



Rekursivní volání

= volání sebe sama (přímo NEBO zprostředkovaně)

Rekursivní definice:

$$\begin{aligned} N! &= 1 && \text{pro } N \leq 1 \\ &N \times (N-1)! && \text{pro } N > 1 \end{aligned}$$

Rekursivní, *adj.*: viz Rekursivní

V Pascalu:

```
function fakt(N: integer): integer;  
begin  
    if N <= 1 then fakt := 1  
        else fakt := N*fakt(N-1)  
end;
```



rekursivní volání

opětovné (rekursivní volání)

=> nový exemplář funkce fakt
s novými parametry
s novými proměnnými...

?

```
function fakt2( N: integer ): integer;  
begin  
    fakt2 := N*fakt2( N-1 )  
end;
```

? co udělá tato procedura:

```
procedure P;  
var  c: char;  
begin  
    read( c );  
    if c<>' ' then  
    begin  
        P;  
        write( c )  
    end  
end;  
end;
```

BP: call stack

Klady a zápory rekurse

- + zjednodušení algoritmu (některých)
- zpravidla náročnější na čas i paměť

Odstrašující příklad:

Fibonacciho posloupnost: $a_i = a_{i-1} + a_{i-2}$

```
function fib( i: integer ): integer;  
begin  
    if i <= 1 then fib := 1  
        else fib = fib(i-1)+fib(i-2)  
end
```

Náprava ukládáním výsledků...

Příklady

Hanoj
variace
kombinace
permutace

(loupežníci)

