CONCEITOS DE ALGORITMOS

Fundamentos da Programação de Computadores - 3ª Ed. 2012
Editora Prentice Hall
ISBN 9788564574168
Ana Fernanda Gomes Ascênsio
Edilene Aparecida Veneruchi de Campos

Algoritmos são passos ordenados em conjunto, que soluciona um problema. Entende-se por ação um evento que acorre num período de tempo finito, estabelecendo um efeito intencionado e bem definido. Quando consideramos um evento como uma sequência de ações, falamos de um processo sequencial. Para descrever um evento usaremos inicialmente a forma de relato de um observador.

Por exemplo:

1) Fazer uma parede de tijolos.
pegar os tijolos
pegar as argamassas
colocar os tijolos em arranjo sobre a argamassa

2) Trocar pneu
tirar o pneu furado
colocar o pneu cheio

Um algoritmo é a descrição de um comportamento, expresso em termos de eventos bem definidos e finitos de ações "primitivas", os quais podem ser executados. Um algoritmo é, em outras palavras, uma norma executável para estabelecer um certo efeito desejado, que na prática será um conjunto de comandos que, quando executados, resultam na solução de um certo tipo de problema.

No cotidiano encontramos constantemente algoritmos, instruções para uso, receitas de cozinha, indicações de montagem, etc.

Computador é uma máquina capaz de seguir uma certa espécie de algoritmo, chamado programa. Ele possui uma memória, capaz de armazenar dados, e uma unidade aritmética, que é capaz de modificar os dados que estão na memória. Além disso, comunica-se com o mundo exterior através de operações de entrada e saída.

Outro exemplo:

3) Calcular a média
obter quatro notas
calcular a média
mostrar a média

Refinando:
obter N1, N2, N3, N4
média é calculada somando-se as quatro notas e dividindo por quatro
mostrar a média calculada

REFINAMENTOS:
Reduzem a complexidade do algoritmo e o torna mais legível. Técnica utilizada para que o algoritmo fique completo e possa ser entendido por outra pessoa. Exemplo: Como trocar um pneu furado

DOCUMENTAÇÃO: Comentários colocados junto ao algoritmo com a finalidade de esclarecer os comandos utilizados no algoritmo. Utilizaremos símbolos especiais para escrever comentário (**{comentário}**).

ITENS FUNDAMENTAIS

A partir de agora veremos um conjunto particular de regras e convenções para o desenvolvimento de algoritmos.

1 – CONSTANTES: Valores que não se modificam ao longo da execução do algoritmo. Uma constante pode ser um valor numérico, um valor lógico ou literal.

- Constante numérica: valor numérico com ou sem parte fracionária

Exemplos:

a) 10
b) $7,8*10^2$
c) 3,14
d) –0,38

2 – VARIÁVEIS: Uma variável na matemática é a representação simbólica dos elementos de um certo conjunto. Nos algoritmos, a cada variável corresponde uma posição de memória e o seu conteúdo pode variar durante a execução do algoritmo.

As variáveis são identificadas por um nome ou identificador.

- Formação de identificadores:

Um identificador é formado por 1 ou mais caracteres, sendo que o primeiro deve, obrigatoriamente, ser uma letra e os seguintes letras ou dígitos. Não são permitidos símbolos especiais com exceção do sublinhado (_).

Exemplos de identificadores **válidos**:

a) A
b) NOTA
c) X5
d) A32B
e) TOT_NOTA

Exemplos de identificadores **não válidos**:

a) 5B
b) E(13)
c) A*B
d) X-Y

- Declaração de variáveis:

Usamos declarar variáveis em um algoritmo para que seja reservado área na memória para armazenar a informação. Além disso, indicamos o tipo de informação que a variável irá conter.

TIPOS BÁSICOS: numérico, lógico e literal

Sintaxe:

```
declare  lista-de-identificadores  tipo
```

onde:
declare: palavra-chave
Lista-de-identificadores: variáveis que serão utilizadas no algoritmo
tipo: qual o conteúdo que a variável terá

Obs.: **Palavra-chave** tem um significado próprio, independente do algoritmo. Não pode ser usada como identificador.

3 – COMENTÁRIOS: É um texto, ou simplesmente uma frase, que aparece sempre delimitado por chaves **{comentário}**, com o objetivo de explicar o algoritmo.

Exemplo:

declare NOTA numérico {Declaração de variáveis}

4 – EXPRESSÕES ARITMÉTICAS: Formadas por operadores aritméticos e operandos que são constantes e/ou variáveis do tipo numérico.

Operações básicas:

Adição: (+)
Subtração: (-)
Multiplicação: (*)
Divisão: (/)

Exemplos:

a) X + Y
b) 2 * NOTA
c) A * B + C
d) SOMA / N

5 – EXPRESSÕES LÓGICAS: Alguma ação durante a execução do algoritmo pode estar sujeita a uma condição. Esta condição é representada por uma expressão lógica.

Os operadores são lógicos e os operandos são relações, constantes e/ou variáveis do tipo lógico.

- RELAÇÕES: Comparação realizada entre dois valores do mesmo tipo básico.

Operadores relacionais:

= (igual a)
<> ou ≠ (diferente de)
> (maior que)
< (menor que)
>= ou ≥ (maior ou igual a)
<= ou ≤ (menor ou igual a)

O resultado é sempre um valor lógico (Falso ou Verdadeiro)

Operadores lógicos: Conectivos que também são operadores em expressões lógicas.

e (é verdadeiro ⇔ ambas as proposições são verdadeiras)
ou (é verdadeiro ⇔ pelo menos uma das proposições é verdadeira)

Exemplos:

a) A + B = 0 e C <> 1
b) TESTE ou A * B > 1

Observe o uso da tabela verdade:

P	Q	p **e** q
V	V	V
V	F	F
F	V	F
F	F	F

p	q	p **ou** q
V	V	V
V	F	V
F	V	V
F	F	F

Onde p e q: são proposições.

V: verdadeiro F: falso

6 – EXPRESSÕES LITERAIS: Formada por operadores literais e operandos que são constantes e/ou variáveis do tipo literal.

As operações dependem muito da linguagem de programação utilizada.

7 – COMANDOS DE ATRIBUIÇÃO: Para atribuição de um valor a uma variável, usaremos o símbolo de atribuição (←). O valor a ser atribuído tem que ser compatível com o tipo da variável.

Sintaxe:

Identificador ← expressão

onde:

Identificador é o nome da variável
← é o símbolo de atribuição - RECEBA
Expressão é um conteúdo do tipo numérico, lógico ou literal

Exemplos:

a) K ← 1 b) COR ← "VERDE" c) TESTE ← falso

8 – COMANDOS DE ENTRADA E DE SAÍDA: Algumas vezes, será necessário fornecer informações ao ambiente externo do computador, por exemplo, vídeo ou impressora. Da mesma maneira, precisa-se obter informações do ambiente externo, por exemplo, o teclado. Em algoritmos, utiliza-se os comandos de E/S para esta finalidade.

- Comandos de Entrada:

Sintaxe:

leia lista-de-identificadores

onde:

leia é uma palavra chave que indica entrada de valores
lista-de-identificadores são as variáveis que estão sendo lidas

Exemplo:

a) leia A, B, SOMA

A, B, SOMA são variáveis do tipo numérico, portanto, 3 valores numéricos serão lidos de um dispositivo de entrada e armazenados na memória.

- Comandos de Saída:

Sintaxe:

escreva lista-de-identificadores

onde:

escreva é uma palavra chave que indica saída de valores
lista-de-identificadores são as variáveis que estão sendo escritas

Exemplos:

a) escreva A, B, SOMA
A, B, SOMA são variáveis do tipo numérico, portanto, 3 valores numéricos serão escritos em um dispositivo de saída.

b) escreva "O nome é: ", NOME
Neste exemplo o algoritmo emitirá uma mensagem seguida do valor existente na variável NOME.

9 – ESTRUTURA SEQUENCIAL: Num algoritmo, declara-se as variáveis primeiramente, e em seguida estão os comandos que, sem indicação do contrário, são executados seqüencialmente, de cima para baixo.

```
algoritmo
          <Declaração de variáveis>
          <Bloco de comandos>
fim algoritmo
```

10 – ESTRUTURA CONDICIONAL: Quando a ação a ser executada depender de um teste, utiliza-se a estrutura de comando **se fim se**. Estes testes representam expressões lógicas que são satisfeitas ou não.

- Estrutura condicional simples:

Sintaxe:

```
Se condição
     então <Bloco de comandos>
fim se
```

O bloco de comandos só será executado se a condição for verdadeira. Entenda-se por bloco de comandos um ou mais comandos.

Exemplo:

```
algoritmo
         declare A, B, C numérico
         leia A, B, C
         se A + B < C
              então escreva "MENSAGEM"
         fim se
fim algoritmo
```

- Estrutura condicional composta:

Sintaxe:

```
Se condição
     então <Bloco de comandos A>
     senão <Bloco de comandos B>
fim se
```

O bloco de comandos **A** só será executado se a condição for verdadeira e o bloco de comandos **B** só será executado se a condição for falsa. Entenda-se por bloco de comandos um ou mais comandos.

Exemplo:

```
algoritmo

        declare A, B, X numérico
        leia A, B
        se A > B
             então X ← A + 2
             senão X ← B - 1
       fim se
             escreva A, B, X
fim algoritmo
```

11 – ESTRUTURA DE REPETIÇÃO: A estrutura de repetição permite que um bloco de comandos seja executado repetidamente até que uma determinada condição de interrupção (condição de parada) seja satisfeita:

- Estrutura de repetição **Enqto**:

Sintaxe:

```
Enqto condição faça
          Bloco de comandos
fim enqto
```

Esta estrutura irá executar o bloco de comandos um número indeterminado de vezes, até que a condição de parada seja satisfeita finalizando a estrutura de repetição.

Exemplo:

```
algoritmo
        declare I, NÚMERO, SOMA numérico
        leia NÚMERO
        enqto NÚMERO  ≠ 0 faça
             SOMA ← SOMA + NÚMERO
             leia NÚMERO
        fim enqto
             escreva SOMA
fim algoritmo
```

- Estrutura de repetição **Para**:

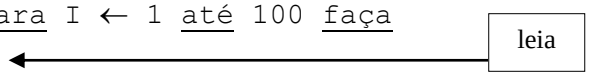
Sintaxe:

```
para variável ← início até fim faça
          Bloco de comandos
fim para
```

Esta estrutura irá executar o bloco de comandos um número determinado de vezes, até que a condição de parada seja satisfeita finalizando a estrutura de repetição.

Exemplo:

```
algoritmo
        declare I, SOMA numérico
        para I ← 1 até 100 faça
                SOMA ← SOMA + 1
        fim para
             escreva SOMA
fim algoritmo
```



Observações:

1. A leitura quando da utilização da estrutura **Para**, deve ser feita na primeira linha após a inicialização da estrutura para conforme indica a seta no exemplo acima;

2. Quando se estiver trabalhando com um intervalo determinado, recomenda-se a utilização da estrutura de repetição **Para**, pois ela conta o número de vezes que irá executar este bloco de comandos, automaticamente, utilizando-se de uma variável de controle.
   Esta variável é chamada de **variável contadora**, usada para indicar o número de vezes que está sendo executado o bloco. Caso contrário, utilize a estrutura de repetição **Enqto**.

3. Lembramos abaixo o caminho a ser seguido para resolução de um algoritmo:

    a) declaração
    b) inicialização
    c) leitura
    d) cálculo
    e) escrita dos resultados