

## Рекурсия. Рекурсивные алгоритмы

Рекурсивным называется объект, который частично определяется через самого себя. Конструкции, которые описываются с помощью рекуррентных формул, иногда называют рекурсивными определениями, например, определение факториала, числа Фибоначчи. Использование рекурсии позволяет запрограммировать вычисления по рекуррентным формулам. Рекурсивное определение позволяет с помощью конечного выражения определить бесконечное множество объектов. С помощью конечного рекурсивного алгоритма можно определить бесконечное вычисление, причем алгоритм не будет содержать повторов. Для создания рекурсивных алгоритмов необходимо и достаточно наличия процедуры или функции. Всякий рекурсивный алгоритм должен состоять из двух фрагментов: в первом фрагменте должен содержаться рекурсивный вызов этой же самой процедуры с измененным значением критерия останова, а во втором фрагменте производится останов алгоритма в случае, когда критерий останова достигает критического значения.

Рассмотрим вычисление факториала.

$$N! = N*(N-1)*(N-2)*\dots*2*1$$

Факториал можно вычислить рекурсивно, поскольку, факториал числа  $N$  можно легко получить из факториала числа  $N-1$ .

$$N! = N * (N-1)!$$

Алгоритм вычисления факториала можно записать так:

$$\text{Факториал}(N) = N * \text{Факториал}(N-1)$$

Это фрагмент рекурсивного обращения функции к самой себе.

С другой стороны, необходимо, чтобы процедура рекурсивного обращения функции к себе самой не была бесконечной. Завершить цепочку рекурсивных вызовов можно условием, что при  $N=0$  функция факториала не должна обращаться к себе, а должна вернуть значение 1. Таким образом, функцию вычисления факториала можно записать так:

Функция ФАКТОРИАЛ( $N$ )

Если  $N > 0$  Тогда

    Возврат  $N * \text{ФАКТОРИАЛ}(N-1)$

Иначе

    Возврат 1

КонецЕсли

Рассмотрим пример использования этой функции для вычисления, например,  $4!$

$$X = \text{ФАКТОРИАЛ}(4)$$

Пошагово опишем процесс вычисления факториала:

$$X = \text{ФАКТОРИАЛ}(4)$$

    Если  $4 > 0$  (истина)

        Возврат  $4 * \text{ФАКТОРИАЛ}(3)$

            Если  $3 > 0$  (истина)

                Возврат  $3 * \text{ФАКТОРИАЛ}(2)$

                    Если  $2 > 0$  (истина)

```

        Возврат 2*ФАКТОРИАЛ(1)
        Если 1>0 (истина)
        Возврат 1*ФАКТОРИАЛ(0)
        Если 0>0 (ложь)
        Возврат 1
        Возврат 1*1
        Возврат 2*1
        Возврат 3*2*1
        Возврат 4*3*2*1
X=4*3*2*1

```

### **Пример реализации рекурсивного вычисления 10 факториалов на языке VFP.**

```

FOR i=1 TO 10
? i,STR(факториал(i))
NEXT

function факториал
lparameters n
if n=0
RETURN 1
else
return n* факториал(n-1)
endif

```

### **Пример рекурсивной процедуры для вычисления факториала на языке VBA.**

```

Function Fact (n As Long) As Long
    If n = 1 Then
        Fact = 1
    Else
        Fact = Fi (n - 1) * n
    End If
End Function

```

### **Пример использования рекурсии для вычисления корня нелинейной функции методом деления отрезка пополам на языке VFP.**

Всякий рекурсивный алгоритм должен состоять из двух фрагментов: в первом фрагменте должен содержаться рекурсивный вызов этой же самой процедуры с измененным значением критерия останова (в нашем случае критерий останова—длина отрезка  $[a,b]$ ), а во втором фрагменте производится останов алгоритма в случае, когда критерий останова достигает критического значения (в данном случае критическое значение критерия останова—точность  $\epsilon$ , с которой определяется корень, т.е.  $|a-b|<\epsilon$ ).

```

PUBLIC i
i=0
x=0
a=3
b=10
fa=f(a)
fb=f(b)

```

```
r= корень(@a,@b,@fa,@fb,0.001)
?r
```

```
function корень
lparameters a,b,fa,fb,eps
LOCAL x, fx
DO case
    CASE fa*fb>0
        RETURN .f.
    case fa=0
        RETURN a
    case fb=0
        RETURN b
    OTHERWISE

    if ABS(a-b)<eps
        RETURN (b+a)/2
    ELSE
        x=(b+a)/2
        fx=f(x)
        i=i+1
? i,x, fx
        IF fx*fa<=0
            fb=fx
            b=x
        ELSE
            fa=fx
            a=x
        ENDIF
        RETURN корень(@a,@b,@fa,@fb,0.001)
    ENDIF
ENDCASE
```

```
FUNCTION f
LPARAMETERS x
RETURN (x-8)*(x-14)
```