



Lógica de Programação



Wagner Kirmse Caldas

Objetivos

- Solucionar problemas computacionais através de uma linguagem algorítmica.
- Implementar algoritmos através de linguagens de programação.

Ementa

- Algoritmos
- Tipos de dados
- Variáveis
- Atribuições e expressões
- Estruturas de controle
- Funções e Procedimentos
- Tipos estruturados básicos
- Ambientes de desenvolvimento e programação

Bibliografia

- Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados. FORBELLONE, A. L. V., EBERSPACHER, H. F. Editora Makron, 2000.
- C Completo e Total. SCHILDT, H. Editora Makron, 1997.
- Estruturas de Dados Usando C. TANENBAUM, A., LANGSAM, Y., AUGENSTEIN, M. J. Editora Makron, 1995.

Introdução

- Desde o início dos tempos o homem tem procurado criar máquinas que o auxiliem em suas atividades, sendo que o computador tem se mostrado uma das mais versáteis, rápidas e seguras.
- Para que um computador execute tarefas, é necessário que seja especificada uma seqüência de passos que o computador deverá seguir ao efetuar essas tarefas.
- Cada passo pode ser definido como uma **instrução** e a seqüência de passos especificada através de uma notação compreendida pelo computador tem o nome de **programa**.

Introdução

- O computador é capaz de auxiliar em qualquer tarefa que lhe seja solicitada, desde que ele receba instruções detalhadas.
- De forma resumida, pode-se dizer que o objetivo de um computador é a manipulação de dados, sendo que é através de programas que o computador consegue processar os dados.



Introdução

- Processar dados significa receber dados por um dispositivo de entrada (teclado, mouse, etc..), realizar operações com esses dados e gerar uma resposta que será expressa em um dispositivo de saída (impressora, vídeo, etc..).
- Desta forma um computador tem duas partes que trabalham juntas: o hardware (partes físicas) e o software (composto pelos programas).
- Para que um computador compreenda e execute esse um programa é necessário que este seja escrito usando uma linguagem que tanto o computador quanto o desenvolvedor entendam. Essa linguagem é chamada ***linguagem de programação***.

Lógica

- A Lógica ensina a colocar “Ordem no Pensamento”, ou seja, usar corretamente as leis do pensamento.
- Lógica é a arte de pensar corretamente.
- Sempre que pensamos, falamos ou escrevemos, utilizamos a Lógica. Então pode-se perceber a importância da Lógica na nossa vida, não só na teoria, como também na prática, pois quando é necessário pensar, falar ou escrever corretamente, precisamos colocar “Ordem no Pensamento”, isto é, utilizar a Lógica.

Lógica

- Exemplo
 - A gaveta está fechada.
 - A bala está na gaveta.
 - Preciso primeiro abrir a gaveta, para depois pegar a bala.
- Com esse exemplo podemos perceber o quanto utilizamos a lógica no nosso dia a dia sem nos atermos a isso. Nós a utilizamos tão cotidianamente que não prestamos atenção para utilizá-la, utilizando-a naturalmente.
- Da mesma forma, a lógica na programação significa o uso correto das leis do pensamento, da “ordem da razão” e do processo de raciocínio e simbolização formais na programação de computadores, para resolvermos com qualidade os problemas que se deseja programar.

Algoritmo

- *Algoritmo* é uma seqüência de passos que visam atingir um objetivo bem definido. [Forbellone].
- *Algoritmo* é uma seqüência finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional. Essa seqüência de passos precisa ser executada em uma “ordem”, portanto, precisamos utilizar a lógica.
- A **seqüenciação** é uma convenção com o objetivo de reger o fluxo de execução, determinando qual operação vem a seguir.

Algoritmo

- Exemplo:

- **Trocar uma lâmpada**

- pegue uma escada
 - posicione-a embaixo da lâmpada
 - busque uma lâmpada nova
 - suba na escada
 - retire a lâmpada velha
 - coloque a lâmpada nova.

Algoritmo

- Involuntariamente, já seguimos uma determinada **seqüência** de ações que, representadas neste algoritmo, fazem com que ele seja seguido naturalmente por qualquer pessoa, estabelecendo um **padrão de comportamento**.
- Reexaminando o algoritmo anterior, notamos que ele tem um objetivo bem definido: trocar uma lâmpada. Porém, e se a lâmpada não estiver queimada? O algoritmo faz com que ela seja trocada do mesmo modo, portanto, o algoritmo não prevê essa situação.

Algoritmo

- Para solucionar esse problema, podemos efetuar um teste, verificando se a lâmpada está ou não queimada:
 - pegue uma escada
 - posicione-a embaixo da lâmpada
 - busque uma lâmpada nova
 - ligue o interruptor
 - se a lâmpada não acender, então:
 - suba na escada
 - retire a lâmpada velha
 - coloque a lâmpada nova
- Agora estamos ligando algumas ações à condição, ou seja, as três últimas ações só serão executadas se a lâmpada velha estiver queimada. O que ocorreu neste algoritmo foi a inclusão de um teste seletivo (condição) que determina qual ou quais ações serão executadas, dependendo do resultado da inspeção da condição.

Algoritmo

- Este algoritmo é funcional, porém pode ser otimizado, uma vez que buscamos uma escada e uma lâmpada sem saber se serão necessárias. Mudemos então o teste condicional para o início da sequência de ações:
 - ligue o interruptor
 - se a lâmpada não acender, então:
 - pegue uma escada
 - posicione-a embaixo da lâmpada
 - busque uma lâmpada nova
 - suba na escada
 - retire a lâmpada velha
 - coloque a lâmpada nova.

Algoritmo

- Existem muitas formas de resolver um mesmo problema, afinal cada pessoa pensa e age de maneira diferente. Isto significa que para este mesmo problema poderíamos ter diversas soluções diferentes e corretas, e portanto somente o bom senso é capaz de indicar qual delas é a mais adequada.
- Esta solução é adequada, porém, não prevê a possibilidade de que a lâmpada nova não funcione e, portanto, não atinja o objetivo nesta situação específica.

Algoritmo

- Devemos ressaltar que qualquer pessoa seria capaz de resolver o problema na prática, envolvendo inclusive circunstâncias inusitadas que pudessem surgir. Todavia, um computador não tem conhecimento prévio e nem adquire experiências, o que implica que devemos determinar em detalhe todas as ações que ele deve executar.
- Tal atividade é realizada pelos programadores que podemos chamar de construtores de algoritmos.
- A representação que utilizamos para os algoritmos até agora foi a representação textual em português coloquial.

Algoritmo

- Algoritmos, porém, podem ser representados de outras maneiras como a representação gráfica em fluxograma tradicional.
- Essas representações gráficas têm algumas desvantagens como a necessidade do conhecimento das convenções para entendê-las.

Algoritmo

- Também é possível utilizarmos pseudocódigo (ou português estruturado), que consiste em analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para a resolução do problema.
- A passagem do algoritmo assim escrito para qualquer linguagem de programação é quase imediata, bastando conhecer as palavras reservadas da linguagem de programação que será utilizada.

Algoritmo

- Para se desenvolver um programa, algumas etapas essenciais devem ser seguidas:
 - Análise: estuda-se o enunciado do problema e extrai-se os dados de entrada, os dados de saída e o processamento (como transformar os dados que entraram na saída desejada).
 - Algoritmo: descrição narrativa, fluxograma ou português estruturado são utilizados para descrever o problema baseado nas soluções analisadas.
 - Codificação: transforma-se o algoritmo em códigos da linguagem de programação escolhida.

Tipos Primitivos

- Informação é a matéria-prima que faz com que seja necessária a existência dos computadores, pois eles são capazes de manipular e armazenar um grande volume de dados com alto desempenho.
- De acordo com o tipo de informação contida nesses dados, eles podem ser classificados em quatro tipos primitivos:
 - Inteiro
 - Real
 - Caracter
 - Lógico

■ Inteiro

- Real

- Toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativa, nula ou positiva). Podem possuir componentes decimais ou fracionários, e podem também ser positivos ou negativos.
- Exemplo: -1,70; 0,00; 100,59
Ela tem 1,73 metro de altura.

Tipos Primitivos

■ Caracter

- Toda e qualquer informação composta por um conjunto de caracteres alfanuméricos:
 - letras (A...Z, a...z)
 - números (0...9)
 - caracteres especiais (exemplos: # ? ~ % _ ! @)
- Este tipo de dados é também muitas vezes chamado de alfanumérico, literais, cadeia de caracteres, ou string (inglês). Usualmente os dados literais são representados nos algoritmos pela coleção de caracteres, delimitada em seu início e término com o caractere aspa ("). Diz-se que o dado do tipo caracter possui um comprimento dado pelo número de caracteres nele contido.
- Exemplo: "QUAL?" - literal de comprimento 5
 - "Use somente caneta!"
 - "Wagner Kirmse"

Tipos Primitivos

■ Lógico

- Toda e qualquer informação que pode apenas assumir duas situações (biestável).
- Estes tipos de dados são chamados booleanos, devido à significativa contribuição de Boole à área da lógica matemática.
- Exemplo: aberto ou fechado verdadeiro ou falso acesa ou apagada

Exercício

- Determine quais são os tipos de informação presente nas sentenças a seguir:
 1. A placa “Pare!” tinha 2 furos de bala.
 2. Josefina subiu 5 degraus para pegar uma maçã boa.
 3. Alberta levou 3,5 horas para chegar ao hospital onde concebeu uma garota.
 4. Maria pintou em sua camisa: “Preserve o meio ambiente”, e ficou devendo R\$ 100,59 ao vendedor de tintas.
 5. A porta está aberta.
 6. José recebeu sua 18º medalha por ter alcançado a marca de 57,3 segundos nos 100 metros rasos

Constantes

- Uma constante é um dado de determinado valor fixo que não se modifica ao longo do tempo, durante a execução do algoritmo (programa).
- Convencionaremos que as informações do tipo lógico poderão assumir um dos seguintes valores constantes: **verdade** (V) ou **falsidade** (F)
- Exemplo: 5, “Não fume”, 2527, -0.58

Variável

- Um dado é classificado como variável quando tem a possibilidade de ser alterado em algum instante no decorrer do tempo, ou seja, durante a execução do algoritmo em que é utilizado, o valor do dado sofre alteração ou o dado é dependente da execução em um certo momento ou circunstância.
- Exemplo: A cotação do dólar, o peso de uma pessoa

Constante x Variável

- Para ilustrar a diferença entre valores constantes e variáveis considere um algoritmo para calcular o valor da área de uma circunferência.
 - Fórmula: πr^2
 - π tem valor **constante** de **3,1416...**, independente da circunferência
 - O valor de r , que representa o **raio**, é dependente da circunferência que se está calculando. Logo é **variável** a cada execução do algoritmo.

Identificadores

- Nomes de informações de caráter variável.
- Os identificadores devem acompanhar as seguintes regras de formação:
 - Devem começar por um caractere alfabético
 - Podem ser seguidos por mais caracteres alfabéticos ou numéricos
 - Não devem ser usados espaços em branco e caracteres especiais
 - Não podem ser utilizadas as palavras reservadas de cada linguagem de programação

Declaração de Variáveis

- Nos computadores, dispositivos chamados de “**memória**” permitem que as instruções a serem executadas e os dados a serem manipulados sejam armazenados.
- Esta memória pode ser vista como um “armário” composto por muitas “gavetas” que são chamadas de **variáveis**.
- A designação de “variáveis” é devido ao fato de podermos substituir seu conteúdo quando necessário.

Declaração de Variáveis

- Todas as variáveis (gavetas) devem receber um nome para sua identificação, estes nomes são denominados **identificadores** (etiquetas ou rótulos)
- Cada variável (gaveta) pode guardar apenas um dado (objeto) de cada vez, sendo sempre de mesmo tipo escalar (material)
- **Declarar variáveis significa definir nomes para determinadas gavetas especificando qual o material dos objetos que lá podem ser armazenados.**
- Mais de uma variável (gaveta) não pode possuir o mesmo identificador (etiqueta), visto que ficaríamos sem saber qual variável utilizar (qual gaveta abrir)

Declaração de Variáveis

- Exemplo:

- X: **inteiro**

- Nome, Endereco, Data: **caractere**

- ABC, Peso, Dolar: **real**

- Resposta: **lógico**

- Resposta é o nome de um local de memória que só pode conter valores do tipo lógico (verdade ou falsidade)

- X é o nome de um local de memória que só pode conter valores do tipo inteiro

Exercícios

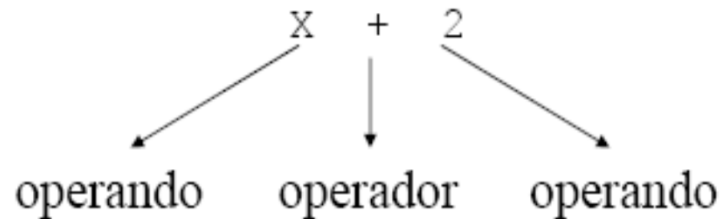
- 1. Assinale os identificadores válidos:
 - a)(x) b)U2 c)AH! d)“Aluno” e)#55 f)KM/L
 - g)UYT h)ASDRUBAL i)AB*C j)O&O
 - l)P{O} m)B52 n)Rua o)CEP p)dia/mes
- 2. Supondo que as variáveis NB, NA, NMat, SX sejam utilizada para armazenar a nota do aluno, o nome do aluno, o número da matrícula e o sexo, declare-as corretamente, associando o tipo primitivo adequado ao dado que será armazenado.

Respostas

- 1. Resposta:
 - b, g, h, m, n, o
- 2. Resposta:
 - NB: real; NA: caractere; Nmat: inteiro; SX: caractere

Expressões Aritméticas

- Denominamos expressão aritmética aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiro ou real).
- Exemplo:



Operadores aritméticos

Operador	Função	Exemplos
+	Adição	$2 + 3$, $X + Y$
-	Subtração	$4.0 - 2.0$, $N - M$
*	Multiplicação	$3 * 4$, $A * B$
/	Divisão	$10/2$, $X1/X2$

Potenciação e Radiciação

Operador	Função	Significado	Exemplos
<code>pot (x, y)</code>	Potenciação	x elevado a y	<code>pot(2,3)</code>
<code>rad (x)</code>	Radiciação	raiz quadrada de x	<code>rad(9)</code>

Operador de Resto e Quociente de Divisão Inteira

Operador	Função	Exemplos
mod	Resta da divisão	9 mod 4 resulta em 1 27 mod 5 resulta em 2
div	Quociente da divisão	9 div 4 resulta em 2 27 div 5 resulta em 5

Prioridade das Operações

Prioridade	Operadores
1ª	Parênteses mais internos
2ª	pot rad
3ª	* / div mod
4ª	+ -

- Em caso de empate (operadores de mesma prioridade, devemos resolver da esquerda para direita, conforme a sequência aritmética. Para alterar a prioridade da tabela pode-se utilizar parênteses mais internos.

Prioridade das Operações

■ Exemplo: a) $5 + 9 + 7 + 8/4$
 $5 + 9 + 7 + 2$
 23

b) $1 - 4 * 3/6 - \text{pot}(3, 2)$
 $1 - 4 * 3/6 - 9$
 $1 - 12/6 - 9$
 $1 - 2 - 9$
 -10

Exercícios

- Supondo que A, B e C são variáveis do tipo inteiro, como valores iguais a 5, 10 e -8, respectivamente, e uma variável real D, com o valor de 1,5, quais os resultados das expressões aritméticas a seguir?
 - a) $2 * A \bmod 3 - C$
 - b) $\text{rad}(-2 * C) \text{ div } 4$
 - c) $((20 \text{ div } 3) \text{ div } 3) + \text{pot}(8,2)/2$
 - d) $((30 \bmod 4) * \text{pot}(3,3))^* -1$
 - e) $\text{pot}(-C,2) + (D * 10)/A$
 - f) $\text{rad}(\text{pot}(A,B/A)) + C * D$

Respostas

■ a)

$$2 * A \bmod 3 - C$$

$$2 * 5 \bmod 3 - (-8)$$

$$10 \bmod 3 - (-8)$$

$$1 + 8$$

$$9$$

■ b)

$$\text{rad}(-2 * C) \text{ div } 4$$

$$\text{rad}(-2 * (-8)) \text{ div } 4$$

$$\text{rad}(16) \text{ div } 4$$

$$4 \text{ div } 4$$

$$1$$

Respostas

■ c)

$$((20 \text{ div } 3) \text{ div } 3) + \text{pot}(8,2)/2$$

$$(6 \text{ div } 3) + \text{pot}(8,2)/2$$

$$2 + \text{pot}(8,2)/2$$

$$2 + 64/2$$

$$2 + 32$$

$$34$$

■ d)

$$((30 \bmod 4) * \text{pot}(3,3))^* -1$$

$$((2) * 27) * -1$$

$$(2 * 27) * -1$$

$$54 * -1$$

$$-54$$

Respostas

e)

$$\text{pot}(-C, 2) + (D * 10)/A$$

$$\text{pot}(8, 2) + (1.5 * 10)/5$$

$$\text{pot}(8, 2) + 15/5$$

$$64 + 3$$

$$67$$

f)

$$\text{rad}(\text{pot}(A, B/A)) + C * D$$

$$\text{rad}(\text{pot}(5, 10/5)) + (-8) * 1.5$$

$$\text{rad}(\text{pot}(5, 2)) + (-8) * 1.5$$

$$\text{rad}(25) + (-8) * 1.5$$

$$5 + (-8) * 1.5$$

$$5 - 12$$

$$-7$$