

02

Criando um objeto-método a partir de um método

1) Vamos primeiro criar uma nova classe dedicada para efetuar os cálculos do preço, chamada `CalculadoraDePrecos` :

```
class CalculadoraDePrecos
{
}
```

2) Então vamos criar um campo privado nessa classe para armazenar uma instância da classe `Produto` :

```
class CalculadoraDePrecos
{
    private readonly Produto produto;
```

3) Então precisamos de um construtor parametrizado para a nova classe. Vamos aproveitar que o Visual Studio permite criar esse construtor facilmente, com CTRL + PONTO + "Create constructor":

```
class CalculadoraDePrecos
{
    private readonly Produto produto;

    public CalculadoraDePrecos(Produto produto)
    {
        this.produto = produto;
    }
}
```

Note que o Visual Studio já fez pra gente a atribuição dos parâmetros aos campos privados da classe!

4) Crie um novo método na classe para calcular o preço. Ele irá retornar um valor decimal:

```
public decimal Calcular()
{
}
```

5) Então copie para dentro do método `Calcular` o corpo do método original `Preco` :

```
public decimal Calcular()
{
    var resultado = precoBase;

    if (this.promocional && desconto > 0)
    {
        throw new Exception("Produto já é promocional e não pode ter desconto!");
    }

    if (desconto > 20)
```

```
{  
    desconto = 20;  
}  
  
if (acrescimo > 15)  
{  
    acrescimo = 15;  
}  
  
return precoBase + precoBase * (acrescimo - desconto);  
}
```

6) Agora precisamos ajustar o algoritmo do método. Os campos `promocional`, `precoBase`, `acrescimo` e `desconto` devem ser substituídos pelas propriedades `produto.Promocional`, `produto.PrecoBase`, `produto.Acrescimo` e `produto.Desconto`

```
public decimal Calcular()  
{  
    var resultado = produto.PrecoBase;  
  
    if (this.produto.Promocional && produto.Desconto > 0)  
    {  
        throw new Exception("Produto já é promocional e não pode ter desconto!");  
    }  
  
    if (produto.Desconto > 20)  
    {  
        desconto = 20;  
    }  
  
    if (produto.Acrescimo > 15)  
    {  
        acrescimo = 15;  
    }  
  
    return produto.PrecoBase + produto.PrecoBase * (produto.Acrescimo - produto.Desconto);  
}
```

7) Vamos substituir a chamada ao método `Preco`, criando uma nova instância da classe `CalculadoraDePrecos` e chamando o método `Calcular()` a partir dela:

```
return new CalculadoraDePrecos(this).Calcular();
```

8) Para encerrar, vamos remover o método `Produto.Preco()`.