





Programação Orientada a Objetos

Ms. Eng. Claudio Ferreira de Carvalho



AULA 03

Comando Write, WriteLine, formatações em tela

Como utilizar o editor de programa do C#

- ✓ O editor do C# com o “Auto Completar” facilita muito a digitação dos comandos da linguagem.
- ✓ Para escrever um comando basta:
 - ✓ Iniciar a digitação do comando, (lembrar que o C# é case sensitivo – letras maiúsculas são diferentes de letras minúsculas).
 - ✓ Imediatamente o “Menu Suspenso Auto Completar” apresentará diversas opções
 - ✓ Procurar utilizando as teclas “cursor para baixo” e “cursor para cima”, ou clicar com o Mouse na barra de rolamento até encontrar o comando desejado.
 - ✓ Quando o comando desejado estiver em evidência, basta digitar a tecla <Enter>.
 - ✓ É possível também digitar a tecla Tab quando o curso estiver sobre a palavra.
 - ✓ Caso o comando seja composto, ou seja, mais instruções devam ser adicionadas, basta digitar ponto (.) e repetir as instruções.

É muito interessante utilizar sempre o recurso Auto Completar, pois ele evita erros, principalmente no caso de diferenças entre letras maiúsculas e minúsculas, muito comum em programadores sem prática.

Comando Comentário

✓ Objetivo

- ✓ Utilizado para o programador colocar comentários.
- ✓ Não é executado pelo Compilador.

✓ Sintaxe

// Comentário

ou

/*Comentário.....

*Comentário...*/

✓ Exemplo

```
1. namespace Aula03_Ex01
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Esta é uma linha de comentário que será desprezada pelo comando
8.             /* Caso se deseje escrever comentários
9.              * com mais de uma linha basta iniciar
10.             * com barra asterisco e fechar com */
11.         }
12.     }
13. }
```

Comando Write e WriteLine

✓ Objetivo

- ✓ Informar dados na tela.
- ✓ Como está sendo utilizado em “ConsoleApplications” => Precedido da palavra “Console”.

✓ Sintaxe

Console.Write(“Texto a ser escrito – deve estar entre aspas”);

Console.WriteLine(“Texto a ser escrito – deve estar entre aspas”);

✓ Exemplo

```
1 namespace Aula03_Ex02
2 {
3     internal class Program
4     {
5         static void Main(string[] args)
6         {
7             // Coloca um conjunto de caracteres na tela
8             Console.WriteLine("o resultado da medida foi 30,9 m");
9         }
10    }
11 }
```

Após escrever o texto o cursor permanece na mesma linha no próximo caracter

Após escrever o texto o cursor passa para o primeiro caracter da próxima linha.

Execução

Console de Depuração do Microsoft Visual Studio

o resultado da medida foi 30,9 m

O C:\D\Aulas_Atuais\Curso_Programacao_Orientada\Aula03_Ex02\bin\Debug\net8.0\Aula03_Ex02.exe

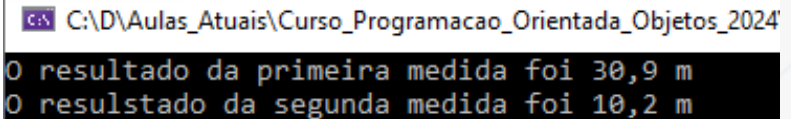
Para fechar o console automaticamente quando o programa terminar, vá em: Opções -> Depuração -> Fechar o console automaticamente. Pressione qualquer tecla para fechar esta

Comando ReadKey

- ✓ Objetivo
 - ✓ Interrompe a programação e espera a digitação de uma tecla.
- ✓ Sintaxe
 - Console.ReadKey();
- ✓ Exemplo

```
1. namespace Aula03_Ex03
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Coloca conjuntos de caracteres na tela
8.             Console.WriteLine("O resultado da primeira medida foi 30,9 m");
9.             Console.Write("O resultado da segunda medida foi 10,2 m");
10.            // Mantém a tela aberta e retira os comentários esperando digitar uma tecla
11.            Console.ReadKey();
12.        }
13.    }
14. }
```

Execução



```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024
O resultado da primeira medida foi 30,9 m
O resultado da segunda medida foi 10,2 m
```

Conectando informações com + no comando Write

✓ Objetivo

- ✓ Colocar mais de uma informação no mesmo comando Write.

✓ Sintaxe

`Console.WriteLine("conjunto de caracteres 1" + "conjunto 2");`

✓ Exemplo

```
1. namespace Aula03_Ex04
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Concatena utilizando a sintaxe =
8.             Console.WriteLine("O valor é: " + 100);
9.             Console.ReadKey();
10.        }
11.    }
12. }
```

Execução

Selecionar C:\D\Aulas_Atuais\C
O valor é: 100

Utilizando o parâmetro {0} para formatar Write

- ✓ Objetivo
 - ✓ Valores após as aspas são colocados no lugar o {0}

- ✓ Sintaxe

```
Console.WriteLine("A palavra é: {0}", Bom);
```

- ✓ Exemplo

```
1. namespace Aula03_Ex05
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             Console.WriteLine("Apalavra é: {0}", "Bom");
8.             Console.ReadKey();
9.         }
10.    }
11.}
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orienta
Apalavra é: Bom
```

Utilizando o parâmetro {0} para formatar Write (mais parâmetros)

✓ Objetivo

- ✓ Valores após as aspas são colocados no lugar o {0}

✓ Sintaxe

```
Console.WriteLine("O primeiro é : {0} e o segundo é {1} ", 120, 130);
```

✓ Exemplo

```
1. namespace Aula03_Ex06
```

```
2. {
```

```
3.     internal class Program
```

```
4.     {
```

```
5.         static void Main(string[] args)
```

```
6.         {
```

```
7.             Console.WriteLine("O primeiro número é: {0}, e o segundo é {1}", 120, 130);
```

```
8.             Console.ReadKey();
```

```
9.         }
```

```
10.     }
```

```
11. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Pro

O primeiro número é: 120, e o segundo é 130

Delimitando a largura de informações com Write

- ✓ Objetivo
 - ✓ Delimitar o espaço onde uma informação é colocada.

- ✓ Sintaxe

Console.Write("Texto desejado {0,10} novo texto desejado {1,10}", valor0, valor1);

- ✓ Exemplo

1. namespace Aula03_Ex07

2. {

3. internal class Program

4. {

5. static void Main(string[] args)

6. {

7. Console.WriteLine("0 primeiro valor é {0,10}, e o segundo é {1,10}", 120, 130);

8. Console.WriteLine("0 primeiro valor é {0,-10}, e o segundo é {1,-10}", 120, 130);

9. Console.ReadKey();

10. }

11. }

12. }

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objetos_2024\Projetos_POO\Aula03.

```
0 primeiro valor é      120, e o segundo é      130
0 primeiro valor é 120      , e o segundo é 130
```

Alinha 10 à direita

Alinha 10 à esquerda

Delimitando o número de casas decimais com Write

- ✓ Objetivo
 - ✓ Determinar o número de casas decimais.

- ✓ Sintaxe

`Console.Write("Texto {0,15:F2}", valor0);`

- ✓ Exemplo

```
1. namespace Aula03_Ex08
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             Console.WriteLine("O primeiro valor é {0,10:F2}, e o segundo é {1,10:F2}", 120, 130);
8.             Console.WriteLine("O primeiro valor é {0,-10:F3}, e o segundo é {1,-10:F3}", 120, 130);
9.             Console.ReadKey();
10.        }
11.    }
12. }
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projetos_POO\Aula03_Ex08>
O primeiro valor é 120,00, e o segundo é 130,00
O primeiro valor é 120,000 , e o segundo é 130,000
```

Mostrar com 2
casas decimais

Mostrar com 3
casas decimais

Comando Write com tabulação

- ✓ Objetivo
 - ✓ Promover uma tabulação dentre ou no início de uma frase ou conjunto de informações

- ✓ Sintaxe

Console.Write("Texto1 \t Texto2 \t Texto3");

- ✓ Exemplo

```
1. namespace Aula03_Ex09
```

```
2. {
```

```
3.     internal class Program
```

```
4.     {
```

```
5.         static void Main(string[] args)
```

```
6.         {
```

```
7.             Console.WriteLine("Produto \t Quantidade \t Valor Unitário \t Valor Total \t");
```

```
8.             Console.WriteLine("Lata \t \t 5,00 \t \t 10,00 \t \t \t 50,00 ");
```

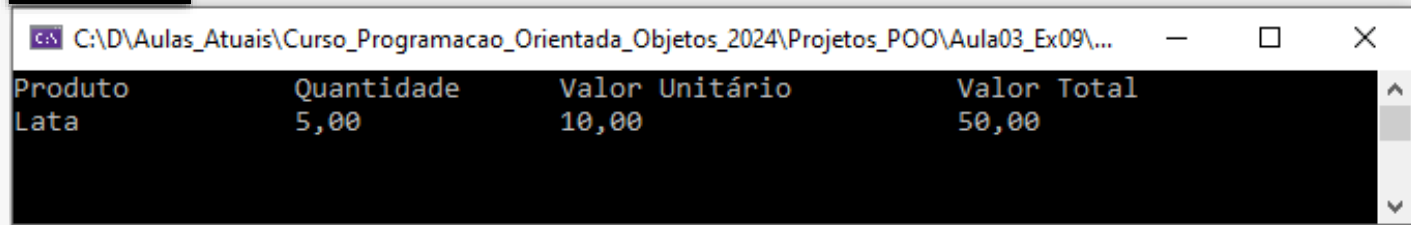
```
9.             Console.ReadKey();
```

```
10.         }
```

```
11.     }
```

```
12. }
```

Execução



Produto	Quantidade	Valor Unitário	Valor Total
Lata	5,00	10,00	50,00

Faz uma tabulação

Dois seguidos fazem duas tabulações

Inserir linhas com um comando Write

- ✓ Objetivo
 - ✓ Insere linha em branco com o comando Write

- ✓ Sintaxe
`Console.Write("Texto1 \n ");`

- ✓ Exemplo

```
1. namespace Aula03_Ex10
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             Console.WriteLine("Primeira linha \n");
8.             Console.WriteLine("Terceira linha");
9.             Console.ReadKey();
10.        }
11.    }
12.}
```

Execução

```
C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Obj
Primeira linha
Terceira linha
```

Insere uma linha em branco

