

Algoritmos e Lógica de Programação

Introdução e formas de representação

Agostinho Brito

Departamento de Engenharia da Computação e Automação
Universidade Federal do Rio Grande do Norte

15 de agosto de 2005

Fazendo café...

Descrição narrativa 1

- 1 Encha de água a chaleira.
- 2 Coloque a água para ferver.
- 3 Prepare o porta-filtro com o filtro sobre o bule.
- 4 Coloque duas colheres de sopa de pó de café no filtro.
- 5 Após a água ter fervido, acrescente aos poucos meio litro de água sobre o filtro.
- 6 Aguarde coar.
- 7 Adoce a gosto.

Descrição narrativa 2

- 1 Encha de água a chaleira.
- 2 Coloque a água para ferver.
- 3 Prepare o porta-filtro com o filtro sobre o bule.
- 4 Se houver café, coloque duas colheres de sopa de pó de café no filtro.
- 5 Senão, desligue o fogo, vá comprar café e repita os itens 2, 3 e 5.
- 6 Após a água ter fervido, acrescente aos poucos meio litro de água sobre o filtro.
- 7 Aguarde coar.
- 8 Adoce a gosto.

Atividade prática

Preparar um bolo de festa

- Preparar massa do bolo.
- Preparar recheio.
- Preparar cobertura.
- Juntar tudo.

Atividade prática

Preparar um bolo de festa

- Preparar massa do bolo.
- Preparar recheio.
- Preparar cobertura.
- Juntar tudo.

Atividade prática

Preparar um bolo de festa

- Preparar massa do bolo.
- Preparar recheio.
- Preparar cobertura.
- Juntar tudo.

Atividade prática

Preparar um bolo de festa

- Preparar massa do bolo.
- Preparar recheio.
- Preparar cobertura.
- Juntar tudo.

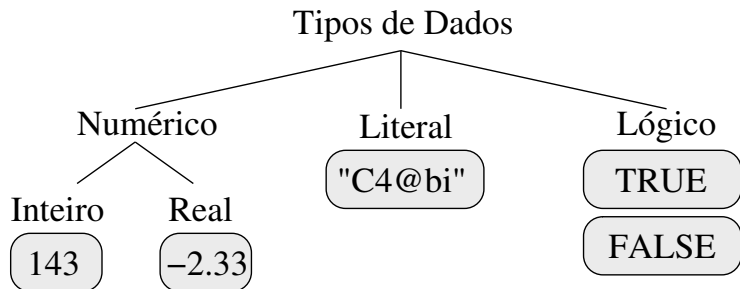
Estruturação de um programa

- ❶ Denominação do programa
- ❷ Declaração e classificação de variáveis
- ❸ Declaração de constantes
- ❹ Início do bloco principal
 - ❶ Iniciar as variáveis
 - ❷ Solicitar a entrada de dados ao usuário
 - ❸ Entrada de dados
 - ❹ Processamento
 - ❺ Saída de informações
- ❺ Fim do bloco principal

Listing 1: Ponderação de um número

```
#include <stdio.h>
int main(void){
    double X, Y;
    Y = 0;
    printf("Digite_X:_");
    scanf("%d", &X);
    Y = 1.3 * X;
    printf("Valor_de_Y:_%d\n", Y);
    return(0);
}
```

Tipos de dados



Tipos de dados numéricos

- Inteiros: não possuem componentes decimais ou fracionários, podendo ser positivos ou negativos.

Exemplos

- 13 ou +13: inteiro positivo.
- 0: inteiro.
- 13: inteiro negativo.

- Reais: podem possuir componentes decimais ou fracionários, e podem também ser positivos ou negativos.

Exemplos

- 24.01: número real positivo com duas casas decimais.
- 144.: número real positivo com zero casas decimais.
- 13.3: número real negativo com uma casa decimal.
- 0.0: número real com uma casa decimal.
- 0.: número real com zero casas decimais.

Tipos de dados literais

- O tipo de dado literal é constituído por uma sequência de caracteres contendo letras, dígitos e/ou símbolos especiais. Também chamado de alfanumérico, cadeia de caracteres ou *string*.
- Usualmente, os dados literais são representados nos algoritmos pela coleção de caracteres, delimitada em seu início e término com o caractere aspas (");
- Diz-se que o dado do tipo literal possui um comprimento dado pelo número de caracteres nele contido.
- Cuidado! "1.2" \neq 1.2.

Exemplos

string	comprimento
"QUAL ?"	6
" "	1
"qUaL ?!\$"	8
"AbCdefGHi "	9
"1-2+3="	6
"0"	1
"1.2"	3

Tipos de Dados Lógicos (booleanos)

O tipo de dados lógico é usado para representar dois únicos valores lógicos possíveis: verdadeiro e falso. É comum encontrar-se em outras referências outros tipos de pares de valores lógicos como 1/0, TRUE/FALSE.

Variáveis

Variáveis são utilizadas para referenciar dados armazenados na memória.

- A todo momento durante a execução de qualquer tipo de programa os computadores estão manipulando informações representadas pelos diferentes tipos de dados descritos anteriormente.
- Para que não se “esqueça” das informações, o computador precisa guardá-las em sua memória.
- Basicamente, uma variável possui três atributos: um nome ou identificador, um tipo de dado associado à mesma e o conteúdo por ela guardada.

Definição de variáveis em algoritmos

- A memória dos computadores é composta por células numeradas ordenadamente denominadas bytes, (ou octetos). Cada byte é constituído por 8 bits.
- Cada tipo de dado requer um número diferente de bytes para armazenar a informação representada por ele na memória. Esta quantidade também pode variar em função do tipo de computador considerado.
- Uma variável é uma entidade dotada de um nome para diferenciá-la das demais e um tipo de dado que define o tipo de informação que ela é capaz de guardar. Uma vez definidos, o nome e o tipo de uma variável não podem ser alterados no decorrer de um programa. Por outro lado, a informação útil da variável é objeto de constante modificação durante o decorrer do programa, de acordo com o fluxo de execução do mesmo.

Regras para criar identificadores

- O primeiro caractere deve ser uma letra ou o caractere underline "_".
- Os nomes devem ser formados por letras [a-z][A-Z], números [0-9] e "_".
- Não podem haver espaços em branco entre os caracteres do identificador.
- Os nomes escolhidos devem ser explicativos do seu conteúdo.
- Nomes longos dificultam a codificação. Os nomes não devem exceder o total de caracteres permitidos pela linguagem.
- Não deve-se usar acentuação ou cedilha.
- Os identificadores não podem ser os mesmos de palavras reservadas da linguagem de programação.

Exemplos

Salario	correto
for	errado
A CASA	errado
SAL/HORA	errado
SAL_HORA	correto
_DESCONTO	correto

Definição de variáveis em algoritmos

- Todas as variáveis devem ser definidas antes de serem utilizadas.
- Este procedimento necessário para permitir que o compilador reserve um espaço na memória para as mesmas.
- Sintaxe: <TIPO_DA_VARIÁVEL> identificador

Exemplos

```
int idade;  
float altura;  
boolean sexo;
```

Expressões

- Espécie de fórmula matemática, onde um conjunto de variáveis e constantes numéricas relacionam-se por meio de operadores aritméticos compostos que, uma vez avaliada, resulta num determinado valor.
- Operadores são elementos funcionais que atuam sobre operandos e produzem um determinado resultado.
- De acordo com o número de operandos sobre os quais atuam, os operadores podem ser classificados em **binários**, **unários** e **relacionais**.

```
area = 0.5*base*altura
```

0.5, base, altura	operandos
*	operador binário
=	operador relacional

- Podem ser literais: `eletro = "refrigera"+"dor"`
- Podem ser lógicas: `teste = TRUE & FALSE (& - "E" lógico).`

Expressões

- Regras são essenciais para a correta avaliação de expressões.
- Operadores de maior prioridade devem ser avaliados primeiro. Em caso de empate, a avaliação se faz da esquerda para a direita.
- O uso de parênteses em sub-expressões força a avaliação das mesmas com maior prioridade.
- Os diversos tipos de operadores devem ser avaliados na seguinte seqüência dentro de uma expressão complexa: primeiro os aritméticos e literais; em seguida, os relacionais e, por último, os lógicos (e/ou).