



# **Linguagem de Programação II**

Ciência da Computação

Prof. Me. Renato Carioca Duarte



# Classes e Objetos

- Uma classe é considerada um tipo de dado, como os tipos básicos pré-definidos (inteiro, real, string, etc.) em todas as linguagens de programação.
- Sendo assim da mesma forma que podemos criar uma variável do tipo “int” também podemos criar uma variável do tipo da classe. Esta variável da classe é chamada de OBJETO.
- Ao definirmos uma classe, estamos fornecendo a **estrutura para o objeto**.
- Toda vez que declaramos uma variável originada de uma classe dizemos que estamos instanciando um objeto.
- Quando definimos uma classe estamos criando um modelo para **criarmos objetos** e/ou outras **classes(herança)**.



# Classe: Cliente de Banco

- Toda classe Cliente vai ter **atributos** nome, cpf, rg e endereço:

```
[-] namespace ContaBancariaConsole2020
    {
        2 referências
        [-] class Cliente
            {
                public string nome;
                public string cpf;
                public string rg;
                public string endereco;
            }
    }
```



# Classe: Conta Bancária

- Toda classe Conta vai ter **atributos** numero, titular e saldo:

```
[-] namespace ContaBancariaConsole2020
    {
        1 referência
        [-] class Conta
            {
                public int numero;
                public Cliente titular;
                public double saldo;
            }
    }
```



# Métodos

- Além de atributos, os objetos também podem possuir métodos. Os métodos são blocos de código que isolam lógicas de negócio do objeto.
- Quando queremos passar um valor para um método, precisamos passar esse valor dentro dos parênteses da chamada do método.
- Para recebermos o valor que foi passado, precisamos declarar um argumento no método.
- O argumento é uma variável declarada dentro dos parênteses do método.
- Um método pode ter qualquer número de argumentos. Precisamos apenas separar a declaração das variáveis com uma vírgula.



# Classe: Conta Bancária

- Toda classe Conta vai ter **métodos**:

```
2 Referências  
public double MostraSaldo()  
{  
    ...  
    return this.saldo;  
}
```



# Classe: Conta Bancária

- Toda classe Conta vai ter **métodos**:

```
public bool Sacar (double valor)
{
    if (this.saldo >= valor)
    {
        this.saldo = this.saldo - valor;
        return (true);
    }
    else
    {
        return (false);
    }
}
```



# Métodos

- No método Sacar, queremos verificar o saldo da conta em que o método foi invocado.
- Para acessarmos a referência em que um determinado método foi chamado, utilizamos a palavra **this** .
- Então para acessarmos o saldo da conta, podemos utilizar `this.saldo` :
- A palavra-chave **this** refere-se à instância atual da classe.
- Quando um método devolve um valor, o tipo do valor devolvido deve ficar antes do nome do método em sua declaração.
- Quando um método não devolve valor algum, utilizamos o tipo **void** .
- Dentro da implementação do método, devolvemos um valor utilizamos a palavra **return** seguida do valor que deve ser devolvido.





# Classe: Conta Bancária

- Toda classe Conta vai ter **métodos**:

1 referência

```
public int MostraNumero()  
{  
    return this.numero;  
}
```

1 referência

```
public string MostraTitular()  
{  
    return titular.nome + " " + titular.cpf;  
}
```



# Executando o programa: MAIN

```
static void Main(string[] args)
{
    Cliente cliente1 = new Cliente();
    cliente1.nome = "Joaquim Jose";
    cliente1.cpf = "988888";
    cliente1.rg = "12345";
    cliente1.endereco = "Rua ZZZZ, 23";

    Conta c1 = new Conta();
    c1.numero = 1;
    c1.titular = cliente1;
    c1.saldo = 1500;

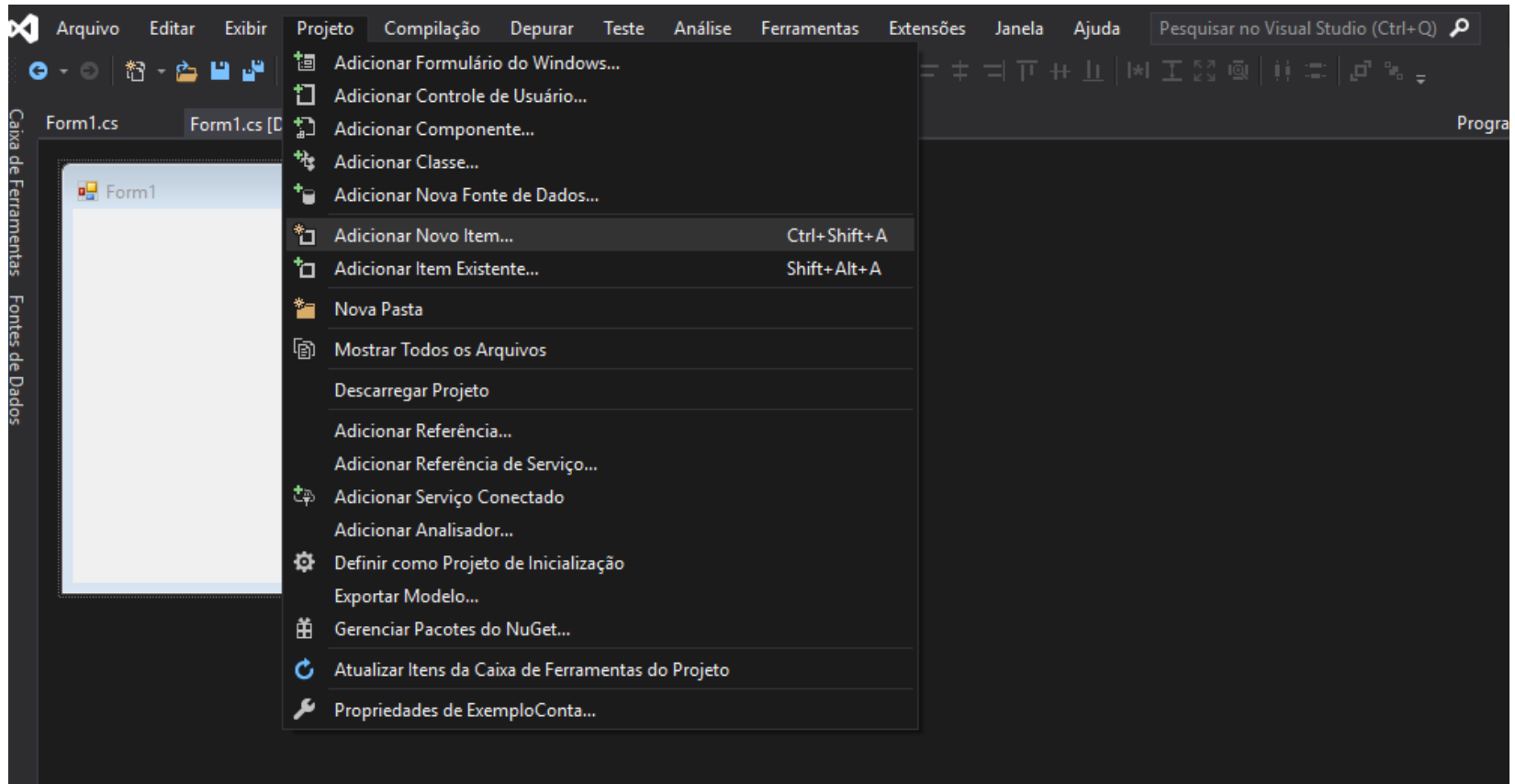
    Console.WriteLine(c1.MostraNumero());
    Console.WriteLine(c1.MostraTitular());
    Console.WriteLine(c1.MostraSaldo() );
    c1.Sacar(350);
    Console.WriteLine(c1.MostraSaldo());

    Console.ReadKey();
}
```



# Criando Classe no Visual Studio

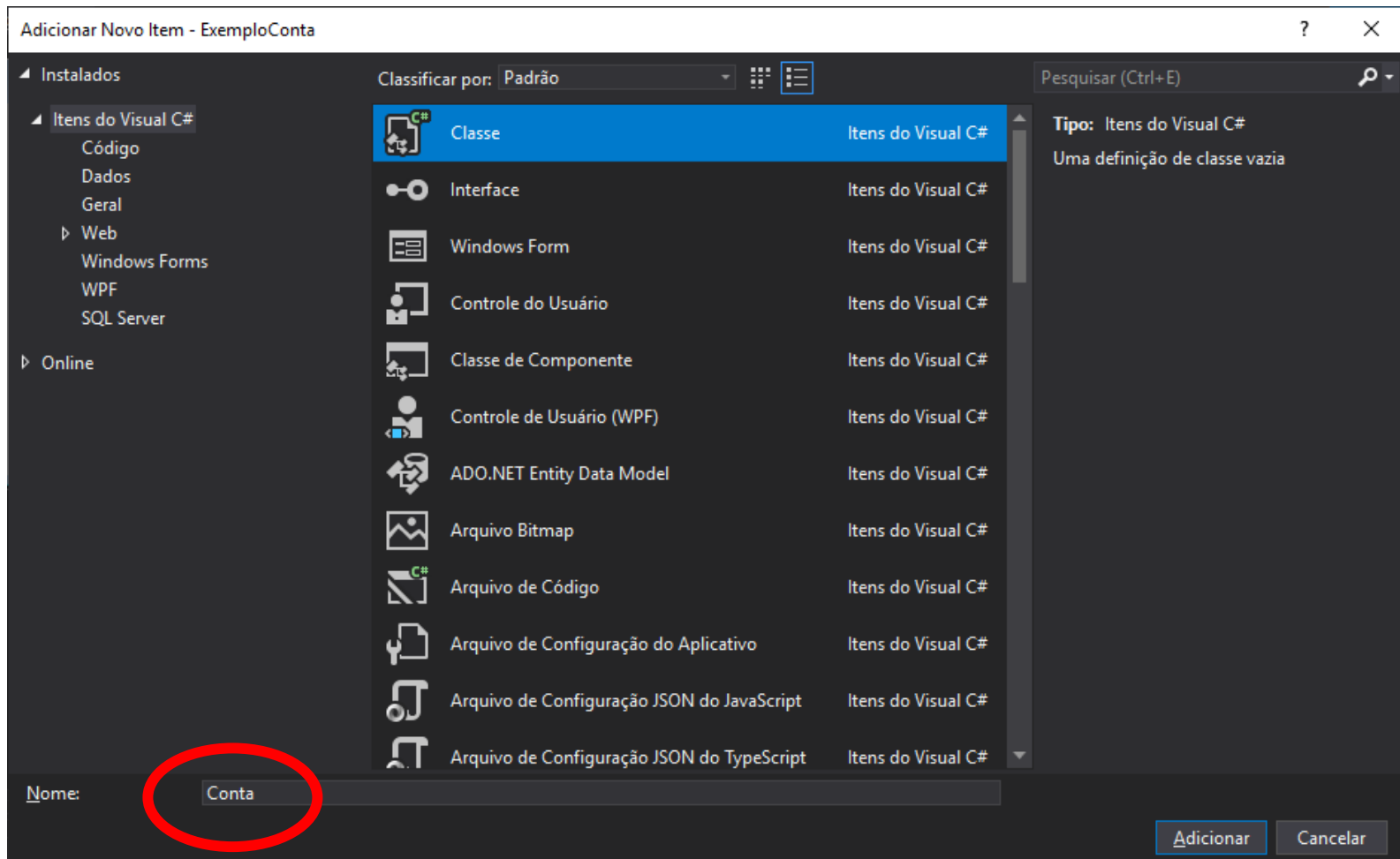
## Projeto / Adicionar Novo Item





# Criando Classe no Visual Studio

Projeto / Adicionar Novo Item / Classe / Nome:





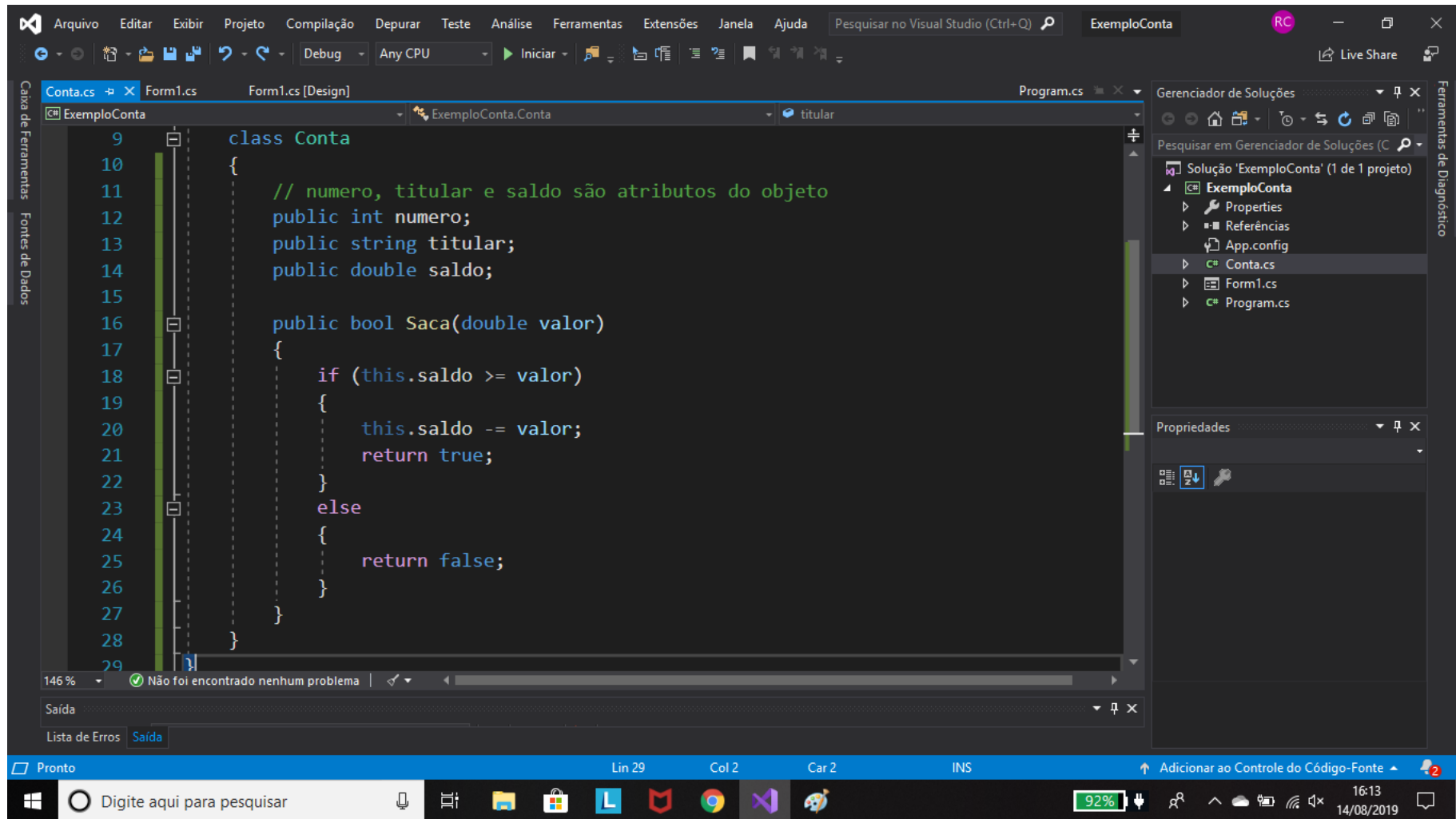
# Criando Classe no Visual Studio

A screenshot of the Visual Studio IDE interface. The top menu bar includes options like 'Arquivo', 'Editar', 'Exibir', 'Projeto', 'Compilação', 'Depurar', 'Teste', 'Análise', 'Ferramentas', 'Extensões', 'Janela', and 'Ajuda'. Below the menu is a toolbar with icons for file operations and debugging. The main window displays a C# code file named 'Conta.cs'. The code defines a namespace 'ExemploConta' and a class 'Conta' within it. The class 'Conta' is currently empty, with only opening and closing curly braces visible. The code is as follows:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ExemploConta
8 {
9     class Conta
10    {
11    }
12 }
13
14
15
```



# Criando Classe no Visual Studio





# Exercícios

1. Fazer o método depositar, informando o valor a ser depositado.
2. Fazer método transferir, informando o valor a ser transferido de uma conta para outra.