UNIDADE II

Linguagem de Programação do VisuAlg

NDDigital

ALGORITMO - PORTUGOL

LAGES
(2014)

Sumário

1 UNIDADE 2 – LINGUAGEM DE PROGRAMAÇÃO DO VISUALG

Nessa unidade discutiremos sobre a linguagem de programação que da ferramenta Visualg 2.0. Ferramenta que vimos na unidade anterior.

1.1 INTRODUÇÃO

A linguagem que o VisuAlg interpreta é bem simples: é uma versão portuguesa dos pseudocódigos largamente utilizados nos livros de introdução à programação, conhecida como "Portugol". Essa linguem contém alguns comandos novos, com o intuito de criar facilidades específicas para o ensino de técnicas de elaboração de algoritmos. Inicialmente, o objetivo era 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, cheguei 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 VisuAlg permite apenas um comando por linha: desse modo, não há necessidade de tokens separadores de estruturas, como o ponto e vírgula em C#. 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. Na versão atual do VisuAlg, com exceção das rotinas de entrada e saída, não há nenhum subprograma embutido, tal como Inc(), Sqr(), Ord(), Chr(), Pos(), Copy() ou outro.

Importante: para facilitar a digitação e evitar confusões, todas as palavras-chave do VisuAlg foram implementadas sem acentos, cedilha, etc. Portanto, o tipo de dados lógico é definido como logico, o comando se..então..senão é definido como se..entao..senao, e assim por diante. O VisuAlg também não distingue maiúsculas e minúsculas no reconhecimento de palavras-chave e nomes de variáveis.

1.2 FORMATO BÁSICO DO PSEUDOCÓDIGO E INCLUSÃO DE COMENTÁRIOS

O formato básico do pseudocódigo é o seguinte:

```
algoritmo "semnome"
// Função :
// Autor :
// Data : 27/02/2014
// Seção de Declarações
var

inicio
// Seção de Comandos
fimalgoritmo
```

‘ A primeira linha é composta pela palavra-chave algoritmo seguida do seu nome delimitado por aspas duplas. Este nome será usado como título nas janelas de leitura de dados (nas futuras versões do VisuAlg, talvez utilizemos este dado de outras formas). A seção que se segue é a de declaração de variáveis, que termina com a linha que contém a palavra-chave *início*. 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 VisuAlg permite a inclusão de comentários: qualquer texto precedido de "//" 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 "//".

1.3 INSTALAÇÃO E REQUERIMENTOS DE HARDWARE

O VisuAlg é um programa simples, que não depende de DLLs, OCXs ou outros componentes. Sua instalação não copia arquivos para nenhuma outra pasta a não ser aquela em que for instalado, e exige cerca de 1 MB de espaço em disco. Pode ser executado sob

Windows 95 ou posterior, e tem melhor aparência com resolução de vídeo de 800x600 ou maior.

1.4 TIPO DE DADOS

O VisuAlg 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.

O VisuAlg permite também a declaração de variáveis estruturadas através da palavra-chave vetor, como será explicado a seguir.

1.4 NOMES DE VARIÁVEIS E SUA DECLARAÇÃO

Os nomes das variáveis devem começar por uma letra e depois conter letras, números ou underline, até um limite de 30 caracteres. As variáveis podem ser simples ou estruturadas (na versão atual, os vetores podem ser de uma ou duas dimensões). Não pode haver duas variáveis com o mesmo nome, com a natural exceção dos elementos de um mesmo vetor.

A seção de declaração de variáveis começa com a palavra-chave var, e continua com as seguintes sintaxes:

<lista-de-variáveis> : <tipo-de-dado>
<lista-de-variáveis> : vetor "["<lista-de-intervalos>"]" de <tipo-de-dado>

Na **<lista-de-variáveis>,** os nomes das variáveis estão separados por vírgulas. Na **<lista-de-intervalos>,** os **<intervalo>** são separados por vírgulas, e têm a seguinte sintaxe:

<intervalo>: <valor-inicial> .. <valor-final>

Na versão atual do VisuAlg, tanto **<valor-inicial>** como **<valor-final>** devem ser inteiros. Além disso, exige-se evidentemente que **<valor-final>** seja maior do que <valor-inicial>.

Exemplos:

```
algoritmo "semnome"
// Função :
// Autor :
// Data : 24/02/2014
// Seção de Declarações

var a: inteiro
    Valor1, Valor2: real
    vet: vetor [1..10] de real
    matriz: vetor [0..4,8..10] de inteiro
    nome_do_aluno: caractere
sinalizador: logico

inicio
// Seção de Comandos
fimalgoritmo
```

Note que não há a necessidade de ponto e vírgula após cada declaração: basta pular linha. A declaração de vetores é análoga à linguagem C#: a variável *vet* acima tem 10 elementos, com os índices de [1] a [10], enquanto matriz corresponde a 15 elementos com índices [0,8], [0,9], [0,10], [1,8], [1,9], [1,10], ... até [4,10]. O número total de variáveis suportado pelo VisuAlg é 500 (cada elemento de um vetor é contado individualmente).

1.5 CONSTANTES E COMANDO DE ATRIBUIÇÃO

O VisuAlg tem três tipos de constantes:

- Numéricos: são valores numéricos escritos na forma usual das linguagens de programação. Podem ser inteiros ou reais. Neste último caso, o separador de decimais é o ponto e não a vírgula, independente da configuração regional do computador onde o VisuAlg está sendo executado. O VisuAlg também não suporta separadores de milhares.
- Caracteres: qualquer cadeia de caracteres delimitada por aspas duplas (").
- Lógicos: admite os valores VERDADEIRO ou FALSO.

A atribuição de valores a variáveis é feita com o operador *"<-"*. Do seu lado esquerdo fica a variável à qual está sendo atribuído o valor, e à sua direita pode-se colocar qualquer expressão (constantes, variáveis, expressões numéricas), desde que seu resultado tenha tipo igual ao da variável.

Alguns exemplos de atribuições, usando as variáveis declaradas acima:

```
a <- 3
Valor1 <- 1.5
Valor2 <- Valor1 + a
vet[1] <- vet[1] + (a * 3)
matriz[3,9] <- a/4 - 5
nome_do_aluno <- "José da Silva"
sinalizador <- FALSO
```

1.5.1 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 \ 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.
`MOD ou %`	Operador de módulo (isto é, resto da divisão inteira). Por exemplo, `8 MOD 3 = 2`. Tem a mesma precedência do operador de divisão tradicional.
`^`	Operador de potenciação. Por exemplo, 5 ^ 2 = 25. Tem a maior precedência entre os operadores aritméticos binários (aqueles que têm dois operandos).

1.5.2 Operadores de Caracteres

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

1.5.3 Operadores Relacionais

`=, <, >, <=, >=, <>`	Respectivamente: igual, menor que, maior que, menor ou igual a, maior ou igual a, diferente de. São utilizados em expressões lógicas para se testar a relação entre dois valores do mesmo tipo. 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 VisuAlg, 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.

1.5.4 Operadores Lógicos

`nao`	Operador unário de negação. `nao VERDADEIRO = FALSO`, e `nao FALSO = VERDADEIRO`. Tem a maior precedência entre os operadores lógicos. Equivale ao `!` do C#.
`ou`	Operador que resulta `VERDADEIRO` quando um dos seus operandos lógicos for verdadeiro. Equivale ao || do C#.
`e`	Operador que resulta `VERDADEIRO` somente se seus dois operandos lógicos forem verdadeiros. Equivale ao `&&` do C#.
`xou`	Operador que resulta VERDADEIRO se seus dois operandos lógicos forem diferentes, e FALSO se forem iguais. Equivale ao ^ do C#.

1.6 COMANDOS DE SAÍDA DE DADOS

`escreval``("<lista-de-expresões>)`

Escreve no dispositivo de saída padrão (isto é, na área à direita da metade inferior da tela do VisuAlg) o conteúdo de cada uma das expressões que compõem <lista-de-expressões>. As expressões dentro desta lista devem estar separadas por vírgulas; depois

de serem avaliadas, seus resultados são impressos na ordem indicada. É equivalente ao comando *Console.WriteLine* do C#.

De modo semelhante à C#, é possível especificar o número de espaços no qual se deseja escrever um determinado valor. Por exemplo, o comando escreva(x:5) escreve o valor da variável x em 5 espaços, alinhado à direita. Para variáveis reais, pode-se também especificar o número de casas fracionárias que serão exibidas. Por exemplo, considerando y como uma variável real, o comando escreva(y:6:2) escreve seu valor em 6 espaços colocando 2 casas decimais.

escreval(<lista-de-expressões>).

Idem ao anterior, com a única diferença que pula uma linha em seguida. É equivalente ao *Console.WriteLine* do C#.

Exemplos:

```
1  algoritmo "exemplo"
2  var x: real
3      y: inteiro
4      a: caractere
5      l: logico
6  inicio
7  x <- 2,5
8  y <- 6
9  a <- "teste"
10 l <- VERDADEIRO
11 escreval ("x", x:4:1, y+3:4)// Escreve: x 2.5    9
12 escreval (a, "ok")          // Escreve: testeok (e depois pula linha) escreval (a, " ok")
13 escreval (a, " ok")         // Escreve: teste ok (e depois pula li-nha) escreval (a + " ok")
14 escreval (a + " ok")        // Escreve: teste ok (e depois pula linha) escreva (l)
15 escreva  (l)                // Escreve: VERDADEIRO
16 fimalgoritmo
17
```

Note que o VisuAlg separa expressões do tipo numérico e lógico com um espaço à esquerda, mas não as expressões do tipo caractere, para que assim possa haver a concatenação. Quando se deseja separar expressões do tipo caractere, é necessário acrescentar espaços nos locais adequados.

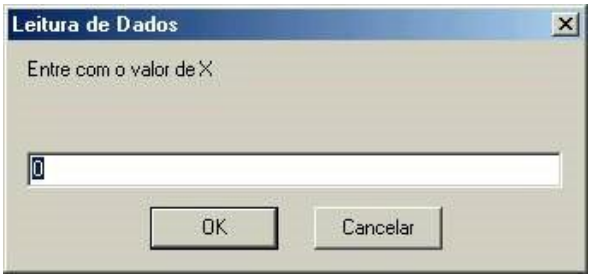
1.7 COMANDO DE ENTRADA DE DADOS

leia (*<lista-de-variáveis>*)

Recebe valores digitados pelos usuário, atribuindo-os às variáveis cujos nomes estão em *<lista-de-variáveis>* (é respeitada a ordem especificada nesta lista). É análogo ao comando *Console.ReadLine* do C#.

Veja no exemplo abaixo o resultado:

```
1  algoritmo "exemplo 1"
2  var x: inteiro
3
4  inicio
5
6  leia (x)
7  escreva (x)
8
9  fimalgoritmo
10
```



O comando de leitura acima irá exibir uma janela como a que se vê ao lado, com a mensagem padrão:

`"Entre com o valor de <nome-de-variável>"`

Se você clicar em *Cancelar* ou teclar *Esc* durante a leitura de dados, o programa será imediatamente interrompido.

Importante: Caso o modo de saída esteja selecionado como DOS. Esses comandos serão exibido em uma tela preta, o famoso *console* ou *cmd*.

1.8EXERCÍCIOS: FIXAÇÃO

1) Faça um algoritmo que receba dois números e exiba o resultado da sua soma.

2) Faça um algoritmo que receba dois números e ao final mostre a soma, subtração, multiplicação e a divisão dos números lidos.

3) Escrever um algoritmo para determinar o consumo médio de um automóvel sendo fornecida a distância total percorrida pelo automóvel e o total de combustível gasto.

4) Escrever um algoritmo que leia o nome de um vendedor, o seu salário fixo e o total de vendas efetuadas por ele no mês (em dinheiro). Sabendo que este vendedor ganha 15% de comissão sobre suas vendas efetuadas, informar o seu nome, o salário fixo e salário no final do mês.

5) Escrever um algoritmo que leia o nome de um aluno e as notas das três provas que ele obteve no semestre. No final informar o nome do aluno e a sua média (aritmética).

6) Ler dois valores para as variáveis A e B, e efetuar as trocas dos valores de forma que a variável A passe a possuir o valor da variável B e a variável B passe a possuir o valor da variável A. Apresentar os valores trocados.

7) Ler uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão é: F=(9*C+160) / 5, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.

8) Elaborar um algoritmo que efetue a apresentação do valor da conversão em real (R$) de um valor lido em dólar (US$). O algoritmo deverá solicitar o valor da cotação do dólar e também a quantidade de dólares disponíveis com o usuário.

9) Faça um algoritmo que receba um valor que foi depositado e exiba o valor com rendimento após um mês. Considere fixo o juro da poupança em 0,70% a. m.

10) A Loja Mamão com Açúcar está vendendo seus produtos em 5 (cinco) prestações sem juros. Faça um algoritmo que receba um valor de uma compra e mostre o valor das prestações.

11) Faça um algoritmo que receba o preço de custo de um produto e mostre o valor de venda. Sabe-se que o preço de custo receberá um acréscimo de acordo com um percentual informado pelo usuário.

12) O custo ao consumidor de um carro novo é a soma do custo de fábrica com a percentagem do distribuidor e dos impostos (aplicados, primeiro os impostos sobre o custo de fábrica, e depois a percentagem do distribuidor sobre o resultado). Supondo que a percentagem do distribuidor seja de 28% e os impostos 45%. Escrever um algoritmo que leia o custo de fábrica de um carro e informe o custo ao consumidor do mesmo.

13) Faça um algoritmo que imprima as seguintes junções (concatenação) de palavras ou caracteres e inteiros. “Curso ” + “NDDigital ” + “Beginner ” + 2014. São 4 variáveis, 3 de texto e uma numérica.

14) Escreva um algoritmo que envie uma mensagem como FALSO caso o cálculo for VERDADEIRO e VERDADEIRO caso for falso. O programa deve ler 2 números e ultilizar todos os tipos de operadores lógicos já mostrados na UNIDADE. Ex: 2 > 1 = FALSO

15) Faça um algoritmo que armazene em um vetor dos 10 nomes dos integrantes da sala. Em uma ordem que comece pelo instrutor e as primeiras fileiras até as últimas. No final, mostre na tela de forma contrária do último ao instrutor.

Desafio:

Escreva um algoritmo que leia três números inteiros e positivos (A, B, C) e calcule a seguinte expressão:

$$D = \frac{R + S}{2}, \text{ onde } R = (A + B)^2 \quad S = (B + C)^2$$

Bons estudos!