

# Métodos

de Moraes, Caique Vinícius. O Guia - Programação Orientada a Objetos:  
Domine os fundamentos da POO (p. 40). Edição do Kindle.

# Introdução

- Lembra o que é um método?
- **A FUNÇÃO QUE A CLASSE TEM!**

# Introdução

- Na programação orientada a objetos, um método é uma função associada a um **objeto** ou a uma **classe**.
- Quando um método é chamado a partir de um objeto, ele age sobre o estado desse objeto, e quando é chamado a partir da classe, ele age sobre a classe como um todo.

# Parâmetros

- Os métodos podem receber parâmetros de entrada e retornar valores de saída.
- Os parâmetros de entrada são os valores que o método precisa para realizar a tarefa desejada;
- **Podemos ter quantos valores de entrada nós quisermos!**
- Os valores de saída são os resultados produzidos pelo método após a execução;
- **Um método tem apenas um valor de saída!**

# Exemplo:

```
public static int Soma(int a, int b)
```

```
{
```

```
    int resultado = a + b;
```

```
    return resultado;
```

```
}
```

```
int resultado = Soma(2, 3);
```

# Modificadores de parâmetro de entrada

- Ref – sinaliza que o parâmetro de entrada **PODE** ser modificado pelo método;
- Out – sinaliza que o parâmetro de entrada **DEVE** ser modificado pelo método;

# Método de classe

- Um método de classe é um método estático que pertence à classe em si e não a uma instância específica dessa classe;
- Isso significa que você pode chamar um método de classe diretamente na classe, sem precisar criar uma instância do objeto;
- Os métodos de classe são definidos com a palavra-chave *static*;

# Método de classe

- Quando um membro da classe é declarado como *static*, ele pode ser acessado diretamente através do nome da classe, sem a necessidade de criar uma instância da classe.
- **Isso significa que não é necessário criar um objeto da classe para chamar o método ou acessar o campo.**



# Formulas da criação

- Método = modificador\_de\_acesso + tipo\_de\_saída +  
nome\_do\_método (tipo\_parâmetro nome\_parâmetro ,  
tipo\_parâmetro nome\_parâmetro...)  
  
{  
  
Conteúdo do método;  
  
Return valor\_saída\_do\_método;  
  
}

# Formulas da criação

```
public static int Soma (int a, int b)
{
    int resultado = a + b;
    return resultado;
}
```

Modificador de acesso: public (todos podem acessar o método) e private (apenas a classe pode acessar o método);

Tipo de saída: int, string, bool, double...

Parâmetros de entrada: int, string, bool, double...

# Construtores

de Moraes, Caique Vinícius. O Guia - Programação Orientada a Objetos:  
Domine os fundamentos da POO (p. 40). Edição do Kindle.

# Construtor

- O construtor é um tipo especial de método presente nas classes.
- Ele é acionado automaticamente assim que uma instância é criada.
- Um construtor sempre será público;
- Um construtor não possui parâmetro de saída;
- O objetivo principal do construtor é inicializar as propriedades e campos do objeto, definir valores padrão para as propriedades ou executar qualquer outra operação necessária para configurar o objeto.

# Construtor

- A sintaxe para criar um construtor em C# é muito semelhante à de um método regular, mas tem algumas diferenças importantes.
- O nome do construtor deve ser **o mesmo que o nome da classe**;
- **não** pode ter um tipo de retorno;

# **Diferença entre passagem de parâmetro e construtor**

# Formulas da criação

Modificador de acesso + mesmo nome da classe (parâmetros de entrada)

{

Atributo = parâmetro de entrada;

}

# Construtores

```
public Conta_Bancaria (double psaldo, string psenha_digitada, string pnome_cliente)

{

    saldo = psaldo;

    senha = psenha_digitada;

    nome_cliente= pnome_cliente;

}
```



# Criação do objeto com o construtor

```
Conta_Bancaria ContadaCarol = new Conta_Bancaria(100, "123456", "Carolina Leite Cardinale");
```

# Método Static

de Moraes, Caique Vinícius. O Guia - Programação Orientada a Objetos:  
Domine os fundamentos da POO (p. 40). Edição do Kindle.

# O método Static

- Eles são comumente usados para realizar tarefas que não dependem do estado de um objeto específico da classe;
- Por exemplo, um método *static* de uma classe de matemática pode ser usado para calcular a raiz quadrada de um número, independentemente dos valores dos outros membros da classe.

# Palavra chave *This*

- Referencia o contexto de um objeto;
- É uma referencia para a instância atual do objeto que está sendo manipulado neste instante;
- O *this* é utilizado para se referir aquilo que pertence a nossa classe;
- Diferencia membros de uma classe dos parâmetros ou variáveis locais com o mesmo nome;