

Projeto e Desenvolvimento de Algoritmos

Procedimentos

Passagem de Parâmetros

Adriano Cruz e Jonas Knopman

cederj

Procedimentos

Passagem de Parâmetros

Objetivos

Definição

Vantagens da utilização de parâmetros

Parâmetros de entrada (Passagem por valor)

Parâmetros de saída (Passagem por referência)

Exemplos

Por que parâmetros?

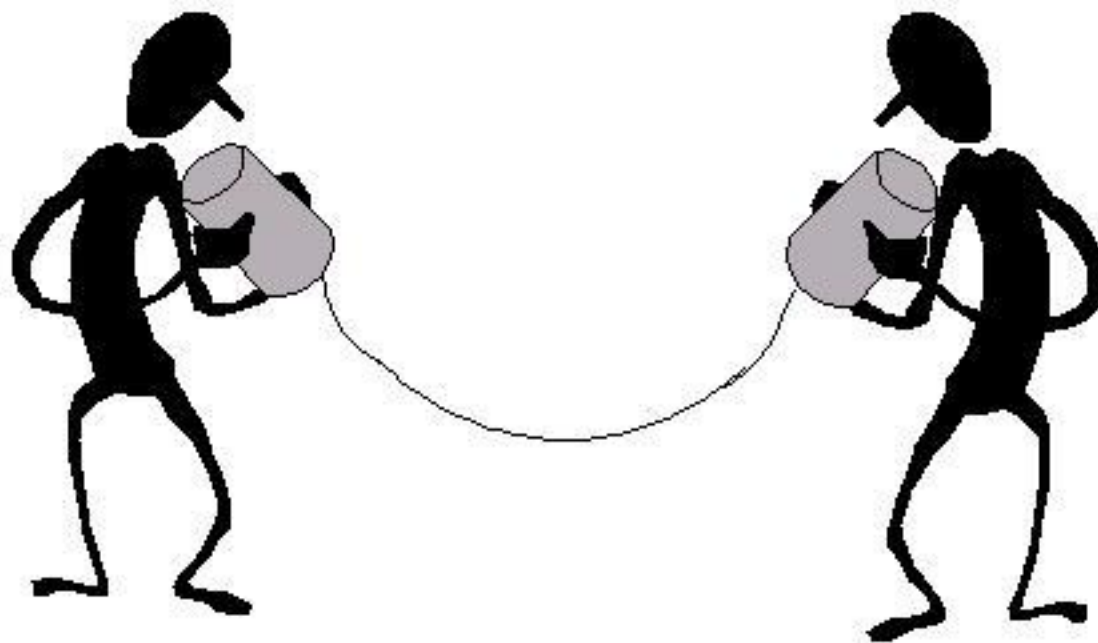
O uso de variáveis públicas expõem perigosamente os dados!



cederj

Parâmetros

*O uso de parâmetros permite a comunicação
privativa entre as partes*



cederj

Parametrização de Procedimentos



X



X



cederj

Entradas e Saídas

*Um procedimento é um bloco de código que opera sobre um conjunto de dados (**suas entradas**) de modo a produzir algum resultado útil para alguém (**suas saídas**)*



Entradas e Saídas

- O conjunto de entradas e saídas define a necessidade de comunicação do procedimento.
- Na seção anterior, usamos este conjunto para definir as variáveis públicas do programa (*solução perigosa!!!*).
- Este mesmo conjunto define os parâmetros do procedimento que têm de ser declarados (*solução melhor!!!*)

Entrada e Saída

- *Os parâmetros de entrada são chamados de parâmetros passados por **valor***
- *Os parâmetros de saída são chamados de parâmetros passados por **referência***

Sintaxe

```
procedimento NONONO (  
    entradas: NONO, NONONO, NONO, ...  
    saídas: NONO, NONONO, NONO, ....)  
início  
    ...  
    ...  
fim
```

Um exemplo

Um procedimento para converter uma quantia em dólares para o equivalente em reais



Procedimento Converte

```
procedimento Converte (  
    entradas: USD  
    saídas: BRL)  
inicio  
    BRL  $\leftarrow$  2.4 * USD  
fim
```


Procedimento Converte

- *A declaração do procedimento e a relação de suas entradas e saídas define o **contrato** de utilização do procedimento*

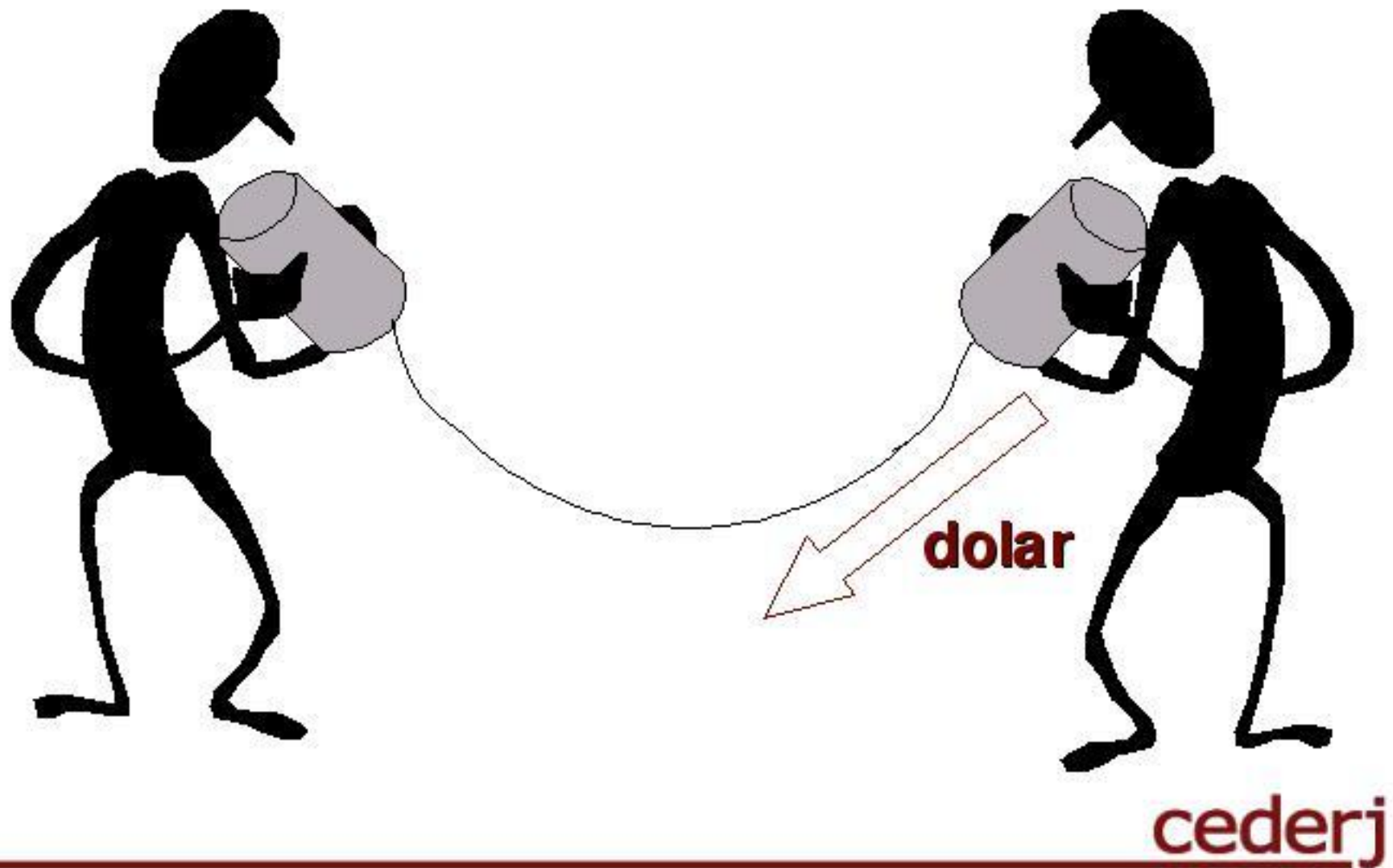
```
procedimento Converte (  
    entradas: USD  
    saídas: BRL)
```

- *Qualquer módulo que queira usar o procedimento Converte, tem de respeitar o **contrato** de utilização.*

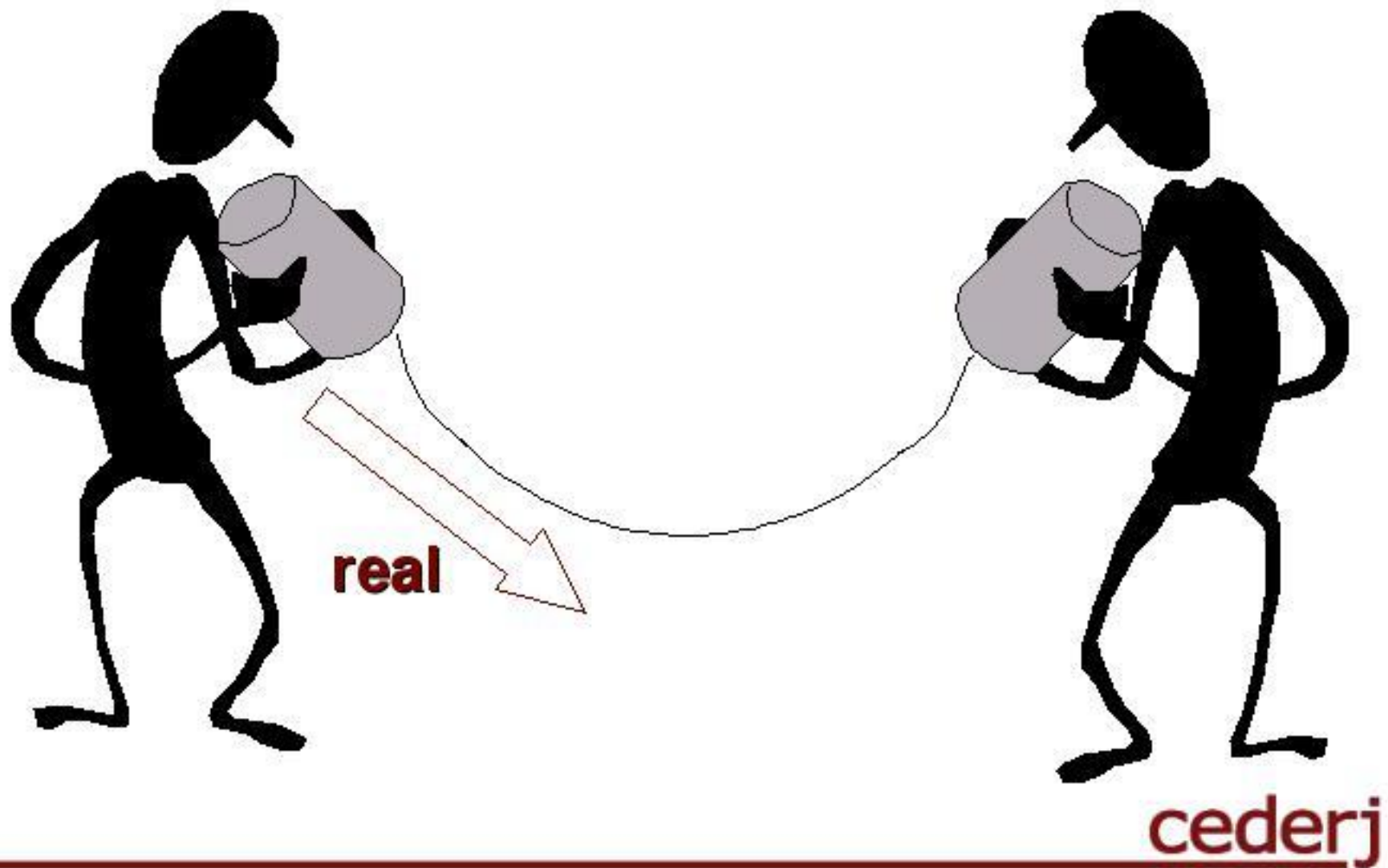
Exemplo de uso do procedimento Converte

```
programa teste
inicio
    acabou ← falso
    enquanto não acabou faça
        leia dolar
        se dolar > 0 então
            Converte (dolar, real)
            imprime real
        senão
            acabou ← verdadeiro
        fim se
    fim enquanto
fim
```

O procedimento Converte



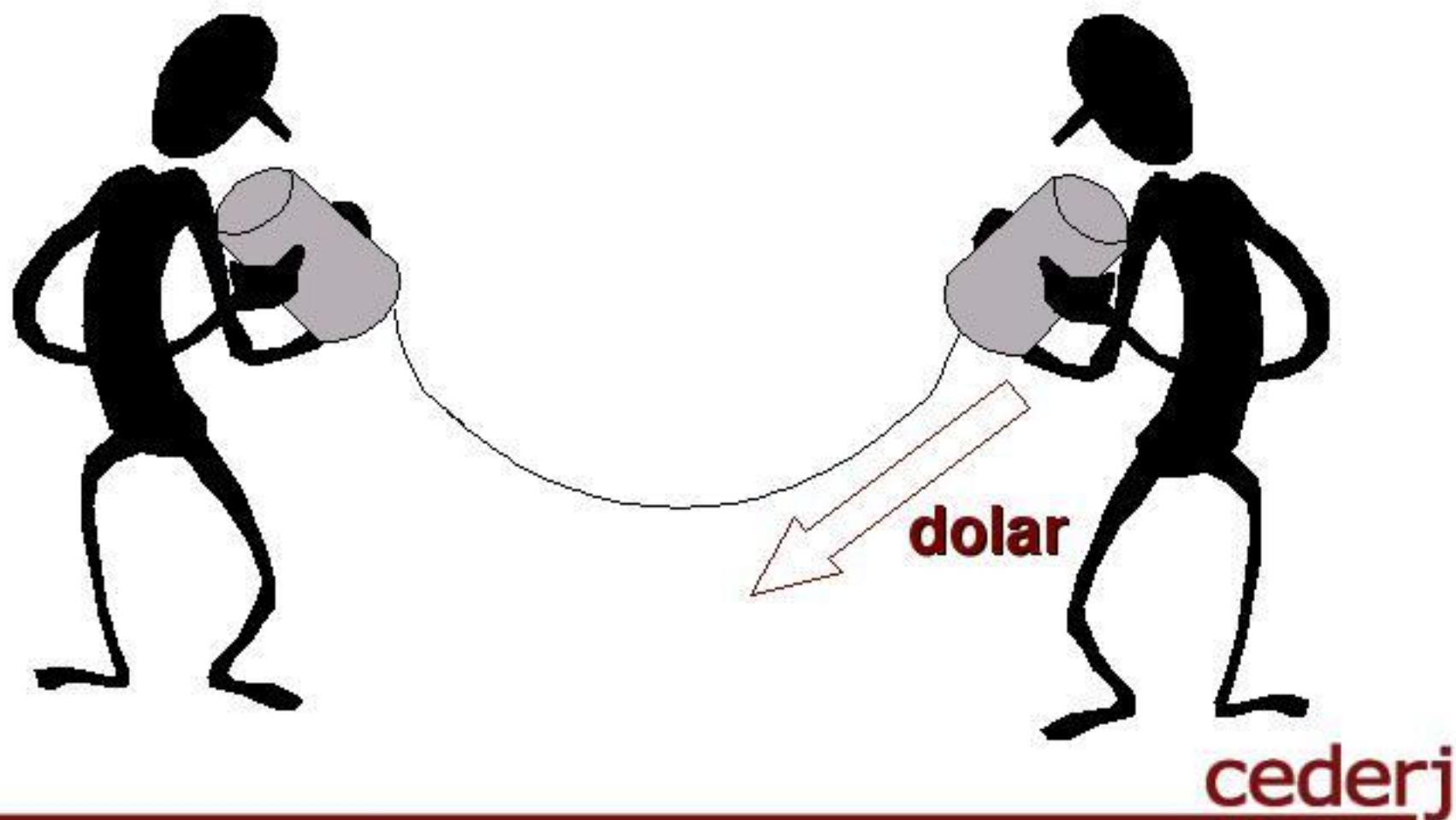
O procedimento Converte



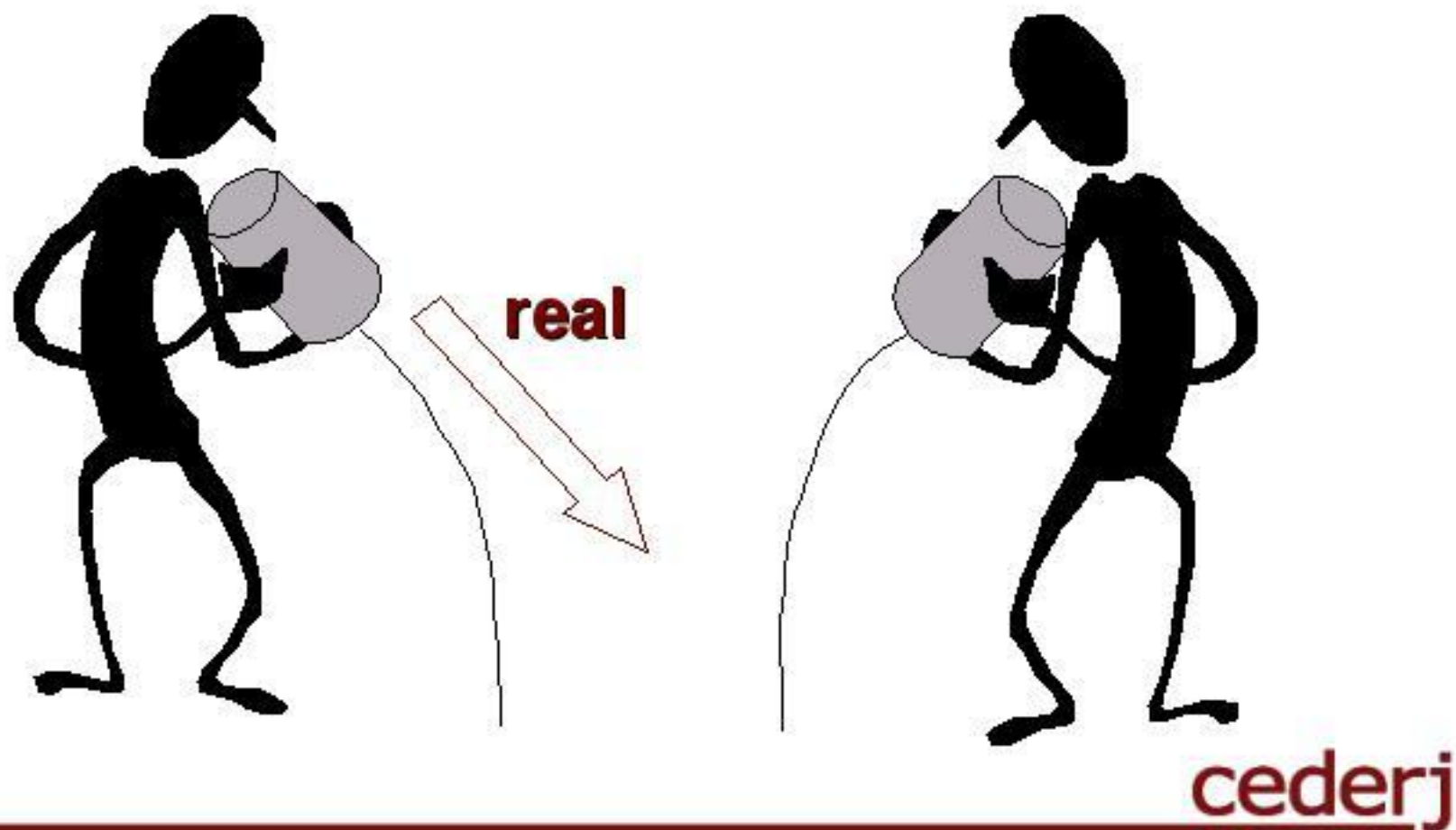
Passagem de Parâmetros

- *O uso de nomes diferentes nas chamadas à procedure e na definição dos procedimentos não apresenta problemas → o número de argumentos deve ser o mesmo.*
- *Os parâmetros são sobrepostos no instante da chamada: o primeiro com o primeiro, o segundo com o segundo, etc.*

O que acontece se eu declarar todos os parâmetros como parâmetros de entrada?



O que acontece se eu declarar todos os parâmetros como parâmetros de entrada?



Um exemplo: O procedimento swap

```
procedimento swap (entradas: x, y)
```

```
inicio
```

```
    imprima 'ponto2> x:',x,' y:',y
```

```
    temp ← x
```

```
    x ← y
```

```
    y ← temp
```

```
    imprima 'ponto3> x:',x,' y:',y
```

```
fim
```


Exemplo de utilização da função swap

```
programa teste
```

```
inicio
```

```
  x ← 5
```

```
  y ← 10
```

```
  imprima 'ponto1> x:',x,' y:',y
```

```
  swap(x,y)
```

```
  imprima 'ponto4> x:',x,' y:',y
```

```
fim
```

Exemplo de utilização da função swap

Saída:

ponto1> x:5 y:10

ponto2> x:5 y:10

ponto3> x:10 y:5

ponto4> x:5 y:10

Por que isto aconteceu???



cederj

Esquema de alocação em memória

Antes de iniciar-se a execução do programa o sistema operacional aloca em memória espaço para as variáveis utilizadas pelo programa principal

?
?

x:teste

y:teste

cederj

Esquema de alocação em memória

Começa o programa principal

$x \leftarrow 5$

$y \leftarrow 10$

10
5

y:teste

x:teste

cederj

Esquema de alocação em memória

chamada do procedimento swap

```
swap (x, y)
```

Esquema de alocação em memória

***Antes de começar a executar o procedimento swap,
o sistema operacional aloca espaço em memória
para os parâmetros e para as variáveis locais***

10
5
?
10
5

y:swap

x:swap

temp

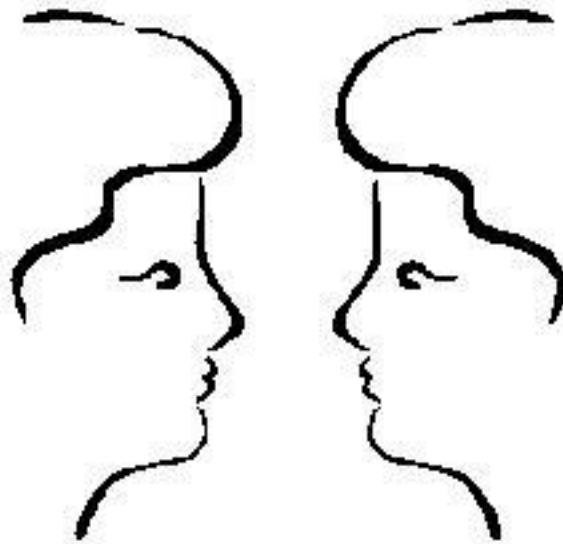
y:teste

x:teste

cederj

Esquema de alocação em memória

Encare a passagem de parâmetros por valor (parâmetros de entrada) como uma cópia xerox do dado original.



Esquema de alocação em memória

começa a execução do procedimento swap

`temp ← x`

10
5
5
10
5

y:swap

x:swap

temp

y:teste

x:teste

cederj

Esquema de alocação em memória

$x \leftarrow y$

10
10
5
10
5

Por isso preciso da variável auxiliar

y:swap

x:swap

temp

y:teste

x:teste

cederj

Esquema de alocação em memória

$y \leftarrow \text{temp}$

5
10
5
10
5

y:swap

x:swap

temp

y:teste

x:teste

Esquema de alocação em memória

termina o procedimento swap

?
?
?
10
5

y:teste

x:teste

As variáveis x e y do programa teste permanecem inalteradas!!!

cederj

Qual é a solução???

Passar o próprio dado e não uma cópia deste.

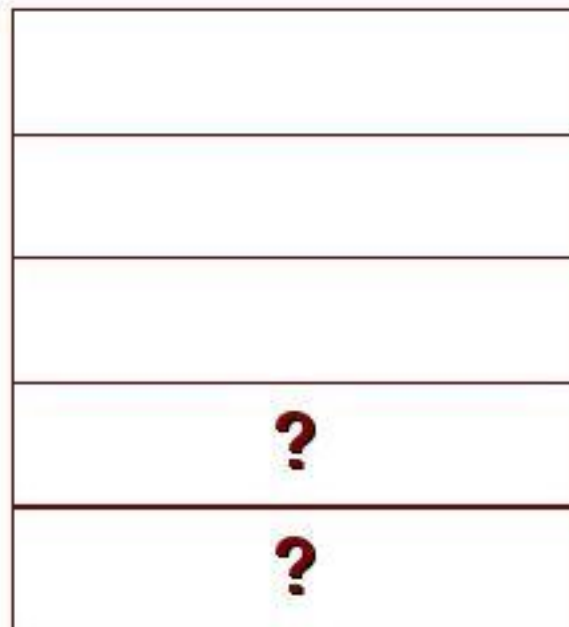


Solução

```
procedimento swap (saidas: x, y)
inicio
    imprima 'ponto2> x:',x,' y:',y
    temp ← x
    x ← y
    y ← temp
    imprima 'ponto3> x:',x,' y:',y
fim
```


Esquema de alocação em memória

Antes de iniciar o programa teste...



x:teste

y:teste

Esquema de alocação em memória

Começa o programa principal

$x \leftarrow 5$

$y \leftarrow 10$

10
5

y:teste

x:teste

cederj

Esquema de alocação em memória

chamada do procedimento swap

```
swap (x, y)
```

Esquema de alocação em memória

***Antes de começar a executar o procedimento swap,
o sistema operacional aloca espaço em memória
para os parâmetros e para as variáveis locais***

	?	temp
y:swap	10	y:teste
x:swap	5	x:teste

Esquema de alocação em memória

Observe como agora o dado é compartilhado...

Ao invés de se entregar ao procedimento uma cópia do dado original, entrega-se o próprio dado

*É verdade, existe um risco no compartilhamento.
Mas, para dados de saída, é o único jeito...*

Esquema de alocação em memória

começa a execução do procedimento swap

temp \leftarrow x

y:swap

x:swap

5
10
5

temp

y:teste

x:teste

cederj

Esquema de alocação em memória

$x \leftarrow y$

y:swap

x:swap

5
10
10

temp

y:teste

x:teste

cederj

Esquema de alocação em memória

$y \leftarrow \text{temp}$

y:swap

x:swap

5
5
10

temp

y:teste

x:teste

Esquema de alocação em memória

termina o procedimento swap

As variáveis x e y do procedimento swap são destruídas, mas o seu conteúdo permanece acessível através das variáveis do programa teste.

?
5
10

y:teste

x:teste

cederj

Valor ou referência???

Ok então. Preciso então usar parâmetros de saída...

Mas e se eu declarar todos os parâmetros como parâmetros de saída? Não funciona? Eu compartilharia os dados referentes a todos os parâmetros e não somente aqueles correspondentes aos parâmetros de saída...



cederj

Valor ou referência???

Existem 2 bons motivos para usar chamadas por valor:

- *Privacidade dos dados*
- *A possibilidade de passar constantes como argumentos de funções*

`x ← raiz(2)`

Outro exemplo: O procedimento CalculaPeso

Escreva um procedimento que calcule o peso ideal de uma pessoa:

homens:

$$\text{peso} = 72.7 * \text{altura} - 58$$

mulheres

$$\text{peso} = 62.1 * \text{altura} - 44.7$$

cederj

Procedimento CalculaPeso

Vamos determinar o conjunto de entradas e saídas do procedimento CalculaPeso



Procedimento CalculaPeso

- **Entradas**

 - sexo*

 - altura*

- **Saída**

 - peso*

Solução utilizando variáveis públicas

Variáveis públicas

sexo, altura, peso

procedimento CalculaPeso

inicio

se sexo = 'M' **entao**

 peso \leftarrow 72.7*altura - 58

senão

 peso \leftarrow 62.1*altura - 44.7

fim se

fim

Contrato

cederj

Um programa que use CalculaPeso

```
programa teste
inicio
    imprima 'Entre com a altura: '
    leia altura
    imprima 'Entre com o sexo: '
    leia sexo
    CalculaPeso
    imprima 'Peso ideal: ', peso
fim
```

Solução utilizando parâmetros

```
procedimento CalculaPeso (  
    entradas: sexo, altura  
    saídas: peso)  
inicio
```

```
    se sexo = 'M' entao
```

```
        peso  $\leftarrow$  72.7*altura - 58
```

```
    senão
```

```
        peso  $\leftarrow$  62.1*altura - 44.7
```

```
    fim se
```

```
fim
```

Contrato

cederj

Um programa que use CalculaPeso

```
programa teste
inicio
    imprima 'Entre com a altura: '
    leia a
    imprima 'Entre com o sexo: '
    leia s
    CalculaPeso (s, a, p)
    imprima 'Peso ideal: ', p
fim
```