

DIFERENCIANDO A PASSAGEM DE PARÂMETROS EM PROCEDIMENTOS NO PORTUGOL

* Ricardo Teixeira Veiga

RESUMO

É proposta uma revisão da notação de procedimentos com parâmetros em Portugol, para representar, de maneira diferenciada, a passagem de parâmetros por valor e por referência.

I- INTRODUÇÃO

O Portugol é uma pseudo-linguagem de programação, baseada em Pascal e ALGOL, ensinada em universidades brasileiras.

Uma pseudo-linguagem adequada deve generalizar os recursos existentes nas linguagens de programação, para servir como veículo apropriado de expressão de algoritmos, facilitando sua posterior codificação nas linguagens de programação objetivo.

Por isso, são representadas no Portugol ferramentas de modularização importantes e encontradas em diversas linguagens - os procedimentos. Inspirado na linguagem Pascal, o Portugol, conforme apresentado por FARRER (1987), divide os procedimentos em duas classes: **sub-rotina** e **função**. Conceitualmente, sub-rotina corresponde à declaração **procedure** da Pascal, enquanto função corresponde à **function**.

As diferenças entre função e sub-rotina são:

- a) A sub-rotina é invocada como um comando isolado e **não** pode ser usada em expressões.
- b) A função é de um tipo definido, que corresponde ao valor resultante da chamada à função. Conseqüentemente, uma função **deve** ser usada em expressões.

(*) Autor

Ricardo Teixeira Veiga

Docente do Centro de Estudos Superiores de Londrina

DIFERENCIANDO A PASSAGEM DE PARÂMETROS EM PROCEDIMENTOS NO PORTUGOL

* Ricardo Teixeira Veiga

RESUMO

É proposta uma revisão da notação de procedimentos com parâmetros em Portugol, para representar, de maneira diferenciada, a passagem de parâmetros por valor e por referência.

I- INTRODUÇÃO

O Portugol é uma pseudo-linguagem de programação, baseada em Pascal e ALGOL, ensinada em universidades brasileiras.

Uma pseudo-linguagem adequada deve generalizar os recursos existentes nas linguagens de programação, para servir como veículo apropriado de expressão de algoritmos, facilitando sua posterior codificação nas linguagens de programação objetivo.

Por isso, são representadas no Portugol ferramentas de modularização importantes e encontradas em diversas linguagens - os procedimentos. Inspirado na linguagem Pascal, o Portugol, conforme apresentado por FARRER (1987), divide os procedimentos em duas classes: **sub-rotina** e **função**. Conceitualmente, sub-rotina corresponde à declaração **procedure** da Pascal, enquanto função corresponde à **function**.

As diferenças entre função e sub-rotina são:

- a) A sub-rotina é invocada como um comando isolado e **não** pode ser usada em expressões.
- b) A função é de um tipo definido, que corresponde ao valor resultante da chamada à função. Conseqüentemente, uma função **deve** ser usada em expressões.

(*) Autor

Ricardo Teixeira Veiga

Docente do Centro de Estudos Superiores de Londrina

Ambas as formas de procedimento podem usar parâmetros e invocarem-se a si mesmas, recursivamente.

II - REPRESENTAÇÃO DE PROCEDIMENTOS EM PORTUGOL

De acordo com FARRER (1987), em Portugol, sub-rotinas e funções são declaradas como a seguir:

Sub-rotina:

sub-rotina NOME (lista-de-parâmetros)

Função:

função tipo NOME (lista-de-parâmetros)

onde:

sub rotina	é uma palavra-chave;
função	é uma palavra-chave;
tipo	é o tipo resultado da função;
lista-de-parâmetros	é a lista de objetos globais que serão usados dentro do procedimento e que têm seus valores passados do algoritmo ou procedimento chamador para esse procedimento, ou vice-versa.

A lista de parâmetros entre parênteses é opcional. Se houver, os parâmetros usados na chamada do procedimento-parâmetros reais devem concordar em número, ordem e tipo com os parâmetros dessa lista, conforme descrita na definição do procedimento.

Vejamos exemplos do uso de procedimentos em Potugol:

(Ex. 1)

Algoritmo Ordena 2 valores ascendentemente

declare X, Y **numérico**

sub-rotina ORDENAR (A,B)

declare A, B, AUX **numérico**

se A B

então AUX ← A

A ← B

B ← AUX

fim se

fim sub rotina

leia X, Y

ORDENAR (X,Y)

escreva X, Y Os valores são escritos em ordem

fim algoritmo.

(Ex. 2)

Algoritmo Acha o maior entre 2 valores

declare X, Y, MAIOR **numérico**

função numérico MAX (A, B)

declare A, B, **numérico**

se A > B

então MAX ← A

senão MAX ← B

fim se

fim função

leia X, Y

MAIOR ← MAX (X, Y)

escreva MAIOR É escrito o maior valor

fim algoritmo.

III - MECANISMO DE PASSAGEM DE PARÂMETROS NO PORTUGOL

Nas linguagens de programação de alto nível usamos basicamente dois mecanismos de passagem de parâmetros para procedimentos. A passagem de parâmetros por **valor** e por **referência**.

Na passagem de **parâmetros por valor**, o parâmetro real atua como parâmetro de entrada, no sentido de fornecer um valor a ser usado procedimento. Entretanto, valores resultantes **não são transferidos** de volta às variáveis usadas como parâmetros reais, isto é, os parâmetros reais **não** podem atuar como parâmetros de saída.

Vejamos um exemplo em Pascal, obtido de WELSH & ELDER (1989):

(EX. 3)

```
procedure EscreverEmMetros (Pes, Polegadas: integer);  
var Metros: real; ComprimentoEmPolegadas: integer;  
begin
```

```
    ComprimentoEmPolegadas: = 12 * Pes + Polegadas;  
    Metros: = ComprimentoEmPolegadas/39.39;  
    write (Metros 6:2)  
end;
```

O procedimento (3) aceita dois valores inteiros, que representam uma distância em pés e polegadas, convertendo a distância para metros. Esses são parâmetros de valor. Exemplo de chamadas a esse procedimento seriam:

```
    EscreverEmMetros (10,5);  
    EscreverEmMetros (Pes, Pols);  
    EscreverEmMetros (Pes+5,0);
```

Na Pascal, parâmetros de valor são transferidos aos parâmetros formais do procedimento e podem inclusive atuar como variáveis locais. Por exemplo, vejamos um outro procedimento obtido de WELSH & ELDER (1989), que elimina a variável ComprimentoEmPolegadas de procedure (3).

(Ex. 4)

```
procedure EscreverEmMetros (Pes, Polegadas: integer);  
var Metros: real;  
begin
```

```
    Polegadas: = 12 * Pes + Polegadas;  
    Metros: = Polegadas/39.39;  
    write (metros:6:2)  
  
    end;
```

Em particular, uma chamada da forma:

```
    EscreverEmMetros (Pes, POLS);
```

não afeta o conteúdo da variável POLS, que deu o valor de entrada de polegadas, pois durante a execução do procedimento o conteúdo do parâmetro real é **copiado** para o parâmetro formal. Este sim pode ser alterado, funcionando como uma variável local.

O outro mecanismo de passagem de parâmetro é a passagem **por referên-**

cia. Em Pascal, conhecido como o uso de parâmetros variáveis. Nesse caso, os parâmetros reais substituem os parâmetros formais diretamente, pois na passagem de parâmetros, são transferidos os endereços dos parâmetros reais para o procedimento. Instruções que alterem o conteúdo dos parâmetros formais atuam diretamente nos parâmetros reais, alterando seu conteúdo.

WELSH & ELDER (1989) apresentam o seguinte exemplo de procedure Pascal em que há passagem de parâmetros por referência:

(Ex. 5)

```
procedure ConverterParaMetros (Pes, Polegadas: integer;  
                               var Metros: real);  
  
begin  
  Metros := (12 * Pes + Polegadas) / 39.39;  
end;
```

Em Pascal, parâmetros variáveis são procedidos pela palavra reservada **var**. Nesse exemplo, a variável Metros é um parâmetro variável.

Parâmetros variáveis podem atuar como parâmetros de entrada/saída. Já os parâmetros de valor só podem atuar como parâmetros de entrada e, eventualmente, como variáveis locais aos procedimentos. E no Portugol, como funciona a passagem de parâmetros?

FARRER (1987) esclarece no trecho seguinte:

“No momento da chamada do procedimento, os valores dos parâmetros de chamada são transferidos para os parâmetros de definição correspondente e, após a sua execução, os valores dos parâmetros de definição são transferidos, de volta, para os parâmetros de chamada.” (pág. 227)

Podemos perceber que o Portugol define um mecanismo de passagem de parâmetros **semelhante** à passagem de parâmetros por referência, já que mudanças nos parâmetros formais são repassadas aos parâmetros reais, caso esses parâmetros sejam variáveis.

O procedimento Pascal (4) seria expresso assim em Portugol:

(Ex. 6)

```
sub rotina EscreverEmMetros (Pes, Polegadas);  
  declare Pes, Polegadas, Metros numérico  
  Polegadas    12 x Pes = Polegadas  
  Metros       Polegadas/39.39  
  escreva Metros
```

fim sub rotina

Deveríamos atentar para o fato de que essa notação indica que a chamada

EscreverEmMetros(Pes, Polegadas)

afetaria o conteúdo da variável **Polegadas**, pois uma das instruções da sub-rotina altera essa variável.

Logo, a sub-rotina **Portugol (6)** acima equivale à seguinte **procedure Pascal**:

(Ex. 7)

```
procedure EscreverEmMetros (Pes: integer;  
                             var Polegadas: integer);  
var Metros: real;  
begin  
    Polegadas := 12 * Pes + Polegadas;  
    Metros := Polegadas/39.39;  
    writeln (Metros)  
end;
```

e não é equivalente à **procedure Pascal (4)** como se poderia pensar.

O mecanismo de passagem de parâmetros no **Portugol** não é exatamente a passagem por referência, pois permite o uso de constante **Portugol ABS (x)**, que retorna o valor absoluto de uma expressão aritmética, pode ser usada em situações como nos trechos de algoritmos seguintes:

(Ex. 8)

```
.  
.   
.   
x      -3  
y      ABS(x)  
.   
.   
.
```

ou

(Ex. 9)

y ABS (-3)

Esse mecanismo sui generis de passagem de parâmetros do Portugol dá-lhe suficiente flexibilidade para a representação geral de procedimentos com parâmetros. Entretanto, obscurece a diferença entre as duas formas de passagem de parâmetros (por valor e por referência), dificultando a codificação imediata dos algoritmos feitos em Portugol nas linguagens de programação. Por exemplo, a codificação em Pascal.

IV - PROPOSTA DE MUDANÇA

Se pretendemos utilizar o Portugol sistematicamente como pseudo-linguagem de programação, julgamos importante redefinir o mecanismo de passagem de parâmetros para procedimentos, diferenciando passagem por valor e por referência. Trata-se de assimilar no Portugol os mecanismos definidos na Pascal.

Como mudança de notação, sugerimos assinalar os parâmetros variáveis do Portugol com um asterisco. Assim, os parâmetros **não** assinalados seriam parâmetros de valor.

A procedure Pascal (5) passaria a ser escrita em Portugol como a seguir:

(Ex. 10)

```
sub rotina ConverterParaMetros (Pes,Polegadas,Metros*)
  declare Pes,Polegadas,Metros numérico
  Metros      (12 x Pes + Polegadas) / 39.39
fim sub rotina
```

V - CONCLUSÃO

A proposta de adotar no Portugol mecanismos de passagem de parâmetros por valor e por referência, de maneira diferenciada, como é na Pascal, visa eliminar a ambigüidade existente no mecanismo de passagem de parâmetros do Portugol, conforme apresentado por FARRER(1987). Uma vantagem adicional é aproximar o Portugol das linguagens de programação usuais, que adotam de maneira diferenciada um ou ambos mecanismos de passagens de parâmetros.

A conseqüente necessidade de mudança de notação para distinguir parâmetros variáveis de parâmetros de valor é resolvida de maneira simples. Basta assinalar os parâmetros variáveis com um asterisco, no cabeçalho do procedimento.

VI - REFERÊNCIAS BIBLIOGRÁFICAS

FARRER, H. et alii. Algoritmos Estruturados. Ed. Guanabara Dois. Rio de Janeiro, 1987. Pg. 65; 220-229.

WELSH, J. & ELDER, J. Introdução à Linguagem Pascal. 2 ed. Ed. Prentice/Hall do Brasil. Rio de Janeiro, 1989, Pg. 82-89.