

Estrutura Repetitiva “Enquanto”

Professor
Antonio Marcos Alvarez

“Enquanto”

Esta **estrutura de controle** é usada para **repetir** um bloco de comandos (conjunto de comandos e instruções contidas em suas chaves{..}) em quanto a expressão (condição) for “**true**” (verdadeira)

Exemplificando:

Vamos desenvolver um projeto (programa) que permita ao usuário entrar com um caractere do teclado, um a um, ou seja, um por vez, logo que esse for inserido, apresentar seu código decimal da tabela ASCII (American Standard Code for Information Interchange), exemplo:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace EX_TabelaASCII
8  {
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             char caracter = '0';
14             int codigo = 0;
15
16             Console.WriteLine("Digite um caracter qualquer do teclado, " +
17                 "para ver seu Código da tabela ASCII, " +
18                 "ou * para Finalizar");
19
20             //Enquanto a variavel (char) caracter for diferente de '*' a rotina {...} será executada
21             while (caracter != '*')
22             {
23                 //caracter = char.Parse(Console.ReadLine());
24                 caracter = Convert.ToIntChar(Console.ReadLine());
25                 codigo = (int)caracter;
26                 Console.WriteLine(codigo);
27             }
28             //Final do While (enquanto)
29
30             Console.WriteLine("Pressione qualquer tecla para Encerrar...");
31             Console.ReadKey();
32         }
33     }
34 }
```

- **while** (Enquanto): A instrução **while** executa uma instrução ou um bloco de instruções enquanto uma expressão booleana especificada é avaliada como true. Como essa expressão é avaliada antes de cada execução do loop, um loop while é executado zero ou mais vezes. Isso difere do loop do, que é executado uma ou mais vezes.

Como utilizar (Sintaxe – Forma Correta)

Sintaxe / regra

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

Regra:

V: executa e volta
F: pula fora

- While (Enquanto): Será executado todo seu “bloco” (Comandos e Instruções entre “{...}”), enquanto a **condição** for Verdadeira, caso contrario o “loop” é encerrado.

Teste de Mesa (DEBUG – Manual)

```
int x = 0;  
int y = 4;  
while (x < 3) {  
    y = y + 2;  
    x = x + 1;  
    Console.WriteLine(x + " | " + y);  
}
```

X	Y
0	4
1	6
2	8
3	10
FIM	

Saída (Tela / Display)
1 6
2 8
3 10

Código (Projeto - EX_TabelaASCII)

```
using System;  
  
namespace EX_TabelaASCII  
{  
  
    internal class Program  
    {  
  
        static void Main(string[] args)  
        {  
  
            //Código do Programa  
        }  
  
    }  
  
}
```

```
//Código do Programa  
  
char caracter = '0';  
  
int codigo = 0;  
  
Console.WriteLine("Digite um caracter qualquer do teclado, " +  
                  "para ver seu Código da tabela ASCII, " +  
                  "ou * para Finalizar");  
  
//Enquanto a variável (char) caracter for diferente de '*' a rotina {...} será executada  
while (caracter != '*')  
{  
  
    //caracter = char.Parse(Console.ReadLine());  
    caracter = Convert.ToChar(Console.ReadLine());  
  
    codigo = (int)caracter;  
  
    Console.WriteLine(codigo);  
}  
  
//Final do While (enquanto)  
  
Console.WriteLine("Pressione qualquer tecla para Encerrar...");  
Console.ReadKey();
```

Exercício Lógico (Desafio) EX_While_IdadeMedia

Desenvolver um programa que entre com "N" idades, até que seja fornecido uma idade não aceita, ou seja, menor que 0 (número negativo), após, interromper a entrada de dados (idades), calcular e exibir a média aritmética das idades informadas.

Obs: Se não houver idades validas, exibir mensagem que não houve dados validos para o calculo da Média de Idades. Pois não podemos dividir um numero por 0 (zero).

Entrada de Dados (Idades)	Saída (Tela / Display)
10	Foram digitadas 3 idade(s) válidas, e a média calculada é 20,0
20	
30	
-5	

Entrada de Dados (Idades)	Saída (Tela / Display)
-5	Não houve entrada de Idade(s) validas, Programa Finalizado...

Codificação

```
double idade=0.0, soma=0.0, media=0.0;  
int c=0;  
  
while (idade >= 0) {  
  
    idade = double.Parse(Console.ReadLine());  
    if (idade < 0) {  
        Console.WriteLine("Idade Inválida! Entrada de Dados Finalizada...");  
    }  
    else {  
        c = c + 1;  
        soma = soma + idade;  
    }  
}
```

```
//Calcula Média das Idades e guarda na variável media (se c > 0), após exibirá os valores calculados

if (c > 0) {
    media = soma / c;
    Console.WriteLine("Foram digitadas " + c + " Idade(s) válidas, e a Média calculada é " + media.ToString("F1"));
}
else {
    Console.WriteLine("Não houve entrada de Idade(s) validas, Programa Finalizado...");
}

Console.WriteLine("Digite [Enter] para encerrar.");
Console.ReadLine();
```

Estrutura Repetitiva “Para”

É uma **estrutura de controle** que **repete** um bloco de comandos **para** um certo **intervalo de valores**.

Quando usar: quando se sabe previamente a quantidade de repetições, ou o intervalo de valores.

Por exemplo:

Fazer um programa que lê um valor inteiro N e depois N números inteiros. Ao final, mostra a soma dos N números lidos

Entrada	Saída
3	
5	
2	
4	

Estrutura “Para” usamos “for”

Sintaxe / regra

Executa somente
na primeira vez

V: executa e volta
F: pula fora

Executa toda vez depois
de voltar

```
for ( início ; condição ; incremento ) {  
    comando 1  
    comando 2  
}
```

- Ex: `for (c=0; c<9; c=c+1 ou c++) {...}`

Codificação do Exemplo: EX_FOR_somaNn

//Declaramos as variáveis N (entradas), n (numero informado), soma (Armazena das soma dos números) e c (contador)

```
int N = 0;  
double n = 0.0, soma =0.0;  
int c = 0;
```

```
Console.WriteLine("Informe o Nº de números a ser digitado:");
```

```
N = int.Parse(Console.ReadLine());
```

```
for (c=1; c<=N; c=c+1)
```

```
{
```

```
    Console.WriteLine("Entre com o " + c + "º numero:");
```

```
    n = double.Parse(Console.ReadLine());
```

```
    soma = soma + n;
```

```
}
```

```
Console.WriteLine("A Soma dos números é = " + soma);
```

```
Console.WriteLine("Digite [Enter] para encerrar...");
```

```
Console.ReadLine();
```

Desafio:

Desenvolver um programa que monte uma tabuada (padrão 10 elementos) de um numero qualquer, após a impressão na tela da Tabuada, solicitar ao usuário se ele gostaria de continuar com o programa para uma nova “TABUADA” de outro numero?

Entrada de Dados (Número a ser Tabuado)	Saída (Tela / display)
25	$25 \times 1 = 25$ $25 \times 2 = 50$ $25 \times 3 = 75$ $25 \times 10 = 250$
_ (Aguardando Resposta)	Deseja efetuar outra operação <s/n>? _

Codificação

```
//Declarando Variáveis
double n = 0.0;
int c = 0;
char resposta = 's';
while (resposta == 's') {
    Console.Clear(); //Limpa a Tela de Console
    Console.Write("Informe o Nº a ser Tabuado:");
    n = double.Parse(Console.ReadLine());
    for (c = 1; c <= 10; c++) {
        Console.WriteLine(n + " X " + c + " = " + (n * c));
    }
    Console.Write("Deseja efetuar outra operação <s/n>? ");
    resposta = char.Parse(Console.ReadLine());
}
Console.WriteLine("Programa Encerrado...");
Console.WriteLine("Tecle [Enter] para finalizar Console C#...");
Console.ReadLine();
```

Estrutura Repetitiva “Faça-Enquanto”

Menos utilizada, mas em alguns casos se encaixa melhor ao problema.

O bloco de comandos executa pelo menos uma vez, pois a condição é verificada no final.

A instrução "do" executa uma instrução ou um bloco de instruções enquanto uma expressão booliana especificada é avaliada como true (verdade). Como essa expressão é avaliada após cada execução do loop, um loop "do-while" é executado uma ou mais vezes. Isso é diferente do loop while, que é executado zero ou mais vezes.

Sintaxe (do-while):

Sintaxe / regra

```
do {  
    comando 1  
    comando 2  
} while ( condição );
```

Regra:

V: volta

F: pula fora

Exercícios

- a) Elaborar um programa que solicite ao usuário entrar um valor relacionado a temperatura em Celsius ($^{\circ}\text{C}$) e converte-la em Fahrenheit ($^{\circ}\text{F}$).
- b) Elaborar um programa que solicite ao usuário entrar um valor relacionado a temperatura em Fahrenheit ($^{\circ}\text{F}$), e converte-la em Celsius ($^{\circ}\text{C}$).
- Ambos programas, irão solicitar ao usuário o desejo de efetuar nova operação (S/N).

✓ Formulas:

Celsius to Fahrenheit	$^{\circ}\text{F} = \left(\frac{9}{5} \times ^{\circ}\text{C}\right) + 32$
Fahrenheit to Celsius	$^{\circ}\text{C} = \frac{5}{9} (^{\circ}\text{F} - 32)$

Exemplo: EX_DoWhile_CelsiusFahrenheit

```
double C = 0.0, F = 0.0;
string repetir;
do {
    Console.Write("Informe o valor em Celsius:");
    C = double.Parse(Console.ReadLine());
    //Conversão de Celsius para Fahrenheit
    F = 9.0 * C / 5.0 + 32.0; //ou F = (9/5)*C + 32 ou F = (1.8*C)+32
    //Não precisa colocar (), pois multiplicação e divisão serão efetuados primeiro
    Console.WriteLine("Equivalente em Fahrenheit: " + F.ToString("F1"));
    Console.Write("Deseja Repetir <s/n>? ");
    repetir = Console.ReadLine();
} while (repetir == "s");
Console.WriteLine("Pressione [Enter] para finalizar o Console...");
Console.ReadLine();
```