

Программирование

Лекция 11 Рекурсия

Петров Александр Владимирович
Фёдоров Станислав Алексеевич

(по материалам Веренинова Игоря Андреевича с
изменениями и дополнениями на Fortran 08 и
UML)

Октябрь 2013

Определение

Рекурсия — частичное определение объекта через себя.

Рекурсия используется, когда можно выделить самоподобие задачи.

Рекурсия - это способ организации вычислительного процесса, при котором процедура или функция в ходе выполнения составляющих ее операторов обращается сама к себе.

Рекурсия в математике

- ❑ Факториал целого неотрицательного числа n .

$$n! = \begin{cases} n \cdot (n-1)!, & n > 0 \\ 1, & n = 0 \end{cases}$$

- ❑ Числа Фибоначчи

$$f = \begin{cases} f(0) = 0; \\ f(n) = n + f(n-1), & n > 0 \end{cases}$$

Рекурсия в физике

- ❑ Классическим примером бесконечной рекурсии являются два поставленные друг напротив друга зеркала: в них образуются два коридора из затухающих отражений зеркал.



Рекурсия в лингвистике

- ❑ Вложенность идеи в идею

«Мыслю, следовательно, существую»

Рене Декарт

Рекурсивная функция Multiply

```
recursive pure integer function Multiply(M, N)
  integer, intent(in)  :: M, N

  if (N = 1) then
    Multiply = M
  else
    Multiply = M + Multiply(M, N-1)
  end if
end function Multiply
```

Трассировка функции Multiply

Multiply(6, 3) — Исходный вызов

М равно 6
N равно 3
3 = 1 равно false
Multiply = 6 + Multiply(6, 2)
Возврат

Рекурсивный
вызов 1

М равно 6
N равно 2
2 = 1 равно false
Multiply = 6 + Multiply(6, 1)
Возврат

Рекурсивный
вызов 2

М равно 6
N равно 1
1 = 1 равно true
Multiply = 6
Возврат

12

6

Рекурсивная функция Factorial

```
recursive integer pure function Factorial(N)
  integer, intent(in) :: N

  if (N == 0) then
    Factorial = 1
  else
    Factorial = N * Factorial(N-1)
  end if
end function Factorial
```


Хвостовая рекурсия

Хвостовая рекурсия — специальный случай рекурсии, при котором любой рекурсивный вызов является последней операцией перед возвратом из функции.

Данный вид рекурсии примечателен тем, что может быть легко заменён на итерацию.

Рекурсивная функция Factorial (2)

```
recursive integer pure function Fact_Times(N, Acc)
  integer, intent(in)  :: N, Acc

  if (N == 0) then
    Fact_Times = Acc
  else
    Fact_Times = Fact_Times(N - 1, Acc * N)
  end if
end function Fact_Times

recursive integer pure function Factorial(N)
  integer, intent(in)  :: N

  Factrorial = Fac_Times (N, 1)
end function
```

Замечания

- ❑ Следует избегать рекурсии, если есть очевидное решение использующее итерацию.
- ❑ Алгоритмы, которые по своей природе являются рекурсивными, а не итерационными, должны программироваться в виде рекурсивных процедур.