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

Variáveis locais e globais

Objetivos da aula

- Explicar o conceito de variáveis locais
- Caracterizar a diferença destes com relação às globais

Motivação

- Vamos retomar o procedimento que troca o valor de duas variáveis
- E também aquela função que conta os dígitos

```
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:= x;
7  end;
```

```
1  function num_digitos (n: longint): longint;
2  begin
3      cont:= 0;          (* variavel global *)
4      while n <> 0 do
5      begin
6          n:= n div 10;
7          cont:= cont + 1;
8      end;
9      num_digitos:= cont;
10 end;
```

Motivação

- Nos dois casos foi necessário usar uma variável global
- Funciona!
- Mas é péssimo do ponto de vista de modularidade

Motivação

- Não se sabe, a princípio, qual outro uso o programador pode estar fazendo destas variáveis
- Em grandes sistemas, se trabalha em equipe, muitas pessoas escrevem partes de códigos que vão constituir o programa completo
- Pode-se estragar o valor de variáveis indevidamente
- Os elementos de um programa podem e devem ser *encapsulados*

Variáveis locais

- Pode-se declarar variáveis localmente nas funções e procedures, assim:

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

```
1  function num_digitos (n: longint): longint;
2  var cont: longint;
3  begin
4      cont:= 0;          (* variavel global *)
5      while n <> 0 do
6      begin
7          n:= n div 10;
8          cont:= cont + 1;
9      end;
10     num_digitos:= cont;
11 end;
```

Programa correto

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

Programa correto

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

Boas práticas

- Todos os elementos que aparecem em um subprograma, seja uma função ou um procedimento, devem estar definidos no próprio subprograma!
- Quando se vê um identificador de variável no corpo do subprograma:
  - Ou ele é um parâmetro
  - Ou ele é uma variável local
- É um grave erro usar variáveis globais!
- É difícil conceber uma situação em que esta regra pode ser violada

Fim do tópico

- este material está no livro no capítulo 8, seção 8.2.8

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/`