





Programação Orientada a Objetos

Ms. Eng. Claudio Ferreira de Carvalho



AULA 07

Estruturas de Repetição

Estruturas de repetição

- ✓ Muitas vezes em um programa é necessário repetir uma série de comandos um determinado número de vezes, ou quantas vezes forem necessárias, enquanto ou até que uma determinada condição seja verdadeira.
- ✓ Para executar estas rotinas o C# possui 3 estruturas:
 - ✓ for valor inicial; valor final; incremento; ← Existe um contador interno
 - ✓ do while; ← Comandos executados enquanto uma condição for verdadeira
 - ✓ while ← Comandos executados até que uma condição for verdadeira
- ✓ Estas estruturas, também chamadas de “estruturas de loop” ou “estruturas de laço”, são muitas vezes classificados como:
- ✓ Estrutura de laço contado (que é a estrutura do tipo for).
- ✓ Estrutura de laço condicional (que são as estruturas do tipo “do while” e “while”).

Estrutura for

✓ Finalidade:

- ✓ Executar o bloco de comandos entre os limitadores {...}, enquanto um contador é acrescido de um valor inicial até um valor final obedecendo a um incremento pré-determinado.

✓ Sintaxe:

- ✓ Executar o bloco de comandos entre os limitadores {...}, enquanto um contador é acrescido de um valor inicial até um valor final obedecendo a um incremento pré-determinado.

```
for (valor_inicial; valor_final; incremento)
{
    <comando>;
    <comando>;
    <comando>;
    :
    :
    :
}
```

Estruturas for – Exemplo

```
1. namespace Aula07_Ex01
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara variáveis
8.             int contador;
9.
10.            // Executa um contador de 1 até 10
11.            for (contador = 1; contador <= 10; contador++)
12.            {
13.                Console.WriteLine("O Contador é : " + contador);
14.            }
15.            // Mantem a tela aberta esperando a digitação de uma tecla
16.            Console.ReadKey();
17.        }
18.    }
19. }
```

Está sendo feita uma atribuição à variável contador (=1)

Está sendo feita uma comparação do valor do contador (<= 10)

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Or
O Contador é : 1
O Contador é : 2
O Contador é : 3
O Contador é : 4
O Contador é : 5
O Contador é : 6
O Contador é : 7
O Contador é : 8
O Contador é : 9
O Contador é : 10
```

Estrutura do....while

✓ Finalidade:

- ✓ Executar o bloco de comandos entre as os limitadores {...}, enquanto uma determinada condição, testada pelo while for verdadeira.
- ✓ É importante notar que o teste é executado no final do comando, então o grupo de comandos será executado pelo menos uma vez, independente da condição ser ou não satisfeita.

✓ Sintaxe:

```
Do
{
    <comando>;
    <comando>;
    <comando>;
    :
    :
    :
}
while (condicao == teste)
```

Estrutura do.....while – Exemplo

```
1. namespace Aula07_Ex02
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara variável
8.             string nome;
9.
10.            // Solicita o nome do usuário
11.            do
12.            {
13.                Console.Write("Forneça seu nome: ");
14.                nome = Console.ReadLine();
15.            }
16.            while (nome == "");
17.
18.            // Cumprimenta o usuário
19.            Console.WriteLine("Bom dia " + nome);
20.
21.            // Mantem a tela aberta esperando a digitação de uma tecla
22.            Console.ReadKey();
23.        }
24.    }
25. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orie
Forneça seu nome: Claudio
Bom dia Claudio
```

Notar que é uma comparação do valor do contador (== "")

Comando executado pelo menos uma vez, pois o teste é no final do comando

Alguns comandos para manipulações de string

- ✓ Calcula o tamanho de uma string:
 - ✓ Permite determinar número de caracteres de uma string.
- ✓ Sintaxe:
 - ✓ `variavel = identificador_string.lenght`
- ✓ Exemplo:
 - ✓ `tamanho = nome.Length;`
- ✓ Identifica caracteres de uma string:
 - ✓ Identifica cada um dos caracteres de uma string.
- ✓ Sintaxe:
 - ✓ `variavel = new string(new char[n])`
- ✓ Exemplo:
 - ✓ `nome = new string(new char[30]);`

a) n é o número de cada um dos caracteres da string.
b) Os caracteres ficam no formato char.
c) À primeira letra é atribuído ao `char[0]`, a segunda ao `char[1]` e assim sucessivamente, até a última letra que é atribuída ao `char[n-1]` (sendo n o total de caracteres da string original).

Alguns comandos para manipulações de string

- ✓ Captura conjuntos de caracteres em uma string:
 - ✓ Cria uma string a partir de outra, pegando somente uma parte dos caracteres da string.
- ✓ Sintaxe:
 - ✓ `variavel = variavel.Substring(inicio,fim)`
 - a) inicio é o número do caracter da string a partir do qual será iniciada a captura.
 - b) fim é o número do caracter da string onde será encerrada a considerada.
- ✓ Exemplo:
 - ✓ `nome = nome.Substring(0, 9);`
 - Pega os 10 primeiros caracteres a string nome.

Exemplo com comandos de manipulação de string

```
1. namespace Aula07_Ex04
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de variáveis
8.             string nome;
9.             char letra;
10.            int c, tamanho;
11.
12.            // Cria a variável nome indexada de 0 à 29
13.            nome = new string(new char[30]);
14.
15.            // Solicita o nome completo do usuário
16.            do
17.            {
18.                Console.Write("Forneça seu nome e sobrenome ");
19.                nome = Console.ReadLine();
20.            }
21.            while (nome == "");
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Pr
Forneça seu nome e sobrenome Claudio Carvalho
Seu primeiro nome é: Claudio
Seu sobrenome nome é: Carvalho
```

```
22.
23.         // Verifica e informa quantas letras tem o nome
completo do usuário
24.         tamanho = nome.Length;
25.
26.         // Separa o primeiro nome
27.         Console.Write("Seu primeiro nome é: \t");
28.         c = 0;
29.         do
30.         {
31.             letra = nome[c];
32.             Console.Write(letra);
33.             c = c + 1;
34.         }
35.         while ((letra != ' ') && (c < tamanho));
36.         if (c < tamanho)
37.         {
38.             // Separa o segundo nome
39.             Console.Write("\nSeu sobrenome nome é: \t");
40.             do
41.             {
42.                 letra = nome[c];
43.                 Console.Write(letra);
44.                 c = c + 1;
45.             }
46.             while ((letra != ' ') && (c < tamanho));
47.         }
48.         // Mantem a tela aberta esperando a digitação de uma
tecla
49.         Console.ReadKey();
50.     }
51. }
52. }
```

Exemplo com comandos de manipulação de string

```
1. namespace Aula07_Ex03
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declarações de Variáveis
8.             string nome;
9.             int tamanho;
10.            char plettra, ulettra;
11.
12.            // Cria a variável nome indexada de 0 à 29
13.            nome = new string(new char[30]);
14.
15.            // Impede que seja informado um nome em branco
16.            do
17.            {
18.                Console.Write("Forneça seu nome: ");
19.                nome = Console.ReadLine();
20.                if (nome == "")
21.                    Console.WriteLine("Por favor não deixe em branco");
22.            }
23.            while (nome == "");
```

```
24.
25.            // Mostra nome
26.            Console.WriteLine("Olá Sr(a) " + nome + " Bem vindo");
27.
28.            // Calcula e mostra a quantidade de letras
29.            tamanho = nome.Length;
30.            Console.WriteLine("Seu nome possui " + tamanho + "
31.            letras");
32.
33.            // Verifica e mostra a primeira e última letra
34.            plettra = nome[0];
35.            ulettra = nome[tamanho - 1];
36.            Console.WriteLine("A primeira letra de seu nome é " +
37.            plettra);
38.            Console.WriteLine("A última letra de seu nome é " +
39.            ulettra);
40.
41.            // Mantem a tela aberta esperando a digitação de uma
42.            tecla
43.            Console.ReadKey();
44.        }
45.    }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_20

```
Forneça seu nome: Claudio
Olá Sr(a) Claudio Bem vindo
Seu nome possui 7 letras
A primeira letra de seu nome é C
A ultima letra de seu nome é o
```

Exemplo com comandos de manipulação de string

```
1. namespace Aula07_Ex05
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de variáveis
8.             string nome;
9.             int tamanho, c;
10.            char letra;
11.
12.            // Cria a variável nome indexada de 0 à 49
13.            nome = new string(new char[50]);
14.
15.            // Solicita o nome completo do usuário
16.            do
17.            {
18.                Console.Write("Qual seu nome completo? (máx. 50
19.                caracteres) ");
20.                nome = Console.ReadLine();
21.                if (nome == "")
22.                    Console.WriteLine("Por favor não deixe em
23.                    branco");
24.            }
25.            while (nome == "");
```

```
24.
25.            // Verifica e informa quantas letras tem o
26.            nome fornecido
27.            tamanho = nome.Length;
28.
29.            // Separa o nome e sobrenomes fornecidos
30.            Console.WriteLine("Seu nome e sobrenomes
31.            são:");
32.            c = 0;
33.            do
34.            {
35.                letra = nome[c];
36.                Console.Write(letra);
37.                c = c + 1;
38.            }
39.            while ((letra != ' ') && (c < tamanho));
40.            Console.WriteLine();
41.        }
42.        while (c < tamanho);
43.
44.        // Mantem a tela aberta esperando a digitação
45.        de uma tecla
46.        Console.ReadKey();
47.    }
48. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projetos_POO\Aula07_Ex05\...
Qual seu nome completo? (máx. 50 caracteres) Claudio Ferreira de Carvalho
Seu nome e sobrenomes são:
Claudio
Ferreira
de
Carvalho
```

Estrutura while

✓ Finalidade:

- ✓ Executar o bloco de comandos entre as os limitadores {...}, enquanto uma determinada condição, testada pelo while for verdadeira.
- ✓ Note que o Comando while, testa a condição antes de entrar no loop. Caso a condição não seja satisfeita, o loop não será executado nenhuma vez. Por este motivo é importante garantir que antes do comando while, seja atribuído um valor para a variável e este valor deve atender as condições de teste.

✓ Sintaxe:

```
while (condição == teste)
{
    <comando>;
    <comando>;
    <comando>;
    :
    :
    :
}
```

Exemplo com comandos de manipulação de string

```
1. namespace Aula07_Ex06
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Delclara variáveis
8.             double saldo;
9.             double saque;
10.
11.             // Solicita o saldo atual
12.             Console.Write("Forneça o saldo atual: ");
13.             saldo = double.Parse(Console.ReadLine());
14.
15.             // Permite retiradas enquanto existir saldo
16.             while (saldo > 0)
17.             {
18.                 Console.Write("Forneça o valor do saque: ");
19.                 saque = double.Parse(Console.ReadLine());
20.
21.                 // Informa o Saldo ou avisa que não pode sacar
22.                 if (saque <= saldo)
23.                 {
24.                     saldo = saldo - saque;
25.                     Console.WriteLine("Seu saldo é: " + saldo);
26.                 }
27.                 else
28.                 {
29.                     Console.WriteLine("Você não pode sacar este
30. valor ");
31.                 }
32.             }
33.         }
34.     }
35. }
```

```
32.
33.         // Informa que o saldo encerrou
34.         (fechou o comando While)
35.         Console.WriteLine("Você já fez todas
36. as retiradas possíveis");
37.
38.         // Mantem a tela aberta esperando a
39.         digitação de uma tecla
40.         Console.ReadKey();
41.     }
42. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Pr
Forneça o saldo atual: 600
Forneça o valor do saque: 200
Seu saldo é: 400
Forneça o valor do saque: 100
Seu saldo é: 300
Forneça o valor do saque: 50
Seu saldo é: 250
Forneça o valor do saque: 250
Seu saldo é: 0
Você já fez todas as retiradas possíveis
```

Neste programa, o “valor do saldo” está sendo atribuído antes de se entrar no comando while.

Este programa não permite que o usuário interrompa os saques enquanto não encerrar o saldo, esta deficiência será sanada no próximo exemplo.

Exemplo com comandos de manipulação de string

```
1. namespace Aula07_Ex07
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara variáveis
8.             double saldo;
9.             double saque;
10.
11.             // Solicita o saldo atual e não permite saldo
negativo ou nulo
12.             do
13.             {
14.                 Console.Write("Forneça o saldo atual: ");
15.                 saldo = double.Parse(Console.ReadLine());
16.
17.                 // fornece mensagem indicando que o saldo deve
ser positivo
18.                 if (saldo <= 0)
19.                 {
20.                     Console.WriteLine("Por favor forneça um
saldo maior que zero");
21.                 }
22.             }
23.             while (saldo <= 0);
```

```
24.
25.             // Testa se existe saldo, caso exista permite mais um
saque.
26.             while (saldo > 0)
27.             {
28.                 Console.Write("Forneça o valor do saque <0 para
encerrar >: ");
29.                 saque = double.Parse(Console.ReadLine());
30.                 // Não permite saque sem saldo.
31.                 if (saque <= saldo)
32.                 {
33.                     saldo = saldo - saque;
34.                     Console.WriteLine("Seu saldo é: " + saldo);
35.
36.                     // Encerra o programa
37.                     if (saque == 0)
38.                     {
39.                         break;
40.                     }
41.                 }
42.                 else
43.                 {
44.                     Console.WriteLine("Você não pode sacar este
valor");
45.                 }
46.             }
47.             Console.WriteLine("Você encerrou retiradas");
48.
49.             // Mantem a tela aberta esperando a digitação de uma
tecla
50.             Console.ReadKey();
51.
52.         }
53.     }
54. }
```


Exemplo com comandos de manipulação de string

```
1. namespace Aula07_Ex07
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declara variáveis
8.             double saldo;
9.             double saque;
10.
11.             // Solicita o saldo atual e não permite saldo
12.             // negativo ou nulo
13.             do
14.             {
15.                 Console.Write("Forneça o saldo atual: ");
16.                 saldo = double.Parse(Console.ReadLine());
17.
18.                 // fornece mensagem indicando que o saldo deve
19.                 // ser positivo
20.                 if (saldo <= 0)
21.                 {
22.                     Console.WriteLine("Por favor forneça um
23. saldo maior que zero");
24.                 }
25.             } while (saldo <= 0);
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objetos_2024\Projetos_POO\Au
Forneça o saldo atual: 0
Por favor forneça um saldo maior que zero
Forneça o saldo atual: 600
Forneça o valor do saque <0 para encerrar >: 300
Seu saldo é: 300
Forneça o valor do saque <0 para encerrar >: 100
Seu saldo é: 200
Forneça o valor do saque <0 para encerrar >: 0
Seu saldo é: 200
Você encerrou retiradas
```

```
24.
25.         // Testa se existe saldo, caso exista permite mais um
26.         saque.
27.         while (saldo > 0)
28.         {
29.             Console.Write("Forneça o valor do saque <0 para
30. encerrar >: ");
31.             saque = double.Parse(Console.ReadLine());
32.             // Não permite saque sem saldo.
33.             if (saque <= saldo)
34.             {
35.                 saldo = saldo - saque;
36.                 Console.WriteLine("Seu saldo é: " + saldo);
37.
38.                 // Encerra o programa
39.                 if (saque == 0)
40.                 {
41.                     break;
42.                 }
43.             }
44.             else
45.             {
46.                 Console.WriteLine("Você não pode sacar este
47. valor");
48.             }
49.         }
50.
51.         Console.WriteLine("Você encerrou retiradas");
52.
53.         // Mantem a tela aberta esperando a digitação de uma
54.         Console.ReadKey();
```

