

Métodos Recursivos

Métodos Recursivos

- **Recursividade** é um termo usado de maneira mais geral para descrever o processo de repetição de um objeto de um jeito similar ao que já fora mostrado. Um bom exemplo disso são as imagens repetidas que aparecem quando dois espelhos são apontados um para o outro;
- Na computação, a recursão especifica métodos muito simples e, a partir dele, formula casos mais complexos;
- A recursão na programação é bem exemplificada quando um método é definido em termos de si mesmo, ou seja, dentro dele há uma chamada para ele mesmo.

Métodos Recursivos

Um exemplo clássico de recursão é a definição do fatorial:

- Fatorial de $N = N! = N * (N - 1)!$
- Fatorial de $0 = 0! = 1$

Portanto:

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

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

$$(N - 2)! = (N - 2) * (N - 3)!$$

$$(N - 3)! = (N - 3) * (N - 4)!$$

$$\vdots$$

$$(N - (N - 1))! = (N - (N - 1)) * (N - N)! = 1 * 0! = 1 * 1 = 1$$

Estruturas de Dados e Análise de Algoritmos

Métodos Recursivos

Exemplo: Cálculo de 4!

- Fatorial de $N = N! = N * (N - 1)!$
- Fatorial de $0 = 0! = 1$

$$4! = 4 * (4 - 1)! = 4 * 3!$$

$$3! = 3 * (3 - 1)! = 3 * 2!$$

$$2! = 2 * (2 - 1)! = 2 * 1!$$

$$1! = 1 * (1 - 1)! = 1 * 0! = 1 * 1 = 1$$

$$2! = 2 * 1! = 2 * 1 = 2$$

$$3! = 3 * 2! = 3 * 2 = 6$$

$$4! = 4 * 3! = 4 * 6 = 24$$

- A definição do método fatorial, dado aqui em pseudocódigo:

```
função fatorial (n)
{
    se (n <= 1)    retorne 1;
    senão  retorne n * fatorial (n-1);
}
```

Estruturas de Dados e Análise de Algoritmos

Métodos Recursivos

- Em Java, tem-se:

```
public static int Fatorial(int iN)
{
    if(iN<=1) return 1
    else return iN * Fatorial(iN-1);
}
```

Métodos Recursivos

- Teste de Mesa:

```
public static int Fatorial(int iN)
{
    if(iN<=1) return 1
    else return iN * Fatorial(iN-1);
}
```

```
//Calcula Fatorial
int iNumero;
iNumero = 4;
int iFatorial = Fatorial(iNumero);
```

Estruturas de Dados e Análise de Algoritmos

Métodos Recursivos

```
public static int Fatorial(int iN)           //Calcula Fatorial
{
    if(iN<=1) return 1
    else return iN * Fatorial(iN-1);
}
```

```
int iNumero;
iNumero = 4;
int iFatorial = Fatorial(iNumero);
```

```
public static int Fatorial(4)
{
    if(4 <= 1) return 1
    else return 4 * Fatorial(3);
}
```

retorna 24

retorna 6

Fatorial(3)

```
public static int Fatorial(3)
{
    if(3 <= 1) return 1
    else return 3 * Fatorial(2);
}
```

retorna 2

Fatorial(2)

```
public static int Fatorial(2)
{
    if(2 <= 1) return 1
    else return 2 * Fatorial(1);
}
```

retorna 1

Fatorial(1)

```
public static int Fatorial(1)
{
    if(1 <= 1) return 1
    else return 1 * Fatorial(0);
}
```


Métodos Recursivos

Outro exemplo clássico de recursão é a definição de Fibonacci:

- Fibonacci de $n = F_n = F_{n-1} + F_{n-2}$
- Fibonacci de $0 = F_0 = 0$
- Fibonacci de $1 = F_1 = 1$

Portanto: $F = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots, \infty$

Métodos Recursivos

Exemplo: O elemento 6 da série de Fibonacci é 8:

- Fibonacci de $n = F_n = F_{n-1} + F_{n-2}$
- Fibonacci de $0 = F_0 = 0$
- Fibonacci de $1 = F_1 = 1$

Portanto: $F = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots, \infty$

$$F_6 = F_5 + F_4 = 5 + 3 = 8 \Rightarrow F_6 = 8$$

$$F_5 = F_4 + F_3 = 3 + 2 = 5$$

$$F_4 = F_3 + F_2 = 2 + 1 = 3$$

$$F_3 = F_2 + F_1 = 1 + 1 = 2$$

$$F_2 = F_1 + F_0 = 1 + 0 = 1$$

$$F_1 = 1$$

$$F_0 = 0$$

Métodos Recursivos

- A definição do método fibonacci, em pseudocódigo:

```
função fibonacci (n)
{
    se (n == 0)    retorne 0;
    senão se (n == 1) retorne 1;
    senão retorne fibonacci (n - 1) + fibonacci (n - 2);
}
```

Estruturas de Dados e Análise de Algoritmos

Métodos Recursivos

- Em Java, tem-se:

```
public static int Fibonacci(int iT)
{
    if(iT == 0) return 0;
    else if(iT == 1) return 1;
    else return Fibonacci(iT-1) + Fibonacci(iT-2);
}
```

Métodos Recursivos

- Teste de Mesa:

```
public static int Fibonacci(int iT)
{
    if(iT == 0) return 0;
    else if(iT == 1) return 1;
    else return Fibonacci(iT-1) + Fibonacci(iT-2);
}

//Calcula
int iTermo;
iTermo = 6;
int iFibonacci=Fibonacci(iTermo);
```

Estruturas de Dados e Análise de Algoritmos

Métodos Recursivos

- Fazer o teste de mesa para o 6º termo da série de Fibonacci:

Métodos Recursivos

- Exercícios:

1º) Desenvolver o método recursivo capaz de calcular o 11º número ímpar, positivo, adotando-se o 1º número ímpar positivo igual a 1.

2º) Desenvolver o método recursivo capaz de calcular o 10º termo de uma Progressão Geométrica – PG (a_{10}), onde o 1º termo (a_1) é igual à 1 e a razão da PG (q) é igual à 2.

Estruturas de Dados e Análise de Algoritmos

Teoria

FIM