

Conceitos Básicos de Algoritmos

TIPOS PRIMITIVOS

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

Conceitos Básicos de Algoritmos

DECLARAÇÃO DE VARIÁVEIS

- Palavra-reservada: var - utilizada para iniciar a seção de declaração de variáveis

Exemplos:

var a: inteiro

nome_do_aluno: caractere

sinalizador: logico

Valor1, Valor2: real

Obs.: O pseudocódigo e o fluxograma não são sensíveis ao caso.

Conceitos básicos de algoritmos

Durante a execução de determinadas tarefas ou durante a execução de um processo para obtenção da solução de um determinado problema são necessários alguns dados e ao final deve-se retornar uma saída.

Da mesma forma, em alguns algoritmos tornam-se necessários mecanismos que possibilitem uma interface com o ambiente externo, ou seja, são necessários comandos que possibilitem a entrada e saída de dados.

Conceitos Básicos de Algoritmos

Entrada de dados

- ▶ Palavra-reservada: leia - utilizada para receber dados externos ao algoritmo e armazená-los em variáveis.

Exemplos:

...

```
var a: inteiro  
    b: real
```

...

```
leia(a)  
leia(b, a)
```

...

Conceitos Básicos de Algoritmos

Entrada de Dados

Tecnicamente podemos utilizar a seguinte definição: O comando de entrada de dados *leia* possui a sintaxe

`leia (<lista-de-variáveis>)`

Onde este recebe valores digitados pelo usuário, atribuindo-os às variáveis cujos nomes estão em <lista-de-variáveis> (é respeitada a ordem especificada nesta lista).

Conceitos Básicos de Algoritmos

Saída de dados

- Palavra-reservada: escreva - utilizada para externar (enviar para o monitor) dados gerados pelo algoritmo.

Exemplos:

...

```
var  x: inteiro
     y: caractere
```

...

```
escreva(y)
escreval(x,y)
```

```
escreva("Estudou e se dedicou tirou ", 10-x)
```

...

Conceitos Básicos de Algoritmos

Saída de Dados

Tecnicamente podemos utilizar a seguinte definição: O comando de saída de dados **escreva** possui a sintaxe

escreva (<*lista-de-expressões*>)

Onde este escreve no dispositivo de saída padrão (monitor) 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.

Conceitos Básicos de Algoritmos

Saída de dados

É possível especificar o número de colunas da tela onde se deseja escrever um determinado valor. Por exemplo, considerando uma variável inteira x , o comando `escreva(x:5)` escreve o valor da variável x em 5 colunas, alinhado-o à direita.

Para variáveis reais, além de especificar o número de colunas a serem reservadas para representar o número, também pode-se especificar o número de casas fracionárias que serão exibidas.

Por exemplo, considerando z como uma variável real, o comando `escreva(z:5:2)`, escreve seu valor em 5 colunas colocando 2 casas decimais.

Conceitos básicos de algoritmos

➔ Operador de Atribuição símbolo <-

Exemplo: var a: inteiro
 valor1, valor2: real

...

A <- 5

Valor1 <- 3.14

valor2 <- valor1

escreva(valor2)

...

Conceitos básicos de algoritmos

➤ Operadores Aritméticos

➤ Unários: +, -

Exemplos: +1
 -5.9
 ...
 var a: inteiro
 ...
 a <- -a

Conceitos básicos de algoritmos

➤ Operadores Aritméticos

➤ Binários: +, -, *, /, \, %, ^

Associação

<i>Símbolo</i>	<i>Operação</i>
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
\	Quociente da divisão inteira
%	Resto da divisão inteira
^	Potenciação

Conceitos básicos de algoritmos

➤ Operadores Aritméticos

➤ Binários

Precedência (Hierarquia nas operações)

Hierarquia	Operação
1	Parênteses
2	Função
3	-, + (unários)
4	\wedge
5	*, /, \, %
6	+, -

Conceitos básicos de algoritmos

Expressões aritméticas

Exemplos:

$$3/4+5 = 5.75$$

$$3/(4+5) = 0.333333333$$

$$3\backslash 2*9 = 9$$

$$11\%3^2 = 2$$

$$11\%(3^2) = 2$$

$$(11\%3)^2 = 4$$

$$3\backslash 2+(65-40)^{(1/2)} = 6$$

Conceitos básicos de algoritmos

➔ Operadores Relacionais

Operador	Ação
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual a
=	igual a
<>	diferente de

Conceitos básicos de algoritmos

➤ Operadores Lógicos

Operador
e
ou
nao
xou

Conceitos básicos de algoritmos

Expressões lógicas

Exemplos:

$3 > 7$ = FALSO

"A" = "a" = VERDADEIRO

"a" > "B" = FALSO

$(3 >= 13 \setminus 4)$ xou $(\text{nao } (5 \% 2 = 0))$ = FALSO

Conceito de Lógica

Para retomarmos à nossa análise dos métodos de representação de algoritmos devemos refletir a respeito do seguinte questionamento:

O que nos orienta a gerar a seqüência de passos para instruir o computador a executar uma tarefa?

A lógica é o ramo da Filosofia e da Matemática que estuda os métodos e princípios que permitem fazer distinção entre raciocínios válidos e não válidos, determinando o processo que leva ao conhecimento verdadeiro.

O uso da lógica é primordial na solução de problemas. Com ela é possível alcançar objetivos com eficiência e eficácia.

Ninguém ensina outra pessoa a pensar, mas a desenvolver e aperfeiçoar esta técnica, com persistência e constância.

Método para Construção de Algoritmos

Os passos necessários para a construção de um algoritmo são:

- ler atentamente o enunciado do problema, compreendendo-o e destacando os pontos mais importantes;
- definir os dados de entrada, ou seja, quais dados serão fornecidos;
- definir os dados de saída, ou seja, quais dados serão gerados depois do processamento;
- definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela obtenção dos dados de saída com base nos dados de entrada;
- definir as variáveis necessárias para armazenar as entradas e efetuar o processamento;
- elaborar o algoritmo;
- testar o algoritmo realizando simulações.

Pseudocódigo

Com base nos conceitos vistos podemos aprofundar nosso estudo sobre os métodos de representação de algoritmos, iniciaremos pelo pseudocódigo.

Pseudocódigo é uma técnica textual de representação de um algoritmo. Também é conhecida como Português Estruturado ou Portugol.

Nele os verbos (ações) disponíveis para utilização são limitados e empregados no imperativo, deve-se evitar as expressões excessivamente longas, estas restrições visam eliminar a possibilidade de ambigüidade.

A técnica é baseada em uma PDL (Program Design Language), que é uma linguagem genérica na qual é possível representar um algoritmo de forma semelhante à das linguagens de programação.

Pseudocódigo – Estrutura

A estrutura de um algoritmo em pseudocódigo pode variar um pouco de acordo com o autor ou com base na linguagem de programação que será utilizada posteriormente, mas essas variações ocorrem apenas na sintaxe, pois a semântica deve ser exatamente a mesma.

A estrutura que empregaremos para a construção de nossos pseudocódigos será a seguinte:

algoritmo “nome” //Tem como objetivo identificar o
//algoritmo, deve-se utilizar um nome o mais significativo
//possível, para facilitar a identificação

var

// Seção de Declarações - Neste ponto são informadas
//quais variáveis, e seus respectivos tipos, serão
// utilizadas no algoritmo

inicio

// Seção de Comandos - Aqui será escrita a seqüência de
// comandos que deve ser executada para solucionar
// o problema em questão

fimalgoritmo //marca o final do algoritmo

Pseudocódigo

As palavras reservadas presentes na estrutura do pseudocódigo apresentado foram:

algoritmo

var

inicio

fimalgoritmo

Pseudocódigo – Exemplo

Observaremos agora um pseudocódigo que recebe um valor inteiro, fornecido pelo usuário, e o retorna no monitor.

algoritmo “exemplo 1”

var x: inteiro

inicio

 leia (x)

 escreva (x)

fimalgoritmo

Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exhibe o resultado desta manipulação.

algoritmo “exemplo 2”

var n: inteiro

inicio

 escreva (“Digite um número inteiro: ”)

 leia (n)

$n \leftarrow n + 2$

 escreval (“Resultado (número + 2): ”, n)

fimalgoritmo

**Slides trabalhados durante a primeira
aula prática**

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo “exercício 1”

var n1, n2, res: inteiro

inicio

 escreva (“Digite o multiplicando inteiro: ”)

 leia (n1)

 escreva (“Digite o multiplicador inteiro: ”)

 leia (n2)

 res <- n1 * n2

 escreva (“Resultado da multiplicação: ”, res)

fimalgoritmo

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo “exercício 1b”

var n1, n2: inteiro

inicio

 escreva (“Digite o multiplicando inteiro: ”)

 leia (n1)

 escreva (“Digite o multiplicador inteiro: ”)

 leia (n2)

 escreva (“Resultado da multiplicação: ”, $n1 * n2$)

fimalgoritmo

Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo “exercício 1c”

var n1, n2: inteiro

inicio

 escreva ("Digite o multiplicando e em seguida ")

 escreva (" o multiplicador (ambos inteiros): ")

 leia (n1, n2)

 escreva ("Resultado da multiplicação: ", n1*n2)

fimalgoritmo

Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.

algoritmo “exercício 2a”

var valor, resultado: real

inicio

 escreva (“Entre com o valor do produto: ”)

 leia (valor)

 resultado <- valor * 0.7

 escreva (“Valor com desconto de 30%: ”)

 escreva (resultado)

fimalgoritmo

Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.

algoritmo “exercício 2b”

var valor: real

inicio

 escreva (“Entre com o valor do produto: ”)

 leia (valor)

 valor <- valor * 0.7

 escreva (“Valor com desconto de 30%: ”)

 escreva (valor:6:2)

fimalgoritmo

Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.

algoritmo “exercício 2c”

var valor: real

inicio

 escreva (“Entre com o valor do produto: ”)

 leia (valor)

 escreva (“Valor com desconto de 30%: ”, valor *
0.7:6:2)

fimalgoritmo