

[◀ VOLTAR](#)

Estruturas de repetição: enquanto

Apresentar os conceitos sobre estruturas de repetição, dando ênfase à estrutura enquanto. São mostrados exemplos em Português Estruturado.

NESTE TÓPICO

- › Introdução
- › Estrutura de Repetição Enquanto
- › Exemplo 1
- › Exemplo 2

Marcar
tópico



Introdução

Em determinadas situações, temos de repetir o programa ou parte dele várias vezes, como no cálculo das médias das notas de um grupo de alunos. Uma forma de solucionar este problema, seria escrever o algoritmo em questão uma vez para cada aluno, ou seja, se forem 10 alunos, teríamos de escrevê-lo 10 vezes, o que é inviável.

Outro modo de resolver essa questão seria utilizar a mesma sequência de comandos novamente, ou seja, teríamos de realizar um retrocesso (ao início dos comandos) para cada aluno, fazendo, portanto, com que o fluxo de execução repetisse certo trecho do algoritmo, o que nessa aplicação corresponderia a repetir o mesmo trecho 30 vezes, sem, no entanto, ter de escrevê-lo 10 vezes.

A esses trechos do algoritmo que são repetidos damos o nome de estruturas de repetição. O número de repetições pode ser indeterminado, porém necessariamente finito.

O conceito de repetição (*looping* ou *laço*) é utilizado quando se deseja repetir um certo trecho de instruções por um número de vezes.

As estruturas de repetição se dividem em ENQUANTO, REPITA, PARA. Para determinarmos qual é a estrutura mais adequada para determinado programa, devemos saber qual o número de vezes que o trecho do programa vai ser executado (laços contados) ou a condição para que ele aconteça (laços condicionais).



Nem todas as estruturas de repetição possuem recursos para fazer a contagem do número de vezes que o laço deverá ser repetido. Por isso, devemos utilizar uma variável de apoio (contador), sempre do tipo inteiro. A seguir é possível observar a utilização de um contador dentro do contexto das estruturas de repetição.

```
1. var contador: inteiro
2.
3. inicio
4.
5.     .....
6.
7.     contador <- contador + 1
8.
9. fim
```

Estrutura de Repetição Enquanto

Consiste de uma estrutura de controle do fluxo de execução que permite repetir diversas vezes um mesmo trecho do algoritmo, porém, sempre verificando **antes** de cada execução se é permitido executar o mesmo trecho.

Desse modo, na estrutura enquanto, a condição de repetição é verificada antes de entrar no laço, isto é, uma condição é testada inicialmente e, se o resultado for verdadeiro, o bloco de instrução será executado.

Em Português Estruturado, a utilização do enquanto é feita a partir do conjunto de instruções enquanto...faça...fim_enquanto, sendo seu funcionamento controlado por decisão.

Assim, enquanto o valor da condição for verdadeiro, as ações dos comandos são executadas. No momento em que a condição se torna falsa, o processamento da rotina é desviado para fora do laço. Se a condição for falsa logo de início, os comandos não são executados nenhuma vez, ou seja, as instruções contidas no laço são ignoradas.

A estrutura enquanto deve ser utilizada da seguinte forma:

```
1. enquanto ( <condicao> ) faça
2.
3.     instruções
4.
5. fimenquanto
```

O enquanto é utilizado quando:

- Não sabemos o número de repetições.
- Quando possuímos uma expressão que deve ser avaliada para que os comandos da estrutura sejam executados.



Quando utilizamos os laços contados, conhecemos previamente quantas vezes o conjunto de instruções será repetido; então, precisamos do auxílio de um contador, que auxiliará no *looping* (laço) da estrutura. A utilização da variável (cont) possibilita a repetição da estrutura até que o contador atinja o limite estipulado na condição da estrutura. Assim, ele deixa de executar as instruções seguindo o fluxo do programa.

Exemplo 1

Um exemplo da utilização de um laço enquanto com contador é o cálculo da média para 10 alunos. O processo de contagem ocorre por meio da expressão aritmética que obtém o valor da variável *cont* e adiciona 1, armazenando esse resultado na própria variável *cont*. Repetindo esse comando várias vezes, perceberemos que a variável vai aumentando gradativamente de valor (de 1 em 1), simulando a contagem das execuções. Este exemplo pode ser observado a seguir:

```
1. algoritmo Calcula_Medias
2. var
3.   cont : inteiro
4.   n1, n2, n3, m : real
5. inicio
6.   cont <- 1
7.
8.   enquanto (cont <= 10) faça
9.
10.     escreva "Digite 3 notas"
11.     leia n1, n2, n3
12.     m <- (n1 + n2 + n3) / 3
13.     escreva "Média: ", m
14.     cont <- cont + 1
15.
16.   fimenquanto
17.
18. fim
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo para o cálculo da média dos 10 alunos.



◀ 52 / 01 ▶

Simulação da execução da estrutura de repetição enquanto - Exemplo 1

Outra forma de utilizarmos a estrutura de repetição enquanto é quando não sabemos, ao certo, o número de repetições. Nesse caso, não há a necessidade da utilização de uma variável como contador, em contrapartida, a variável utilizada na condição deve ser controlada dentro do laço. Tal controle é feito, normalmente, por leitura dessa variável.

Exemplo 2

Para exemplificar esta forma de utilização do enquanto, no exemplo a seguir é feito cálculo do somatório (soma acumulada) de apenas número positivos, os quais deverão ser definidos pelo usuário (por leitura). O somatório é feito enquanto o usuário digitar um número positivo e, quando ele digitar um número negativo, o laço é encerrado. Por fim, o valor do somatório é mostrado no final do algoritmo.



```
1. algoritmo Calcula_Somatorio
2. var
3.   num, soma: inteiro
4. inicio
5.   escreva "Digite um numero inteiro:"
6.   leia num
7.   soma <- 0
8.
9.   enquanto (num >= 0) faça
10.
11.     soma <- soma + num
12.     escreva "Digite um numero inteiro:"
13.     leia num
14.
15.   fimenquanto
16.
17.   escreva "Somatorio:", soma
18. fim
```

Na animação a seguir, é possível observar a execução passo a passo do algoritmo para o cálculo do somatório dos números positivos.

《 16 / 01 》

Simulação da execução da estrutura de repetição enquanto - Exemplo 2

CONTADOR E ACUMULADOR

Nos exemplos apresentados, dois conceitos foram utilizados: o contador e o acumulador.



O **contador** é uma variável cujo conteúdo é alterado pelo seu próprio valor adicionado ou subtraído de uma constante. Pode ser utilizado para acumular a quantidade de vezes que o programa executa as instruções do *looping*. O contador deve ser iniciado com um valor e deve ser incrementado ou decrementado. No primeiro exemplo é possível observar a utilização de um contador denominado *cont*.

O **acumulador**, também conhecido como somador, é utilizado para obter somatórios. Em estruturas de repetição, um acumulador ajuda a acumular valores cada vez que um laço é executado. Um acumulador é uma variável que recebe ela mesma mais o valor de outra variável. O acumulador deve ser iniciado, normalmente, com um valor nulo. O valor adicionado ao acumulador a cada execução de laço é variável, diferente do contador, cujo valor adicionado é sempre constante. No segundo exemplo é possível observar a utilização de um acumulador denominado *soma*.

Exercícios complementares

1. Escreva o português estruturado que verifique se uma "senha" numérica digitada pelo usuário está correta. O programa deve repetir o pedido até que o usuário escreva o valor correto. A senha deve estar definida no próprio programa.
2. Escreva o português estruturado que mostre na tela todos os números múltiplos de 3 até um certo número N digitado pelo usuário. Exemplo: usuário digita 14, aparece em tela: 3, 6, 9, 12.

Quiz

Exercício Final

Estruturas de repetição: enquanto

INICIAR >



Referências

FORBELLONE, A. L. V; EBERSPACHER, H. F. Lógica de Programação: a construção de algoritmos e estruturas de dados (3a. edição). São Paulo: Prentice Hall, 2005.

RISSETTI, G.; PUGA, S. Lógica de Programação e estruturas de dados, com aplicações em Java. São Paulo: Prentice Hall, 2003.

SOUZA, M. A. F. e outros. Algoritmos e lógica de programação: um texto introdutório para Engenharia (2a. edição). São Paulo: Cengage Learning, 2011.



Avalie este tópico



ANTERIOR

Implementação de desvios condicionais em C

Índice

Biblioteca
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

Ajuda?

PRÓXIMO

Estruturas de repetição: loops

(<https://ava.uninove.br/ava/estrutura-de-repeticao-loops>)

© Todos os direitos reservados