**Estruturas de Repetição**

Em muitas situações durante o desenvolvimento de programas, existe a necessidade de repetir por inúmeras vezes um determinado trecho de código, ou seja, repetir por várias vezes algumas instruções.

Anda em outras palavras, os comandos de repetição permitem que um conjunto de instruções seja executado até que satisfaça uma determinada condição.

Sendo assim, devemos utilizar estruturas de repetição para que isso seja possível.

As estruturas de repetição são:

•**for** (para)

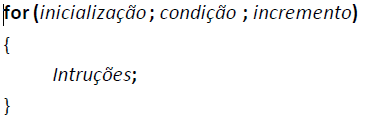
•**while** (Enquanto)

•**do...while** (faça...enquanto)

**Estrutura de Repetição: Comando FOR**

É uma estrutura muito utilizada principalmente quando se sabe quantas vezes deve ser executado um conjunto de instruções.

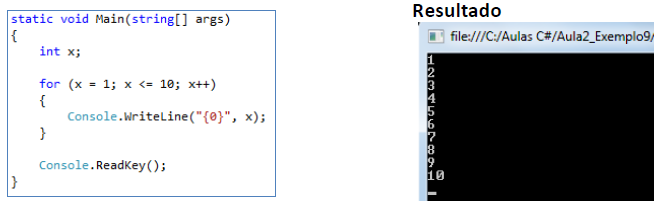
A forma geral do comando *for* é:



Normalmente, a **inicialização** é realizada através de um comando de atribuição, que é usado para colocar um valor na variável de controle do laço. A **condição** determina quando a repetição acaba. O **incremento ou decremento** define como a variável de controle do laço varia cada vez que o laço é repetido.

**Exemplo 9 - Estrutura de Repetição: Comando FOR (incremento)**

Neste exemplo serão exibidos na tela, todos os números de 1 a 10 em ordem crescente



Onde:

•***x = 10*** é a inicialização da variável, que começa o laço de repetição valendo ***10***.

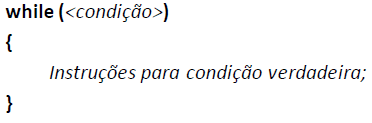
•***x >= 1*** é a condição de parada do laço de repetição, onde enquanto ***x*** for maior ou igual a ***1***, o bloco de instruções será executado.

•***x--*** é o decremento, onde a cada iteração a variável ***x*** passa a ser decrementada

**Estrutura de Repetição: Comando WHILE**

Esta estrutura de laço de repetição caracteriza-se por efetuar um teste lógico no início do laço de repetição, verificando se é permitido executar o trecho de instruções subordinado a ele.

Sintaxe:



**Estrutura de Repetição: Comando WHILE**

A estrutura ***while*** tem o seu funcionamento controlado por condição. Desta forma, pode executar um determinado conjunto de instruções enquanto a condição verificada permanecer *Verdadeira.* No momento em que esta condição se torna *Falsa,* o processamento da rotina é desviado para fora do laço de repetição, ou seja, a execução do bloco de instruções é encerrada.

Caso seja a condição *Falsa* logo no início do laço de repetição, as instruções contidas nele são ignoradas.

Caso seja necessário executar mais de uma instrução para uma condição verdadeira dentro de um laço, elas devem estar definidas dentro de um bloco por meio dos símbolos de chaves.

**Exemplo 11 - Estrutura de Repetição: Comando WHILE**

Neste exemplo serão exibidos na tela, todos os números de 1 a 20.

O valor de num inicial é ***1***. Na sequencia será realizado o teste condicional do comando ***while***, que será verdadeiro, pois o valor de ***num*** é menor ou igual a ***20***. Sendo assim, será exibido o valor de ***num*** na tela e em seguida haverá um incremento de ***num*** que passa a valer ***2***, e retorna para o teste condicional, que sendo verdadeiro, continuará a execução do bloco de instruções. Caso contrário será encerrada a execução do bloco de instruções.



**Exemplo 12 - Estrutura de Repetição: Comando WHILE**

Neste exemplo o usuário deverá digitar um número inteiro e exibir todos os números pares em ordem crescente menores que 20, a partir no número digitado pelo usuário.

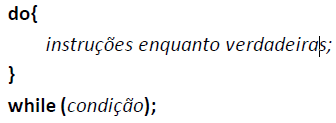


**Estrutura de Repetição: Comando DO...WHILE**

Esta estrutura faz um teste lógico no final de um laço de repetição. Ela é parecida com a estrutura ***while***. Seu funcionamento é controlado também por condição. Esse tipo de laço executa um conjunto de instruções pelo menos uma vez antes de verificar a validade da condição estabelecida.

Desta forma ***do...while*** irá processar um conjunto de instruções, no mínimo uma vez, até enquanto a condição for *verdadeira.*

Sintaxe:



**Exemplo 13 - Estrutura de Repetição: Comando DO...WHILE**

Neste exemplo, o bloco de repetição será executado pelo menos uma vez. Sendo iniciado o usuário digitará um número inteiro. Se este número for diferente de zero, exibirá o resultado do dobro deste número digitado. Senão informará o fim do programa. Ou seja, será executado o bloco de instruções enquanto a condição do ***while*** for verdadeira.



**Estruturas de Repetição, qual usar?**

Conforme conseguimos observar nos exemplos, ambas as estruturas possuem a mesma finalidade, que é executar um bloco de instruções várias vezes. Mas como saber qual utilizar? É simples, não existe uma regra que diz qual é a melhor forma. O que pode existir é um estudo sobre o desempenho de cada estrutura para cada problema a ser resolvido.

O que podemos concluir em linhas gerais para auxiliar na tomada de decisão em qual estrutura utilizar, é analisar o problema e usar:

•**for:** quando se sabe quantas vezes um bloco de instruções deverá ser executado.

•**while:** quando desejar que um bloco de instruções seja executado por várias vezes somente se a condição inicial for verdadeira (teste no início).

•**do...while:** quando desejar que um bloco de instruções seja executado pelo menos uma vez (teste no final).

**Exemplo 14 – Utilizando Diversas Estruturas**

Escreva um programa que exiba o seguinte menu na tela:

1 – Par/Ímpar

2 – Decrescente até 0 (zero)

3 - Sair

Para cada item deverá ser implementado o que se pede, sendo que se o usuário escolher a opção:

1, deverá ser digitado um número e o programa informará se este número é par ou ímpar.

2, deverá ser digitado um número e o programa exibirá todos os números em ordem decrescente até 0 (zero)

3, deverá encerrar o programa.



Neste exemplo foram utilizadas várias estruturas e instruções já vistas no curso até o momento:

switch...case, while, do...while, operadores ternários e decremento

Este foi apenas um exemplo de resolução deste exercício, pois poderia ainda ter usado ***if...else*** , em vez de usar operadores ternários. Poderia utilizar ***for***, me vez de ***while.***

\*\*\* Tente fazer o mesmo programa de outras formas, porém devendo chegar ao mesmo resultado.

***Console.Clear();*** instrução para limpar a tela.

*Obs.: Nas linhas de instrução para o usuário digitar valores, foi colocado o comando de entrada de dados na mesma linha, apenas para colocar em um só slide o código fonte.*

**Exemplo 15 – Contagem do número de elementos de um conjunto**

Neste exemplo, o usuário terá que digitar vários números inteiros e ao final o programa irá informar a quantidade de números que foram digitados. A condição de parada deste laço de repetição, será quando o usuário digitar o valor 0 (zero), que não deverá entrar para a contagem dos números válidos digitados.



**Exemplo 15 – Acúmulo de resultados parciais**

Neste exemplo, o usuário terá que digitar vários números reais e ao final o programa irá informar a soma dos valores que foram digitados. A condição de parada deste laço de repetição, será quando o usuário digitar o valor 0 (zero).



**EXERCÍCIOS**

1-) Faça um programa que exiba todos os números pares até 100. (Um para cada estrutura)

2-) Faça um programa que exiba todos os números ímpares até 100. (Um para cada estrutura)

3-) Faça um programa que leia um número e divida-o por dois (sucessivamente) até que o resultado seja igual a 1.

4-) Faça um programa que mostre a tabuada (0 à 10) do número 7.

5-) Faça um programa que leia um número digitado pelo usuário e mostre a tabuada referente a ele.

6-) Leia duas notas de 10 alunos, calcule e exiba a média de cada aluno . Ao final do programa exiba a média da classe.

7-) Faça um programa que leia 20 números inteiros positivos ou negativos. Ao final do programa mostre a quantidade de números positivos e negativos digitados.

8-) Escreva um programa que leia vários números inteiros e ao final informe quantos números pares, quantos números ímpares, quantos números positivos e quantos números negativos foram digitados pelo usuário. O programa só deve parar de rodar quando o usuário responder "S" na seguinte pergunta, "Deseja encerrar o programa?".