





Programação Orientada a Objetos

Ms. Eng. Claudio Ferreira de Carvalho



AULA 08

Métodos – Método Main() Variáveis Globais

Conceitos de Programação Estruturada x POO

- ✓ A linguagem C# foi concebida sob o paradigma de “Programação Orientada a Objetos” (POO).
- ✓ Esta filosofia de programação pode ser considerada como uma evolução do conceito de “Programação Estruturada”.
- ✓ Por sua vez, os conceitos de programação estruturada facilitou a escrituração de programas razoavelmente complexos dividindo-os em módulos ou subprogramas.
- ✓ Os módulos ou subprogramas são normalmente chamados em “Programação Estruturada” de Procedimentos e Funções.
 - ✓ **Procedimentos**, (Procedures) são subprogramas ou sub-rotinas que tem como objetivo realizar ações.
 - ✓ **Funções** (Functions) são subprogramas ou sub-rotinas que normalmente calculam e fornecem valores.

Método

- ✓ Método é um conjunto de códigos de uma linguagem de programação que realizam operações predefinidas pela linguagem ou criadas pelo programador.
- ✓ Basicamente existem dois tipos de Métodos:
 - ✓ **Métodos Internos:** São aqueles já existentes na linguagem. Estes métodos podem ser considerados como recursos da linguagem.
 - ✓ **Métodos externos:** São aqueles criados pelos programadores. Programadores caprichosos podem criar seus métodos e formar suas próprias bibliotecas, assim como incorporar à suas bibliotecas, módulos criados por outros programadores.
 - ✓ Obs: em C# um namespace é um pacote de métodos.
- ✓ Funcionamento
 - ✓ A construção de um método é feita com os mesmos comandos disponibilizados pela linguagem para escrituração de programas

Método

✓ Sintaxe

```
<qualificador> <tipo> <identificador> (parâmetro, parâmetro....)  
    <comando>;  
    <comando>;
```

qualificador:

Define como será o método para o C#. Pode ser:

private: Só pode ser acessado dentro da classe que foi criado.

public: Pode ser acessado dentro ou fora da classe que foi criado.

protect: Pode ser acessado dentro da classe que foi criado assim como pelas classes filhos das classes que estejam herdando as características da classe pai (classe que criou).

tipo:

Define o tipo de dado que irá retornar pela classe (Função). São utilizados os tipos de variáveis apresentados em aulas anteriores.

Se for um Procedimento (não manipula valores), neste caso, deve ser utilizada a palavra void.

identificador:

É o nome do Método, segue as mesmas regras dos identificadores de variáveis.

parâmetro:

São os tipos e identificadores que são passados pelo programa que chamou o método.

Método tipo procedimento

- ✓ Finalidade
 - ✓ Executar uma rotina
 - ✓ Não recebe nem passa valores
- ✓ Exemplo.
 - ✓ Método tipo procedimento para desenhar uma linha na tela

```
1. public static void Linha()  
2.     {  
3.         Console.WriteLine("-----");  
4.     }
```

- ✓ Exemplo.
 - ✓ Método tipo procedimento para encerrar um programa

```
1. // Mantém a tela aberta aguardando a digitação de uma tecla  
2. public static void Aguarda()  
3.     {  
4.         Console.WriteLine("\nDigite qualquer tecla para encerrar o programa.");  
5.         Console.ReadKey();  
6.     }
```

Método tipo procedimento - Exemplo

```
1. namespace Aula08_Ex01
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de variáveis
8.             string nome;
9.             char continuar;
10.            double compra, venda;
11.
12.            // solicitação de dados
13.            do
14.            {
15.                Console.Write("Forneça o nome do produto: ");
16.                nome = Console.ReadLine();
17.                Console.Write("Forneça o custo do produto: ");
18.                compra = double.Parse(Console.ReadLine());
19.                // Chama subprograma Linha
20.                Linha();
21.
22.                // Calcula o preço de venda
23.                if (compra <= 100)
24.                    venda = compra * 1.8;
25.                else
26.                    venda = compra * 1.5;
27.
28.                // Informa o preço de venda
29.                Console.WriteLine("Preço venda será: {0} ",
venda);
```

```
30.
31.            // Chama subprograma linha
32.            Linha();
33.
34.            Console.Write("Deseja calcular outro produto (S/N) ");
35.            continuar = char.Parse(Console.ReadLine());
36.            if ((continuar == 'S') || (continuar == 's'))
37.            {
38.                // Chama subprograma linha
39.                Linha();
40.                // Chama subprograma linha
41.                Linha();
42.            }
43.        }
44.        while ((continuar == 'S') || (continuar == 's'));
45.
46.        // Mantem tela aberta
47.        Aguarda();
48.    }
49.
50.    // Subprograma linha
51.    public static void Linha()
52.    {
53.        Console.WriteLine("-----
-----");
54.    }
55.
56.    // Mantém a tela aberta aguardando a digitação de uma tecla
57.    public static void Aguarda()
58.    {
59.        Console.WriteLine("\nDigite qualquer tecla para encerrar o
programa.");
60.        Console.ReadKey();
61.    }
62. }
63. }
```


Método tipo procedimento - Exemplo

Execução

```
1. namespace Aula08_Ex01
2. {
3.     internal class Programa
4.     {
5.         static void Main()
6.         {
7.             // Declaração de variáveis
8.             string nome;
9.             char continuar;
10.            double compra;
11.
12.            // solicitação de dados
13.            do
14.            {
15.                Console.WriteLine("Forneça o nome do produto: ");
16.                nome = Console.ReadLine();
17.                Console.WriteLine("Forneça o custo do produto: ");
18.                compra = double.Parse(Console.ReadLine());
19.                // Chama subprograma Linha
20.                Linha();
21.
22.                // Calcula o preço de venda
23.                if (compra <= 100)
24.                    venda = compra * 1.8;
25.                else
26.                    venda = compra * 1.5;
27.
28.                // Informa o preço de venda
29.                Console.WriteLine("Preço venda será: {0} ",
```

```
Forneça o nome do produto: Sapato
Forneça o custo do produto: 100
-----
Preço venda será: 180
-----
Deseja calcular outro produto (S/N) S
-----
Forneça o nome do produto: Camisa
Forneça o custo do produto: 200
-----
Preço venda será: 300
-----
Deseja calcular outro produto (S/N)
```

```

// Chama subprograma Linha
Linha();

Console.WriteLine("Deseja calcular outro produto (S/N) ");
continuar = char.Parse(Console.ReadLine());
if ((continuar == 'S') || (continuar == 's'))
{
    // Chama subprograma Linha
    Linha();
    // Chama subprograma Linha
    Linha();
}
while ((continuar == 'S') || (continuar == 's'));

// Mantém tela aberta
Aguarda();
}

// Subprograma Linha
public static void Linha()
{
    Console.WriteLine("-----");
    Console.WriteLine("-----");
}

// Mantém a tela aberta aguardando a digitação de uma tecla
public static void Aguarda()
{
    Console.WriteLine("\nDigite qualquer tecla para encerrar o
programa.");
    Console.ReadKey();
}
}
}
```

Método tipo função

- ✓ Finalidade
 - ✓ Executar uma rotina
 - ✓ Pode passar valores para o programa principal

Método tipo função – Exemplo

```
1. namespace Aula08_Ex02
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de variáveis
8.             string nome;
9.             char continuar;
10.            double compra;
11.
12.            // solicitação de dados
13.            do
14.            {
15.                Console.Write("Forneça o nome do produto: ");
16.                nome = Console.ReadLine();
17.                Console.Write("Forneça o custo do produto: ");
18.                compra = double.Parse(Console.ReadLine());
19.
20.                // Chama subprograma Linha
21.                Linha();
22.
23.                // Chama subprograma Calculo
24.                Calculo(compra);
25.
26.                Console.Write("Deseja calcular outro produto (S/N) ");
27.                continuar = char.Parse(Console.ReadLine());
28.                if ((continuar == 'S') || (continuar == 's'))
29.                {
30.                    // Chama subprograma Linha
31.                    Linha();
32.                    // Chama subprograma Linha
33.                    Linha();
34.                }
35.            }
36.        }
37.        while ((continuar == 'S') || (continuar == 's'));
```

```
38.
39.            // Mantém tela aberta
40.            Aguarda();
41.        }
42.        // Subprograma Linha
43.        public static void Linha()
44.        {
45.            Console.WriteLine("-----");
46.        }
47.        // Subprograma Calculo
48.        public static void Calculo(double compra)
49.        {
50.            double venda;
51.            if (compra <= 100)
52.                venda = compra * 1.8;
53.            else
54.                venda = compra * 1.5;
55.            Console.WriteLine("Preço venda {0} ", venda);
56.        }
57.
58.        // Mantém a tela aberta aguardando a digitação de uma
59.        tecla
60.        public static void Aguarda()
61.        {
62.            Console.WriteLine("\nDigite qualquer tecla para
63.            encerrar o programa.");
64.            Console.ReadKey();
65.        }
66.    }
```

Método tipo função – Exemplo

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projetos_1

```
1. namespace Aula08_Ex02
2. {
3.     internal class Program
4.     {
5.         static void Main(string[] args)
6.         {
7.             // Declaração de variáveis
8.             string nome;
9.             char continuar;
10.            double compra;
11.
12.            // solicitação de dados
13.            do
14.            {
15.                Console.WriteLine("Forneça o nome do produto: ");
16.                nome = Console.ReadLine();
17.                Console.WriteLine("Forneça o custo do produto: ");
18.                compra = double.Parse(Console.ReadLine());
19.
20.                // Chama subprograma Linha
21.                Linha();
22.
23.                // Chama subprograma Calculo
24.                Calculo(compra);
25.
26.                Console.WriteLine("Deseja calcular outro produto (S/N) ");
27.                continuar = char.Parse(Console.ReadLine());
28.                if ((continuar == 'S') || (continuar == 's'))
29.                {
30.                    // Chama subprograma Linha
31.                    Linha();
32.                    // Chama subprograma Calculo
33.                    Linha();
34.                }
35.            }
36.        }
37.        while ((continuar == 'S') || (continuar == 's'));
```

```
Forneça o nome do produto: Calça
Forneça o custo do produto: 100
-----
Preço venda 180
Deseja calcular outro produto (S/N) S
-----
Forneça o nome do produto: Camisa
Forneça o custo do produto: 200
-----
Preço venda 300
Deseja calcular outro produto (S/N) N
-----
Digite qualquer tecla para encerrar o programa.
```

```
        // Mantém tela aberta
        Aguarda();
    }
    // Subprograma Linha
    public static void Linha()
    {
        Console.WriteLine("-----");
    }
    // Subprograma Calculo
    public static void Calculo(double compra)
    {
        double venda;
        if (compra <= 100)
            venda = compra * 1.8;
        else
            venda = compra * 1.5;
        Console.WriteLine("Preço venda {0} ", venda);
    }
    // Mantém a tela aberta aguardando a digitação de uma
    // tecla
    public static void Aguarda()
    {
        Console.WriteLine("\nDigite qualquer tecla para
        encerrar o programa.");
        Console.ReadKey();
    }
}
```

Variáveis Locais e Variáveis Globais

✓ Variáveis locais

- ✓ Só podem ser utilizadas no programa ou subprograma que as declara.
- ✓ Variável local é declarada no programa principal main(),
- ✓ Caso se deseje utilizar o conteúdo destas variáveis em subprogramas elas tem que ser passadas para o subprograma e este deverá atribuir os conteúdos a outras variáveis locais a este subprograma.

✓ Variáveis globais

- ✓ São aquelas que podem ser utilizadas tanto no programa principal main() como nos subprogramas do mesmo namespace.
- ✓ As variáveis globais são declaradas como public de maneira a tornarem-se públicas, após o item class (que abre a classe), antes do início do programa principal (static void main(string[] args)).

Variáveis Locais e Variáveis Globais

- ✓ Sintaxe declaração de variáveis Globais
`public static <tipo> <identificador>.`

public: determina que o recurso se torna público ou seja visível de forma global.

Nota: caso a variável não seja global não é necessário digitar o domínio (public). Esta é a maneira como as variáveis foram declaradas nos programas anteriores do curso.

static: indica que o conteúdo permanecerá em uma parte da memória, sendo visível a outros métodos deste namespace.

tipo Define o tipo de dado que será declarado.

identificador: É o nome da variável.

Variável Global – Exemplo

```
1. namespace Aula08_Ex03
2. {
3.     internal class Program
4.     {
5.         // Declaração de variável global
6.         public static double venda;
7.
8.         static void Main(string[] args)
9.         {
10.            // Declaração de variáveis
11.            string nome;
12.            char continuar;
13.            double compra;
14.
15.            // solicitação de dados
16.            do
17.            {
18.                Console.Write("Forneça o nome do produto: ");
19.                nome = Console.ReadLine();
20.                Console.Write("Forneça o custo do produto: ");
21.                compra = double.Parse(Console.ReadLine());
22.                Console.WriteLine("Produto: {0}, Preço Compra {1:F2}, ", nome,
23. compra);
24.
25.                // Chama subprograma Linha
26.                Linha();
27.
28.                // Chama subprograma calculo
29.                Calculo(compra);
30.                Console.WriteLine("Preço venda {0} ", venda);
31.
32.                Console.Write("Deseja calcular outro produto (S/N) ");
33.                continuar = char.Parse(Console.ReadLine());
34.
35.                if ((continuar == 'S') || (continuar == 's'))
36.                {
37.                    // Chama subprograma Linha
38.                    Linha();
39.                    // Chama subprograma Linha
40.                    Linha();
41.                }
42.            }
43.            while ((continuar == 'S') || (continuar == 's'));
```

```
44.
45.         // Manté tela aberta
46.         Aguarda();
47.     }
48.
49.     // Subprograma Linha
50.     public static void Linha()
51.     {
52.         Console.WriteLine("-----");
53.         Console.WriteLine("-----");
54.     }
55.
56.     // Subprograma Calculo
57.     public static void Calculo(double compra)
58.     {
59.         if (compra <= 100)
60.             venda = compra * 1.8;
61.         else
62.             venda = compra * 1.5;
63.     }
64.
65.     // Mantém a tela aberta aguardando a digitação de uma
66.     // tecla
67.     public static void Aguarda()
68.     {
69.         Console.WriteLine("\nDigite qualquer tecla para
70. encerrar o programa.");
71.         Console.ReadKey();
72.     }
```


Variável Global – Exemplo

```
1. namespace Aula08_Ex03
2. {
3.     internal class Program
4.     {
5.         // Declaração de variável global
6.         public static double venda;
7.
8.         static void Main(string[] args)
9.         {
10.            // Declaração de variáveis locais
11.            string nome;
12.            char continuar;
13.            double compra;
14.
15.            // solicitação de dados
16.            do
17.            {
18.                Console.Write("Forneça o nome do produto: ");
19.                nome = Console.ReadLine();
20.                Console.Write("Forneça o custo do produto: ");
21.                compra = double.Parse(Console.ReadLine());
22.                Console.WriteLine("Produto: {0}, Preço Compra {1:F2}, ", nome,
23.                                compra);
24.
25.                // Chama subprograma Linha
26.                Linha();
27.
28.                // Chama subprograma calculo
29.                Calculo(compra);
30.                Console.WriteLine("Preço venda {0} ", venda);
31.
32.                Console.Write("Deseja calcular outro produto (S/N) ");
33.                continuar = char.Parse(Console.ReadLine());
34.
35.                if ((continuar == 'S') || (continuar == 's'))
36.                {
37.                    // Chama subprograma Linha
38.                    Linha();
39.                    // Chama subprograma Linha
40.                    Linha();
41.                }
42.            }
43.            while ((continuar == 'S') || (continuar == 's'));
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto_2024\Projetos_POO\Aul

```
Forneça o nome do produto: Calça
Forneça o custo do produto: 100
Produto: Calça, Preço Compra 100,00,
Preço venda 180
Deseja calcular outro produto (S/N) S
-----
Forneça o nome do produto: Camisa
Forneça o custo do produto: 200
Produto: Camisa, Preço Compra 200,00,
Preço venda 300
Deseja calcular outro produto (S/N) N
Digite qualquer tecla para encerrar o programa.
```

```
// Mantém tela aberta
Aguarda();
}

// Subprograma Linha
public static void Linha()
{
    Console.WriteLine("-----");
}

// Subprograma Calculo
public static void Calculo(double compra)
{
    if (compra <= 100)
        venda = compra * 1.8;
    else
        venda = compra * 1.5;
}

// Mantém a tela aberta aguardando a digitação de uma
tecla
public static void Aguarda()
{
    Console.WriteLine("\nDigite qualquer tecla para
encerrar o programa.");
    Console.ReadKey();
}
}
```


Variável Global – Exemplo

```
1. namespace Aula08_Ex04
2. {
3.     internal class Program
4.     {
5.         // Declaração de variáveis globais
6.         public static float r;
7.
8.         static void Main(string[] args)
9.         {
10.            // Declaração de variáveis locais
11.            float v1, v2;
12.
13.            // Tomada de valores
14.            Console.Write("Forneça o primeiro valor ");
15.            v1 = float.Parse(Console.ReadLine());
16.            Console.Write("Forneça o segundo valor ");
17.            v2 = float.Parse(Console.ReadLine());
18.            r = calculo(v1, v2);
19.            Console.WriteLine("O soma entre {0} e {1} é {2} ", v1, v2, r);
20.
21.            Console.ReadKey();
22.        }
23.        public static float Calculo(float a, float b)
24.        {
25.            r = a + b;
26.            return r;
27.        }
28.    }
29. }
```

Execução

C:\D\Aulas_Atuais\Curso_Programacao_Orientada_Objeto

```
Forneça o primeiro valor 6
Forneça o segundo valor 9
O soma entre 6 e 9 é 15
```

return:

Retorna o cálculo na variável publica r

