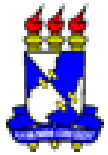
Expressões

Prof. Alberto Costa Neto
alberto@ufs.br

Linguagens de Programação



Departamento de Computação
Universidade Federal de Sergipe

Expressões

- Expressão é uma frase de um programa que ao ser avaliada retorna um valor
- Tipos
  - Literais
  - Agregados
  - Chamadas de Função
  - Operadores
  - Expressões Condicionais

Expressões

- Literais
  - valores fixos de algum tipo
  - Ex: 365, "casa", 'c', 2.3e-34

Expressões

- Agregados
  - constrói um valor composto de seus componentes
  - os valores dos componentes são determinados avaliando-se sub-expressões

Haskell `(a*2.0, b/2.0)` *(valor do tipo tupla Real x Real)*

ML
```
[31, if bisexto(anoatual) then 29 else 28, 31, 30, 31,
30, ..., 31]
```
(valor do tipo Lista de Inteiros)

Ada
```
anonovo := (y => anoatual+1, m => jan, d => 1);
```

Pascal
```
anonovo.y := anoatual+1;
anonovo.m := jan;
anonovo.d :=1;
```

não permite construção automática de agregados

Expressões

- Aritméticas

C

```
float f;
int num = 9;
f = num/6;      // 1.0
f = num/6.0;    // 1.5
```

- Relacionais
  - usadas para comparar os valores de seus operandos
- Booleanas
  - Realizam as operações de negação, conjunção e disjunção da álgebra de Boole

Expressões

- Binárias

C
```
void main() {
    int j = 10;
    char c = 2;
    printf("%d\n", j & c);     /* imprime  2 */
    printf("%d\n", j | c);     /* imprime 10 */
    printf("%d\n", j ^ c);     /* imprime  8 */
    printf("%d\n", j << c);    /* imprime 40 */
    printf("%d\n", j >> c);    /* imprime  2 */
}
```

Expressões

- Condicionais

ML
```
val c = if a > b  then a - 3 else b + 5
```

Java
```
max = x > y ? x : y;
par  = z % 2 == 0 ? true : false
```

  – Algumas LPs (como Pascal e ADA) não oferecem expressões condicionais – forçam o uso de comandos condicionais

ADA
```
if x > y then max := x; else max := y; end if;
```

Expressões

- Chamadas de Funções
  - Operador => nome da função
  - Operandos => parâmetros
  - Resultado => retorno da função

*Uma expressão que retorna (gera) uma função chamando uma função **desconto** ou **multa***

ML

```
val taxa =
  (if difPgVenc > 0 then desconto else multa) (difPgVenc)
```

C

```
double (*p)(double);
p = difPgVenc < 0 ? desconto: multa;
taxa = (*p) (difPgVenc);
```

Funcionalidade semelhante, mas usando ponteiros para funções

Expressões

- Com Efeitos Colaterais

C
```
x = 3.2 * ++c;
```

  - Podem gerar indeterminismo

C
```
x = 2;
y = 4;
z = (y = 2 * x + 1) + y;
```

Qual o valor de y?

Depende da implementação do compilador!

Expressões

- Referenciamento
  - Usadas para acessar o conteúdo ou retornar referência para variáveis ou constantes

C

```
*q                  = *q + 3;
const float pi      = 3.1416;
int raio            = 3;
float perimetro     = 2*pi*raio;
p[i]                = p[i + 1];
*q                  = *q + 3;
r.ano               = r.ano + 1;
s->dia              = s->dia +1;
t                   = &m;
```

Expressões de referenciamento do lado:
- esquerdo: retornam Endereço
- direito: retornam Conteúdo

Expressões

- Categóricas
  - Realizam operações sobre tipos de dados
    - Tamanho do Tipo

```c
float * p = (float *) malloc (10 * sizeof (float));
```

•Operador sizeof obtém o tamanho em bytes do tipo.
•A expressão inteira aloca um tamanho de memória para 10 elementos float

    - Conversão de Tipo

```c
float f;
int num = 9, den = 5;
f = (float)num/den;          // f = 1,8
```

Avaliação de Expressões

- Precedência de Operadores
  - Escolha inadequada pode afetar a redigibilidade

```
/* if a > 5 and b < 10 then */
if (a > 5) and (b < 10) then a := a + 1;
```

Pascal

- Associatividade de Operadores
  - Mesma Precedência: esquerda para a direita

```
x = a + b - c;
y = a < b < c;
```

C

  - Exceções à regra

```
a = b = c;
```

C

Avaliação de Expressões

- Com Curto Circuito
  - Resultado da expressão é determinado antes
  - Situação Potencial (uma das expressões retornasse 0)

C
```
z = (x - y) * (a + b) * (c – d);
```

  - Usado em Expressões Booleanas

Java
```
int[] a = new int [n];
i = 0;
while (i < n && a[i] != v) i++;
```

Curto-circuito quando v não existe no vetor a
--> i == n e fim
Em Pascal daria erro !!!

Sugestões de Leitura

- Concepts of Programming Languages (Robert Sebesta)
  - Capítulo 7
- Programming Language Concepts and Paradigms (David Watt)
  - Seção 2.6
- Linguagens de Programação (Flávio Varejão)
  - Seção 5.1