



Aula 20

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 (polimorfismo)
 - (vamos imaginar alguns animais)

Todo animal emite um som.

Se pensarmos em uma arara e em um gato, ambos são animais, porém ambos emitem um som diferente.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (polimorfismo)
 - (vamos imaginar alguns animais)

Trazendo isso para dentro de um contexto de POO, a classe *Animal* possui o método *EmitirSom*, e as classes *Arara* e *Gato* herdam a classe *Animal*.

Contudo, é preciso que a lógica do método *EmitirSom* seja diferente para a classe *Arara* e para a classe *Gato*.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (polimorfismo)

O polimorfismo é um dos principais fundamentos da POO, que permite que métodos com o mesmo nome tenham lógicas diferentes.

Ou seja, o polimorfismo permite existir dois métodos com o mesmo nome, mas que funcionam de forma diferente.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (polimorfismo)

Precisamos entender primeiro que todo método possui uma assinatura, que funciona como uma forma de identificar qual é aquele método.

Essa assinatura é composta pelo nome do método e pelos seus parâmetros.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (polimorfismo)

Uma vez compreendido a assinatura de um método, podemos partir para entender que, temos duas formas de trabalhar com o polimorfismo:

- Sobrescrita
- Sobrecarga

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (polimorfismo: sobrescrita)

A primeira maneira de trabalharmos com o polimorfismo é através da sobrescrita, é a maneira mais comum de se trabalhar com polimorfismo, e é a maneira a qual normalmente os desenvolvedores enxergam o polimorfismo.

Ela permite que os métodos que foram definidos nas superclasses, tenham sua implementação refeita pelas suas subclasses.

Programação Orientada a Objetos (POO)

- Quatro pilares da POO (polimorfismo: sobrescrita)

Ou seja, a **superclasse** define um **método** com uma lógica genérica, e a **subclasse** vai reescrever esse **método** com uma lógica mais específica.

Através da **sobrescrita**, permitimos que um mesmo **método** tenha várias formas de ser executado sem alterar a sua assinatura.

Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (polimorfismo: sobrescrita)

No C#, para utilizarmos a sobrescrita, precisamos fazer o uso de duas palavras-chaves.

Primeiramente a palavra-chave **virtual** no **método** da **superclasse**, para identificar que aquele **método** pode ser **sobrescrito**.

Depois, a palavra-chave **override** no **método** da **subclasse**, para identificar que aquele **método** está sendo **sobrescrito**.

Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (polimorfismo: sobrescrita)

```
public class Animal
{
    public virtual void EmitirSom()
    {
        // Bloco de código do método
    }
}
```

Programação Orientada a Objetos (POO)

- C#: Quatro pilares da POO (polimorfismo: sobrescrita)

```
public class Gato : Animal
{
    public override void EmitirSom()
    {
        // NOVO bloco de código do método
    }
}
```

Exercício

Exercício

gg.gg/SenacP0020

github.com/hmaltaurodev/slides