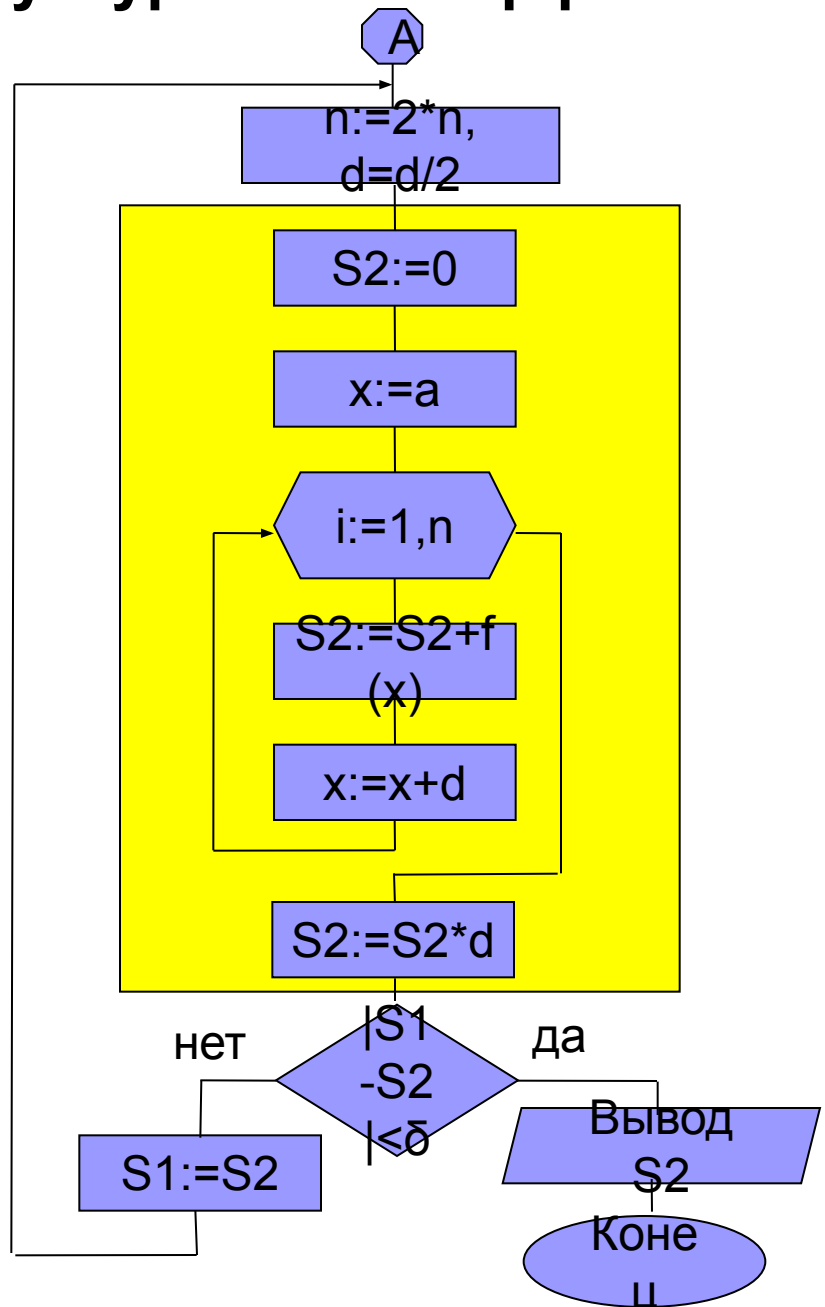
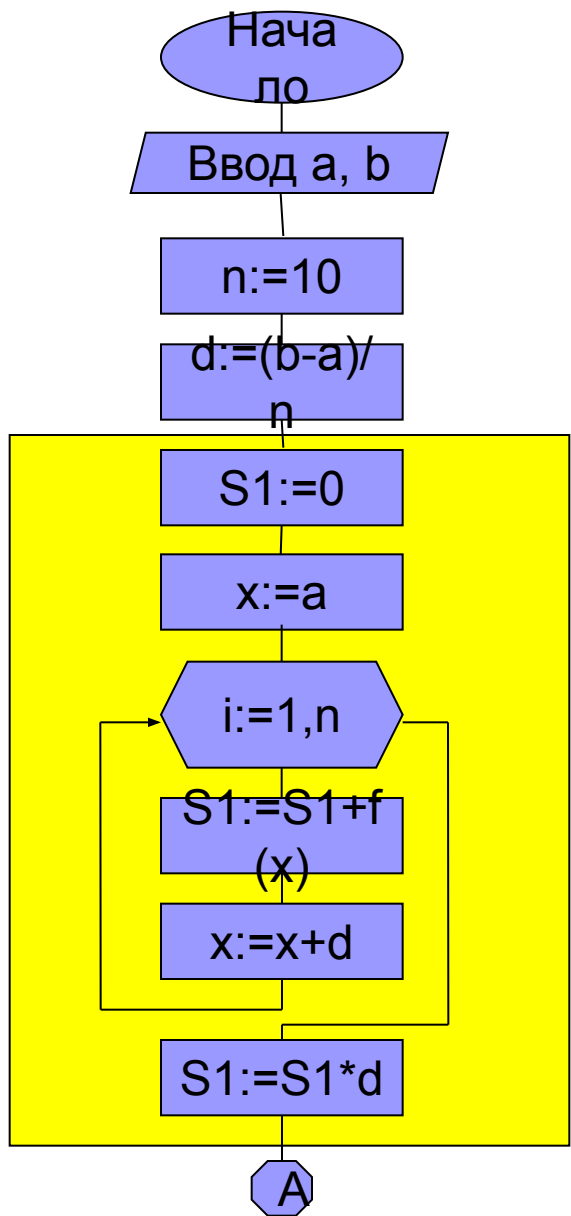
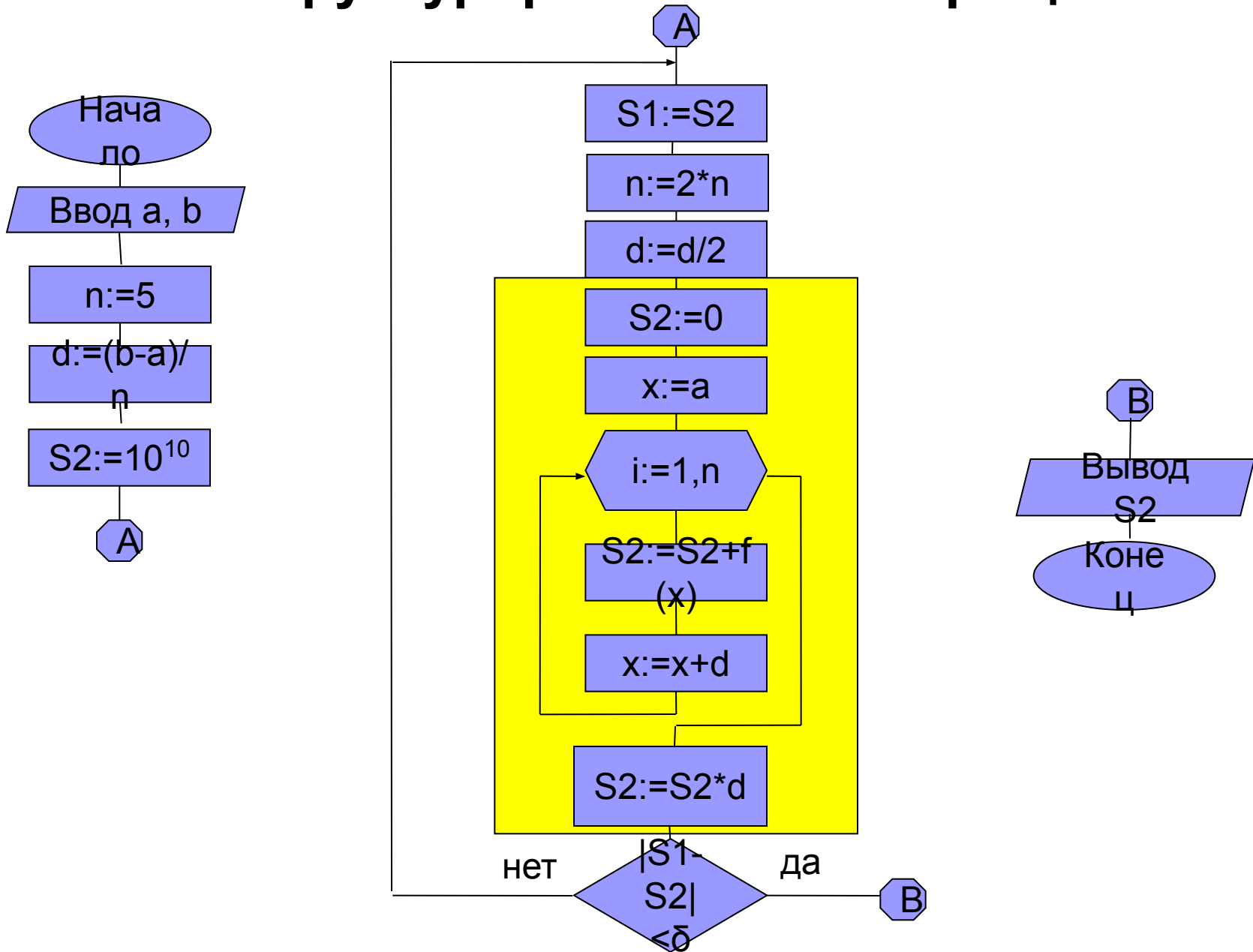


Схема структурированная и сокращенная



Программа

```
program Ex2_5;  
{$APPTYPE CONSOLE}
```

```
uses SysUtils;
```

```
Var a,b,S1,S2,d,eps,x:Single;  
    n,i:Integer;
```

```
Begin
```

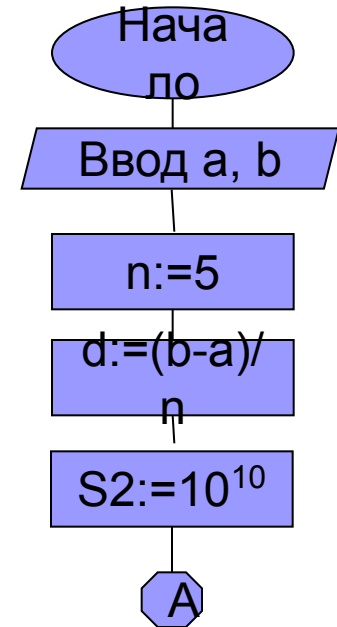
```
    WriteLn('Input a, b and eps:');
```

```
    ReadLn(a,b,eps);
```

```
    n:= 5;
```

```
    d:=(b-a)/n;
```

```
    S2:=1E+10;
```



Программа (2)

```
repeat
  S1 := S2 ;
  n := n * 2 ;
  d := d / 2 ;
  S2 := 0 ;
  x := a ;
  for i := 1 to n do
    begin
      S2 := S2 + x * x - 1 ;
      x := x + d ;
    end ;
  S2 := S2 * d ;
until abs (S2 - S1) < eps ;
WriteLn (' I = ' , S2 : 10 : 6 ) ;
ReadLn ;
```

End.

