

Algoritmos e Programação I

Desenvolvimento de Algoritmos (termos técnicos)

Quando desenvolvemos algoritmos, trabalhamos, tipicamente, com sete tipos de componentes: **estruturas de dados, variáveis, constantes, instruções de manipulação de dados, expressões condicionais, estruturas de controle e módulos.**

Tipos de Dados

- ❖ O tipo de um dado define o conjunto de valores ao qual o valor do dado pertence, bem como o conjunto de todas as operações que podem atuar sobre qualquer valor daquele conjunto de valores.
- ❖ Por exemplo, o tipo de dados numérico pode ser imaginado como o conjunto de todos os números e de todas as operações que podem ser aplicadas aos números.
- ❖ Os tipos de dados manipulados por um algoritmo podem ser classificados em dois grupos:
 - ❖ Atômicos (numéricos, caracteres e lógico)
 - ❖ Complexos (cadeia de caracteres*)

Tipos de Dados

- ❖ O tipo numérico deve ser dividido em dois grupos:
 - ❖ inteiros (números naturais)
 - ❖ Reais (números reais)
- ❖ Distingue-se o tipo caracter do tipo cadeia através do uso de aspas simples e duplas respectivamente.
 - ❖ 'a' ≠ "a"
- ❖ O tipo lógico possui apenas os valores verdadeiro e falso. Nos algoritmos, estes valores podem ser escritos como V e F.

Variáveis

- ❖ Um algoritmo manipula dados, que podem ser dados **variáveis** ou **constantes**.
- ❖ Por exemplo, em um algoritmo que calcula a área de um círculo, o raio do círculo é um dado de entrada variável, pois o valor do raio pode variar de círculo para círculo. Por outro lado, o valor do número π , utilizado no cálculo da área do círculo*, é uma constante.
- ❖ Uma variável pode ser imaginada como um “depósito” para armazenar valores de dados, para o qual existe um nome, conhecido como **identificador**, e cujo conteúdo pode ser alterado pelo algoritmo.

* A área do círculo é dada por πr^2 ,
onde r é o raio do círculo.

Variáveis

- ❖ O identificador de uma variável deve ser distinto daquele das demais variáveis do algoritmo, pois é através do identificador da variável que o algoritmo a distingue das demais e tem acesso ao seu conteúdo.
- ❖ O ato de criar uma variável é conhecido como **declaração de variável**.
 - ❖ Cada variável utilizada em um algoritmo deve ter sido declarada antes de ser utilizada pela primeira vez.
 - ❖ Ao criarmos uma variável, temos de, explicitamente, associar-lhe um tipo de dados.

Variáveis

- ❖ Para se declarar uma variável, segue-se o formato abaixo*:

`<lista-de-variáveis> : <tipo-de-dado>`

- ❖ Onde lista de variáveis é uma lista de identificadores de variáveis separados por vírgula e tipo de dado é o nome do tipo ao qual as variáveis estarão associadas.

Ex.

`nota_conceito_aluno : caracter
n : real`

- ❖ Quando executamos um algoritmo em um computador, a cada variável corresponde uma posição distinta de memória.

Variáveis

- ❖ Uma constante faz exatamente o que o nome sugere: representa um dado cujo valor não muda durante todo o algoritmo.
- ❖ Ex*.

```
defina PI 3.14159  
defina MENSAGEM "A área do círculo é:"
```


Operadores

- ❖ Uma vez que declaramos as variáveis e constantes, temos a nossa disposição vários tipos de operadores, com os quais podemos atribuir valor a uma variável e manipular os valores armazenados em variáveis e constantes. Há três categorias básicas de operadores:
 - ❖ operadores de atribuição.
 - ❖ operadores aritméticos.
 - ❖ operadores de entrada e saída.

Operadores de Atribuição

- ❖ O ato de atribuir ou copiar um valor para uma variável é conhecido como **atribuição**.
- ❖ Utilizaremos o operador de atribuição (\leftarrow) como um símbolo para esta operação.
- ❖ Ex.

$x \leftarrow 4$

$y \leftarrow 'a'$

Operadores Aritméticos

- ❖ Os operadores aritméticos básicos são quatro:
 - ❖ Adição, representado pelo símbolo +;
 - ❖ Subtração, representado pelo símbolo -;
 - ❖ Multiplicação, representado pelo símbolo *; e
 - ❖ Divisão, representado pelo símbolo / e o resto por MOD.

- ❖ Desta forma, podemos criar expressões como:

$$a + b + c$$

$$a - b * c/2$$

Operadores Aritméticos

- ❖ Assim como na aritmética padrão, a **precedência** de operadores nas expressões aritméticas dos algoritmos também é governada pelo uso de parênteses.

$x \leftarrow a + b * c$ é **diferente** de $x \leftarrow (a + b) * c$

Entrada e Saída

- ❖ Qualquer algoritmo requer a obtenção de dados do “mundo” (entrada) e também um meio de comunicar ao “mundo” o resultado por ele obtido (saída).
- ❖ Para tal, existem duas operações, denominadas entrada e saída, realizadas, respectivamente, pelos operadores `leia` e `escreva`.

leia <lista de variáveis>

escreva <lista de variáveis e/ou constantes e/ou expressões >

Estrutura Geral de um Algoritmo

- ❖ Características que encontraremos nos algoritmos que estudaremos neste curso:
 - ❖ Linha de cabeçalho
 - ❖ Declaração de constantes e variáveis
 - ❖ Corpo do algoritmo
 - ❖ Linha final

Estrutura Geral de um Algoritmo

- ❖ Temos ainda alguns detalhes que servirão para deixar o algoritmo mais claro e mais fácil de ler:
 - ❖ Identação
 - ❖ Comentários
 - ❖ Linhas em branco

Estrutura Geral de um Algoritmo

```
// Função : algoritmo para calcular a área do círculo
// Autor : Fernando Maia da Mota
// Data : 03/2015
// Seção de Declarações
algoritmo "área do círculo"

var
valor_pi, area, raio : real
msg : caracter

inicio
// Seção de Comandos

//atribuição do valor de pi
valor_pi <- 3.14

//atribuição de mensagem final
msg <- "O valor da área do círculo é:"

// leitura do raio do círculo
leia(raio)

// cálculo da área do círculo
area <- valor_pi * raio * raio

// comunicação do resultado
escreval(msg, area)

finalgoritmo
```


Referências

- ❖ **SIQUEIRA, Marcelo F. Algoritmos e Estrutura de Dados. Mato Grosso do Sul: CCET/CPCX - UFMS, 2007.**