

Aula 4

Estrutura de Repetição

Prof. Sandro de Araujo

Conversa Inicial

Essa aula apresenta a seguinte estrutura de conteúdo:

- Estrutura de repetição
- Estrutura de repetição *while* (enquanto)
- Estrutura de repetição *do-while* (repita-até)
- Estrutura de repetição *for* (para)
- Estrutura de repetição aninhada

- O objetivo desta aula é conhecer os principais conceitos e aplicações das estruturas de repetições, como: *while* (enquanto), *do-while* (repita-até) e *for* (para). Também é objetivo representá-los em fluxograma, pseudocódigo e linguagem C para resolver problemas computacionais

Estrutura de repetição

Estrutura de repetição

- Também é conhecida como laços (*loops*) a repetição de um conjunto de ações uma vez, várias vezes ou nenhuma vez
- Depende de uma condição verdadeira ou falsa
- Condicionada a um controle ou condição de parada

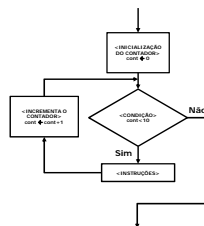
Quatro elementos fundamentais de uma estrutura de repetição

- Inicialização – determina a condição inicial da repetição
- Condição – é a expressão booleana que avalia após cada leitura do corpo e determina se uma nova leitura deverá ser feita ou se a estrutura de repetição pode ser encerrada

Quatro elementos fundamentais de uma estrutura de repetição

- Corpo – formado por todas as instruções que serão executadas repetidamente
- Iteração – é a repetição de um conjunto de instruções juntamente com a condição de terminação do laço

Exemplo - estrutura de repetição



Fonte: Elaboração própria (2018).

Como resultado do contador, teremos...

INÍCIO DO LAÇO

cont = 0

cont = 1

cont = 2

cont = 3

...

cont = 9

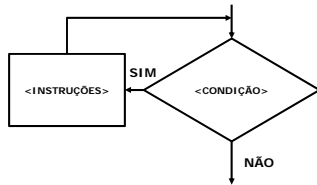
- Terminou de contar!

Estrutura de repetição *while* (enquanto)

Estrutura de repetição *while* (enquanto)

- Na estrutura de repetição *while* (enquanto), a execução de *loop* depende de uma condição que é verificada no início do laço
- Enquanto o resultado da condição for verdadeiro, o bloco de instruções é executado; caso contrário, ocorre o desvio para a primeira linha após esse bloco

**Exemplo - estrutura de repetição
while (enquanto)**



Fonte: Elaboração própria (2018).

■ **Pseudocódigo - Sintaxe**

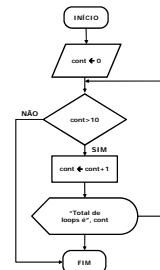
1. enquanto <EXPRESSÃO BOOLEANA> faça
2. <comandos>
3. fimenquanto

■ **Linguagem C - Sintaxe**

1. while (<condição>)
2. {
3. <comandos>
4. }

**Exemplo - estrutura de repetição
while (enquanto)**

- Considere um algoritmo que vai imprimir na tela o número de vezes que foi executado o *loop* usando a estrutura de repetição *while*



Fonte: Elaboração própria (2018).

■ **Pseudocódigo**

1. algoritmo "Exemplo1"
2. var
3. cont : inteiro
4. início
5. cont ← 0
6. enquanto cont < 10 faça
7. escreval (Total de loops é: ", cont)
8. cont ← cont + 1
9. fimenquanto
10. fimalgoritmo

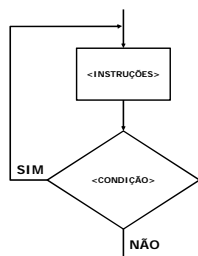
■ **Linguagem C**

1. ...
2. {
3. int cont = 0; //declarando e inicializando a variável de controle
- 4.
5. while (cont < 10) // Testando a condição
6. {
7. printf("Total de loops é: %d ", cont); //Instruções do laço
- 8.
9. cont++; //atualizando a variável de controle
10. }

Estrutura de repetição *do-while* (repita-até)

Estrutura de repetição *do-while* (repita-até)

- É bastante similar ao laço *while* (enquanto)
- Utilizada em simulações em que se faz necessário executar o corpo antes do teste, para ter garantia de que o laço será executado uma vez

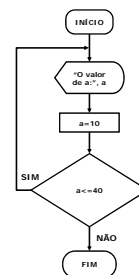


Fonte: Elaboração própria (2018).

- Pseudocódigo - Sintaxe
 1. repita
 2. <INSTRUÇÕES
 3. ate <EXPRESSÃO BOOLEANA>
- Linguagem C - Sintaxe
 1. do{
 2. <comandos>
 3. } while (<condição>;

Exemplo - estrutura de repetição *do-while* (repita-até)

- Considere um algoritmo que vai pegar dois números inteiros, comparar se o primeiro número é diferente do segundo e imprimir na tela uma mensagem com o resultado dessa comparação



Fonte: Elaboração própria (2018).

■ Pseudocódigo

```
1. ...
2. var
3.   a : inteiro
4. inicio
5.   a <- 10
6.   repita
7.     escreva ("O valor de a: ", a)
8.     a <- a+1
9.   ate ( a <= 40 )
10. fimalgoritmo
```

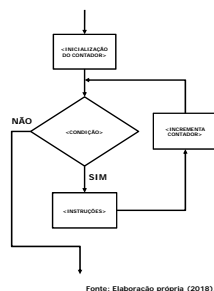
■ Linguagem C

```
1. ...
2. int a = 10;
3. do {
4.   printf("O valor de a: %d\n", a);
5.   a = a + 1; //ou a++;
6. } while( a <= 40 );
7. ...
```

Estrutura de repetição *for* (para)

Estrutura de repetição *for* (para)

- A estrutura de repetição *for* (para) é uma estrutura que reúne no cabeçalho os elementos de inicialização, condição e iteração
- No *for*, o laço vai repetir se o resultado da condição *for* verdadeira



■ Pseudocódigo - Sintaxe

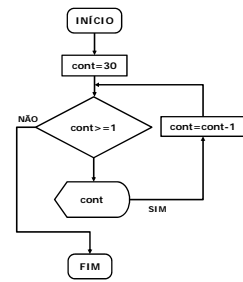
```
1. para <variável> de <início> ate <fim> <passo +1> faça
2.   <instruções>
3. fimpara
```

■ Linguagem C - Sintaxe

```
for (inicialização ; teste; interação)
{
  <instruções>;
}
```

Exemplo - estrutura de repetição for (para)

- Considera um algoritmo que faz a contagem decrescente de 1 até 30 e imprime na tela o resultado



Pseudocódigo

- ...
- var
- cont: inteiro
- inicio
- para cont de 30 ate 1 passo -1 faca
- escreval (cont)
- fimpara
- fimalgoritmo

Linguagem C

- ...
- ```
int cont;
for (cont = 30; cont >= 1; cont--)
{
 printf("%d\n", cont);
}
system("pause");
return 0;
}
```

### Estrutura de repetição aninhada

### Estrutura de repetição aninhada

- A linguagem C permite a utilização de um laço dentro de outro laço
- Quando um laço faz parte do corpo de outro laço, dizemos que o laço interno é um laço aninhado (Mizrahi, 2008)

#### ■ Sintaxe *while*

```
1. while (<condição>)
2. {
3. while (<condição>)
4. {
5. <comandos>
6. }
7. <comandos>
8. }
```

#### ■ Sintaxe *do-while*

```
1. do
2. {
3. <comandos>
4. do
5. {
6. <comandos>
7. } while (<condição>);
8. } while (<condição>);
```

#### ■ Sintaxe *for*

```
1. for (<valor inicial>; <condição>; <incremento>)
2. {
3. for (<valor inicial>; <condição>; <incremento>)
4. {
5. <instruções>;
6. }
7. <instruções>;
8. }
```

## Referências

■ MIZRAHI V. V. Treinamento em Linguagem C.  
2. ed. São Paulo: Makron Books, 2008.