
C#
.NET
Core

Alex Sander Resende de Deus

A 25 anos ensinando programação a jovens e adultos.

Apaixonado por tecnologia é atualmente coordenador de cursos na ETEC Albert Einstein. Na FIAP atua como professor da FIAP School, lecionando C#, SQLServer e Desenvolvimento Mobile

AULA 7

Métodos

Assinatura de métodos

A primeira linha do método é chamada de assinatura e ela indica:

- Visibilidade
- Retorno de valor
- O nome do método
- Passagem de parâmetros

- Exemplo: Sem retorno e sem parâmetros.
  - Na classe de modelagem:

```
public void cadastraProduto(){

}
```

  - No chamado do método:

```
cadProd.cadastraProduto();
```

Exemplo: Sem retorno e com parâmetros.

- Na classe de modelagem:

```
public void cadastraProduto(int quantidade){

}
```

- No chamado:

```
cadProd.cadastraProduto(qtde);
```

- Exemplo: Com retorno e sem parâmetros.
  - Na classe de modelagem:

```
public boolean cadastraProduto(){

    return true;
}
```

  - No chamado

```
boolean resposta = cadProd.cadastraProduto();
```

- Exemplo: Com retorno e com parâmetros.
  - Na classe de modelagem:

```
public boolean cadastraProduto(int quantidade){

    return true;
}
```

  - No chamado

```
boolean resposta = cadProd.cadastraProduto(qtde);
```

- Exemplo: Passando um objeto por parâmetro.
  - Na classe de modelagem:

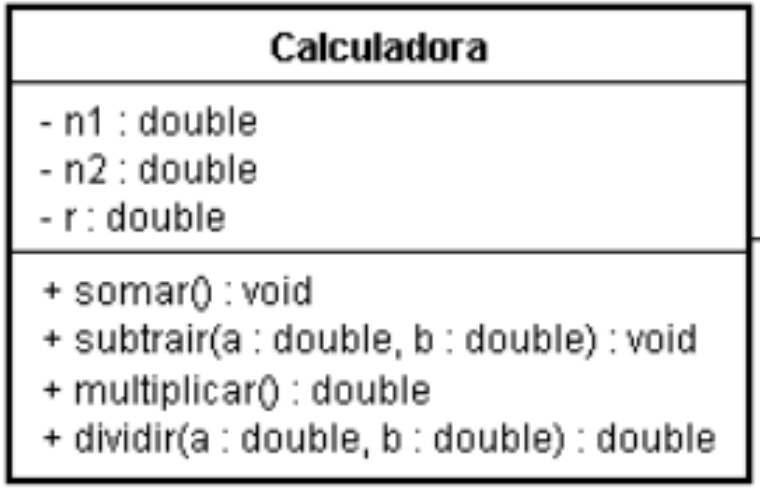
```
public void cadastraProduto(Produto p){

}
```

  - No chamado:

```
cadProd.cadastraProduto(prod);
```

Momento Hands On

- Refazendo a Calculadora



Classe Calculadora

```
class Calculadora
{
    public double n1 { get; set; }
    public double n2 { get; set; }
    public double res { get; set; }

    public Calculadora()
    {

    }

    public void somar()
    {
        res = n1 + n2;
    }
```

```
public void subtrair(double a, double b)
{
    res = a - b;
}

public double multiplicar()
{
    res = n1 * n2;
    return res;
}

public double dividir(double a, double b)
{
    res = n1 / n2;
    return res;
}
```

Código dos botões

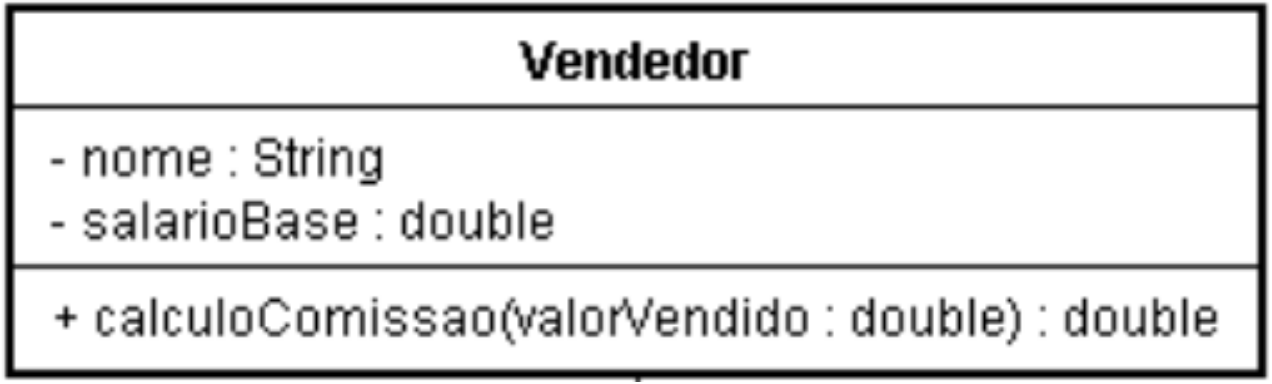
```
private void btnSomar_Click(object sender, EventArgs e)
{
    Calculadora calc = new Calculadora();
    calc.n1 = Convert.ToDouble(txtN1.Text);
    calc.n2 = Convert.ToDouble(txtN2.Text);
    calc.somar();
    lblResp.Text = calc.res.ToString();
}

private void btnSubtrair_Click(object sender, EventArgs e)
{
    Calculadora calc = new Calculadora();

    calc.subtrair(Convert.ToDouble(txtN1.Text),Convert.ToDouble(txtN2.Text));
    lblResp.Text = calc.res.ToString();
}
```

Código dos botões

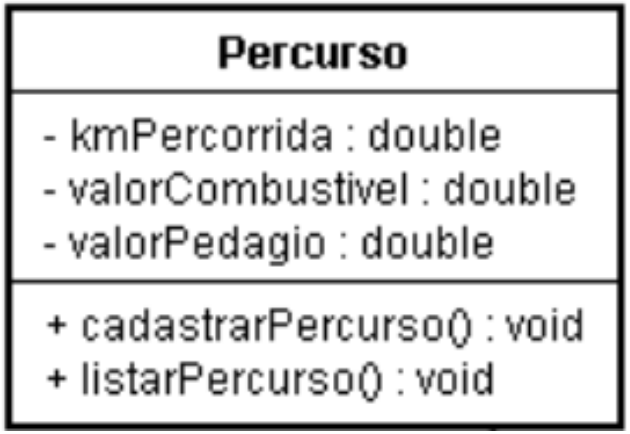
```
private void btnMultiplicar_Click(object sender, EventArgs e)
{
    Calculadora calc = new Calculadora();
    calc.n1 = Convert.ToDouble(txtN1.Text);
    calc.n2 = Convert.ToDouble(txtN2.Text);
    lblResp.Text=calc.multiplicar().ToString();
}

private void btnDividir_Click(object sender, EventArgs e)
{
    Calculadora calc = new Calculadora();
    lblResp.Text = calc.dividir(Convert.ToDouble(txtN1.Text), Convert.ToDouble(txtN2.Text)).ToString();
}
```

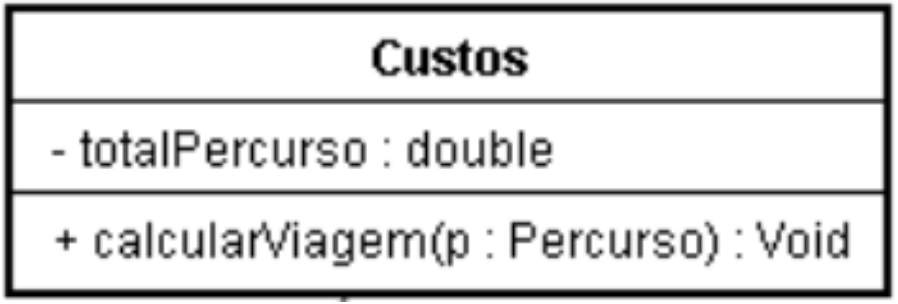
Para treinar mais

Exercício 01



Exercício 02





Tratando erros
Bloco Try Catch

É possível que uma destas tarefas gere um erro ou exceção, isto é, uma ocorrência que faz com que o fluxo normal do programa seja alterado. Esse erro ou exceção pode ser simples ou complexo: em algumas situações, a execução do programa simplesmente continua, em outras ela pode ser interrompida abruptamente.

Por estes motivos, devemos dar atenção muito especial à detecção e manipulação de erros em nossas aplicações. No C#, contamos com um mecanismo sofisticado que nos auxilia a produzir códigos de manipulação organizados e muito eficientes, que é a manipulação de exceções.

O manipulador de exceções é um código que captura a exceção lançada e a manipula. Sua principal vantagem é que precisamos escrever apenas uma vez o código de manipulação de uma exceção que pode ocorrer em uma região controlada.

O bloco de código onde as exceções podem ocorrer é chamado de região protegida. Ele é indicado pelo comando try. Para associarmos uma determinada exceção a um bloco de código que a manipulará, usamos uma ou mais cláusulas catch

Exemplos

```
try

        {

            //linha de código que pode disparar uma exceção

            int numero = Convert.ToInt32(txtValor.Text);

            int resultado = 100 / numero;

        }

        catch

        {

            //linha de código que trata o erro

            lblMensagem.Text = "Número inválido!";

        }
```

```
try

        {

            int numero = Convert.ToInt32(txtValor.Text);

            int resultado = 100 / numero;

        }

        catch (Exception ex)

        {

            lblMensagem.Text = ex.Message;

        }
```

Momento Hands On

Agora que já sabemos tratar exceções, que tal inserir o bloco try catch nos exercícios feitos na aula de hoje?

Vamos lá! É com vocês!

OBRIGADO

profalex.deus@fiap.com.br

linkedin.com/in/alexsanderresende

FIAP MBA+


FIAP