

Recursividade

Estruturas de Dados e Algoritmos

Prof. Vinicius Ramos

Slides do prof. **Cristian Cechinel**



Recursividade

- ⦿ Subrotina ou método que pode chamar (invocar) a si mesma.
- ⦿ Utilizada na construção e no percorrimto de diversas estruturas de dados existentes, como as listas e as árvores.
- ⦿ Um método é recursivo quando se utiliza dele próprio em sua definição.
- ⦿ Os métodos recursivos são internamente divididos entre os casos considerados base (critérios de parada) e os casos recursivos (chamadas ao próprio método).

Recursividade

- ⦿ A execução de um método recursivo faz com que o mesmo método seja chamado sucessivas vezes até que o critério de parada estabelecido seja encontrado.
- ⦿ As diferentes execuções do método são empilhadas na memória sucessivamente até que uma última chamada ao método seja realizada e encontre o critério de parada.

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de $N-1$

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de $N-1$

fatorial(5)

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de $N-1$

fatorial(5)

$5 * \text{fatorial}(4)$

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de N-1

fatorial(5)

5 * fatorial(4)

5 * 4 * fatorial(3)

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de N-1

fatorial(5)

5 * fatorial(4)

5 * 4 * fatorial(3)

5 * 4 * 3 * fatorial(2)

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de N-1

fatorial(5)

5 * fatorial(4)

5 * 4 * fatorial(3)

5 * 4 * 3 * fatorial(2)

5 * 4 * 3 * 2 * fatorial(1)

Exemplo de definição recursiva - Fatorial

- ☉ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de $N-1$

fatorial(5)

$5 * \text{fatorial}(4)$

$5 * 4 * \text{fatorial}(3)$

$5 * 4 * 3 * \text{fatorial}(2)$

$5 * 4 * 3 * 2 * \text{fatorial}(1)$

$5 * 4 * 3 * 2 * 1 *$

fatorial(0)

Exemplo de definição recursiva - Fatorial

- ⊙ O cálculo do fatorial de um número N pode ser definido como o valor de N multiplicado pelo fatorial de N-1

fatorial(5)

5 * fatorial(4)

5 * 4 * fatorial(3)

5 * 4 * 3 * fatorial(2)

5 * 4 * 3 * 2 * fatorial(1)

5 * 4 * 3 * 2 * 1 *

fatorial(0)

5 * 4 * 3 * 2 * 1 * 1

5 * 4 * 3 * 2 * 1

5 * 4 * 3 * 2

5 * 4 * 6

5 * 24

120

Exemplo de definição recursiva - Fatorial

- Qual foi o critério de parada para a definição recursiva abaixo ?

fatorial(5)

5 * fatorial(4)

5 * 4 * fatorial(3)

5 * 4 * 3 * fatorial(2)

5 * 4 * 3 * 2 * fatorial(1)

5 * 4 * 3 * 2 * 1 *

fatorial(0)

5 * 4 * 3 * 2 * 1 * 1

5 * 4 * 3 * 2 * 1

5 * 4 * 3 * 2

5 * 4 * 6

5 * 24

120

Exemplo de definição recursiva - Fatorial

- ☉ Qual foi o critério de parada para a definição recursiva abaixo ?

fatorial(5)

5 * fatorial(4)

5 * 4 * fatorial(3)

5 * 4 * 3 * fatorial(2)

5 * 4 * 3 * 2 * fatorial(1)

5 * 4 * 3 * 2 * 1 *

fatorial(0)

5 * 4 * 3 * 2 * 1 * **1**

5 * 4 * 3 * 2 * 1

5 * 4 * 3 * 2

5 * 4 * 6

5 * 24

120

Exemplo de definição recursiva - Fatorial

```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```

Exemplo de definição recursiva - Fatorial

🎯 Pilha de Execução

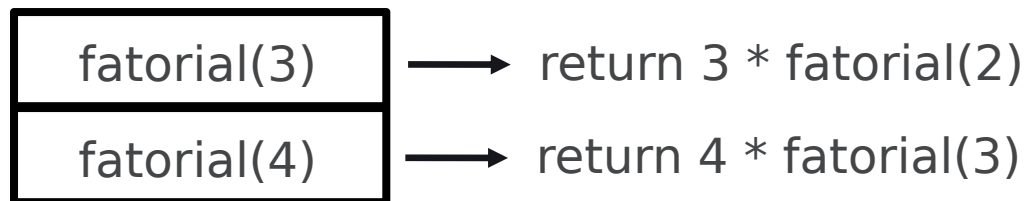
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```

fatorial(4) → return 4 * fatorial(3)

Exemplo de definição recursiva - Fatorial

🕒 Pilha de Execução

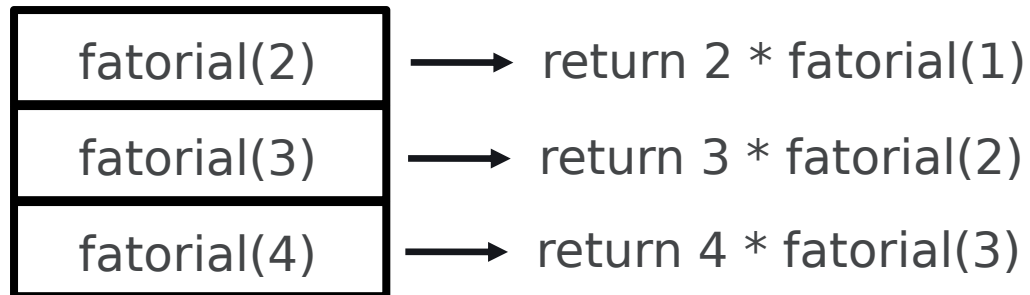
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🕒 Pilha de Execução

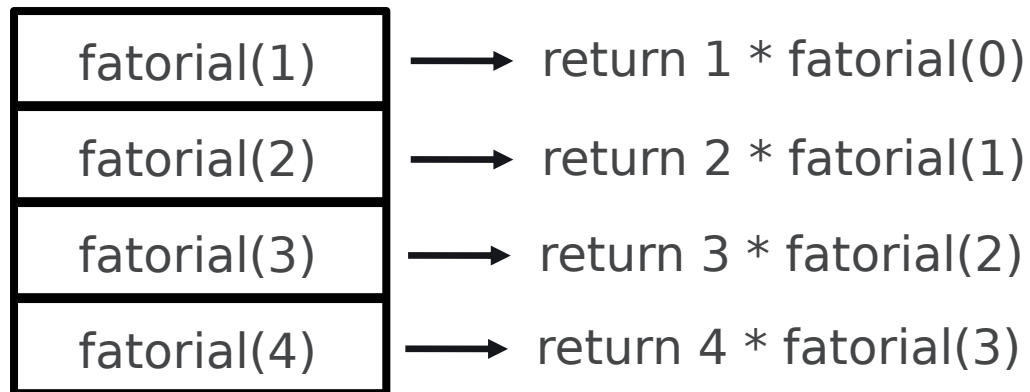
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🕒 Pilha de Execução

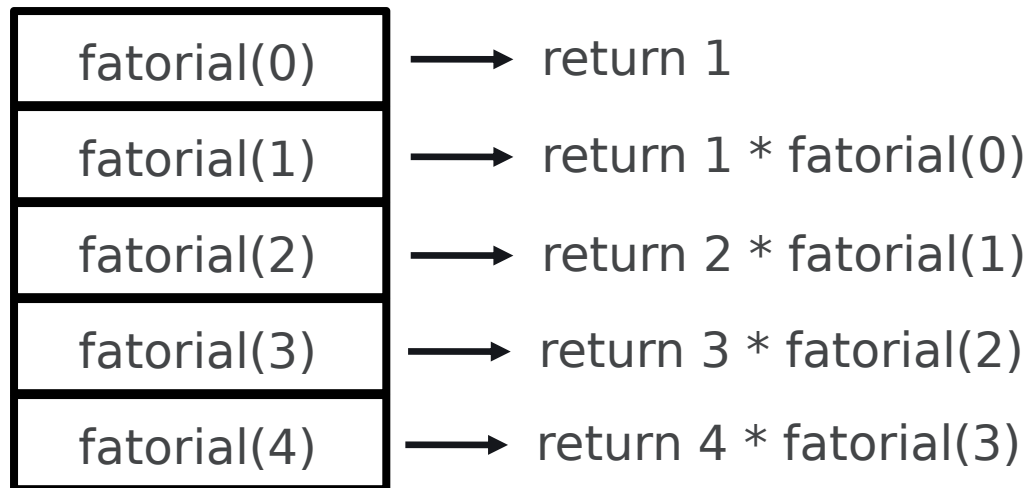
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🕒 Pilha de Execução

```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```

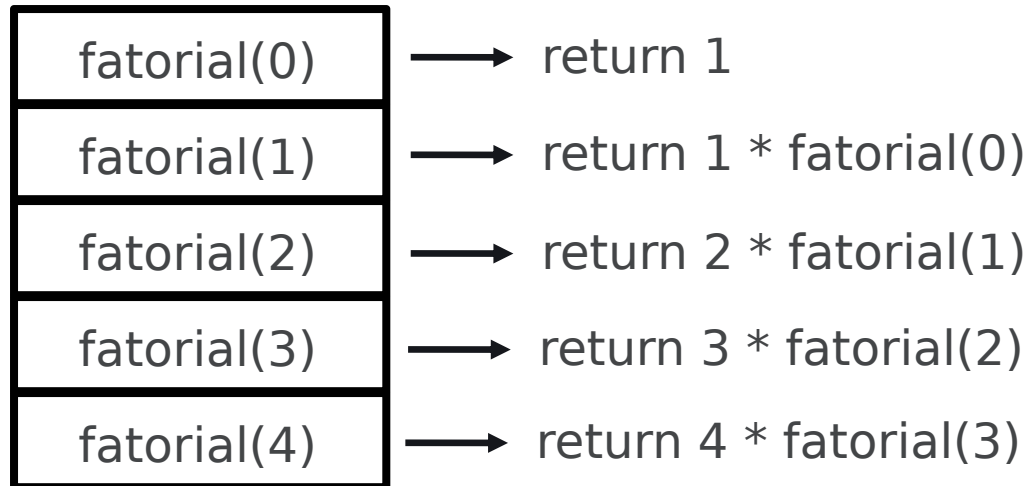


Exemplo de definição recursiva - Fatorial

🕒 Pilha de Execução

```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```

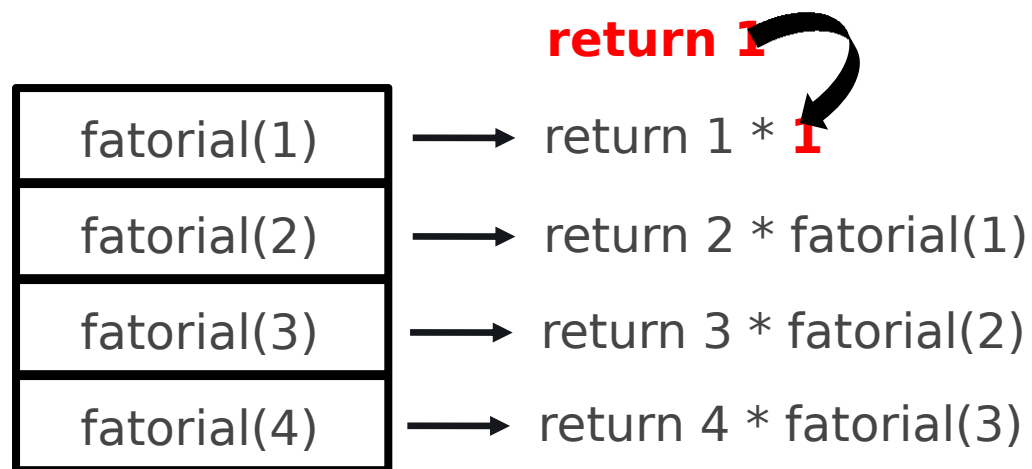
Desempilha



Exemplo de definição recursiva - Fatorial

🎯 Pilha de Execução

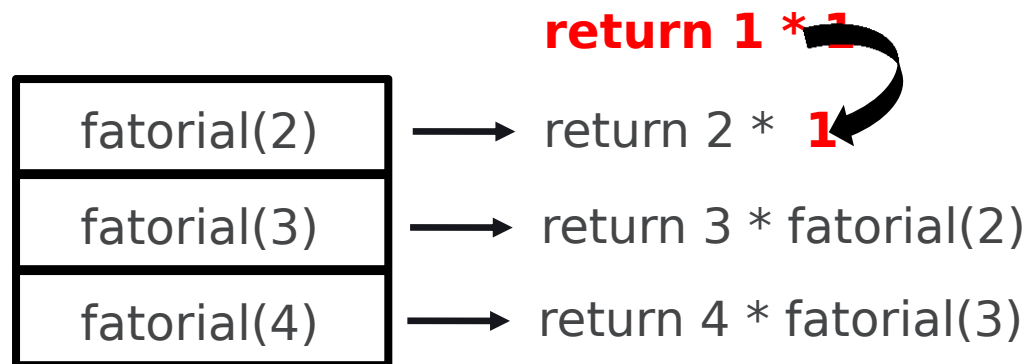
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🎯 Pilha de Execução

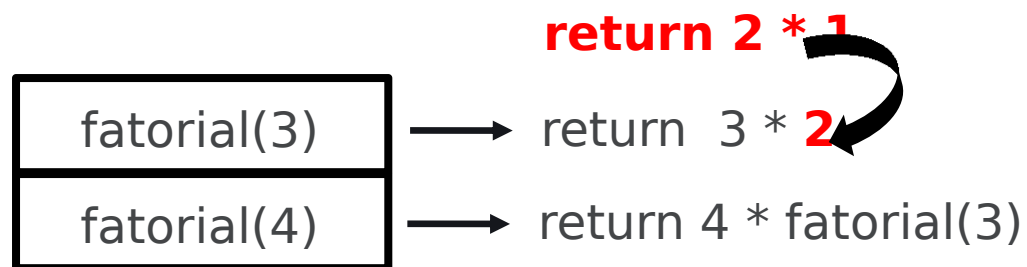
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🎯 Pilha de Execução

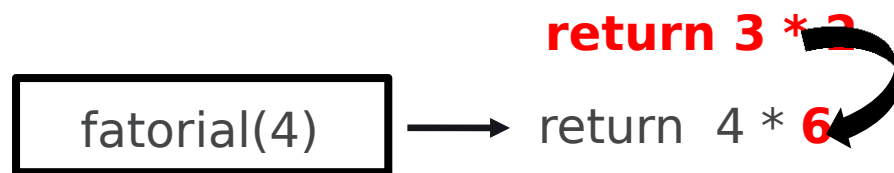
```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🎯 Pilha de Execução

```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```



Exemplo de definição recursiva - Fatorial

🎯 Pilha de Execução

```
public int fatorial(int n){  
    if (n == 0){  
        return 1;  
    }  
    else {  
        return n* fatorial(n-1);  
    }  
}
```

Resultado = **24**

Exercício - Máximo Divisor Comum

- ⦿ Faça uma função recursiva para calcular o Máximo Divisor Comum (MDC) entre dois números. Uma forma de se calcular o MDC entre dois número x e y é:
 - ⦿ $\text{MDC} (X, Y) =$
 - ⦿ (a) Y se $Y \leq X$ e $X \bmod Y = 0$
 - ⦿ (b) $\text{MDC} (Y, X)$ se $X < Y$
 - ⦿ (c) $\text{MDC} (Y, X \bmod Y)$ se caso contrário

Exercício - Máximo Divisor Comum

```
public int mdc (int x, int y){  
    if (y <= x && x % y == 0 ) {  
        return y;  
    }  
    else if ( x < y ){  
        return mdc( y , x );  
    }  
    else {  
        return mdc(y, x % y);  
    }  
}
```

Exercício - Fatorial Duplo

- ⊗ O fatorial duplo de um número natural n é o produto de todos os números de 1 (ou 2) até n , contados de 2 em 2. Por exemplo, o fatorial duplo de 8 é $8 \times 6 \times 4 \times 2 = 384$, e o fatorial duplo de 7 é $7 \times 5 \times 3 \times 1 = 105$.
- ⊗ Defina uma função recursiva para calcular o fatorial duplo

Exercício - Fatorial Duplo

```
public int fatduplo ( int n ){  
    if (n == 1) {  
        return 1;  
    }  
    else if (n == 2) {  
        return 2;  
    }  
    else {  
        return fatduplo ( n - 2 ) * n;  
    }  
}
```

Exercícios

- ④ Desenvolva um procedimento recursivo que imprima os números de 10 a 1
- ④ Defina uma função recursiva que recebe dois números naturais m e n e retorna o produto de todos os números no intervalo $[m,n]$: $m \times (m+1) \times \dots \times (n-1) \times n$
- ④ Desenvolva uma função recursiva que calcule a potência de um número inteiro elevado a um expoente também inteiro.
- ④ Desenvolva uma função recursiva que retorne o n ésimo termo da sequência de fibonacci

Referências utilizadas

- ⦿ <https://www.slideshare.net/skosta/aula-recursividade>
- ⦿ <http://www.decom.ufop.br/romildo/2012-1/bcc222/slides/06-recursividade.pdf>