

Algoritmos e Programação

Daniel de Sousa Moraes
danielmoraes14@gmail.com

Ementa

- Conceitos de Computação e Computadores
 - Origens da computação
 - A evolução dos computadores
 - Representação da informação
 - Arquitetura de um computador
- Introdução à Lógica de Programação
 - O que é Lógica
 - O que é um Algoritmo

Ementa

- Algoritmos e fluxogramas
 - Aplicabilidade, propriedades
 - Tipos de dados
 - Variáveis
 - Expressões aritméticas e lógicas
 - Atribuição
 - Estruturas de Controle
 - Estruturas de Dados
 - Modularização e Recursividade
 - Manipulação de Arquivos

Relembrando

- O que é um Algoritmo?
 - Sequência de passos que visam atingir um objetivo bem definido
 - Conjunto de regras para a solução de um problema
- Como precisamos especificar um sequência de passos, precisamos utilizar ordem, ou seja, “pensar em ordem”

Relembrando

- Um algoritmo representa o caminho de solução para um problema.
- Uma sequência de regras que devem ser executadas em uma ordem preestabelecida.
- Cada algoritmo possui um conjunto finito de regras que devem ter um significado e ser formalizadas segundo alguma convenção.

Relembrando

- Deve-se saber qual o problema a ser resolvido. Extrair todas as informações a respeito do problema (dados e operações), relacioná-las com o conhecimento atual.
- Abstração
- Definir claramente os passos para se chegar à solução
- Uma vez concebida uma solução algorítmica, deve-se ser possível traduzi-la para qualquer linguagem de programação

Aplicabilidade

- Existe um algoritmo em toda tarefa, independentemente de ela ser relacionada a um programa de computador
- Exemplo não computacional: RECEITA
- Exemplo computacional: MDC

Propriedades de um algoritmo

- Todo algoritmo deve possuir:
 - Valores de entrada: 0 ou mais entradas
 - Valores de saída: 1 ou mais saídas que simbolizam o resultado
 - Finitude: todo algoritmo deve ser finito.
 - Passos elementares: operações elementares que crie diferenças na interpretação
 - Correção: as saídas devem ser coerentes com as entradas.

Fluxogramas

- Representação gráfica de um algoritmo
- A partir do símbolo Início, permite a execução das instruções contidas nos símbolos subsequentes sem desvio algum na direção até alcançar o símbolo fim.

Fluxogramas - Símbolos



FLUXO DE DADOS

Indica o sentido do fluxo de dados. Conecta os demais símbolos



TERMINAL

Indica o INÍCIO ou FIM de um processamento
Exemplo: Início do algoritmo



PROCESSAMENTO

Processamento em geral
Exemplo: Cálculo de dois números



ENTRADA/SAÍDA (Genérica)

Operação de entrada e saída de dados
Exemplo: Leitura e Gravação de Arquivos



DESVIO (conector)

Permite o desvio para um ponto qualquer do programa

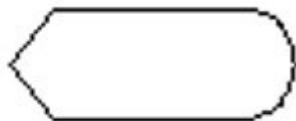
Fluxogramas - Símbolos



ENTRADA MANUAL

Indica entrada de dados via Teclado

Exemplo: Digite a nota da prova 1



EXIBIR/SAÍDA

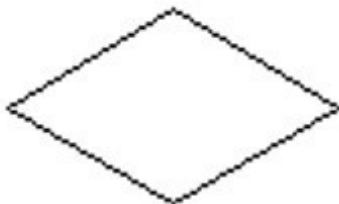
Mostra informações ou resultados

Exemplo: Mostre o resultado do cálculo



DECISÃO

Permite elaborar processos de decisão



Fluxogramas - Exemplos

- Dado o valor do raio (r) de uma circunferência, elaborar um programa para calcular e imprimir sua área (A) e o seu comprimento (C). A fórmula da área do círculo é $A = \pi r^2$ e do comprimento é $C = 2\pi r$.
- Passo 1 – Símbolo terminal de Início
- Passo 2 – Leitura dos valores das variáveis: r
- Passo 3 – cálculo direto
- Passo 4 – Exibir o valor calculado de A e C

Exercícios

- Elaborar um programa para calcular e imprimir o volume (V) de uma esfera e a área (A) de sua superfície, dado o valor de seu raio (R). A fórmula do volume da esfera é $V = \frac{4}{3} \pi R^3$
- Faça um programa para calcular a média final de um aluno, supondo-se que há quatro notas bimestrais durante o ano e que esta é calculada através de uma média aritmética simples (todos os bimestres têm o mesmo peso).
- Faça um programa que calcule a média e imprima se o aluno está aprovado ou reprovado
- Faça um programa que calcule e imprima o MDC de dois números

Convenções para tipos de dados

- Dados:
 - Informações em estado primitivo que servem de base para a formulação de algoritmos que, por sua vez, geram informação útil para o usuário
 - Como a natureza do que é armazenado em memória muda, os dados possuem um **tipo** identificando que valores podem ser armazenados

Tipos de dados - Inteiro

- Representa o conjunto dos número inteiros.
 - Ex: 2, 10, -8, 43223
 - Utilização:
 - Número de alunos na sala,
 - Número de produtos no estoque

Tipos de dados - Real

- Representa o conjunto dos números reais.
 - Ex: 2, -3, 7.3, 43.212
 - Utilização:
 - Saldo bancário
 - Salário de um funcionário

Tipos de dados - Lógico

- Armazena verdadeiro ou falso (de acordo com a lógica de Boole)
- Valores possíveis: **verdadeiro** ou **falso**
 - Utilização:
 - Estado de funcionamento de um carro: ligado ou desligado
 - Condição de pagamento de um boleto: pago ou não pago

Tipos de dados - Caractere

- Armazena um caractere
- Exemplo: Letras (A-Z ou a-z), dígitos (0-9), ou outros símbolos (% \$ # *). É usado com apóstrofes. 'a', '9'.
 - Utilização:
 - Primeira letra do nome de fulano

Tipos de dados – Cadeia de Caractere

- Armazena um conjunto de caracteres
- As cadeias de caractere podem conter qualquer símbolo, inclusive os espaços.
- Exemplo: 'Bom dia'. 'Olá', '987'
 - Utilização:
 - Nome completo de um aluno
 - Telefone da empresa

Operadores Aritméticos

Operadores Aritméticos	
Adição	+
Subtração	-
Divisão	/
Multiplicação	*
Resto de divisão	%

Operadores Relacionais

DESCRIÇÃO	SÍMBOLO
igual a	=
maior que	>
menor que	<
maior ou igual a	>=
menor ou igual a	<=
diferente de	!=

Operadores Lógicos

- E – (and) - &&
 - Relação de conjunção
- Ou – (or) - ||
 - Relação de disjunção
- Não – (not) - !
 - Relação negação

Operadores Lógicos - AND

A	B	Resultado da Conjuncção
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

Operadores Lógicos - OR

A	B	Resultado da Disjunção
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

Operadores Lógicos - NOT

A	Resultado da Negação
Verdadeiro	Falso
Falso	Verdadeiro

Funções

- Similar a funções matemáticas
- Recebem um ou mais parâmetros, aplicam um determinado processamento e retornam um valor resultante.
- Sintaxe de funções:
<tipo de retorno> <nome> (<parametro1>, <parametro2>, ...)

Funções prédefinidas

Nome da função	Semântica
QUAD (x)	Quadrado de X
RAIZ (x)	Raiz quadrada de X
TRUNC (x)	Valor inteiro de X sem parte decimal
ARRED (x)	Valor inteiro mais próximo de X
ABS (x)	O valor de X sem sinal

Expressões – precedência

- $4+4 * 2$
- $(4+4) * 2$
- $(-4+4) * 2$

Expressões – precedência

- Similar a expressões algébricas
 - 1 – troca de sinal (“-” unário)
 - 2 – parênteses mais internos
 - 3 – funções
 - 4 – operadores aritméticos (*, /, abs, +, -)
 - 5 – operadores relacionais
 - 6 – operadores lógicos (not, and, or)

Exemplos de expressões

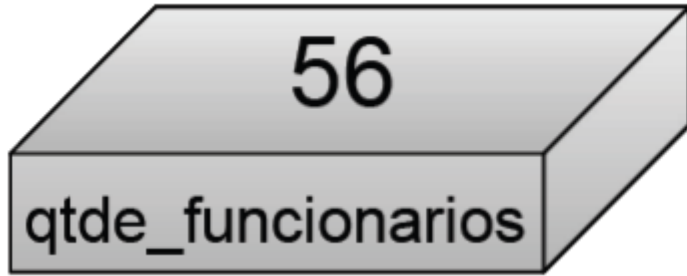
$$\frac{4+4}{2*2} + 3^2 \Rightarrow (4+4)/(2*2) + QUAD(3)$$

$$\frac{3*2+1}{9+5} + \sqrt{4} \Rightarrow (3*2+1)/(9+5) + RAIZ(4)$$

$$3 \leq x < 5 \vee x \leq 4 \Rightarrow (x \geq 3 \text{ E } x < 5) \text{ OU } x \leq 4$$

Variáveis

- Área da memória para armazenamento de dados



- Variáveis diferem sutilmente do mesmo conceito em matemática, porque em algoritmos seu valor podem variar com o tempo.

Nomenclatura de Variáveis

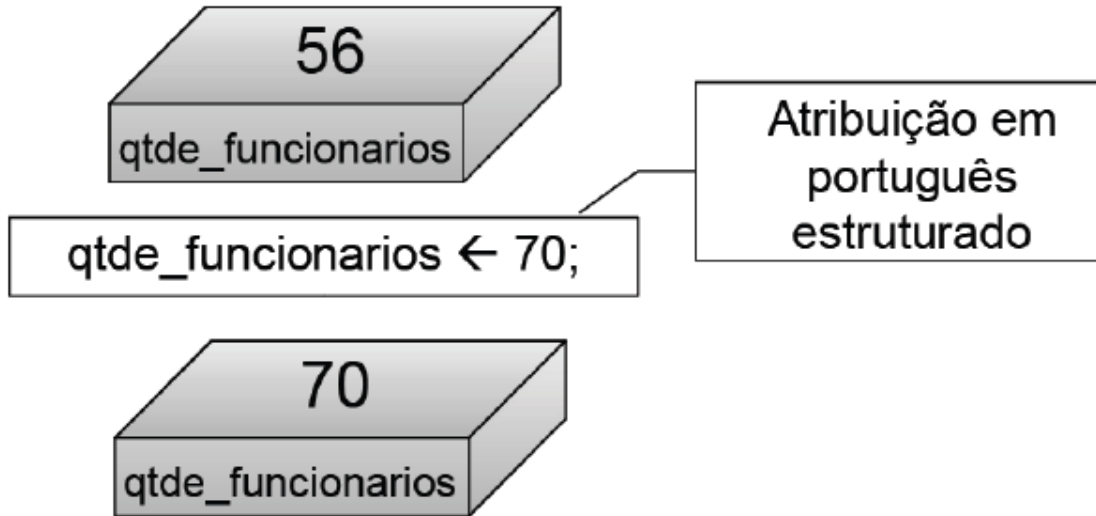
- Os nomes de variáveis devem começar com uma letra seguida de letras, dígitos ou do símbolo “underscore” ‘_’
- Toda variável possui um tipo associado
- Os nomes devem ser autoexplicativos
 - num_pessoas
 - idade
 - a,b, hga (são nomes ruins)

Declaração de Variáveis

- A declaração de uma variável serve identificá-la (nomear) e definir seu tipo.
- Sintaxe da declaração:
 <tipo> < **variável 1**>, <**variável 2**> ...
 ou
 <tipo> <**variável 1**>
 <tipo> <**variável 2**>
 ⋮

Atribuindo valores a Variáveis

- Uma variável armazena sempre **um único**



- Na atribuição, o novo valor substitui o valor anterior da variável

Atribuindo valores a Variáveis

- Atribuir valores de tipos é um erro
 inteiro quantidade
 quantidade ← 'Muitos'

Comando de Entrada de dados

- O comando **ler** é a forma mais rudimentar de entrada de dados por parte do usuário
- A entrada de dados em português estruturado é feita exclusivamente por este comando
- Sintaxe do comando:
ler <variável1>, <variável 2>, ...

Entrada de dados - exemplo

```
inteiro clientes  
ler clientes
```

```
string nome_cliente  
inteiro idade_cliente  
ler nome_cliente, idade_cliente
```

Saída de dados

- O comando **escrever** é a forma mais rudimentar de saída de dados
- A saída de dados em português estruturado é feita exclusivamente por este comando
- Sintaxe do comando:
escrever <variável1 ou expressão ou mensagem>

Entrada de dados - exemplo

escrever 'Olá. Bom dia

exibe na tela:

Olá. Bom dia

inteiro idade ← 20

string nome ← 'Daniel'

escrever 'O cliente', nome, ' tem ', idade, ' anos de idade.'

exibe na tela:

O cliente Daniel tem 20 anos de idades.

Sintaxe Geral de um português estruturado

Início

```
<tipo> <variavel1>, <variavel2>  
<tipo> <variavel3>  
//Isso é um comentário que será ignorado  
<comando 1>  
⋮  
<comando 2>
```

Fim

Referências

SOUZA, Marco Antonio Furlan de et al. **Algoritmos e lógica de programação: um texto introdutório para Engenharia**. 2ª ed rev. e ampl. -- São Paulo: Cengage Learning, 2011. ISBN 978-85-221-1129-9

FORBELLONE, André Luiz Villar EBERSPACHER, Henri Frederico. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3ª ed – São Paulo: Prentice Hall, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: lógica para desenvolvimento de programação de computadores** . 21. ed. São Paulo: Érica, 2008. 240 p. ISBN 978-85-7194-718-4(broch)

Carlos de Salles Soares Neto – Notas de Aula da Disciplina de Algoritmos I - UFMA