COMANDO DE DECISÃO EM LÓGICA DE PROGRAMAÇÃO

Decision´s Command for Logic Programming

Daiana Graciela Galane RABELLO
Faculdade de Jaguariúna

Resumo: Este trabalho tem como objetivo central mostrar a utilização e aplicação do comando de decisão na lógica de programação através dos algoritmos, abordando suas formas de aplicação e as exemplificando de acordo com o problema proposto. Outro objetivo desejado ao final deste artigo é o aumento do conhecimento e o desenvolvimento do raciocínio lógico mais aprimorado a resolver problemas computacionais ou não computacionais, visando sempre uma resolução rápida e segura.
Palavras-chave: Algoritmo; Raciocínio Lógico; Decisão.

Abstract: The main purpose of this article is to present the application of the decision´s command applied for logical computing programming through the use of algorithms, discussing about the possible utilization forms and providing examples about the proposed topic. Another desired objective at the end of this article is the personal knowledge increase and the logical thinking development in order to solve computing or non computing issues, always seeking the faster and safer solution.
Key - words: Algorithm; Logical Thinking; Decision.

1. Introdução

A estrutura de decisão possibilita escolher uma ação dentre um conjunto de alternativas que foram especificadas, podendo então mudar o curso de execução de um programa.

Apesar de conseguir criar algoritmos sem a utilização das estruturas de controles, esses se tornam restritos, pois em algum momento podemos nos deparar com problemas onde deverá ser tratado um determinado dado (MANZANO, 2001).

A estrutura de decisão irá nos ajudar a desenvolver um algoritmo executando suas instruções de forma seqüencial e efetuar um tratamento mais adequado para as condições impostas nos problemas apresentados.

Estruturas de Decisão

As estruturas possuem diferentes formas de serem desenvolvidas e saber qual será a forma mais adequada para ser aplicada dependerá do problema proposto e das execuções para atender as condições apontadas.

Para melhor entendimento você deve ter aprendido a trabalhar com entrada, processamento e saída de dados e ter conhecimento dos operadores utilizados, como segue:

- Operadores Relacionais:

Ao ser utilizado a instrução se..senao..fim_se, ela implica na utilização de condições para verificar o estado de uma determinada variável quanto verdadeiro ou falso. (MANZANO, 2001, p.42)

Símbolos	Significado
>	Maior
<	Menor
>=	Maior/igual
<=	Menor/igual
==	Igual
!=	Diferente

- Operador lógico .e. (&&):

O operador do tipo **.e.** é utilizado quando dois ou mais relacionamentos lógicos de uma determinada condição necessitam ser verdadeiros. (MANZANO, 2001, p.48)

Condição x	Condição y	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

- Operador lógico .ou. (||).

O operador do tipo **.ou.** é utilizado quando pelo menos um dos relacionamentos lógicos (quando houver mais de um relacionamento) de uma condição necessita ser verdadeiro. (MANZANO, 2001, p.49).

Condição x	Condição y	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

2.1 Condicional Simples

O comando de decisão é especificado em uma expressão lógica que retorna um valor booleano (verdadeiro ou falso). Vamos realizar a aplicação utilizando como exemplo um algoritmo que leia dois valores inteiros e apresente o resultado da divisão do maior valor pelo menor valor.

Os algoritmos possuem 3 formas de serem representados para demonstrar o exemplo acima utilizaremos o fluxograma como demonstra a figura 1.

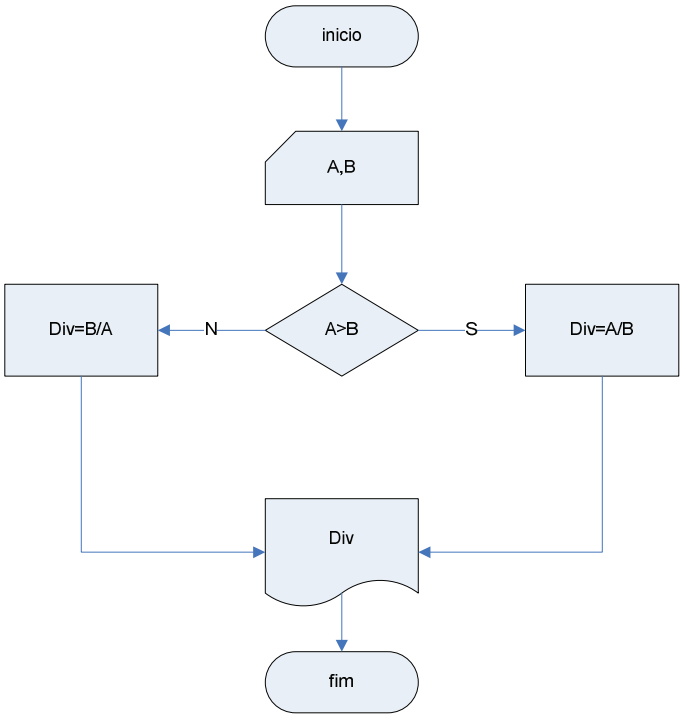

Figura 1 – Exemplo de Utilização de Algoritmo

A condição A>B é uma expressão lógica cujo resultado é um valor booleano. Se o valor de **A** for **maior** que o valor de **B**, o resultado será o valor verdadeiro e o cálculo **Div=A/B** será executado. Do contrário, ou seja, se o valor de **A** for **menor** que o valor de **B**, o resultado desta expressão lógica será falso e o cálculo **Div=B/A** será executado.

Utilizando um exemplo prático:

Assuma que A tenha o valor 8 e B tenha o valor 4. Como 8 é maior que 4, a expressão lógica A>B dá como resultado um valor verdadeiro e o cálculo será Div=8/4 será executado pelo computador. Suponhamos agora, que o valor de A seja 2. Neste caso, o resultado da expressão lógica será falso e o computador executará o calculo Div=4/2. Através do fluxograma na figura 1, observamos as duas possíveis opções que o computador pode tomar, dependendo do resultado avaliado na expressão lógica.

A escolha de uma ação é executada após avaliar uma condição, que nada mais é que uma expressão lógica que nos dá como resultado um valor lógico. A forma geral do comando de decisão em linguagem algorítmica é dada por:

```
SE ( condição ) ENTAO
   comando 1
SENAO
   comando 2
```

Se a condição for verdadeira, o comando 1 será executado e o comando 2 será ignorado. Do contrário, o computador executará o comando 2, ignorando o comando 1.

Como vimos no exemplo anterior, que compara os valores de A e B para verificar qual o maior entre eles, em notação algorítmica ficaria:

```
Início
   A,B:inteiro
   Div: real
   escreva("Digite dois valores:");
   leia( A,B );
     se ( A>B ) entao
          Div=A/B;
   senao
          Div=B/A;
   escreva("Resultado",Div);
```

```
fimse;

fim.
```

O comando SE pode ser utilizado sem o SENAO deixando assim o comando simplificado, conforme abaixo:

SE (condição) **ENTAO**

comando 1

O comando 1 apenas será executado se o resultado da condição for verdadeira. Como exemplo, iremos solicitar um valor inteiro fornecido pelo usuário e fazer a sua apresentação se o valor NÃO for maior que 5.

```
inicio;
A:inteiro
escreva("Digite um valor:");
leia(A);
    se (A<5)
        escreva("Valor menor:",A);
    fimse;
fim.
```

Comandos Ses Aninhados

Quando um comando SE é executado dentro de outro comando SE, isso significa que o comando interno SE, está aninhado. O exemplo abaixo mostra uma utilização de comandos ses e a aplicação do aninhamento ou encadeamento (MANZANO, 2001):

```
inicio;
A:inteiro
se(A>=0) entao
   se(A<=100) entao
escreva("Digitou um entre 0 e 100");
fimse;
fim.
```

Condicional com Seqüência de Comandos

Uma seqüência de comandos nada mais é que um conjunto de comandos delimitados por um determinado símbolo. Na linguagem algorítmica vamos delimitar o bloco de comandos pelos símbolos { (abre chave) indicando o início do bloco e } (fecha chave) para indicar o final do bloco (MIZRAHI, 1990).

Podemos ter um conjunto de instruções sob um comando de decisão. Podendo ser representada pela forma abaixo:

```
SE  ( condição ) ENTÃO
  {
    comando 1.1
    comando 1.2
    comando 1.3
  }
SENAO
```

```
{
   comando 2.1
   comando 2.2
   comando 2.3
}
```

Modificando o primeiro exemplo utilizado iremos aplicar um algoritmo que se o valor de A for maior que B efetuar a soma e subtração, caso contrário se B for maior que A executar a multiplicação e a média.

```
inicio;
A,B,R,R1,R2:inteiro
R3:real
leia(A,B);
se(A>B)
        R=A+B;
        R1=A-B;
escreva(R,R1);
senao
        R2=A*B;
        R3=(A+B)/2;
escreva(R2,R3);
fimse;
```

fim.

Portanto, se a condição for verdadeira, a primeira seqüência de comandos será executada. Do contrário, a segunda seqüência de comandos será executada. Isto se faz necessário, pois se não utilizarmos o conjunto de instruções como um bloco de comandos, apenas a primeira instrução estaria sob o comando de decisão, estando as demais instruções independentes do comando de decisão.

Considerações

Com a utilização da Estrutura de Decisão podemos criar algoritmos cada vez mais elaborados e realizar diferentes tipos de tratamentos de dados dependendo é lógico de suas condições. Existem outras formas de se tratar um mesmo problema utilizando outras formas de Estruturas como a de Seleção e Repetição, juntamente com a forma vista neste artigo, ou seja, utilizando uma encadeada à outra. À medida que desenvolvermos nosso conhecimento em lógica de programação teremos cada vez mais raciocínio lógico necessário que nos proporcionará boas condições. Isso vale também para aplicarmos em nossa vida, pois você poderá utilizar seus conhecimentos e encontrar soluções rápidas caso tenha um problema para analisar, mesmo que estes não sejam resolvidos por meio de algoritmos computacional.

Referências Bibliográficas

MANZANO, J. A. N. G. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 11. ed. São Paulo: Érica, 2001.

MIZRAHI, V. V. **Treinamento em Linguagem C: Curso Completo Módulo 1**. São Paulo: McGraw-Hill, 1990.