



SENAI

Métodos

Fundamentos de Programação Orientada a Objeto

Método

Definição

Métodos são ações que podemos executar na classe.

Método

Exemplo

Em um Banco podemos realizar diferentes operações tais como:

- ✓ Depositar;
- ✓ Sacar;
- ✓ Consultar saldo;
- ✓ Etc.



Método

Classe Conta

Observem o método Depositar na classe Conta

0 referências

```
public class Conta
```

```
{
```

```
    //Atributos
```

0 referências

```
    public double saldo { get; set; }
```

0 referências

```
    public double limite { get; set; }
```

0 referências

```
    public int numero { get; set; }
```

```
    //Métodos
```

0 referências

```
    void Depositar(double valor)
```

```
{
```

```
        //implementação
```

```
}
```

```
}
```

Método

Anatomia

Um método contém:

- ✓ Nome – **verbo**;
- ✓ Lista de parâmetros – entre parênteses;
- ✓ Corpo / Implementação - código;
- ✓ Retorno – retorna ou não;
- ✓ Acesso – quem pode acessar este método.

```
public void Depositar(double valor)
{
    //Implementação
}
```

Acesso

Retorno

Nome

Parâmetro(s)

Corpo

```
public void Depositar(double valor)
{
    this.saldo += valor;
}
```

```
public class Conta
{
    //Atributos
    1 referência
    public double saldo { get; set; }
    0 referências
    public double limite { get; set; }
    0 referências
    public int numero { get; set; }
```

Método

Como usar?

Para realizar um depósito devemos chamar o método Depositar() usando a variável de instância do objeto.



```
Conta novaConta = new Conta();
```



```
6  class Program
7  {
8      0 referências
9      static void Main(string[] args)
10     {
11         Conta novaConta = new Conta();
12
13         Console.WriteLine($"Saldo inicial      = {novaConta.saldo}");
14
15         Console.Write("Digite o valor do depósito: ");
16         double vlr_dep = Convert.ToDouble(Console.ReadLine());
17         novaConta.Depositar(vlr_dep);
18
19         Console.WriteLine($"Saldo após depósito = {novaConta.saldo}");
20         Console.WriteLine("Depósito efetuado com sucesso!!!");
21         Console.ReadKey();
22     }
23 }
```

Método

Explicando melhor



```
6 class Program
7 {
8     0 referências
9     static void Main(string[] args)
10    {
11        Conta novaConta = new Conta();
12
13        Console.WriteLine($"Saldo inicial      = {novaConta.saldo}");
14
15        Console.Write("Digite o valor do depósito: ");
16        double vlr_dep = Convert.ToDouble(Console.ReadLine());
17
18        novaConta.Depositar(vlr_dep);
19
20        Console.WriteLine($"Saldo após depósito = {novaConta.saldo}");
21        Console.WriteLine("Depósito efetuado com sucesso!!!");
22        Console.ReadKey();
23    }
```

```
0 referências
public class Conta
{
    //Atributos
    0 referências
    public double saldo { get; set; }
    0 referências
    public double limite { get; set; }
    0 referências
    public int numero { get; set; }

    //Métodos
    0 referências
    void Depositar(double valor)
    {
        //implementação
    }
}
```

Linha 10: Criando a referência do objeto - **novaConta**

Linha 12: Exibindo o saldo inicial

Linhas 14 e 15: Obtendo o valor do depósito - **vlr_dep**

Linha 17: Através da referência do objeto, **CHAMAMOS** o método **Depositar(vlr_dep)** passando o valor do depósito

Linha 19: Exibindo o novo valor do **saldo**

Método

Observações:

- ✓ O método Depositar() está alterando o valor armazenado no atributo da classe: **novaConta.Depositar(vlr_dep);**
- ✓ Dentro de um método, para acessar os atributos da classe devemos usar a palavra reservada **this**;
- ✓ O método **não** retorna nenhum valor para quem o chamou - **void**

```
public class Conta
{
    //Atributos
    3 referências
    public double saldo { get; set; }
    0 referências
    public double limite { get; set; }
    0 referências
    public int numero { get; set; }

    //Métodos
    1 referência
    public void Depositar(double valor)
    {
        this.saldo += valor;
    }
}
```

Método

Podemos também ...

No exemplo anterior, o método Depositar() **NÃO** retorna nenhum valor para quem o chamou.

Concordam?

Observem a palavra chave **void**.

```
public class Conta
{
    //Atributos
    3 referências
    public double saldo { get; set; }
    0 referências
    public double limite { get; set; }
    0 referências
    public int numero { get; set; }

    //Métodos
    1 referência
    public void Depositar(double valor)
    {
        this.saldo += valor;
    }
}
```

Como ficaria o código com o método **retornando** o valor do saldo?

Entenderam a pergunta?

Observem o código a seguir ...

Método sem retorno

```
public class Conta
{
    //Atributos
    3 referências
    public double saldo { get; set; }
    0 referências
    public double limite { get; set; }
    0 referências
    public int numero { get; set; }

    //Métodos
    1 referência
    public void Depositar(double valor)
    {
        this.saldo += valor;
    }
}
```

Método com retorno

```
public class Conta
{
    //Atributos
    3 referências
    public double saldo { get; set; }
    0 referências
    public double limite { get; set; }
    0 referências
    public int numero { get; set; }

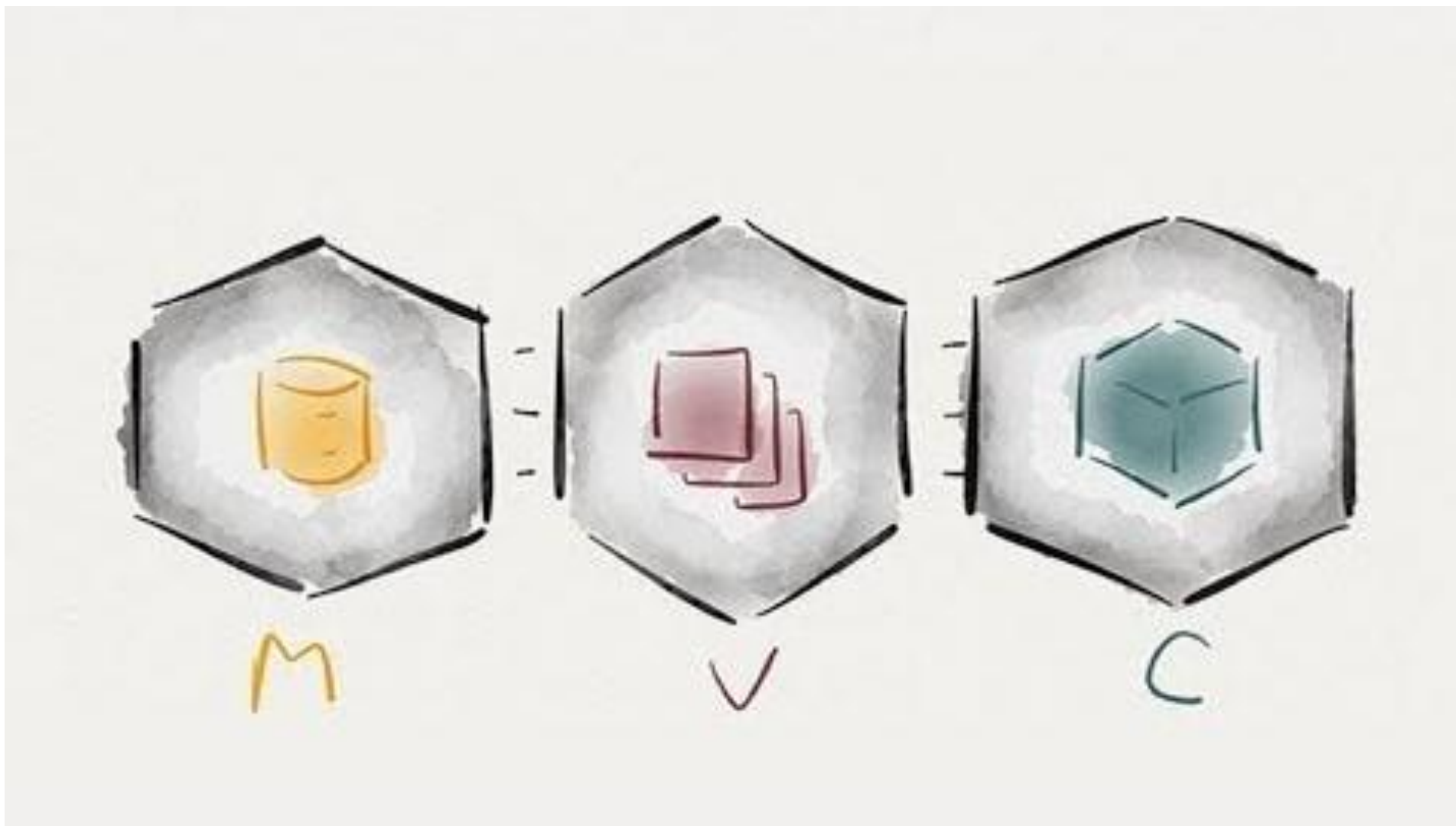
    //Métodos
    1 referência
    public double Depositar(double valor)
    {
        return this.saldo += valor;
    }
}
```

Método

Exercício de fixação ...

Implementar alguns métodos na classe Conta:

- ✓ Sacar();
- ✓ ConsultarSaldo();
- ✓ TransferirValor();





Atila Andreatti Olivi

atila.olivi@senaisp.edu.br

(11) 5642-3400

Escola SENAI Suíço Brasileira “Paulo E. Tolle”

R. Bento Branco de Andrade Filho, 379