

C++

Básico ao Avançado

Vire o papel

Heitor Rodrigues Savegnago

UFABC Rocket Design

2017.3

1 Polimorfismo

2 Recursão

3 Hora de brincar

Qual nome?

Qual nome?

- Fazer função para calcular média

Qual nome?

- Fazer função para calcular média
- Média de 2 valores

Qual nome?

- Fazer função para calcular média
- Média de 2 valores, 3 valores

Qual nome?

- Fazer função para calcular média
- Média de 2 valores, 3 valores, 4 valores

Qual nome?

- Fazer função para calcular média
- Média de 2 valores, 3 valores, 4 valores. . .

Qual nome?

- Fazer função para calcular média
- Média de 2 valores, 3 valores, 4 valores. . .
- Qual nome utilizar?

Qual nome?

- Fazer função para calcular média
- Média de 2 valores, 3 valores, 4 valores...
- Qual nome utilizar?

```
double mediaDois(double A, double B)
{
    return (A+B)/2.0;
}
```

```
double mediaTres(double A, double B, double C)
{
    return (A+B+C)/3.0;
}
```

```
double mediaQuatro(double A, double B, double C, double
    D)
{
    return (A+B+C+D)/4.0;
}
```

Qual nome?

- Fazer função para calcular média
- Média de 2 valores, 3 valores, 4 valores...
- Qual nome utilizar?

```
double mediaDois(double A, double B)
{
    return (A+B)/2.0;
}
```

```
double mediaTres(double A, double B, double C)
{
    return (A+B+C)/3.0;
}
```

```
double mediaQuatro(double A, double B, double C, double
    D)
{
    return (A+B+C+D)/4.0;
}
```

- E se você pudesse utilizar o mesmo nome?

O mesmo nome

O mesmo nome

- Você pode!

O mesmo nome

- Você pode!
- *E se ligar agora, tem o incrível desconto de...*

O mesmo nome

- Você pode!
- *E se ligar agora, tem o incrível desconto de...*
- Como?

O mesmo nome

- Você pode!
- *E se ligar agora, tem o incrível desconto de...*
- Como?
- Simples, faça!

O mesmo nome

- Você pode!
- *E se ligar agora, tem o incrível desconto de...*
- Como?
- Simples, faça!

```
double media(double A, double B)
{
    return (A+B)/2.0;
}
```

```
double media(double A, double B, double C)
{
    return (A+B+C)/3.0;
}
```

```
double media(double A, double B, double C, double D)
{
    return (A+B+C+D)/4.0;
}
```

O mesmo nome

- Você pode!
- *E se ligar agora, tem o incrível desconto de...*
- Como?
- Simples, faça!

```
double media(double A, double B)
{
    return (A+B)/2.0;
}
```

```
double media(double A, double B, double C)
{
    return (A+B+C)/3.0;
}
```

```
double media(double A, double B, double C, double D)
{
    return (A+B+C+D)/4.0;
}
```

- O compilador resolve (argumentos)

Explicando recursão

Explicando recursão

- Como explicar recursão?

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão

Explicando recursão

- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- Como explicar recursão?
- Explicar fazendo recursão
- Mas o que é recursão?
- Fazer recursão sem saber recursão
- ...

É o que?

É o que?

- Um procedimento que chama a si próprio

É o que?

- Um procedimento que chama a si próprio
- Precisa de uma limitação

É o que?

- Um procedimento que chama a si próprio
- Precisa de uma limitação
- *stack overflow*

É o que?

- Um procedimento que chama a si próprio
- Precisa de uma limitação
- *stack overflow*
- Simplificar problema

É o que?

- Um procedimento que chama a si próprio
- Precisa de uma limitação
- *stack overflow*
- Simplificar problema
- Condição de parada

É o que?

- Um procedimento que chama a si próprio
- Precisa de uma limitação
- *stack overflow*
- Simplificar problema
- Condição de parada
- Passo de recursão

É o que?

- Um procedimento que chama a si próprio
- Precisa de uma limitação
- *stack overflow*
- Simplificar problema
- Condição de parada
- Passo de recursão
- Caso trivial

Você já viu um

Você já viu um

- A função recursiva mais famosa é o fatorial!

Você já viu um

- A função recursiva mais famosa é o fatorial!
- O que é o fatorial?

Você já viu um

- A função recursiva mais famosa é o fatorial!
- O que é o fatorial?

$$f(n) = n! \equiv n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 2 \cdot 1 \quad (1)$$

Você já viu um

- A função recursiva mais famosa é o fatorial!
- O que é o fatorial?

$$f(n) = n! \equiv n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \quad (1)$$

$$n! = \begin{cases} n \leq 1 & \Rightarrow 1 \\ n \not\leq 1 & \Rightarrow n \cdot (n-1)! \end{cases} \quad (2)$$

Você já viu um

- A função recursiva mais famosa é o fatorial!
- O que é o fatorial?

$$f(n) = n! \equiv n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \quad (1)$$

$$n! = \begin{cases} n \leq 1 & \Rightarrow 1 \\ n \not\leq 1 & \Rightarrow n \cdot (n-1)! \end{cases} \quad (2)$$

$$f(n) = n! = n \cdot (n-1)! = n \cdot f(n-1) \quad (3)$$

Você já viu um

- A função recursiva mais famosa é o fatorial!
- O que é o fatorial?

$$f(n) = n! \equiv n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \quad (1)$$

$$n! = \begin{cases} n \leq 1 & \Rightarrow 1 \\ n \not\leq 1 & \Rightarrow n \cdot (n-1)! \end{cases} \quad (2)$$

$$f(n) = n! = n \cdot (n-1)! = n \cdot f(n-1) \quad (3)$$

- E como fica um código disso?

Você já viu um

- A função recursiva mais famosa é o fatorial!
- O que é o fatorial?

$$f(n) = n! \equiv n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \quad (1)$$

$$n! = \begin{cases} n \leq 1 & \Rightarrow 1 \\ n \not\leq 1 & \Rightarrow n \cdot (n-1)! \end{cases} \quad (2)$$

$$f(n) = n! = n \cdot (n-1)! = n \cdot f(n-1) \quad (3)$$

- E como fica um código disso?

```
unsigned int fatorial(unsigned int N)
{
    if(N<=1) return 1;
    else return N*fatorial(N-1);
}
```


Vamos testar!