

Aula 04

Introdução ao Português Estruturado

A linguagem que o Portugol (Português Estruturado) interpreta é bem simples: é uma versão portuguesa dos pseudocódigos largamente utilizados nos livros de introdução à programação, conhecida como "Portugol". Foi tomada a liberdade de acrescentar alguns comandos novos, com o intuito de criar facilidades específicas para o ensino de técnicas de elaboração de algoritmos. Inicialmente, pensava-se em criar uma sintaxe muito simples e "liberal", para que o usuário se preocupasse apenas com a lógica da resolução dos problemas e não com as palavras-chave, pontos e vírgulas, etc. No entanto, chegou-se, depois, à conclusão de que alguma formalidade seria não só necessária como útil, para criar um sentido de disciplina na elaboração do "código-fonte".

A linguagem do Portugol (Português Estruturado) permite apenas **um comando por linha**: desse modo, não há necessidade de *tokens* separadores de estruturas, como o ponto e vírgula em Pascal. Também não existe o conceito de blocos de comandos (que correspondem ao *begin* e *end* do Pascal e ao { e } do C), nem comandos de desvio incondicional como o *goto*. No nosso Portugol (Português Estruturado), com exceção das rotinas de entrada e saída, **não** há nenhum subprograma (função ou procedimento) embutido, tal como *Inc()*, *Sqr()*, *Ord()*, *Chr()*, *Pos()*, *Copy()* ou outro.

Importante: para facilitar a digitação e evitar confusões, todas as palavras-chave do Portugol (Português Estruturado) foram implementadas sem acentos, cedilha, etc. Portanto, o tipo de dados *lógico*, por exemplo, é definido como *logico*, o comando *se..então..senão* é definido como *se..entao..senao*, e assim por diante. O Portugol (Português Estruturado) também não distingue maiúscula e minúscula no reconhecimento de palavras-chave e nomes de variáveis, ou seja, não é case-sensitive.

Formato Básico do Pseudocódigo e Inclusão de Comentários

O formato básico do nosso pseudocódigo é o seguinte:

```
algoritmo "semnome"  
// Função :  
// Autor :  
// Data :  
// Seção de Declarações  
inicio  
// Seção de Comandos  
fimalgoritmo
```

A primeira linha é composta pela palavra-chave *algoritmo* seguido do nome do algoritmo delimitado por aspas duplas. Este nome será usado como título nas janelas de leitura de dados. A seção que se segue é a de declaração de variáveis, que termina com a linha que contém a palavra-chave *inicio*. Deste ponto em diante está a seção de comandos, que continua até a linha em que se encontre a palavra-chave *fimalgoritmo*. Esta última linha marca o final do pseudocódigo: todo texto existente a partir dela é ignorado pelo interpretador.

O Portugol (Português Estruturado) permite a inclusão de comentários: qualquer texto após o "//" é ignorado, até se atingir o final da sua linha. Por este motivo, os comentários não se estendem por mais de uma linha: quando se deseja escrever comentários mais longos, que ocupem várias linhas, cada uma delas deverá começar por "//".

Sintaxe e Semântica

Sintaxe é o nome dado ao conjunto de regras a serem seguidas para a escrita dos algoritmos. Semântica refere-se ao que é efetuado pelo computador quando ele encontra um comando ou uma instrução. A sintaxe está relacionada à forma e a semântica ao conteúdo.

Tipos de Dados

Todo o trabalho realizado por um computador é baseado na manipulação das informações contidas em sua memória. Grosso modo, estas informações podem ser classificadas em dois tipos:

- As instruções, que comandam o funcionamento da máquina e determinam a maneira como devem ser tratados os dados. As instruções são específicas para cada modelo de computador, pois são funções do tipo particular de processador utilizado em sua implementação;
- Os dados propriamente ditos, que correspondem à porção das informações a serem processadas pelo computador.

O objetivo desta aula é justamente o de classificar os dados de acordo com o tipo de informação contida neles. A classificação apresentada não se aplica a nenhuma linguagem de programação específica; pelo contrário, ela sintetiza os padrões utilizados na maioria das linguagens.

Aula 05

Tipos de Dados (Continuação)

Dados Numéricos

Antes de apresentar formalmente os tipos de dados numéricos, é conveniente recordar alguns conceitos básicos relacionados à teoria dos números e conjuntos.

O conjunto dos números naturais é representado por N e é dado por:

$$N = 0, 1, 2, 3, 4, \dots$$

Algumas correntes de matemáticos teóricos convencionam que o número 0 está contido neste conjunto; contudo, não convém perder tempo em tais discussões filosóficas, uma vez que isto não influenciará de forma alguma este estudo.

Na seqüência, encontramos o conjunto dos números inteiros:

$$Z = \dots, -3, -2, -1, 0, 1, 2, 3, \dots$$

O conjunto Z contém todos os elementos de N , bem como alguns números que não pertencem a N (os números negativos e o zero). Portanto, dizemos que N está contido em Z .

Englobando o conjunto dos números inteiros, existe o conjunto dos números fracionários (Q), dado pelo universo dos números que podem ser expressos na forma de uma fração, isto é, um quociente onde o numerador e o denominador são números inteiros.

Mais formalmente:

$$Q = \{p/q \mid (p, q) \in Z\}$$

Por último, surge o conjunto dos números reais (R), formado pela união do conjunto dos números fracionários Q com o conjunto dos números que não podem ser expressos na forma de uma fração (os números irracionais).

$$\text{Ex.: } \sqrt{2} = 1, 41421356 \dots, \mathbf{p} = 3, 14159265 \dots$$

Os números inteiros são aqueles que não possuem componentes decimais ou fracionários, podendo ser positivos ou negativos.

Os elementos pertencentes aos conjuntos N e Z , apesar de serem representáveis na classe dos números reais, são classificados como dados do tipo inteiro, por não possuírem parte fracionária. Esta possibilidade é interessante por permitir uma economia do espaço de memória, como veremos adiante.

Por sua vez, os elementos dos conjuntos Q e R , por possuírem parte fracionária, não podem ser representados na classe inteira, pertencendo necessariamente aos tipos de dados ditos reais.

Ou seja, números inteiros são representados sem parte fracionária e sem ponto (chamado também de número de ponto fixo).

Exemplo: 86 0 -19 23456

Números reais: representados com parte fracionária e com ponto (chamado também de número de ponto flutuante).

Exemplo: 85.3 -9.453 10.0 6. 0.00

Importante: Para representar a vírgula decimal usamos o ponto (.). Assim 3,14 é representado por 3.14.

Dados Literais

Pode ser um único caractere ou pode ser uma seqüência de caracteres contendo letras, dígitos e/ou símbolos especiais. São chamados, também, "*alfanuméricos*", "*cadeia de caracteres*" ou "*strings*". Serão representados nos algoritmos entre aspas. O comprimento de um dado literal é a quantidade de caracteres que ele tem.

Exemplo:

"UFRN"	comprimento	4
" "	comprimento	1
""	comprimento	0
"18/04/99"	comprimento	8

Dados Lógicos

Usados para representar dois únicos valores lógicos possíveis: verdadeiro e falso. Representados nos algoritmos como: VERDADEIRO e FALSO.

RESUMO DOS TIPOS DE DADOS

inteiro	ADMITE SOMENTE NÚMEROS INTEIROS. GERALMENTE É UTILIZADO PARA REPRESENTAR UMA CONTAGEM (QUANTIDADE).
real	ADMITE NÚMEROS REAIS (COM OU SEM CASAS DECIMAIS). GERALMENTE É UTILIZADO PARA REPRESENTAR UMA MEDIÇÃO, UM VALOR MONETÁRIO.
caractere	ADMITE CARACTERES ALFANUMÉRICOS. OS NÚMEROS QUANDO DECLARADOS COMO CARACTERES TORNAM SE REPRESENTATIVOS E PERDEM A ATRIBUIÇÃO DE VALOR.
lógico	ADMITE SOMENTE VALORES LÓGICOS (VERDADEIRO/FALSO).

O nosso Portugol prevê quatro tipos de dados: **inteiro**, **real**, **cadeia de caracteres** e **lógico** (ou *booleano*). As palavras-chave que os definem são as seguintes (observe que elas não têm acentuação):

- `inteiro`: define variáveis numéricas do tipo inteiro, ou seja, sem casas decimais.
- `real`: define variáveis numéricas do tipo real, ou seja, com casas decimais.
- `caractere`: define variáveis do tipo *string*, ou seja, cadeia de caracteres.
- `logico`: define variáveis do tipo *booleano*, ou seja, com valor VERDADEIRO ou FALSO.

Exercício Aula 05

Classifique os dados especificados abaixo de acordo com seu tipo, assinalando com 'I' os dados do tipo inteiro, com 'R' os reais, com 'L' os literais, com 'B' os lógicos (booleanos), e com 'N' aqueles para os quais não é possível definir a priori um tipo de dado.

- | | | | |
|---------------------------------|----------------------------------|-------------------------------------|---------------------------------|
| <input type="checkbox"/> 0.21 | <input type="checkbox"/> "T" | <input type="checkbox"/> "V" | <input type="checkbox"/> 0,35 |
| <input type="checkbox"/> 1 | <input type="checkbox"/> +3257 | <input type="checkbox"/> VERDADEIRO | <input type="checkbox"/> "F" |
| <input type="checkbox"/> W | <input type="checkbox"/> "a" | <input type="checkbox"/> "abc" | <input type="checkbox"/> +36 |
| <input type="checkbox"/> "0" | <input type="checkbox"/> "+3257" | <input type="checkbox"/> FALSO | <input type="checkbox"/> q |
| <input type="checkbox"/> 0.01 | <input type="checkbox"/> +3257 | <input type="checkbox"/> "C" | <input type="checkbox"/> ±3 |
| <input type="checkbox"/> "Jose" | <input type="checkbox"/> "-0.0" | <input type="checkbox"/> "Maria" | <input type="checkbox"/> -0.001 |

Aula 06

Operadores

Operadores são elementos funcionais que atuam sobre **operandos** e produzem um determinado resultado. Por exemplo, a expressão $3 + 2$ relaciona dois operandos (os números 3 e 2) por meio do operador (+) que representa a operação de adição.

De acordo com o número de operandos sobre os quais os operadores atuam, os últimos podem ser classificados em:

- Binários: quando atuam sobre dois operandos. Ex.: os operadores das operações aritméticas básicas (soma, subtração, multiplicação e divisão);
- Unários: quando atuam sobre um único operando. Ex.: o sinal de (-) na frente de um número, cuja função é inverter seu sinal.

Outra classificação dos operadores é feita considerando-se o tipo de dado de seus operandos e do valor resultante de sua avaliação. Segundo esta classificação, os operadores dividem-se em aritméticos, lógicos e literais. Esta divisão está diretamente relacionada com o tipo de expressão onde aparecem os operadores.

Um caso especial é o dos operadores relacionais, que permitem comparar pares de operandos de tipos de dados iguais, resultando sempre num valor lógico.

OPERADORES ARITMÉTICOS

+ , -	Operadores unários, isto é, são aplicados a um único operando. São os operadores aritméticos de maior precedência. Exemplos: -3 , $+x$. Enquanto o operador unário $-$ inverte o sinal do seu operando, o operador $+$ não altera o valor em nada o seu valor.
\	Operador de divisão inteira. Por exemplo, $5 \setminus 2 = 2$. Tem a mesma precedência do operador de divisão tradicional.
+ , - , * , /	Operadores aritméticos tradicionais de adição, subtração, multiplicação e divisão. Por convenção, $*$ e $/$ têm precedência sobre $+$ e $-$. Para modificar a ordem de avaliação das operações, é necessário usar parênteses como em qualquer expressão aritmética.
%	Operador de módulo (isto é, resto da divisão inteira). Por exemplo, $8 \% 3 = 2$. Tem a mesma precedência do operador de divisão tradicional.
^	Operador de potenciação. Por exemplo, $5 \wedge 2 = 25$. Tem a maior precedência entre os operadores aritméticos binários (aqueles que têm dois operandos).

OPERADORES RELACIONAIS

>	Maior que
<	Menor que
>=	Maior ou Igual
<=	Menor ou Igual
=	Igual
<>	Diferente
Exemplos	$3 = 3$ (3 é igual a 3?) resulta em VERDADEIRO ; "A" > "B" ("A" está depois de "B" na ordem alfabética?) resulta em FALSO.

Importante: No nosso Portugal, as comparações entre *strings* **não diferenciam** as letras maiúsculas das minúsculas. Assim, "ABC" é igual a "abc".

Valores lógicos obedecem à seguinte ordem: FALSO < VERDADEIRO.

Operadores de Caracteres

+	Operador de concatenação de <i>strings</i> (isto é, cadeias de caracteres), quando usado com dois valores (variáveis ou constantes) do tipo "caractere". Por exemplo: "Rio " + " de Janeiro" = "Rio de Janeiro".
---	---

Operadores Lógicos

nao	Operador unário de negação. nao VERDADEIRO = FALSO, e nao FALSO = VERDADEIRO. Tem a maior precedência entre os operadores lógicos. Equivale ao NOT do Pascal.
ou	Disjunção - Operador que resulta VERDADEIRO quando um dos seus operandos lógicos for verdadeiro. Equivale ao OR do Pascal.
e	Conjunção - Operador que resulta VERDADEIRO somente se seus dois operandos lógicos forem verdadeiros. Equivale ao AND do Pascal.
xou	Disjunção Exclusiva - Operador que resulta VERDADEIRO se seus dois operandos lógicos forem diferentes, e FALSO se forem iguais. Equivale ao XOR do Pascal.

TABELA-VERDADE DOS OPERADORES LÓGICOS

A	B	A e B	A ou B	nao (A)	A xou b
VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	FALSO
VERDADEIRO	FALSO	FALSO	VERDADEIRO	FALSO	VERDADEIRO
FALSO	VERDADEIRO	FALSO	VERDADEIRO	VERDADEIRO	VERDADEIRO
FALSO	FALSO	FALSO	FALSO	VERDADEIRO	FALSO

Expressões

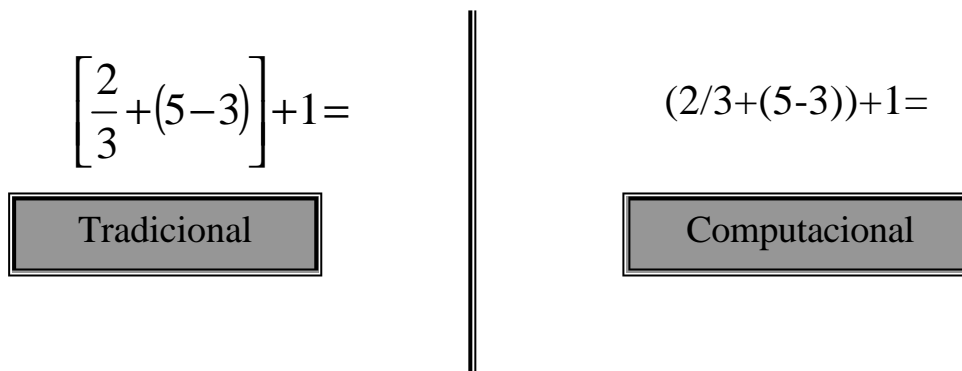
Expressão é uma combinação de variáveis, constantes e operadores que, uma vez avaliada, resulta num valor.

Avaliação de Expressões

Regras:

- Observar a prioridade dos operadores (os de maior prioridade devem ser avaliados primeiros) . Se houver empate, considera-se a expressão da esquerda para a direita.
- Os parênteses alteram a prioridade, forçando a avaliação da subexpressão em seu interior.

Para a construção de Algoritmos todas as expressões aritméticas devem ser linearizadas, ou seja, colocadas em linhas. É importante também ressaltar o uso dos operadores correspondentes da aritmética tradicional para a computacional.



A tabela a seguir apresenta a prioridade assumida para o cálculo de expressões.

PRIORIDADE NO CÁLCULO DE EXPRESSOES

1º. lugar	Parênteses mais internos		
2º. Lugar	Funções		
3º. Lugar	Operadores aritméticos	1º. lugar	Exponenciação (^)
		2º. lugar	Multiplicativos (*, /, \, %)
		3º. lugar	Aditivos (+, -)
4º. Lugar	Operadores relacionais		
5º. lugar	Operadores lógicos	1º. lugar	nao
		2º. lugar	e
		3º. lugar	ou, xou

Havendo empate, como já observado na página anterior, deve-se efetuar o cálculo mais à esquerda.

Conversão de Notação

Veja a formula abaixo:

$$\frac{a+b}{c}d$$

Assumindo que seus valores são reais, em notação algorítmica, ela seria:

$$(a+b) / c * d$$

Outro exemplo:

$$\frac{\sqrt{x+y}}{b^2}$$

Assumindo valores reais, seria:

$$((x+y)^{0.5}) / (b^2)$$