

# Aula 1: Introdução a C# - Fundamentos da Linguagem

Professor(a): Virgínia Fernandes Mota  
[virginia@teiacoltec.org](mailto:virginia@teiacoltec.org)

TECNOLOGIAS DE PROGRAMAÇÃO - SETOR DE INFORMÁTICA

# Introdução a C#

- Ao término desta aula, você será capaz de:
  - ▶ Seus primeiros passos com C#
  - ▶ Saiba as diferenças de variáveis primitivas
  - ▶ Tome decisões com condicionais
  - ▶ Facilite a repetição com o FOR e WHILE
  - ▶ Representando entidades com classes
  - ▶ Extraíndo comportamentos usando métodos

# Introdução a C#

- `MessageBox.show("Oi mundo");`

# Introdução a C#

- Mas e se eu quiser só imprimir na Console??

```
1 public class Hello
2 {
3     static void Main()
4     {
5         System.Console.WriteLine("Hello World");
6     }
7 }
```

# Variáveis e Operações

- Queremos também armazenar e processar informações.
- Como funcionam as variáveis e operações no C#?

```
1  int  numero;  
2  double  outronumero;  
3  
4  numero = 42;  
5  outronumero = 10.0 + numero;  
6  
7  MessageBox.show("Meus n meros s o " + numero + "e " +  
    outronumero);
```

- As operações são as mesmas que já conhecemos +, -, \*, /
- Em C# também podemos utilizar +=, ++, /= e cia.

# Variáveis e Operações

- As strings são também muito parecidas com Java.

```
1  string mensagem1 = "Oi, sou uma string";  
2  string mensagem2 = "e eu tb";  
3  
4  //juntamos strings com +  
5  MessageBox.show(mensagem1 + mensagem2);
```

# Variáveis e Operações

- Este código compila?

```
1 int numero = 3.14;  
2 double outronumero = Math.Sqrt(2);
```

- O código acima não compila, pois não é possível representar o número 3.14 dentro de um inteiro.
- O C# nunca permitirá que você tente guardar algo "maior" do que cabe na variável que você está usando.
- Ao menos que façamos um casting (int) 3.14.

# Estruturas de Controle

- Tomando decisões

```
1 int idade = 18;
2 //a condi o uma express o booleana
3 if (idade > 18)
4     MessageBox.show("Pode ir na festa");
5 else if (idade == 18)
6     MessageBox.show("Pode ir na festa na rabeira");
7 else
8     MessageBox.show("Fique em casa");
```

- Operações booleanas &&, ||
- Demais operações <, <=, >, >=, ==



# Estruturas de Repetição

- Fazendo os amigos loops

```
1 //investindo dinheiro
2 double valorinvestido = 2000.0;
3 for (int i = 1; i <= 12; i++){
4     valorinvestido *= 1.01;
5
6 }
7 MessageBox.show("Valor investido agora      :" + valorinvestido)
8 ;
```

```
1 //investindo dinheiro
2 double valorinvestido = 2000.0;
3 int i;
4 while (i <= 12) {
5     valorinvestido *= 1.01;
6     i+=1;
7 }
8 MessageBox.show("Valor investido agora      :" + valorinvestido)
9 ;
```

- Também temos o do ... while ;)

# Classes e objetos

- Declarando uma classe em C#. Lembram da nossa conta em Java? Como ela ficaria aqui?

```
1 //equivalente aos imports
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text
6 //equivalente a namespace do C++ e using package do Java
7 using namespace CaixaEletronico{
8     //nossa classe Conta
9     class Conta{
10         public int numero;
11         public string titular;
12         public double saldo;
13     }
```

- E para instanciar?

```
1 Conta c1 = new Conta();
2 c1.numero = 42;
3 c1.titular = "Seu fulaninho muito rico";
4 c1.saldo = 189023809128.0;
```

# Classes e objetos

- Lembrando que objetos são apenas referências!

```
1  Conta c1 = new Conta();
2  c1.numero = 42;
3  c1.titular = "Seu fulaninho muito rico";
4  c1.saldo = 189023809128.0;
5
6  Conta c2 = new Conta();
7  c2.numero = 42;
8  c2.titular = "Seu fulaninho muito rico";
9  c2.saldo = 189023809128.0;
10
11 if (c1 == c2){
12     //nunca ser o iguais pois s o refer ncias      objetos
13     diferentes
14 }
15 c1 = c2; // agora c1 e c2 referenciam a mesma coisa.
```

# Métodos

```
//equivalente aos imports
(...)
//equivalente a namespace do C++ e using package do Java
using namespace CaixaEletronico{
    //nossa classe Conta
    class Conta{
        //atributos
        (...)

        //m todos
        public void Sacar(double valor){
            this.saldo -= valor;
        }
        public void Deposita(double valor){
            this.saldo += valor;
        }
        public void Transfere(double valor, Conta destino){
            this.Sacar(valor);
            destino.Deposita(valor);
        }
    }
}
```

# Métodos

- E para validação dos nossos saques e depósitos?
- Podemos usar as condicionais como fizemos em Java!
- E para métodos com retorno? Mesma coisa!!

```
1     public double CalculaRendimentoAnual(){
2         rendimento = this.saldo * 1.0007;
3         return rendimento;
4     }
```

# Composição de Classes

```
class Cliente{  
    public string nome;  
    public string rg;  
    public string endereco;  
}
```

```
class Conta{  
    public Cliente titular;  
    public int numero;  
    public double saldo;  
}
```

- Lembrar sempre: Instanciar todos os objetos!!!

Na próxima aula...

Orientação a Objetos com C#