



Colégio  
Pedro II

# PROGRAMAÇÃO O.O. (C#)

---



Classes de Controle x Classes de Entidade: static  
**Professor:** João Luiz Lagôas



# Classe de Controle x Classe de Entidade

## Classe de Controle

Quando temos uma classe composta por diversos atributos e métodos que:

- não precisa de um objeto (de uma entidade) para fazer sentido dentro do contexto do sistema;
- serve como “uma pasta” para agregar várias funções;
- realiza puramente tarefas de controle e gerenciamento;
- é comum o uso da palavra reservada **static** na assinatura dos atributos e métodos.

## Classe de Entidade

Quando temos uma classe composta por diversos atributos e métodos que:

- precisa de um objeto (de uma entidade) para fazer sentido dentro do contexto do sistema;
- serve como “um modelo” para gerar (instanciar) novos objetos;
- realiza tarefas que mudam o estado dos objetos (como, por exemplo, alterar seus atributos);
- não é tão comum o uso da palavra reservada **static** na assinatura dos atributos e métodos.

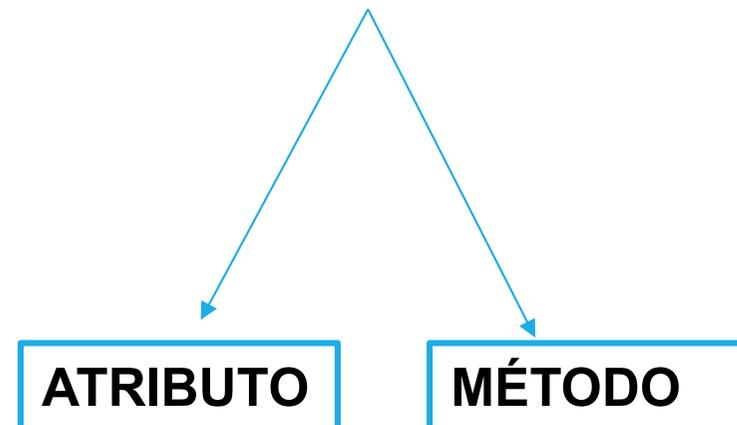
# Palavra Reservada: *static*



Um atributo ou método assinado com a palavra reservada *static* existe independentemente de um objeto. Isto é, você pode acessar este atributo ou chamar este método SEM PRECISAR criar um objeto da classe, basta preceder o acesso ou a chamada com o nome da classe.

- Note que tais atributos e métodos pertencem à classe onde foram declarados como um todo e não a um objeto.
- Por esse motivo, costumamos chamar esses atributos e métodos como campos de classe e não de objeto.

*static*



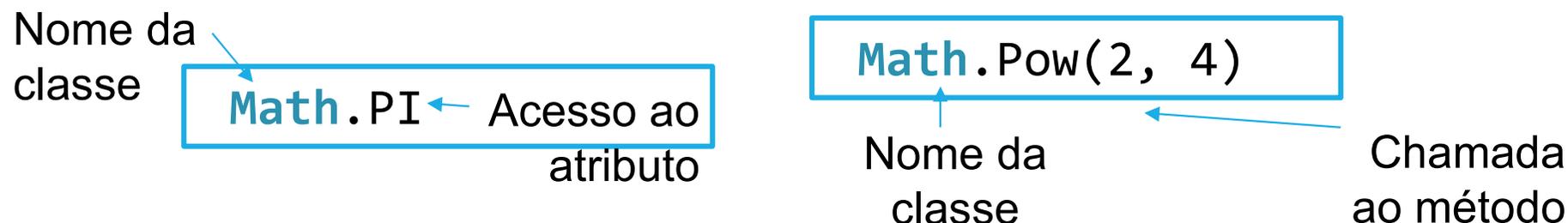
Não precisam de uma instância para serem usados!



# Exemplo

## Atributos e Métodos Estáticos

- Até aqui já utilizamos vários métodos estáticos! Observe o exemplo:



- Você pode chamar qualquer atributo ou método estático especificando o nome da classe onde o método foi declarado, seguido pelo operador de acesso a membro (.) e o nome do método como.

`Console.ReadLine()`

`Convert.ToInt32("78")`

`Console.WriteLine("LP2 é irado")`

`Convert.ToDouble("2")`



# Exemplo

## Métodos Estáticos

- Nós usamos vários métodos estáticos das classes **Math**, **Console** e **Convert**.
- A classe **Math**, por exemplo, provê uma coleção de métodos que te habilitam a realizar cálculos comuns matemáticos.

**Mas como o nosso programa entende que esses comandos já programados `Math.Pow()`, `Math.Sqrt()`, `Console.WriteLine()`, etc, podem ser usados no nosso código??**

# The .NET Framework Class Library



- Muitas classes predefinidas estão agrupadas em categorias de classes relacionadas chamadas de **namespaces**. Juntos, esses namespaces são referidos como .NET Framework Class Library (**FCL**).
- Através de diretivas precedidas pela palavra reservada **using**, especificamos qual namespace queremos que esteja presente em nosso projeto.

```
using System;
```

Diretiva para importar todas as classes que estão no namespace **System**

No site: [https://msdn.microsoft.com/pt-br/library/system\(v=vs.110\).aspx](https://msdn.microsoft.com/pt-br/library/system(v=vs.110).aspx)  
há uma referência completa de todas as classes presentes no namespace System



# namespace System

- Até agora já utilizamos algumas das classes presentes no namespace System, são elas:

	<b>Console</b>	Representa os fluxos de entrada, saída e erro padrão para aplicativos de console.
	<b>Convert</b>	Converte um tipo de dados base em outro tipo de dados base.
	<b>Math</b>	Fornece constantes e métodos estáticos para trigonométricas, logarítmicas e outras funções matemáticas comuns.

namespace  
System