



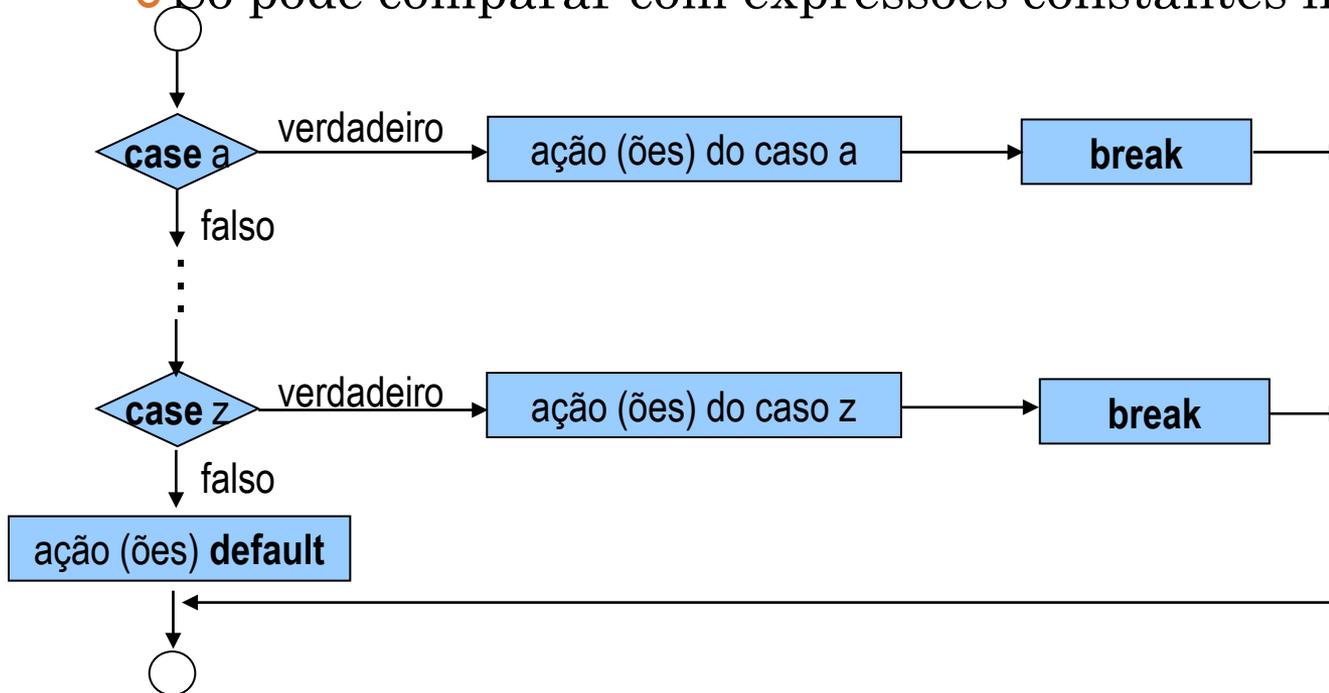
PROCESSAMENTO DA INFORMAÇÃO

LAÇOS E REPETIÇÕES

Prof. Dr. Francisco Isidro Massetto

ESTRUTURAS DE CONTROLE

- **switch** (*estrutura de seleção múltipla*)
 - Trata uma série de condições nas quais uma variável ou expressão particular é comparada com valores que ela pode assumir e diferentes ações são tomadas
 - `break` depois das instruções para cada `case`: faz com que o controle saia imediatamente da estrutura `switch`
 - Só pode comparar com expressões constantes integrais



ESTRUTURAS DE CONTROLE

- Algumas considerações sobre o switch
 - Só pode ser usado para comparar IGUALDADES
 - Outras operações de comparação (maior, maior ou igual, menor, menor ou igual, diferença) não são usadas
 - Aceita tipos específicos
 - Booleano, caractere e inteiros
 - Float e double não são aceitos
 - Uma vez uma comparação resulta em verdadeiro, o resultado (Verdadeiro) é propagado para todas as demais comparações.
 - Por isso da necessidade do break



ESTRUTURAS DE CONTROLE

○ Estruturas de repetição

- while

- Especifica que uma ação deve ser repetida enquanto alguma condição permanecer verdadeira

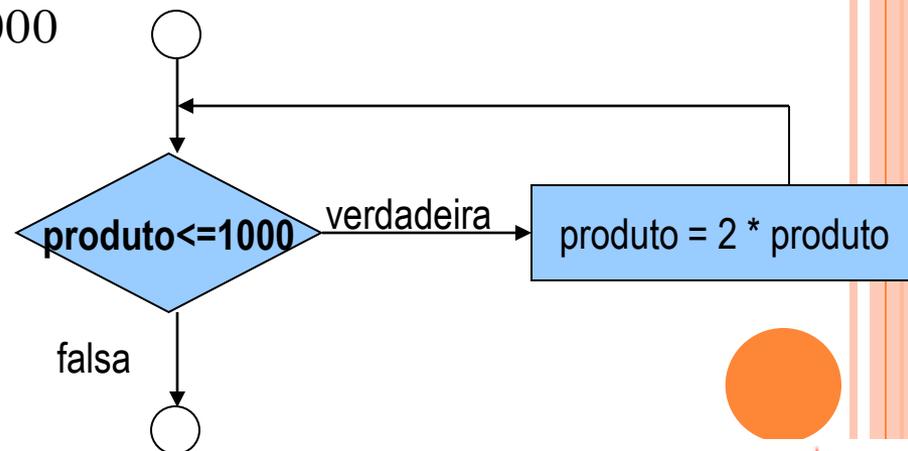
`while` (*condição*)

instrução

- Exemplo: segmento de programa projetado para encontrar a primeira potência de 2 maior que 1000

```
int produto = 2;
```

```
while (produto <= 1000)  
    produto = 2 * produto;
```



CUIDADO!!!!

- Apesar de sempre chamarmos a condição do while de “condição de parada”, na verdade queremos dizer “condição de continuidade”, pois o laço continua quando a condição for verdadeira
- O laço só termina quando a condição for falsa



ACUMULADORES E CONTADORES

- A idéia dos acumuladores é usar os conceitos de repetições para criar ferramentas para contar quantas vezes uma operação deve ser feita ou para acumular valores em uma variável.
- Um contador é no fundo, um acumulador que é incrementado (ou decrementado) de 1 unidade por vez



EXEMPLO DE ACUMULADOR

- Imagine um sistema de PDV (Ponto De Venda) de um supermercado. Como fazer um algoritmo para calcular o valor total da compra de um cliente, uma vez que os preços são informados para o sistema?
- Considere que a compra termina quando o operador do caixa digitar o valor 0.00



EXEMPLO DE ACUMULADOR

```
programa pdv
  declaracoes
    real precoProduto, totalCompra
    inteiro numItens
  inicio
    numItens <- 0
    totalCompra <- 0

    escreva("digite o preco do produto")
    leia(precoProduto)
    totalCompra <- totalCompra + precoProduto
    numItens <- numItens + 1
    enquanto (precoProduto != 0) faca
      escreva("digite o preco do produto")
      leia(precoProduto)
      totalCompra <- totalCompra + precoProduto
      numItens <- numItens + 1
    fimenquanto
    escreva ("total da compra R$", totalCompra)
    escreva ("itens comprados ", numItens)
  fim
```



ESTRUTURAS DE CONTROLE

○ Exemplos:

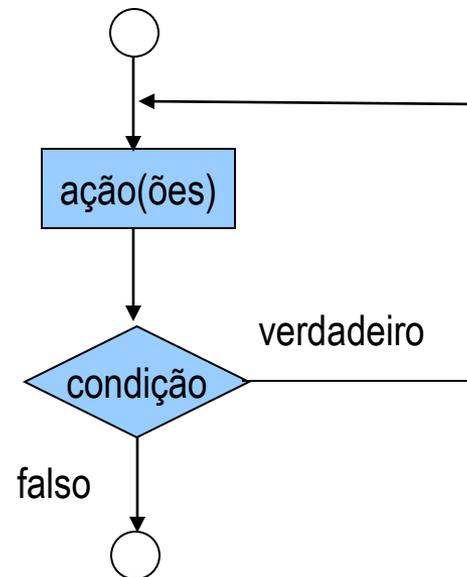
- Faça um algoritmo que receba um valor e imprima sua tabuada
- Faça um algoritmo que receba 2 valores inteiros: uma base e um expoente e calcule a potência
 - Antes, verifique se o expoente é positivo
- Faça um algoritmo que calcule a série harmônica até um determinado valor (n)
 - $S = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$



ESTRUTURAS DE CONTROLE

- `do/while`
 - Testa a condição de continuação do laço no final do laço
 - O corpo do laço será executado pelo menos uma vez

```
do {  
    instrução  
} while (condição);
```



ESTRUTURAS DE CONTROLE

- for
 - trata de todos os detalhes da repetição controlada por contador e exige o seguinte:
 - o *nome* de uma variável de controle;
 - o *valor inicial* da variável de controle;
 - o *incremento* (ou *decremento*) pelo qual a variável de controle é modificada a cada passagem pelo laço (cada iteração do laço); e
 - A condição que teste o *valor final* da variável de controle.

`for` (*expressão1*; *expressão2*; *expressão3*)

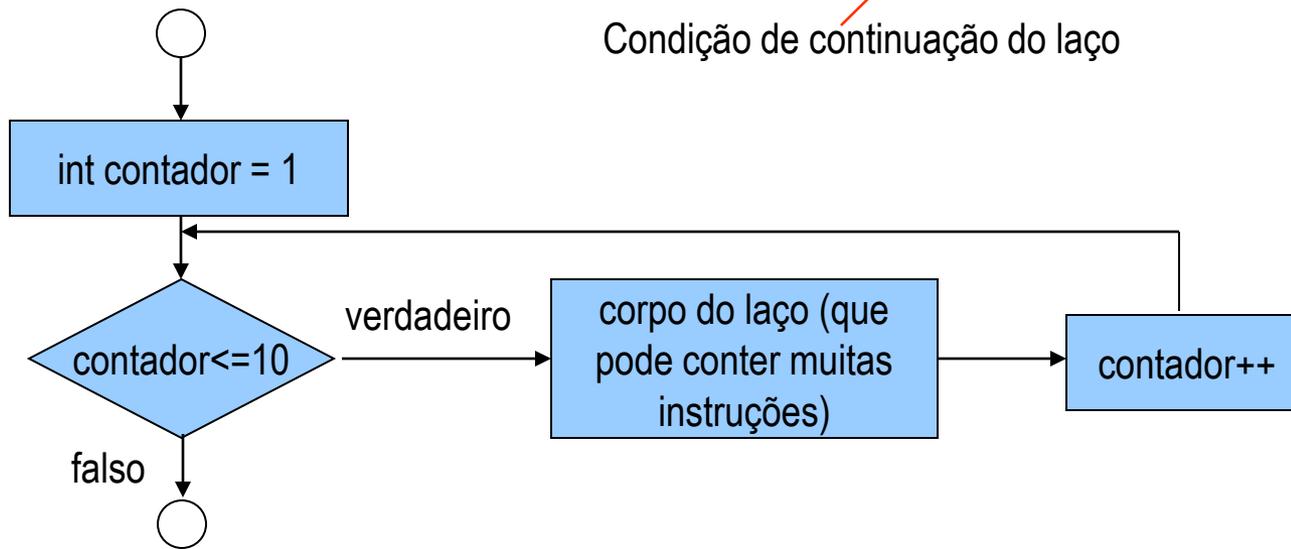
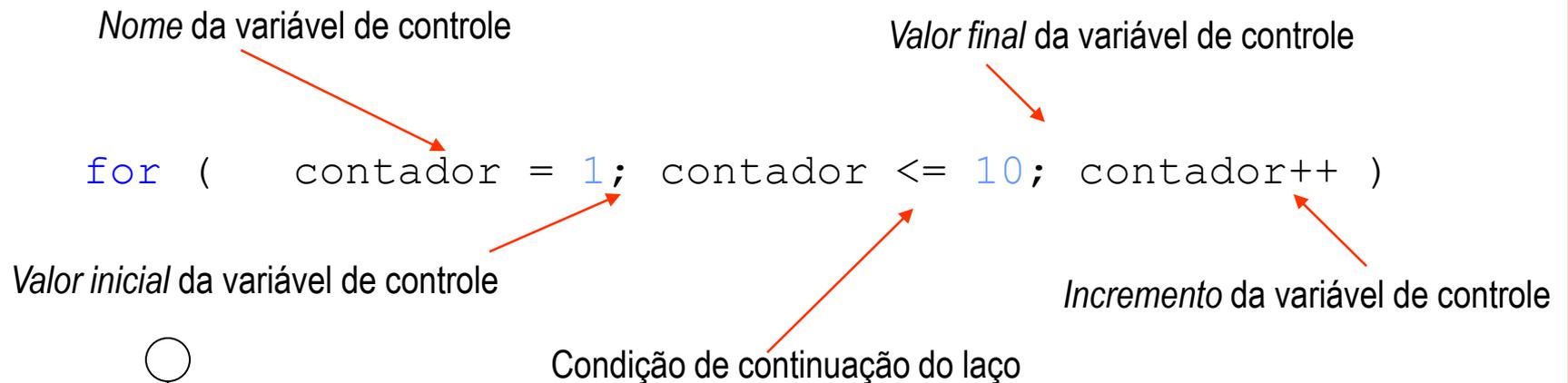
instrução

- *Expressão1*: inicializa a variável de controle do laço;
- *Expressão2*: é a condição de continuação do laço;
- *Expressão3*: incrementa a variável de controle, até que a condição de continuação do laço se torne falsa



ESTRUTURAS DE CONTROLE

- Componentes de um cabeçalho de **for** típico



ESTRUTURAS DE CONTROLE

- Exemplos com a estrutura **for**:
 - Variável de controle de 1 a 100 em incrementos de 1
`for (int i = 1; i <= 100; i++)`
 - Variável de controle de 100 a 1 em incrementos de -1
`for (int i = 100; i >= 1; i--)`
 - Variável de controle de 7 a 77 em incrementos de 7
`for (int i = 7; i <= 77; i += 7)`
 - Variável de controle de 20 a 2 em incrementos de -2
`for (int i = 20; i >= 2; i -= 2)`
 - Variável de controle assumir a seguinte seqüência de valores: 2, 5, 8, 11, 14, 17, 20.
`for (int j = 2; j <= 20; j += 3)`
 - Variável de controle assumir a seguinte seqüência de valores: 99, 88, 77, 66, 55, 44, 33, 22, 11.
`for (int j = 99; j > 0; j -= 11)`



ESTRUTURAS DE CONTROLE

○ **break**

- Quando executada em uma das instruções de repetição (**while**, **for**, **do/while** ou **switch**), causa saída imediata dessa estrutura
- A execução continua com a primeira instrução depois da estrutura

○ **continue**

- Quando executada em uma das instruções de repetição (**while**, **for**, **do/while**), pula qualquer instrução restante no corpo da estrutura e prossegue com o teste para a próxima iteração do laço
 - Nas estruturas **while** e **do/while**, o programa avalia o teste de continuação do laço imediatamente depois da instrução **continue** ser executada
 - Em estruturas **for**, a expressão de incremento é executada e depois o programa avalia o teste de continuação do laço



DIFERENÇAS ENTRE FOR/WHILE/DO..WHILE

○ While

- Executa no mínimo 0 vezes e no máximo infinitas vezes
 - Nunca executa → o teste é falso já de cara
 - Nunca termina → o teste NUNCA é falso

○ Do... While

- Executa no mínimo 1 vez e no máximo infinitas vezes
 - O teste é feito no final, mesmo que seja falso, o bloco vai executar 1 vez

○ For

- Usado em geral (não é regra) quando se sabe precisamente o número de repetições

