

AULA 08 – PARADIGMAS DE PROGRAMAÇÃO

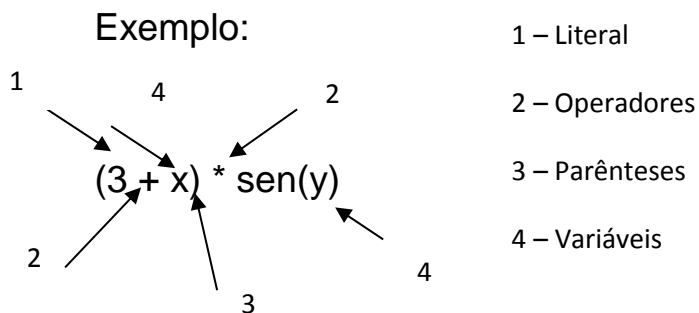
06/03/2013

PROFESSOR GLAUBER FERREIRA CINTRA
SEMESTRE 2012.2 – ENGENHARIA DE COMPUTAÇÃO IFCE

EXPRESSÕES E OPERADORES

Uma expressão denota uma sequência de operações que quando executadas resultam em um valor.

Uma expressão é constituída de operadores, variáveis, literais, parênteses e chamadas a funções.



Os parênteses servem para alterar a ordem com que as operações contidas em uma expressão serão executadas.

As operações dentro de um parêntese são executadas antes das operações que estão fora dos parênteses.

A ordem com que as operações contidas em uma expressão serão realizadas depende da **precedência** dos operadores. Os operadores de maior precedência são avaliados antes dos operadores que possuem menor precedência. Operadores distintos de mesma precedência são analisados da esquerda para a direita.

Se um mesmo operador se repete consecutivamente em uma expressão, ordem de execução depende da associatividade do operador. Se a associatividade for à esquerda, que é o caso mais comum, as operações serão executadas da esquerda para a direita. Se for a direita, que é um caso mais raro em linguagens de programação, as operações serão executadas da direita para a esquerda.

OPERADORES ARITMÉTICOS

- Adição (+)
- Subtração (-)
- Multiplicação (*)
- Divisão (/ , DIV)
- Resto da divisão (%, MOD)
- Exponenciação (** , ^)
- Menos unário (-)
- Mais unário (+)

OPERADORES LÓGICOS

11/03/13

- Conjunção (E): &&, AND
- Disjunção (OU): ||, OR
- Negação (Não): !, NOT
- Disjunção exclusiva: XOR (Pascal)
- Implicação: :- (Prolog)

OPERAÇÕES RELACIONAIS

- Igualdade: ==, =, eq*
- Desigualdade: !=, <>, ne*
- Maior: >, gt*
- Menor: <, lt*
- Maior ou igual: >=, ge*
- Menor ou igual: <=, le*
- Comparação: ⇔, cmp* (Perl)

* - Comparação entre strings: ordem alfabética.

MANIPULAÇÃO DE BITS

- Deslocamento de bits para esquerda (*Shift left*): \ll , SHL

Exemplo: $3 \ll 2 = 11_B \ll 10_B = 1100_B = 12$

- Deslocamento de bits para direita (*Shift right*): \gg , SHR

Exemplo: $13 \gg 2 = 1101_B \gg 10_B = 11_B = 3$

Note que:

$$a \ll b = a * 2^b$$

$$a \gg b = a / 2^b \text{ (Divisão inteira)}$$

- E Bit-a-Bit: $\&$

Exemplo: $5 \& 3 = 101_B \& 011_B = 011_B = 3$

- OU Bit-a-Bit: $|$

Exemplo: $5 | 3 = 101_B | 011_B = 111_B = 7$

- NÃO Bit-a-Bit: \sim

- XOR Bit-a-Bit: \wedge

Exemplo: $5 \wedge 3 = 101_B \wedge 011_B = 110_B = 6$

Aplicação XOR: Fazer swap de variáveis sem o uso de uma variável auxiliar.

MANIPULAÇÃO DE PONTEIROS

13/03/13

- Referência: &

Resulta no endereço do operando.

- Dereferência: *, ^

Resulta no conteúdo da região de memória apontada pelo operando.

- Dereferência do campo: ->, .

Resulta no valor do campo apontado pelo ponteiro.

Sintaxe: Ponteiro -> Campo

MANIPULAÇÃO DE STRINGS

- Concatenação: +, .
- Busca
- Substituição: ~s
- Translação: ~tr

OUTROS OPERADORES

- Enumeração: ..
- Replicação: x
- If ternário: ?:

Exemplo: $x = x < 0 ? -x : x$

- Atribuição: =, :=, is

Sintaxe: <lvalue> <operador de atribuição> <rvalue>

O valor do rvalue é armazenado na região de memória associada ao lvalue. O tipo de rvalue deve ser “compatível” com o tipo de lvalue. Se o rvalue for de tipo diferente, mas compatível com o tipo do lvalue o tipo do rvalue será convertido para o tipo lvalue.

A conversão de um tipo para um tipo mais restritivo para um tipo menos restritivo é chamado de **alargamento de tipo**. A conversão oposta é chamada de **estreitamento de tipo**. Algumas linguagens não fazem estreitamento de forma automática.

Algumas linguagens possuem o recurso da atribuição múltipla.

Exemplo:

$$(\$x, \$y) = (8, 5)$$
$$(\$a, @V) = (1 \dots 10)$$
$$(\$x, \$y) = (\$y, \$x)$$

OPERADORES ATRIBUITIVOS

- Incremento: ++
- Decremento: --
- Atribuição aditiva: +=
- Atribuição substrativa: -=
- Atribuição multiplicativa: *=
- Atribuição divisiva: /=
- Atribuição do resto: %=
- Atribuição do bit-a-bit: &=
- Atribuição do OU bit-a-bit: |=
- Atribuição do E bit-a-bit: &=
- Atribuição do XOR bit-a-bit: ^=
- Atribuição do Shift Left: <<=
- Atribuição do Shift Right: >>=