

Linguagem de Programação C#

Comandos de Repetição: While / Do...While

Laço com Condição no Início: 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.

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.

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. Desta forma, a instrução while deve ser escrita:

```
while (<condição>
{
    Instruções para condição verdadeira;
}
```

Exemplo 19

/* Programa que imprime os números de 1 a 20.*/

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int num=1;

            while (num<=20)
            {
                Console.WriteLine ("{0} ", num);
                num=num+1;    // igual num++
            }

            Console.ReadKey();
        }
    }
}
```

Exemplo 20

/*Programa que imprime os números de 1 a 20, identificando-os sendo par ou ímpar.*/

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int num=1;

            while (num<=20)
            {
                if ( ( num % 2 ) == 0 ) {
                    Console.WriteLine ( "\n O número {0} = par", num);
                }
                else {
                    Console.WriteLine("\n O número {0} = impar", num);
                }
                num++;
            }
            Console.ReadKey();
        }
    }
}
```

Em muitos algoritmos é necessário processar-se um conjunto de dados, e ao mesmo tempo contar o número de elementos deste conjunto.

Para realizarmos esta contagem, devemos utilizar uma variável especialmente criada para este fim. Uma variável utilizada para este fim é chamada de cont.

Exemplo 21

/*Programa que verifica os números pares e ímpares existentes de 0 a 500, e mostre o resultado de quantos números pares e ímpares existem neste intervalo.*/

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int num=0, cont_par=0, cont_impar=0;

            while (num<=500)
            {
                if ( ( num % 2 ) == 0 ) {
                    cont_par++;
                }
                else {
                    cont_impar++;
                }
                num++;
            }

            Console.Write ("No intervalo de 0 a 500 existem: \n");
            Console.Write ("{0} números pares e {1} números ímpares.",cont_par, cont_impar);

            Console.ReadKey();
        }
    }
}
```

Laço com Condição no Fim: 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. Diferente da estrutura while que executa somente um conjunto de instruções enquanto a condição é verdadeira.

Desta forma do...while irá processar um conjunto de instruções, no mínimo uma vez, até enquanto a condição for verdadeira. Então, a instrução do... while deve ser escrita:

```
do{
    Realiza instruções enquanto verdadeiras;
}
while (<condição>);
```

Exemplo 22

/* Programa que pede a leitura de dois valores inteiros para o usuário, efetua a adição e apresenta o resultado por cinco vezes. */

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int A, B, RESULT, i=6;

            do
            {
                Console.Write ("Digite um valor para A: ");
                A = int.Parse(Console.ReadLine());

                Console.Write ("Digite um valor para B: ");
                B = int.Parse(Console.ReadLine());

                RESULT = A + B;
                Console.WriteLine ("\n O resultado correspondente = {0} ", RESULT);
                i++;
            } while (i<=5);

            Console.ReadKey();
        }
    }
}
```