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

Procedimentos

Objetivos da aula

- Explicar o conceito de procedimentos
- Caracterizar a diferença destes com relação às funções

Procedimentos

- Assim como as funções, procedimentos também são subprogramas
- Contrariamente às funções, que podem constituir expressões, os procedimentos são como novos comandos definidos pelo programador
- Os procedimentos, diferentemente das funções, **não** retornam valores ao programa que a ativou
- A passagem de parâmetros é idêntica ao caso das funções

Exemplo

```
1  (* Troca os conteudos de duas variaveis *)
2  procedure troca (var x, y: real);
3  begin
4      temp:= x;    (* temp eh uma variavel global do tipo real *)
5      x:= y;
6      y:= temp;
7  end;
```

- Para haver a troca dos valores é necessário passar os parâmetros por referência, senão a troca seria feita nas cópias, sem efeito no programa que ativou o procedimento
- Por enquanto, usaremos a variável global `temp`, que é necessária para o correto funcionamento do procedimento

Exemplo completo

```
1  program exemplo_procedimento;
2  var a, b, temp: real;
3
4  (*
5     Troca os conteudos de duas variaveis
6  *)
7  procedure troca (var x, y: real);
8  begin
9      temp:= x;    (* temp eh uma variavel global do tipo real *)
10     x:= y;
11     y:= temp;
12 end;
13
14 begin
15     read (a, b);
16     writeln ('a= ',a,' b= ', b);
17     troca (a, b);
18     writeln ('a= ',a,' b= ', b);
19 end.
```

Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

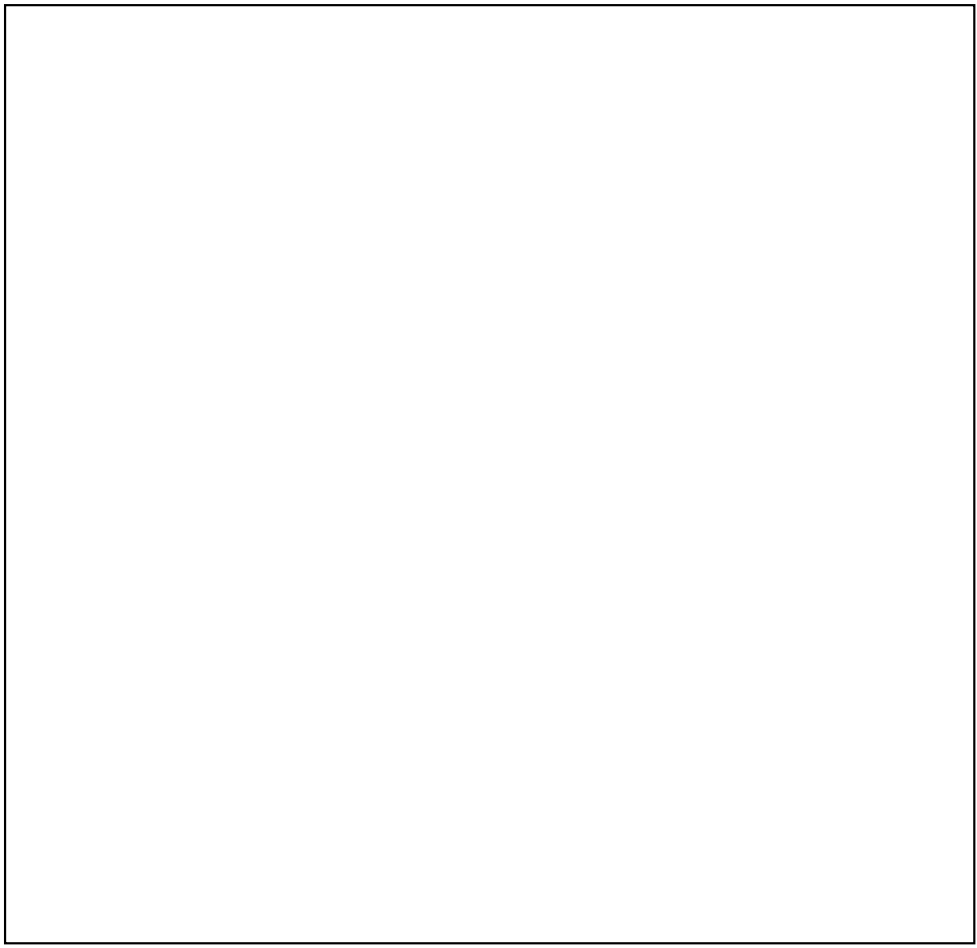


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

Inicia folha de papel do programa principal

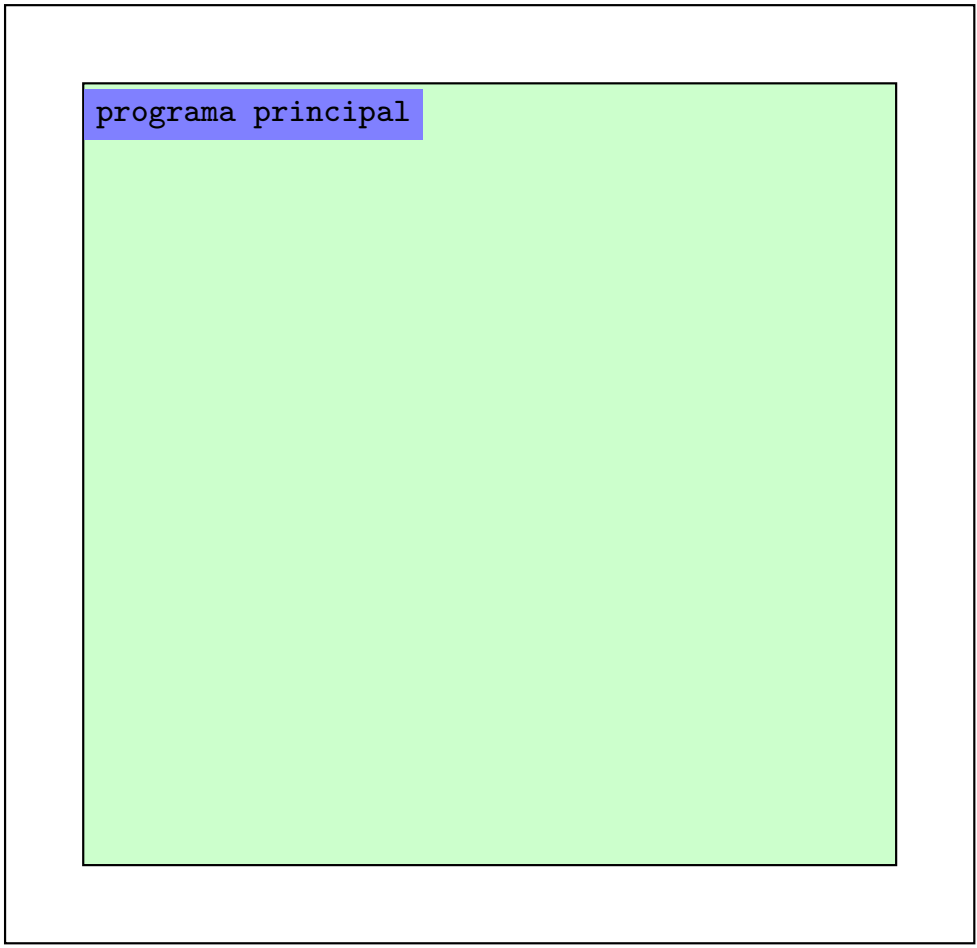


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

Cria espaço para as variáveis
do programa principal

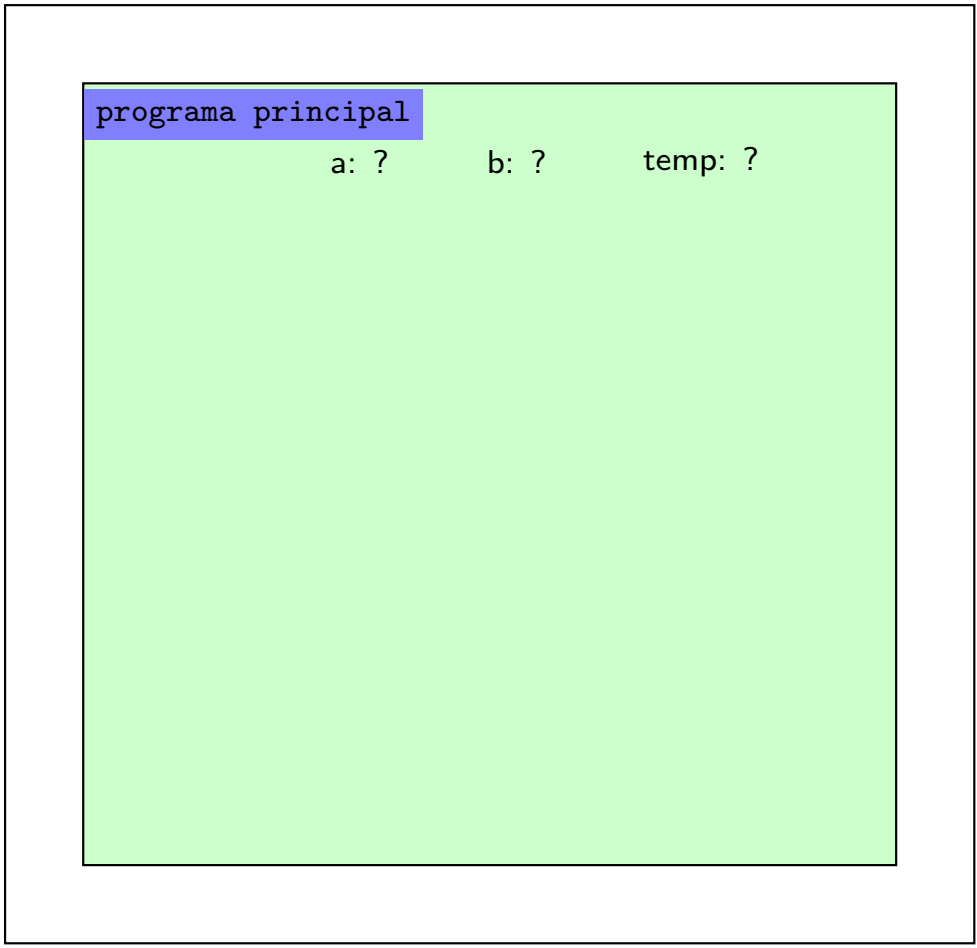


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

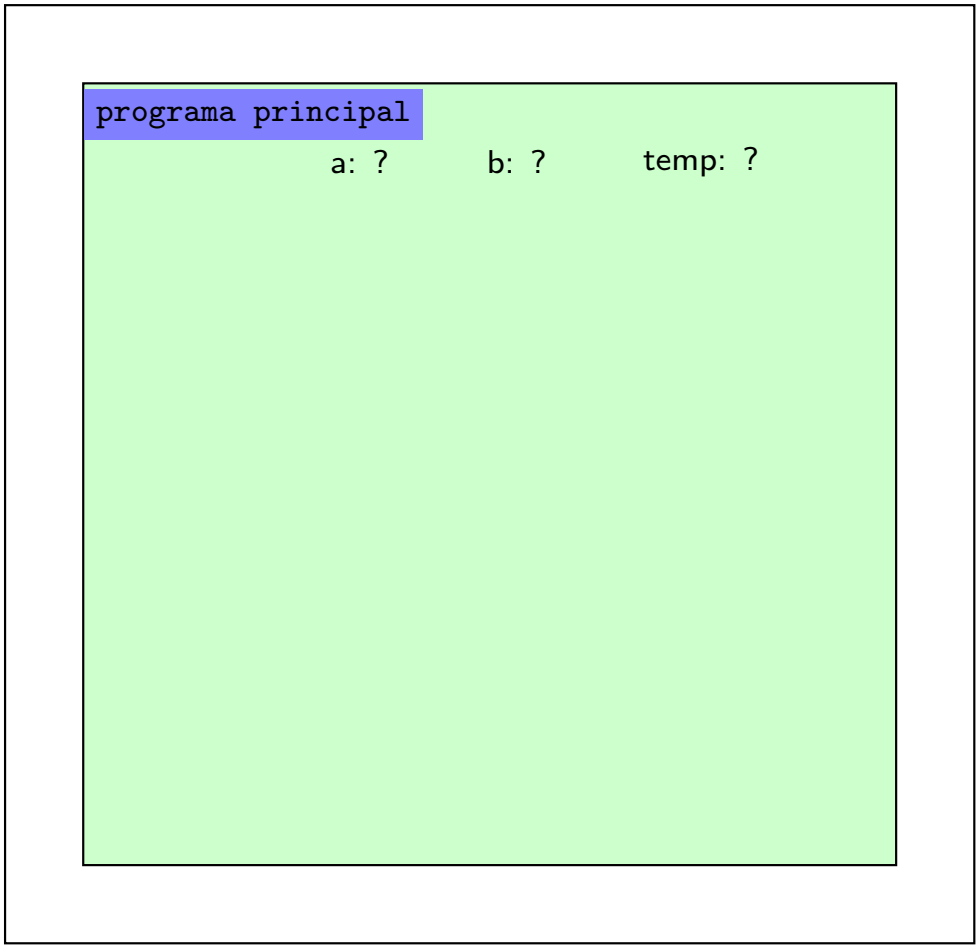


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

usuário digita 2,3

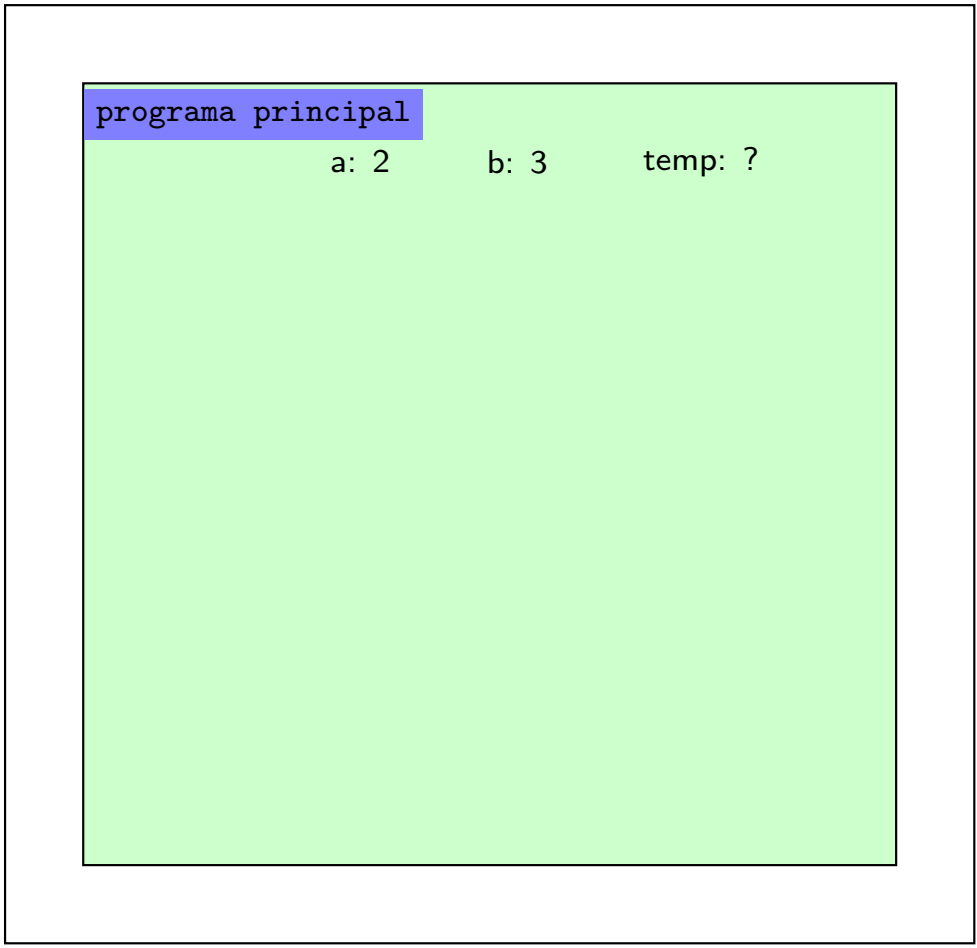


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

```
imprime "a= 2 b= 3 "
```

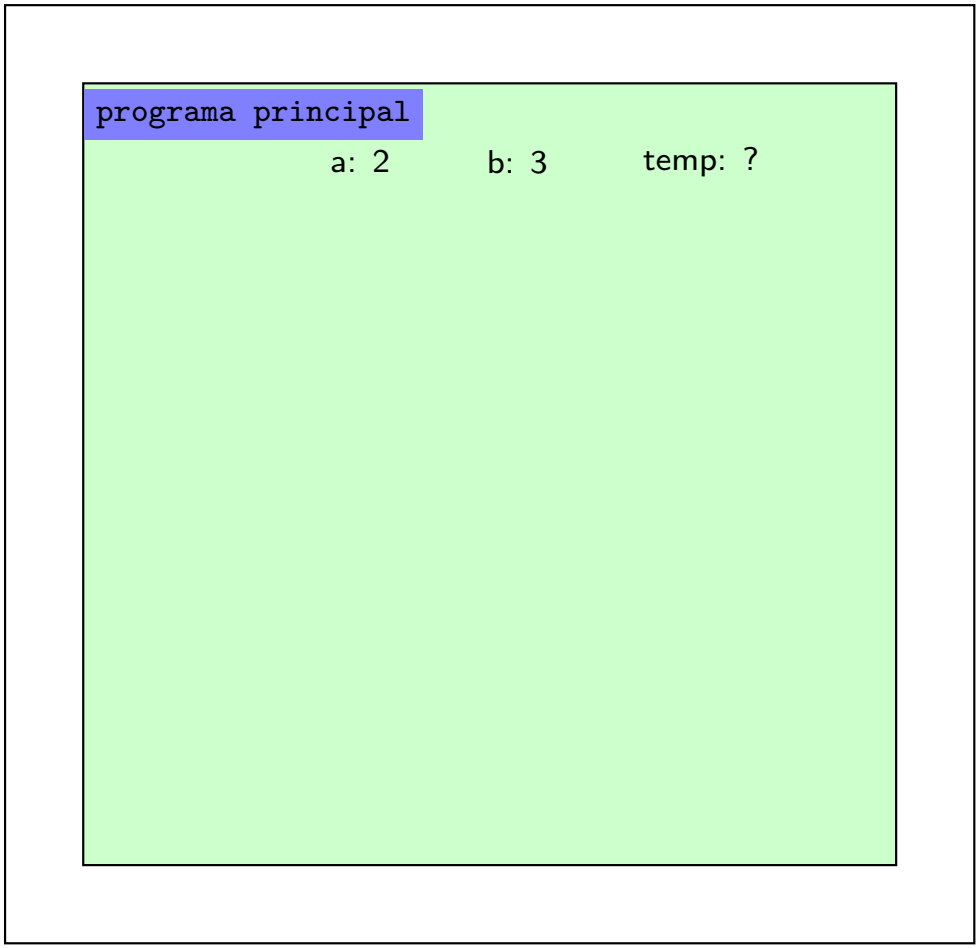


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

1. Desvia fluxo
2. Cria nova folha
3. Aloca variáveis: x, y
4. Indica onde retornar ao finalizar

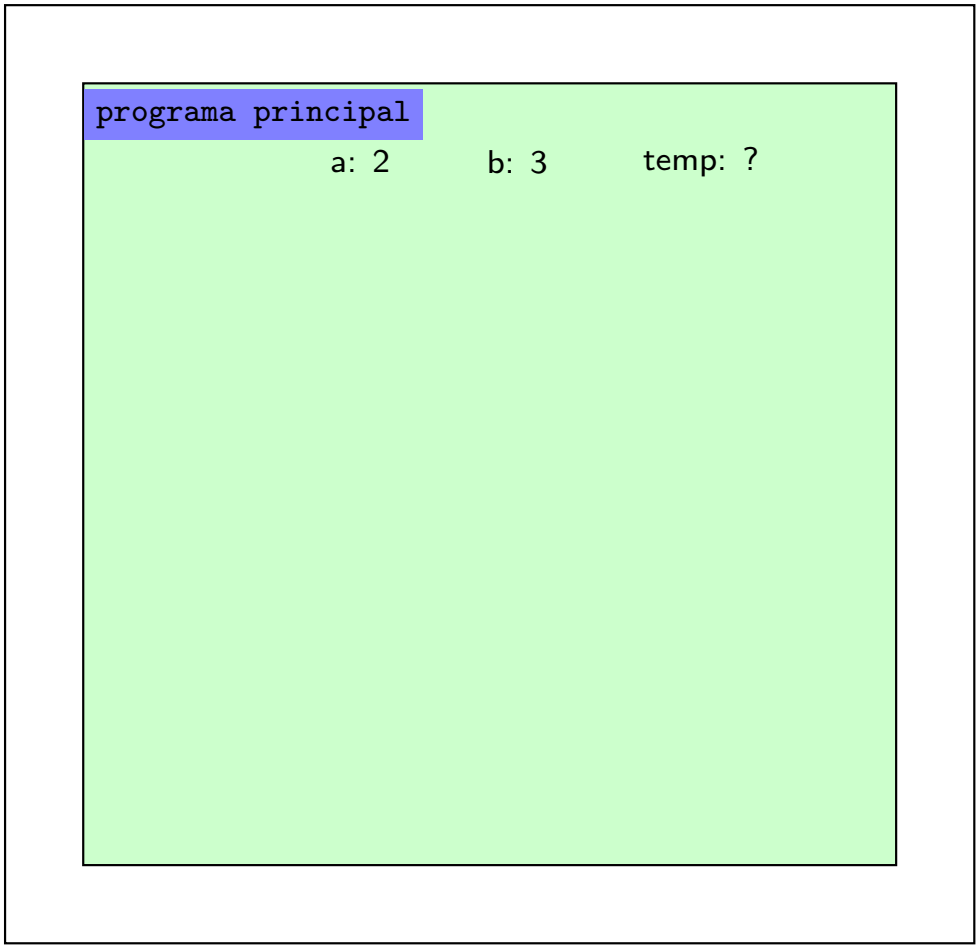


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);①
    writeln ('a= ',a,' b= ', b);
end.
```

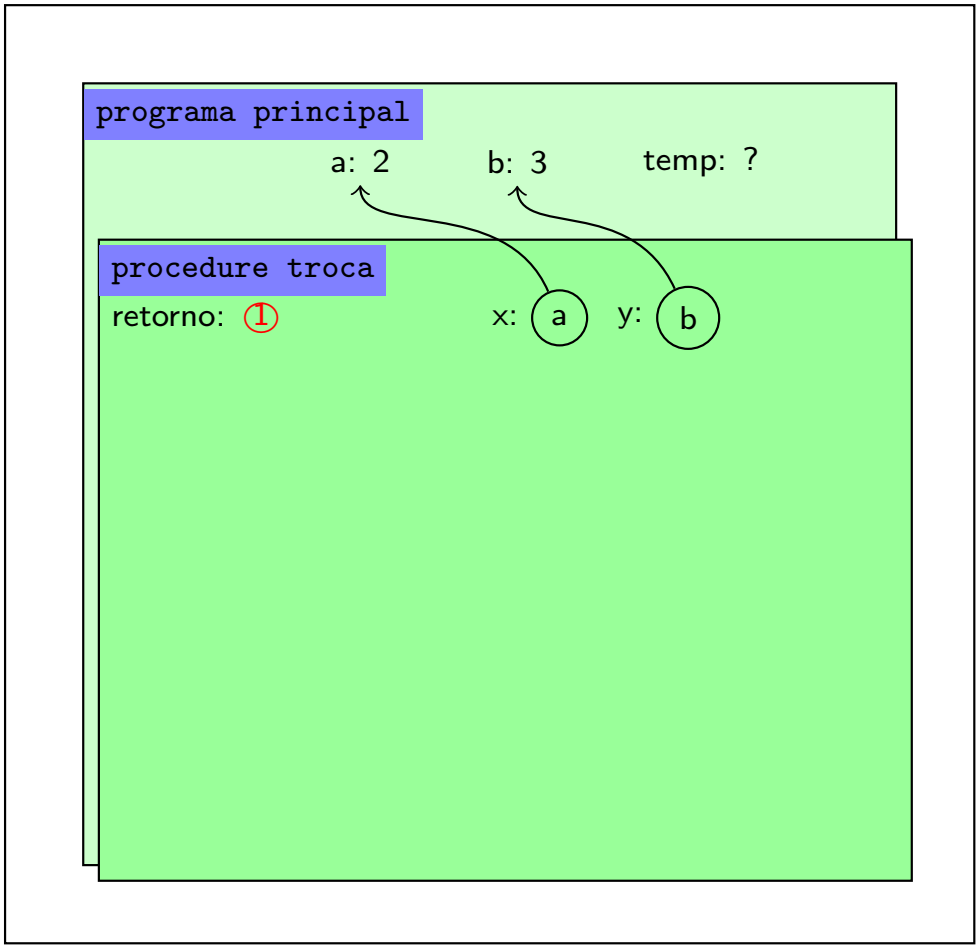


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);①
    writeln ('a= ',a,' b= ', b);
end.
```

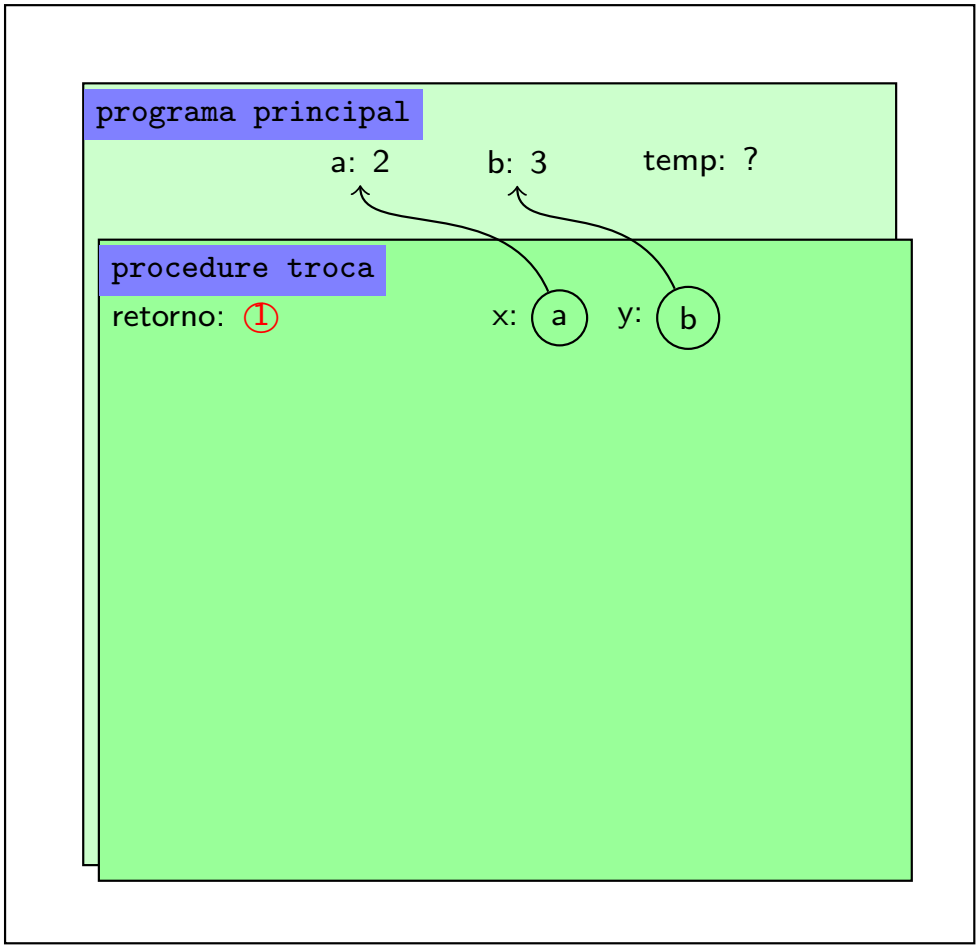


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);①
    writeln ('a= ',a,' b= ', b);
end.
```

temp := ⓐ
Como localizou temp?

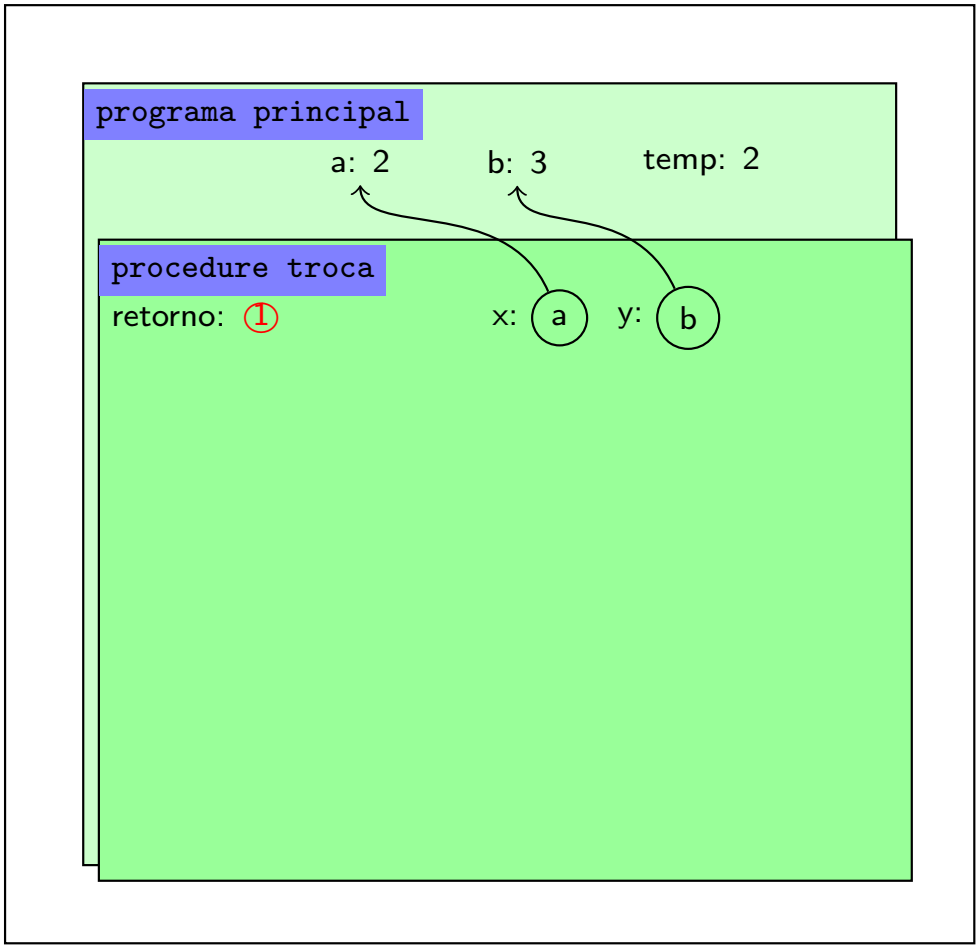


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);①
    writeln ('a= ',a,' b= ', b);
end.
```

ⓐ := ⓑ

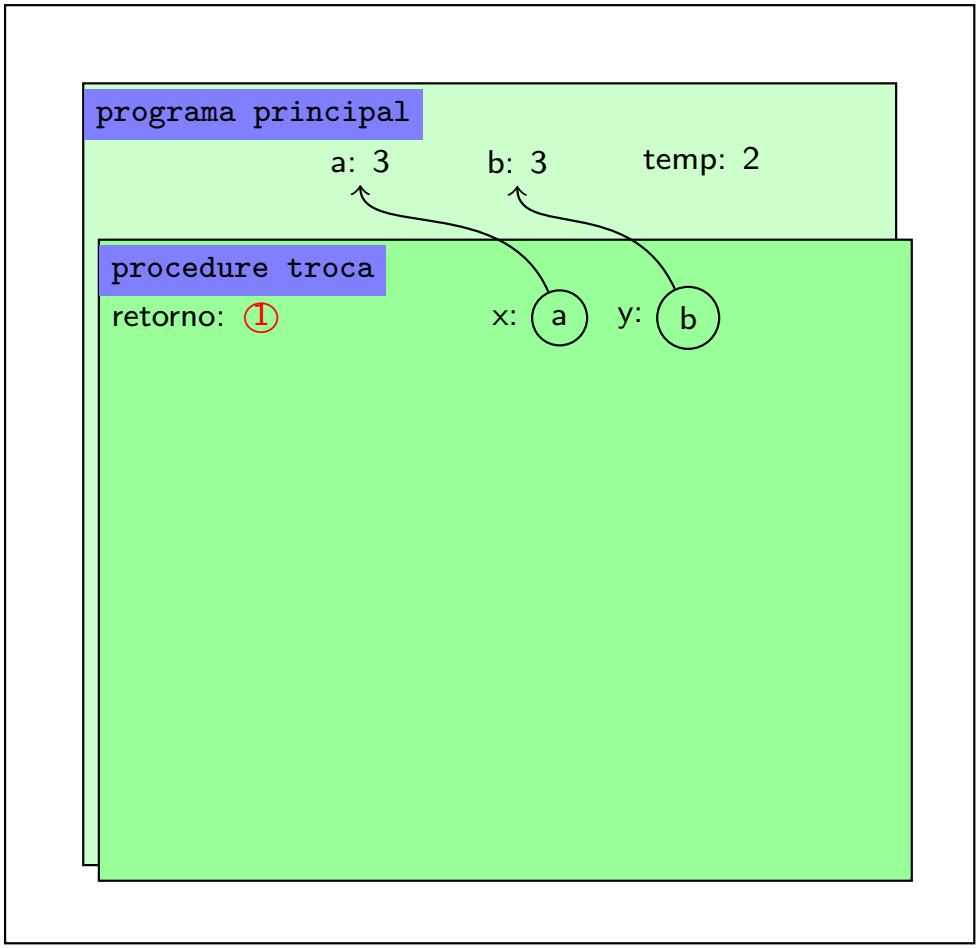


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);①
    writeln ('a= ',a,' b= ', b);
end.
```

ⓑ := temp

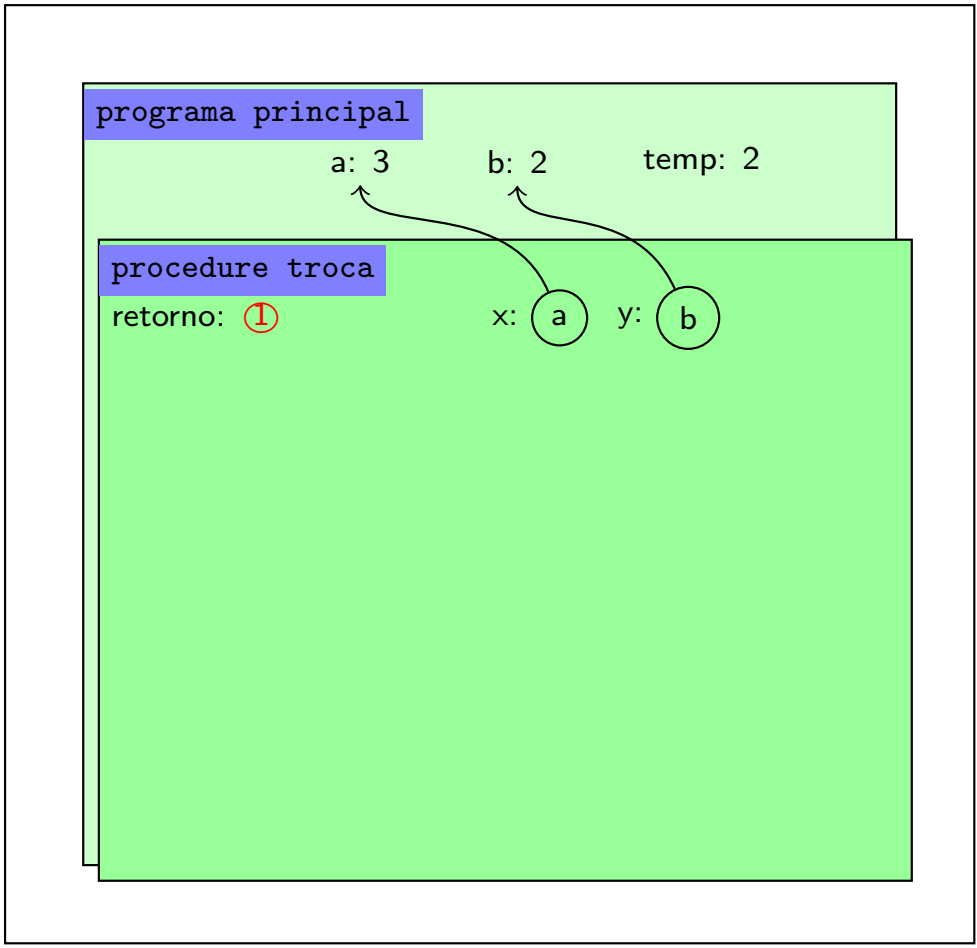


Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);①
    writeln ('a= ',a,' b= ', b);
end.
```

1. desvia fluxo para o local indicado em retorno
2. retira folha

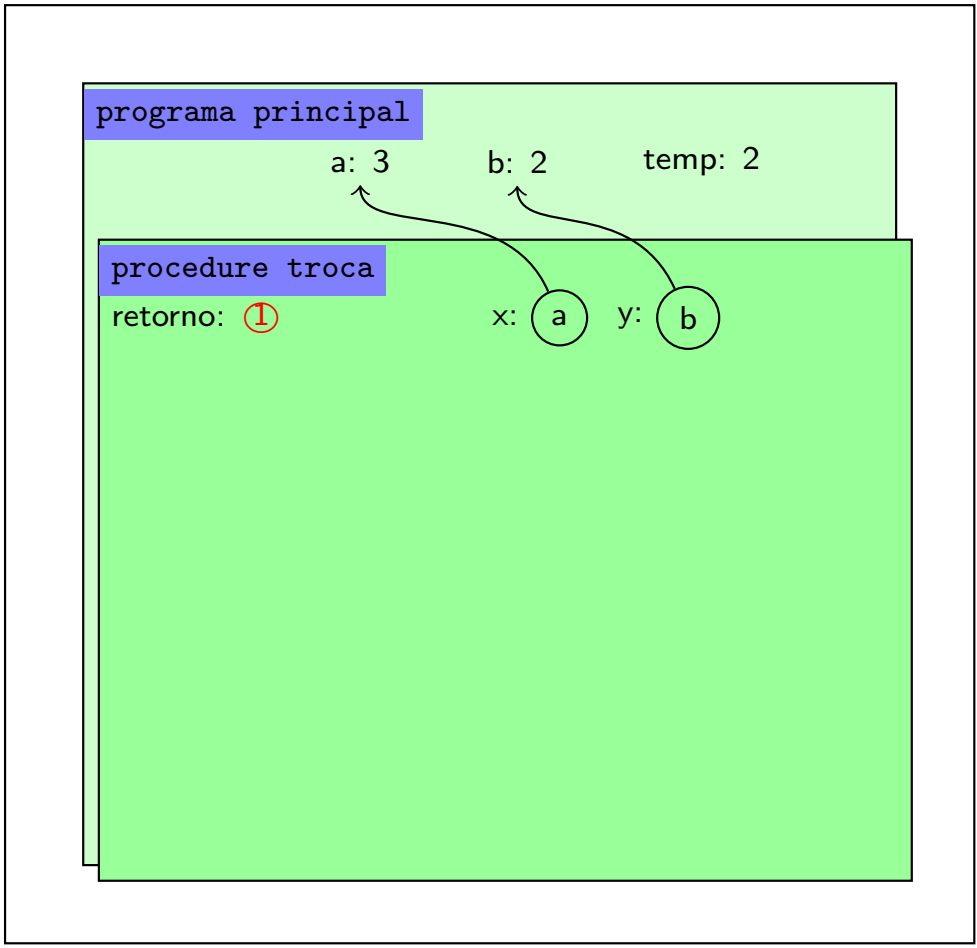


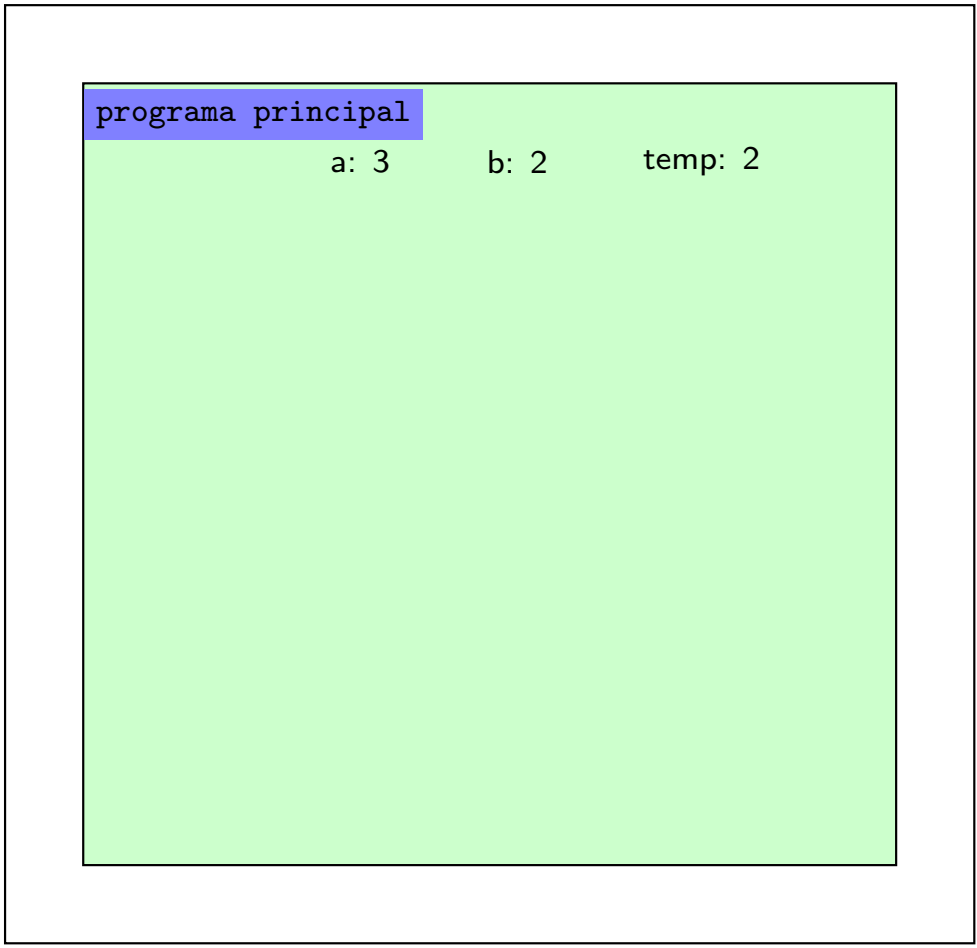
Diagrama de Execução

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

imprime "a= 3 b= 2"



Discussão

```
program exemplo_procedimento;
var a, b, temp: real;

procedure troca (var x, y: real);
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

```
program exemplo_procedimento;
var a, b: real;

procedure troca (var x, y: real);
var temp: real;
begin
    temp:= x;
    x:= y;
    y:= temp;
end;

begin
    read (a, b);
    writeln ('a= ',a,' b= ', b);
    troca (a, b);
    writeln ('a= ',a,' b= ', b);
end.
```

Observações

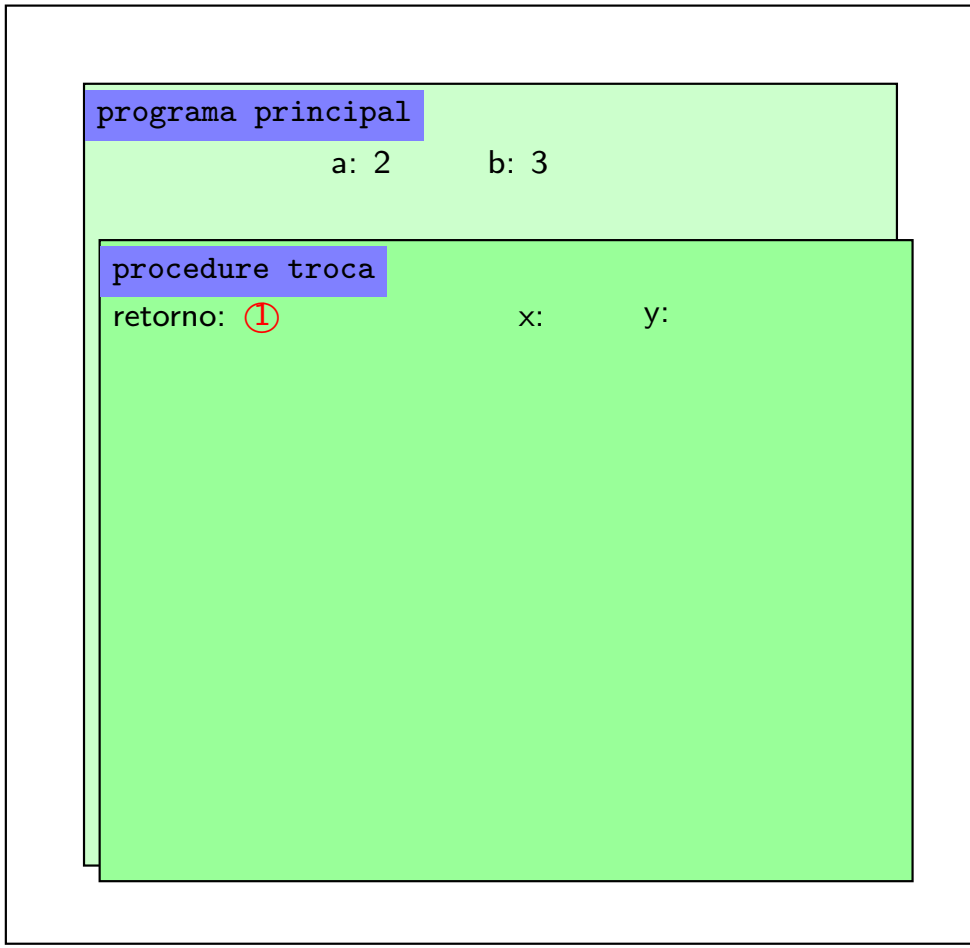
- Conforme explicado, procedimentos são, semanticamente, novos comandos
- Por este motivo, não se pode ativar um procedimento em uma expressão nem se pode imprimir
- Exemplos errados (o compilador acusa erro):
  - `x:= troca (a, b);`
  - `writeln (troca (a, b));`

Experimentos com os parâmetros

- O que aconteceria se o protótipo do procedimento `troca` fosse:
  - `procedure troca (x, y: real);`
  - `procedure troca (var x: real; y: real);`
  - `procedure troca (x: real; var y: real);`

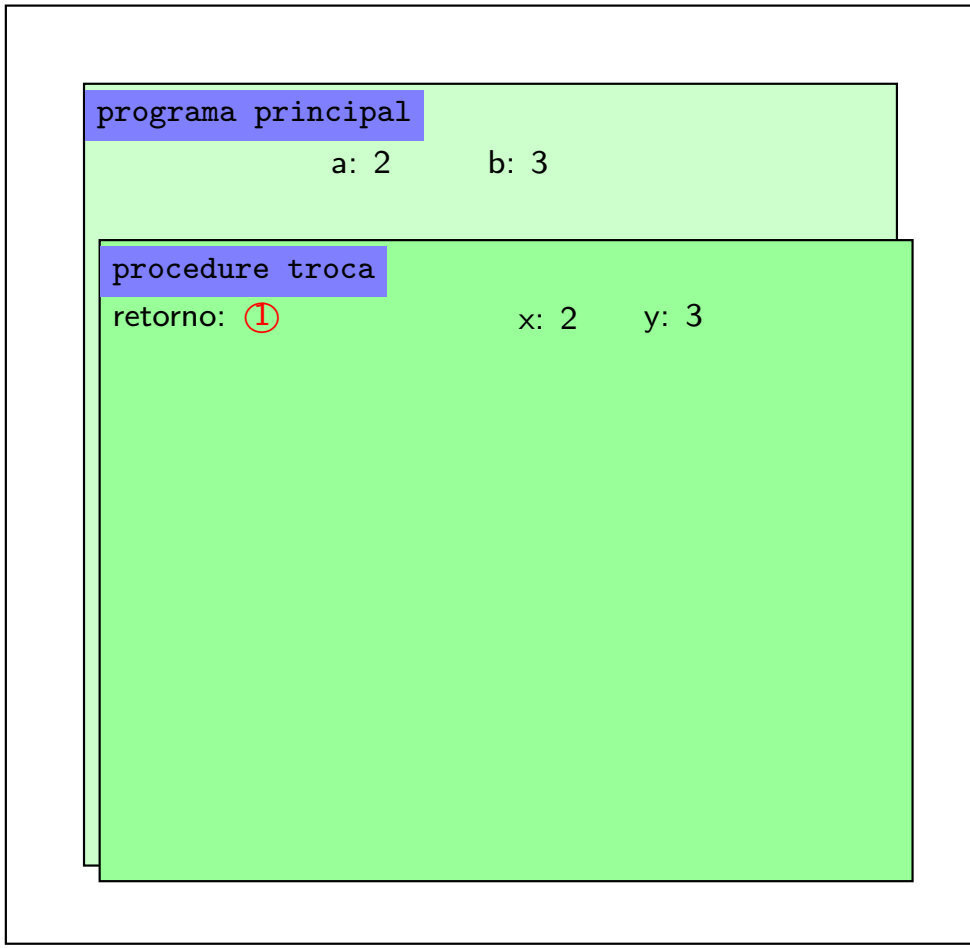
Caso 1

```
procedure troca (x, y: real);
...
a:=2;
b:=3;
troca(a,b);
write (a, b);
```



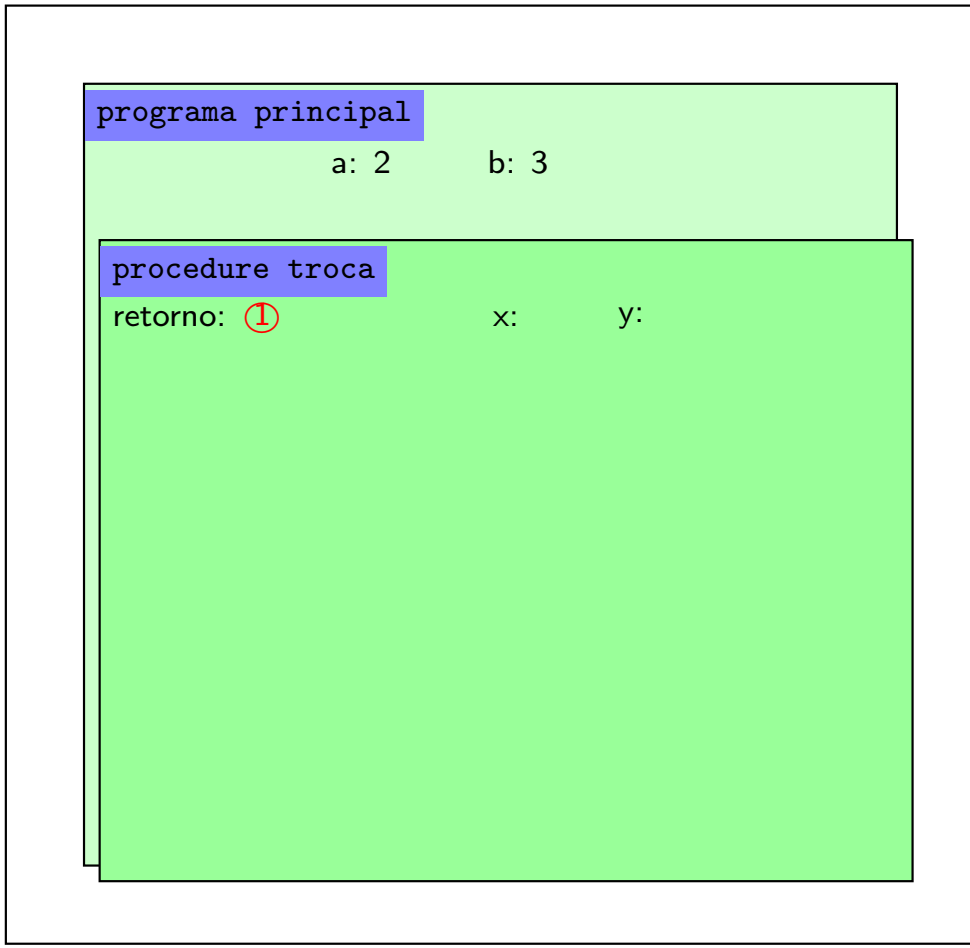
Caso 1

```
procedure troca (x, y: real);
...
a:=2;
b:=3;
troca(a,b);
write (a, b);
(* 2, 3 *)
```



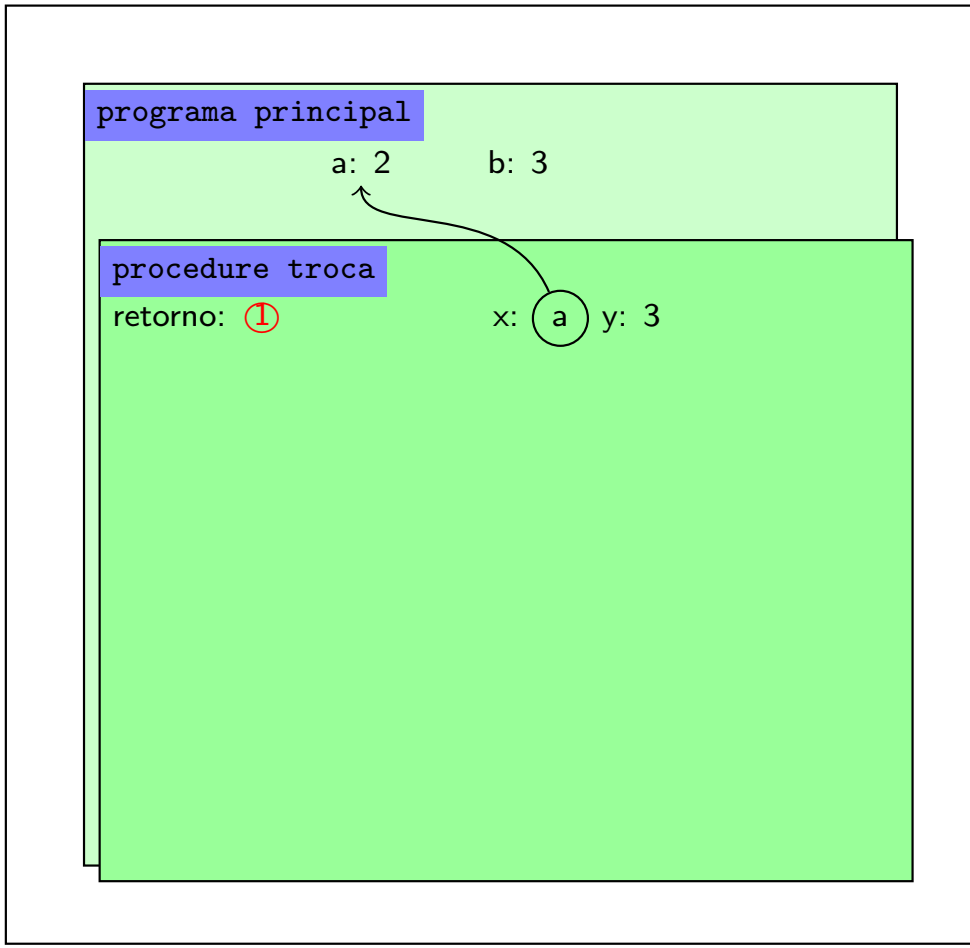
Caso 2

```
procedure troca (var x: real; y: real);
...
a:=2;
b:=3;
troca(a,b);
write (a, b);
```



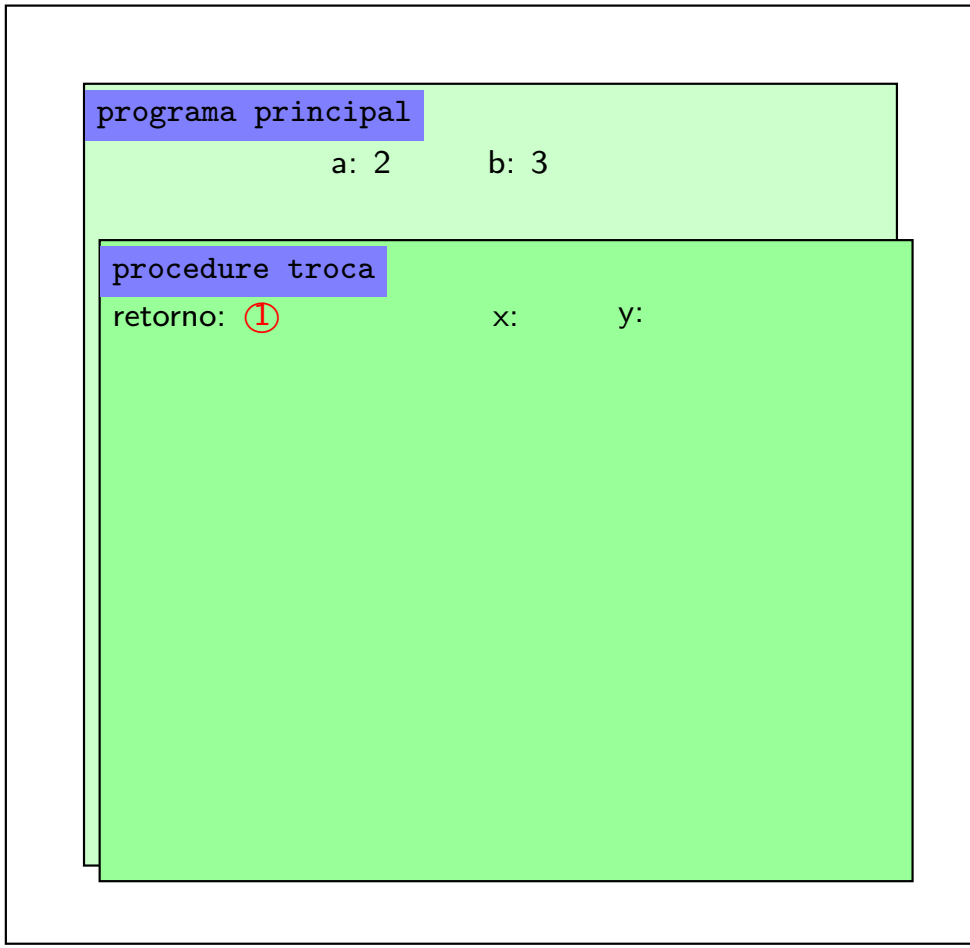
Caso 2

```
procedure troca (var x: real; y: real);
...
a:=2;
b:=3;
troca(a,b);
write (a, b);
(* 3, 3 *)
```



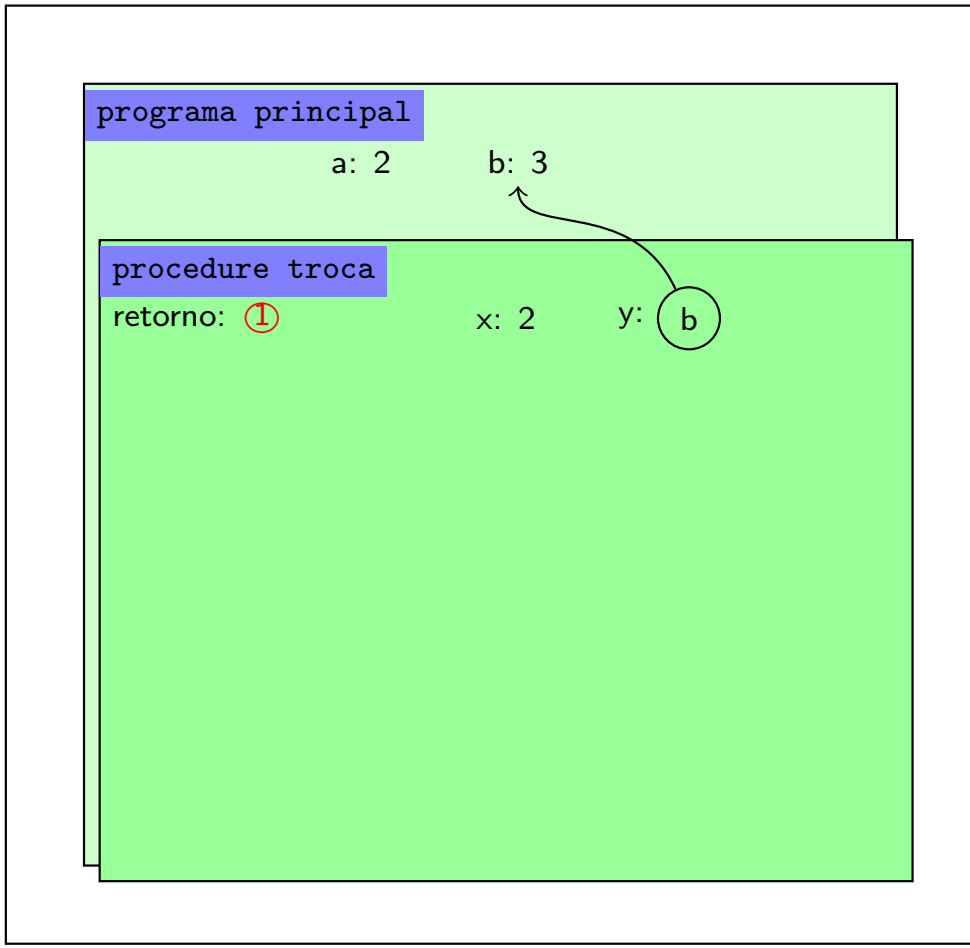
Caso 3

```
procedure troca (x: real; var y: real);!
...
a:=2;
b:=3;
troca(a,b);
write (a, b);
```



Caso 3

```
procedure troca (x: real; var y: real);!
...
a:=2;
b:=3;
troca(a,b);
write (a, b);
(* 2, 2 *)
```



Boas práticas

- Normalmente, as boas práticas para procedures são as mesmas daquelas para funções
- É possível "simular" uma função usando um procedimento com passagem de parâmetros por referência
- Existem algumas situações onde isso é necessário
- Porém, não é muito bonito, em geral

Exemplo

```
1  function eh_par (n: integer): boolean;
2  begin
3      eh_par:= true;
4      if n mod 2 <> 0 then
5          eh_par:= false
6  end;
```

Pode ser simulado assim:

```
1  procedure eh_par (n: integer; var retorno: boolean);
2  begin
3      retorno:= true;
4      if n mod 2 <> 0 then
5          retorno:= false
6  end;
```

O programa principal seria diferente

```
1    read (a);
2    if eh_par (a) then
3        writeln (a,' eh par')
4    else
5        writeln (a,' eh impar')
```

Com procedure, ficaria assim, pois não se pode colocar o nome da procedure no `if`:

```
1    read (a);
2    eh_par (a, retorno); (* retorno: variavel booleana global *)
3    if retorno then
4        writeln (a,' eh par')
5    else
6        writeln (a,' eh impar')
```

Bem feio, não?

Fim do tópico

- este material está no livro no capítulo 8, seção 8.2.7

Licença

- 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/`