

PROGRAMAÇÃO ORIENTADA A OBJETOS

Recursão (continuação)

Prof. Emanuel Barreiros



Recursão vs. Iteração

- Estudamos dois problemas naturalmente recursivos e vimos como podem ser facilmente implementados usando-se a recursão
- Tais problemas podem, no entanto serem implementados iterativamente
- Iremos analisar a diferença entre essas duas abordagens

Estruturas de controle

- Tanto iteração quanto recursão são baseadas em estruturas de controle
- Iteração usa laços de repetição: for, while ou do...while
- Recursão usa comandos de seleção: if, if...else ou switch
- Tanto iteração quanto recursão envolvem repetição, embora usem mecanismos diferentes
- Podemos identificar as diferenças básicas?

Fatorial recursivo

```
1 package br.upe.rekursao;  
2  
3 public class Fatorial {  
4  
5     public int fatorial(int n) {  
6         if(n == 1) {  
7             return 1;  
8         } else {  
9             return n * fatorial(n - 1);  
10        }  
11    }  
12 }
```

Fatorial iterativo

- Como podemos implementar o fatorial de maneira iterativa?

Fibonacci iterativo

- Como podemos implementar o Fibonacci de maneira iterativa?

Fechando

- Iteração e recursão são baseados em estruturas de controle: iteração usa repetição e recursão usa seleção
- Ambas envolvem repetição
- Ambas envolvem testes de terminação: a iteração termina quando o loop encerra, a recursão termina quando o caso base é reconhecido
- Ambas caminham passo a passo em direção à solução
- Tanto iteração quanto recursão podem ocorrer indefinidamente
- Recursão repetidamente invoca o mesmo método repetidamente, logo, adicionando este *overhead*

Fechando

- Qualquer problema que pode ser resolvido recursivamente, pode ser resolvido iterativamente
- Uma abordagem recursiva é preferida em detrimento da iterativa quando a primeira modela melhor a maneira de pensar e gera, no fim das contas, um programa mais fácil de ler
- Uma solução recursiva geralmente usa menos linhas de código que a sua análoga iterativa