



# Aula 17

Prof: Henrique Augusto Maltauro

# Desenvolvendo Algoritmos

# Programação Orientada a Objetos (POO)

- Quatro pilares da POO

Principais fundamentos da POO.

- Abstração ✓
- Herança
- Encapsulamento
- Polimorfismo

# Programação Orientada a Objetos (POO)

- Quatro pilares da POO (herança)

A herança é um dos principais fundamentos da POO, que permite que classes herdem atributos e métodos de outras classes.

Ou seja, eu posso compartilhar atributos e métodos entre as classes.

# Programação Orientada a Objetos (POO)

- Quatro pilares da POO (herança)
  - (vamos imaginar um carro e uma moto)

Um carro tem atributos e funcionalidades diferentes de uma moto.

Contudo, ao mesmo tempo, eles têm atributos e funcionalidades similares entre eles, e isso se deve ao fato de que ambos são veículos.

# Programação Orientada a Objetos (POO)

- Quatro pilares da POO (herança)
  - (vamos imaginar um carro e uma moto)

Nesse contexto, se fossemos trazer isso para dentro da estrutura de POO, podemos dizer que as classes carro e moto, compartilham atributos e funcionalidades através da classe veículo.

# Programação Orientada a Objetos (POO)

- Quatro pilares da POO (herança)

Ela é usada na intenção de reaproveitar código e de criar um comportamento generalizado entre as classes.

**Uma classe só pode herdar de uma única classe.**

# Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (herança)

```
public class Veiculo
{
    public string Modelo;
    public string Motor;

    public void Ligar()
    {
        // Bloco de código do método Ligar
    }
}
```

# Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (herança)

```
public class Carro : Veiculo
{
    // Agora a classe Carro tem
    // todos os atributos e
    // métodos da classe Veiculo
}
```



# Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (herança)

```
public class Moto : Veiculo
{
    // Agora a classe Moto tem
    // todos os atributos e
    // métodos da classe Veiculo
}
```

# Programação Orientada a Objetos (POO)

- Quatro pilares da POO (herança)

Quando a classe herdada possui somente métodos construtores com parâmetros, o método construtor da herdeira deve chamar o método construtor original através da palavra-chave `base`.

# Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (herança)

```
public class Veiculo
{
    public string Modelo;

    public Veiculo(string modelo)
    {
        Modelo = modelo;
    }
}
```

# Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (herança)

```
public class Carro : Veiculo
{
    public Carro(string modelo) : base(modelo)
    {
        // Bloco de código do método construtor
    }
}
```

# Exercício

## Exercício

1 - Imagine um sistema de cadastro escolar, e a partir disso, fazer um processo de **abstração** e criar as **classes** aluno e professor, porém, essas **classes** devem ter uma **herança** da classe pessoa.

**Não pode usar a classe pessoa da aula passada, porque agora ela deve ser pensada em um conceito de cadastro escolar.**