

COMANDOS ESTRUTURADOS – DE REPETIÇÃO (laços)



COMANDOS ESTRUTURADOS – DE REPETIÇÃO (laços)

Para resolver um problema, frequentemente deve-se repetir a mesma operação muitas vezes, e para isso deve-se usar estrutura de repetição (laço ou *loop*).

Neste sentido, há um conjunto de comandos será executado zero, uma ou mais vezes. Este conjunto de comandos chama-se corpo do laço.

Tipos de laços existentes:

1. **LAÇO CONTROLADO LOGICAMENTE:** o corpo do laço é repetido indefinidamente enquanto uma expressão booleana for satisfeita.
2. **LAÇO CONTROLADO POR CONTADOR:** o corpo do laço é repetido um número predeterminado de vezes.

Iteração = cada repetição do conjunto de comandos. Logo, cada execução do corpo do laço, juntamente com a condição de terminação do laço, é uma iteração.

LAÇOS CONTROLADOS LOGICAMENTE

1) REPITA-ATÉ

Formato: REPITA <comandos> ATE (condição de parada)	Exemplo: REPITA ESCREVA ("Sexo (F/M) : ") LEIA (sexo) ATE (sexo = "F" OU sexo = "M")
---	---

- muito adequado para validar dados de entrada
- os comandos serão executados pelo menos 1 vez (pois a condição de parada está no final do laço)
- a(s) variável(is) de controle do laço pode(m) ser de qualquer tipo primitivo

2) ENQUANTO-FAÇA

Formato: ENQUANTO (condição) FAÇA <comandos> FIMENQUANTO	Exemplo: ESCREVAL ("Informe x e y:") LEIA (x, y) ENQUANTO (x < y) FAÇA x <- x + 0.5 FIMENQUANTO
--	---

- adequado para repetir comandos um nº indeterminado de vezes (não se sabe exatamente quantas)
- os comandos do bloco podem não ser executados (pois a condição controladora está no início do laço)
- a(s) variável(is) de controle do laço pode(m) ser de qualquer tipo primitivo

LAÇO CONTROLADO POR CONTADOR

1) PARA-FAÇA

Formato:

PARA var DE valorInicial ATE valorFinal **PASSO** x **FACA**
 <comandos>

FIMPARA

O **PASSO** pode ser omitido quando for 1

Exemplos:

Com incremento (passo positivo):

PARA contador DE 1 ATE 10 **FACA**

ESCREVAL (contador)

// 1 2 3 4 5 6 7 8 9 10

FIMPARA

Com decremento (passo negativo):

PARA contador DE 10 ATE 1 **PASSO** -1 **FACA**

ESCREVAL (contador)

// 10 9 8 7 6 5 4 3 2 1

FIMPARA

Com incremento duplo:

PARA contador DE 1 ATE 9 **PASSO** 2 **FACA**

ESCREVAL (contador)

// 1 3 5 7 9

FIMPARA

- adequado para repetir comandos um nº determinado e conhecido de vezes (seja via valor constante ou informado pelo usuário anteriormente)
- a variável de controle do laço (contador nos exemplos acima) é automaticamente atualizada pelo laço conforme o passo indicado (que pode ser positivo ou negativo)
- a variável de controle do laço deve ser do tipo INTEIRO ou CHARACTER
- *valorInicial*, *valorFinal* e *PASSO* podem ser informados pelo usuário, para flexibilizar a execução do laço

COMPARAÇÃO ENTRE OS COMANDOS DE REPETIÇÃO

Comando	Expressão	Número de execuções	Condição de parada
ENQUANTO-FACA	Início	?	Expressão falsa
REPITA-ATE	Fim	No mínimo 1	Expressão verdadeira
PARA-FACA	Não tem	ABS (valorFinal - valorInicial + passo)	Variável = valorFinal

ATENÇÃO:

- Todo ENQUANTO-FACA pode ser convertido para REPITA-ATE e vice-versa
- Todo PARA-FACA pode ser convertido para ENQUANTO-FACA, **MAS** nem todo ENQUANTO-FACA pode ser convertido para PARA-FACA.

EXEMPLOS – Em VisuAlg:

ALGORITMO "ex_repita-ate"

// controle de programa via menu

VAR

opcao: INTEIRO

INICIO

REPITA

ESCREVAL("1 - Dizer olá!")

ESCREVAL("2 - Dizer oi! ")

ESCREVAL("0 - Sair do programa")

LEIA(opcao)

ESCOLHA (opcao)

CASO 1

ESCREVAL("Olá!")

CASO 2

ESCREVAL("Oi!")

FIMESCOLHA

ATE (opcao = 0)

FIMALGORITMO

ALGORITMO "ex_enquanto-faca"

// controle de erro numérico

VAR

erro, x, y: REAL

INICIO

ESCREVA ("Informe x e y:")

LEIA (x,y)

erro <- ABS(x - y)

ENQUANTO (erro > 0.0001) FACA

x <- x / 2 *// divisão real pois x e y já são tipo real*

y <- y / 2

erro <- ABS(x - y)

FIMENQUANTO

ESCREVAL ("Valores finais: x= ", x, " e y= ", y)

FIMALGORITMO

ALGORITMO "ex_para-faca"

// contagem regressiva

VAR

contador: INTEIRO

INICIO

PARA contador DE 10 ATE 1 PASSO -1 FACA

ESCREVAL ("Detonação em: ", contador)

FIMPARA

ESCREVAL ("BOOOM!!!")

FIMALGORITMO

COMANDOS ESTRUTURADOS DE REPETIÇÃO EM C++

LAÇOS CONTROLADOS LOGICAMENTE

1) do-while () => traduzindo: faça-enquanto

Formato: <pre>do { <comandos> } while (condição);</pre>	EXEMPLO: <pre>// validação de entrada: idade entre [18,25] do { cout << "Informe idade:"; cin >> idade; } while (idade < 18 or idade > 25);</pre>
---	---

- neste comando a condição é de continuidade (pois o do-while não é tradução de REPITA-ATE), portanto cuidado/atenção nas traduções de VisuAlg para C++.

2) while ()

Formato: <pre>while (condição) { <comandos> }</pre>	EXEMPLO: <pre>cout << "Informe x e y:" << endl; cin >> x >> y; while (x < y) { x = x + 0.5; // ou x+=0.5; }</pre>
---	---

- o laço while é a tradução literal do ENQUANTO-FAÇA.

LAÇO CONTROLADO POR CONTADOR

1) for

Formato <pre>for ([tipo] var = valor1; var <= valorN; var++) { <comandos> }</pre>
EXEMPLOS: <pre>for(contador=1; contador<=10; contador++) { // ++ INCREMENTO cout << contador << " "; // 1 2 3 4 5 6 7 8 9 10 } for(contador=10; contador>=1; contador--) { // -- DECREMENTO cout << contadora << " "; // 10 9 8 7 6 5 4 3 2 1 } for(contador=1; contador<=9; contador+=2) { // INCREMENTO DUPLO cout << contador << " "; // 1 3 5 7 9 } for(int i=0,j=5; i<j; i++, j--) { // USANDO 2 VAR cout << i << " " << j << " - "; // 0 5 - 1 4 - 2 3 }</pre>

- usando [tipo] var no for uma variável local de bloco será criada, conhecida e acessível apenas dentro do bloco (laço) – após o bloco ela é “destruída”.

- o for do C++ é diferente e bem mais flexível que o para-faça