

Рекурсивное программирование

Recursio (лат.) - возвращение

Рекурсия – это метод, сводящий общую задачу к некоторым задачам более узкого, простого типа

Рекурсивное программирование

Рекурсивный алгоритм – это алгоритм, который в процессе своей работы обращается сам к себе.

Рекурсивное программирование

Суть заключается в том, что при каждом вызове создается новая копия со своими переменными, но как только она заканчивает свою работу, то память, занятая этими локальными переменными, освобождается, а полученные результаты передаются в точку вызова.

Задачи с рекурсивной формулировкой

Пример: вычисление факториала натурального числа

$$N! = \begin{cases} 1, & \text{при } N = 1 \\ N * (N - 1)!, & \text{при } N > 1 \end{cases}$$

```
Function factorial(n: integer): longint;
```

```
Begin
```

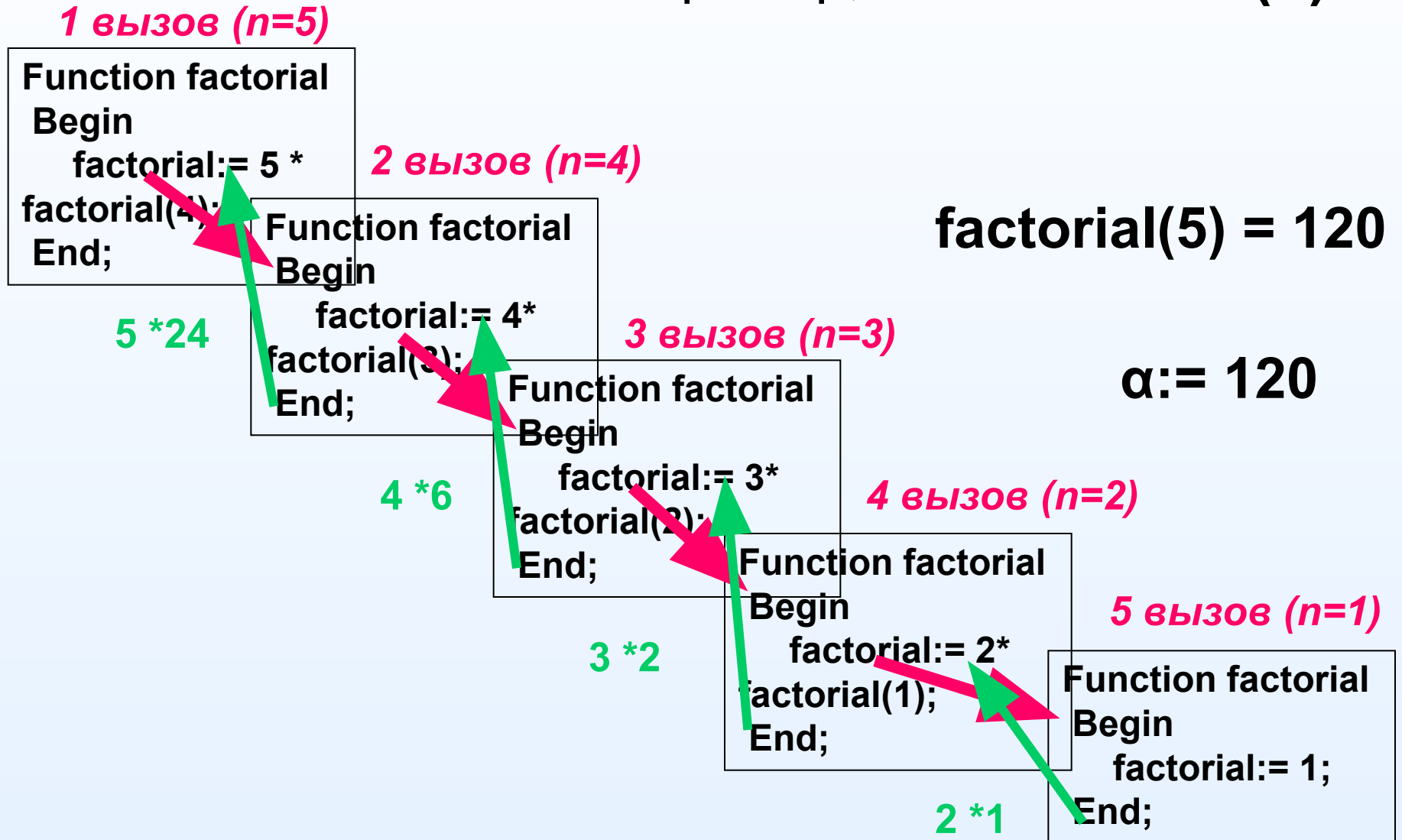
```
  If n = 1 then factorial:=1
```

```
  else factorial:= n * factorial (n - 1);
```

```
End;
```

Найдем 5!

Первый вызов этой функции будет из основной программы. Например, $\alpha := \text{factorial}(5)$



Задания

1. Составить рекурсивную программу ввода с клавиатуры последовательности чисел (окончание ввода - 0) и вывода ее на экран в обратном порядке.
2. Найти первые N чисел Фибоначчи. Каждое число равно сумме двух предыдущих чисел при условии, что первые два равны 1 (1, 1, 2, 3, 5, 8, 13, 21, ...)

$$\Phi(n) = \begin{cases} 1, & \text{если } n = 1 \text{ или } n = 2 \\ \Phi(n - 1) + \Phi(n - 2), & \text{при } n > 2 \end{cases}$$

Пример: перевод натурального числа из десятичной системы счисления в двоичную.

$$39_{10} = 100111_2$$

```
Procedure Rec(n: integer);  
begin  
  If n > 1 then Rec(n Div 2);  
  Write(n Mod 2);  
End;
```

1 вызов (n = 39)

```
Procedure Rec  
begin  
  Rec(n Div 2);  
  Write(n Mod 2);  
End;
```

2 вызов (n = 19)

```
Procedure Rec  
begin  
  Rec(n Div 2);  
  Write(n Mod 2);  
End;
```

3 вызов (n = 9)

```
Procedure Rec  
begin  
  Rec(n Div 2);  
  Write(n Mod 2);  
End;
```

4 вызов (n = 4)

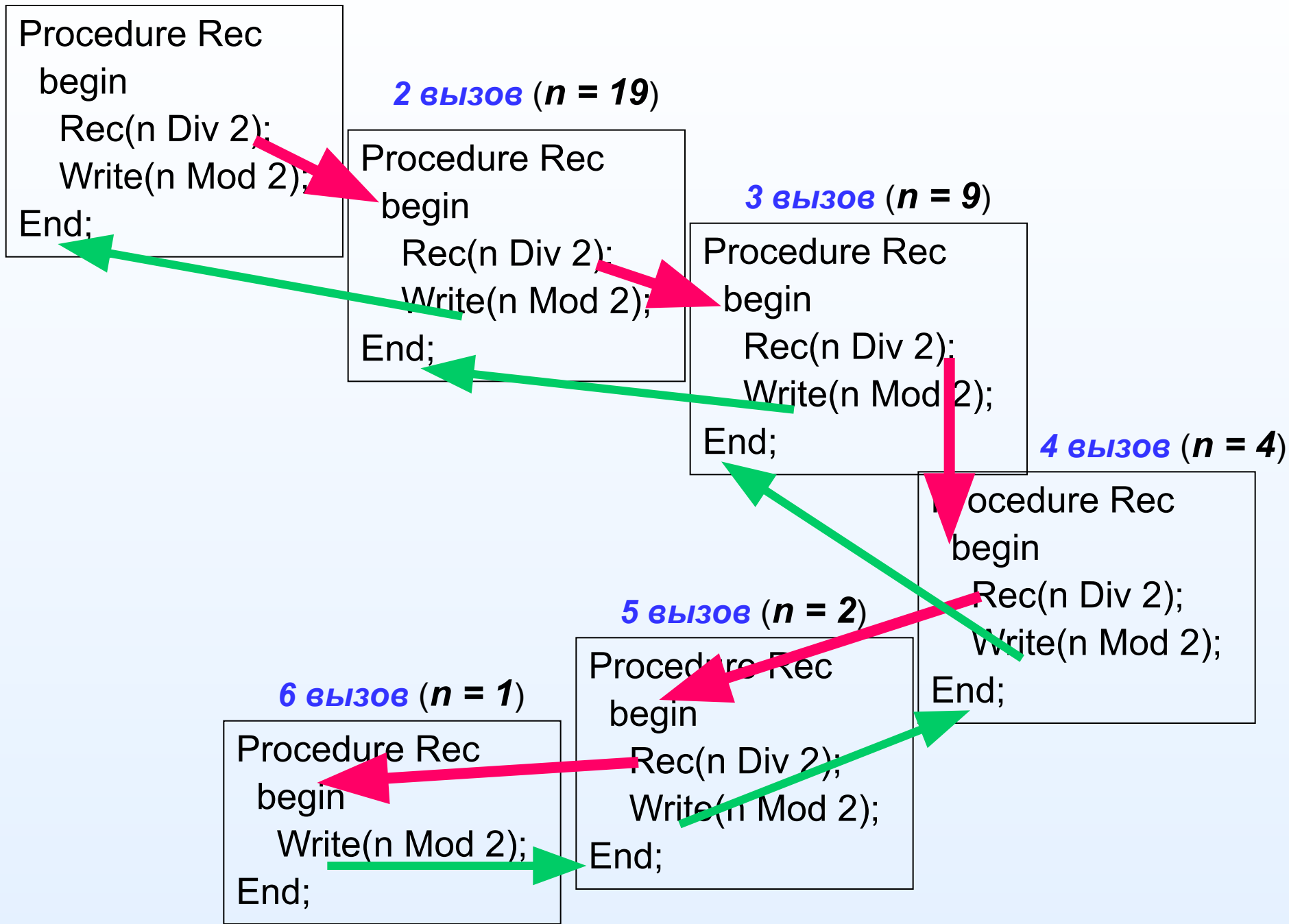
```
Procedure Rec  
begin  
  Rec(n Div 2);  
  Write(n Mod 2);  
End;
```

5 вызов (n = 2)

```
Procedure Rec  
begin  
  Rec(n Div 2);  
  Write(n Mod 2);  
End;
```

6 вызов (n = 1)

```
Procedure Rec  
begin  
  Write(n Mod 2);  
End;
```



Задание

1. Написать процедуру перевода из десятичной системы в N -ю, при условии, что $2 \leq N \leq 16$ и его значение вводить с клавиатуры. Каким будет условие завершения входа в рекурсию?

```
Procedure Picture1(x,y,r,r1,n:integer);  
Var x1,y1:integer; i:integer;  
Begin  
  if n > 0 then {"заглушка"}  
    begin  
      circle(x,y,r);r1:=trunc(r*k2); {рисование окружности}  
      r1:=trunc(r*k2)           {вычисление радиуса орбиты}  
      For i:=1 to 4 do  
        begin  
          x1:=trunc(x+r1*cos(pi/2*i); { координаты центра }  
          y1:=trunc(y+r1*sin(pi/2*i); { i-ой окружности }  
          Picture1(x1,y1,trunc(r*k1),r1,n-1);  
        end;  
      end;  
    end;  
end;
```



```
Uses Graph;
Var x,y,n,r,r1,cd,gm:integer; k1,k2:real;
Procedure Picture1(x,y,r,r1,n:integer);
...
End;
Begin
  Writeln('Введите количество уровней n'); Readln(n);
  x:=600 Div 2; y:=400 Div 2;
  Writeln('Введите радиус первой окружности r');
  readln(r);
  k1:=0.3; k2:=2;
  Cd:=detect; gm:=1;
  Initgraph(cd,gm,'c:\tp7\bin');
  Picture1(x,y,r,r1,n);
  Readln;
  Closegraph;
End.
```