

CI1055: Algoritmos e Estruturas de Dados I

Profs. Drs. Marcos Castilho, Bruno Müller Jr, Carmem Hara

Departamento de Informática/UFPR

11 de agosto de 2020

Resumo

Passagem de parâmetros por referência

Objetivos da aula

- Explicar o conceito de passagem de parâmetros por referência
- Caracterizar a diferença da passagem de parâmetro por valor

Parâmetros por referência

- Diferentemente da passagem de parâmetros por valor, não é passado uma cópia do valor da variável do programa que ativou a função
- O que é passado é uma *referência* àquela variável
- Uma referência nada mais é do que o endereço da variável
 - Obs.: ver capítulo sobre o modelo Von Neumann
- A diferença *sintática* é a palavra **var** antes do identificador do parâmetros
- A diferença *semântica* é brutal!

Exemplo

```
1  (*
2   Retorna true se este inteiro n for positivo.
3   Se for o caso, incrementa o valor de n.
4  *)
5  function incrementa_se_positivo (var n: integer): boolean;
6  begin
7     if n > 0 then
8     begin
9         n := n + 1;
10        incrementa_se_positivo := true;
11    end
12    else
13        incrementa_se_positivo := false;
14  end;
```

- O incremento, neste caso, é uma espécie de “efeito colateral” da função
- Passagem de parâmetros por referência normalmente é recomendada para procedures, que veremos mais tarde
- O importante agora é *entender* o conceito

Exemplo completo

```
1 program exemplo_parametro_por_referencia;
2 var a: longint;
3
4 (*
5  Retorna true se este inteiro n for positivo.
6  Se for o caso, incrementa o valor de n.
7 *)
8 function incrementa_se_positivo (var n: longint): boolean;
9 begin
10     if n > 0 then
11         begin
12             n:= n + 1;
13             incrementa_se_positivo:= true
14         end
15     else
16         incrementa_se_positivo:= false;
17 end;
18
19 begin (* programa principal *)
20     read (a);
21     if incrementa_se_positivo (a) then
22         writeln (a,' foi incrementado')
23     else
24         writeln (a,' nao foi incrementado');
25 end.
```

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

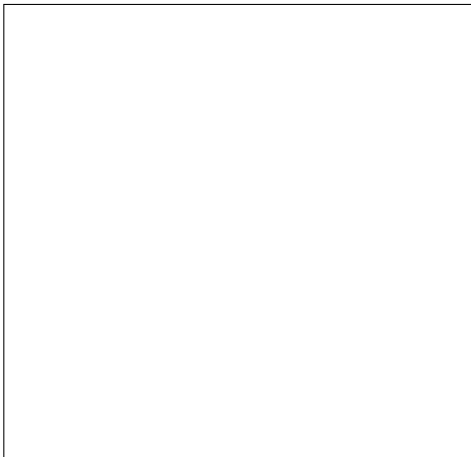
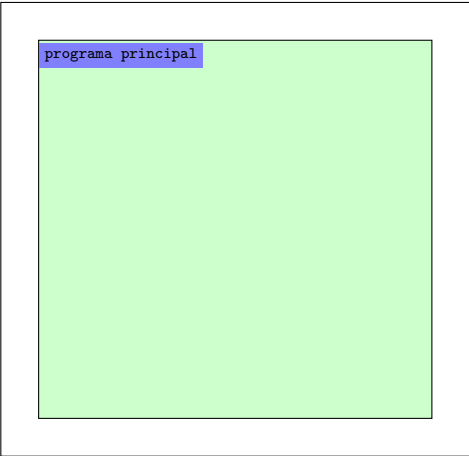


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

Inicia folha de papel do
programa principal



programa principal

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

Cria espaço para as variáveis
do programa principal

programa principal

a: ?

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

programa principal

a: ?

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.  
Usuário digita 2
```

programa principal

a: 2

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.  
1. Desvia fluxo  
2. Cria nova folha  
3. Aloca variáveis  
4. Indica onde retornar ao finalizar
```

programa principal

a: 2

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
  begin  
    n:= n + 1;  
    increm_se_pos:= true  
  end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) ① then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

valor de n "aponta" para variável
que ela representa (ler/escrever)

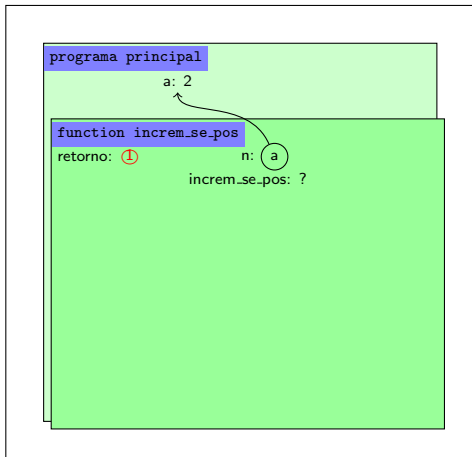


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) ① then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.  
  
if @ > 0 then
```

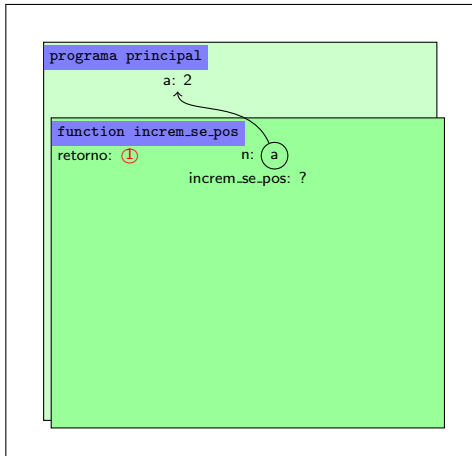


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
  end;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a)① then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.  
  
if ② := ③ + 1
```

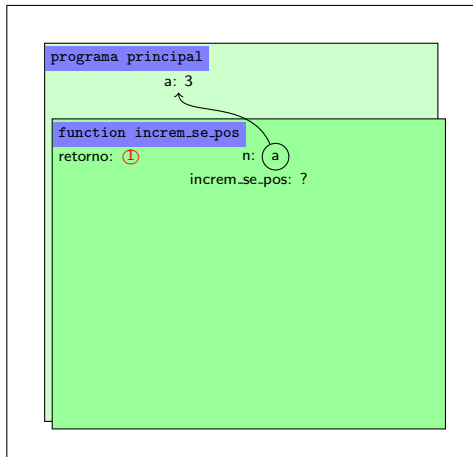


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
  end;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) ① then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

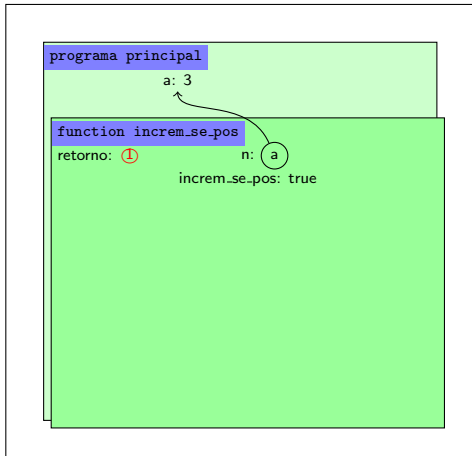


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

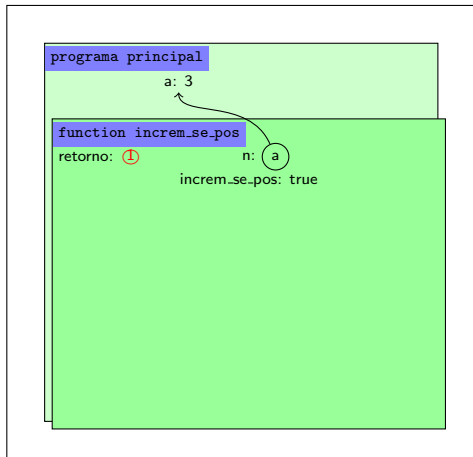


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) ① then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

1. coloca o valor de increm_se_pos no chamador
2. desvia fluxo para o local indicado em retorno
3. retira folha

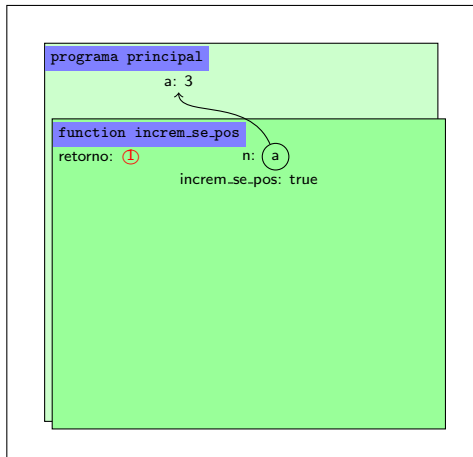


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a)(* true *) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

programa principal

a: 3

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

programa principal

a: 3

- Considere agora a execução resumida, onde somente analisamos o ponto de vista do programa principal.

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

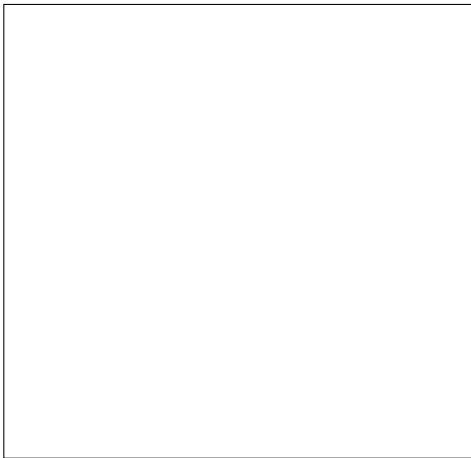
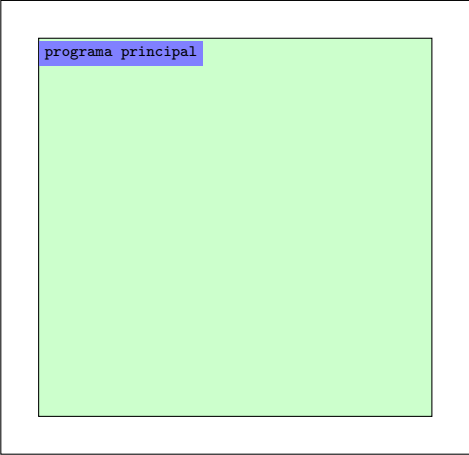


Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

Inicia folha de papel do
programa principal



programa principal

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

Cria espaço para as variáveis
do programa principal

programa principal

a: ?

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

programa principal

a: ?

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.  
Usuário digita 2
```

programa principal

a: 2

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

Considere a execução da function
e o que ocorre após o retorno

programa principal

a: 2

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a)(* true *) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

programa principal

a: 3

Diagrama de Execução

```
program exemplo_parametro_por_referencia;  
var a: longint;  
  
function increm_se_pos (var n: longint): boolean;  
begin  
  if n > 0 then  
    begin  
      n:= n + 1;  
      increm_se_pos:= true  
    end  
  else  
    increm_se_pos:= false;  
end;  
begin (* programa principal *)  
  read (a);  
  if increm_se_pos (a) then  
    writeln (a,' foi incrementado')  
  else  
    writeln (a,' nao foi incrementado');  
end.
```

programa principal

a: 3

Exercício Mental

- O programa abaixo inclui uma modificação no programa principal.
- O que ele imprime para as entradas abaixo:
 - (-1,-1)
 - (3,-3)
 - (-5,5)
 - (7,7)
- E se `n` for passado por valor?

```
program exemplo_parametro_por_referencia;
var a, b: longint;

function increm_se_pos (var n: longint): boolean;
...
begin (* programa principal *)
  read (a, b);
  if increm_se_pos (a) and increm_se_pos (b)
    write('Positivos ');
  writeln(a, b);
end.
```

Exercício Mental

- O programa abaixo inclui uma modificação no programa principal.
- O que ele imprime para as entradas abaixo:
 - (-1,-1)
 - (3,-3)
 - (-5,5)
 - (7,7)
- E se `n` for passado por valor?

```
program exemplo_parametro_por_referencia;
var a, b: longint;

function increm_se_pos (var n: longint): boolean;
...
begin (* programa principal *)
  read (a, b);
  if increm_se_pos (a) and increm_se_pos (b)
    write('Positivos ');
  writeln(a, b);
end.
```

Exercício Mental

- O programa abaixo inclui uma modificação no programa principal.
- O que ele imprime para as entradas abaixo:
 - (-1,-1)
 - (3,-3)
 - (-5,5)
 - (7,7)
- E se `n` for passado por valor?

```
program exemplo_parametro_por_referencia;
var a, b: longint;

function increm_se_pos (var n: longint): boolean;
...
begin (* programa principal *)
  read (a, b);
  if increm_se_pos (a) and increm_se_pos (b)
    write('Positivos ');
  writeln(a, b);
end.
```

Exercício Mental

- O programa abaixo inclui uma modificação no programa principal.
- O que ele imprime para as entradas abaixo:
 - (-1,-1)
 - (3,-3)
 - (-5,5)
 - (7,7)
- E se `n` for passado por valor?

```
program exemplo_parametro_por_referencia;
var a, b: longint;

function increm_se_pos (var n: longint): boolean;
...
begin (* programa principal *)
  read (a, b);
  if increm_se_pos (a) and increm_se_pos (b)
    write('Positivos ');
  writeln(a, b);
end.
```


Exercício Mental

- O programa abaixo inclui uma modificação no programa principal.
- O que ele imprime para as entradas abaixo:
 - (-1,-1)
 - (3,-3)
 - (-5,5)
 - (7,7)
- E se `n` for passado por valor?

```
program exemplo_parametro_por_referencia;
var a, b: longint;

function increm_se_pos (var n: longint): boolean;
...
begin (* programa principal *)
  read (a, b);
  if increm_se_pos (a) and increm_se_pos (b)
    write('Positivos ');
  writeln(a, b);
end.
```

Exercício Mental

- O programa abaixo inclui uma modificação no programa principal.
- O que ele imprime para as entradas abaixo:
 - (-1,-1)
 - (3,-3)
 - (-5,5)
 - (7,7)
- E se `n` for passado por valor?

```
program exemplo_parametro_por_referencia;
var a, b: longint;

function increm_se_pos (var n: longint): boolean;
...
begin (* programa principal *)
  read (a, b);
  if increm_se_pos (a) and increm_se_pos (b)
    write('Positivos ');
  writeln(a, b);
end.
```

- Novamente, o nome do identificador do parâmetro na função não importa
- O que importa é o tipo do parâmetro, que deve ser o mesmo do tipo da variável do programa que chamou a função.

Tipos variados de parâmetros

- É possível definir quantos parâmetros quisermos
- Eles são associados segundo sua posição na chamada
- É possível que alguns sejam por valor e outros por referência
- Neste caso, deve-se separar com um ponto-e-vírgula

Exemplo

```
1 program misturando_tipos_de_parametros;  
2 var a, b: longint;  
3  
4 function esquisita (var n: longint; x: longint): real;  
5 begin  
6     n:= n + 1;  
7     x:= x + 1;  
8     esquisita:= (n + x) / 2;  
9 end;  
10  
11 begin (* programa principal *)  
12     read (a, b);  
13     writeln (esquisita(a, b));  
14     writeln (a, b); (* a teve seu valor alterado, b nao *)  
15 end.
```

- a é associado a n por referência
- b é associado a x por valor

Os cálculos **não** são feitos em cópias!

- Os cálculos são feitos na própria variável do programa que ativou a função
- Por isso deve-se tomar cuidado, pois a função pode alterar o valor da variável do programa que ativou a função
- Isto pode ser perigoso, o programador deve ter muita atenção

Exemplo

```
1  program contando_digitos_e_estragando_a;  
2  var a, cont: longint;  
3  
4  (* funcao que retorna quantos digitos n possui *)  
5  function num_digitos (var n: longint): longint;  
6  begin  
7      cont:= 0;          (* variavel global *)  
8      while n > 0 do  
9          begin  
10         n:= n div 10;  
11         cont:= cont + 1;  
12     end;  
13     num_digitos:= cont;  
14 end;  
15  
16 begin (* programa principal *)  
17     read (a);  
18     writeln (a, ' possui ', num_digitos (a), ' digitos');  
19     writeln (a); (* a teve seu valor alterado *)  
20 end.
```

- Normalmente, não se deve passar parâmetros por referência para funções
- Isto se justifica em casos especiais que ainda veremos
- Mas, se o programador estiver seguro do que está fazendo, então pode usar

- este material está no livro no capítulo 8, seção 8.2.6

- Slides feitos em \LaTeX usando beamer
- Licença

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>