

Programação 2

Jordana S. Salamon

jssalamon@inf.ufes.br

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Recursividade

- ▶ De forma geral a recursividade ocorre quando algo é definido a partir de si mesmo;
- ▶ Os elementos básicos para se definir uma recursão são:
 - ▶ Uma ou mais relações recursivas: quando o próximo elemento é definido a partir do anterior;
 - ▶ Uma ou mais bases de recursão: pontos de parada;

$\text{lista} = \text{lista_vazia} \Rightarrow \text{base da recursao}$

$\text{lista} = \text{elemento} + \text{lista} \Rightarrow \text{relação recursiva}$

- ▶ Em programação, um método recursivo pode ser usado para definir uma função;

```
1  int fatorial (int n){  
2      if (n == 0){  
3          return 1;  
4      }else{  
5          return n * fatorial (n-1);  
6      }  
7  }
```

Recursividade

- ▶ As chamadas recursivas não geram código otimizado, porém um código recursivo pode ser a única forma de resolver um problema.
- ▶ Exemplo da implementação de chamadas recursivas:

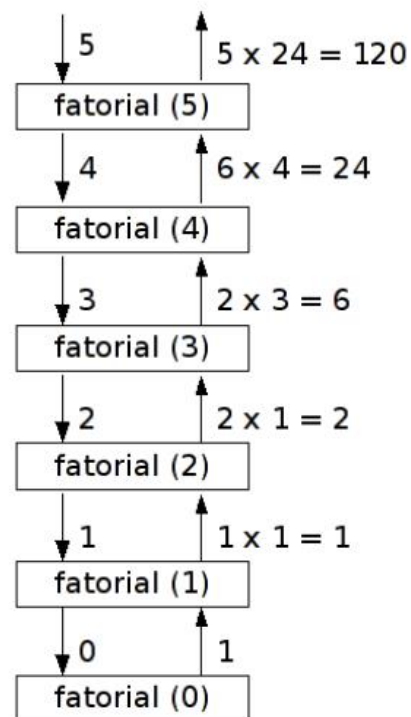


Figura 3.6: Fatorial utilizando recursão.

Exercícios

1. Escreva uma função recursiva que calcule o n-ésimo termo da sequência de fibonacci.
2. Escreva uma função recursiva para calcular a soma dos n primeiros números inteiros, onde n será lido do teclado.
3. Escreva uma função recursiva para calcular o valor de uma base x elevada a um expoente y.
4. Escreva uma função recursiva que calcule o número de grupos distintos com k pessoas que podem ser formados a partir de um conjunto de n pessoas. A definição abaixo da função $Comb(n,k)$ define as regras:

$$Comb(n, k) = \begin{cases} n & \text{se } k = 1 \\ 1 & \text{se } k = n \\ Comb(n-1, k-1) + Comb(n-1, k) & \text{se } 1 < k < n \end{cases}$$

That's all Folks!



nemo