

INSTITUTO FEDERAL

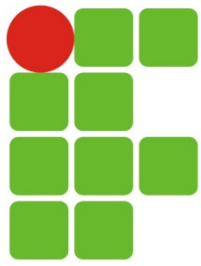
Farroupilha

Campus Uruguaiana

POO - Herança

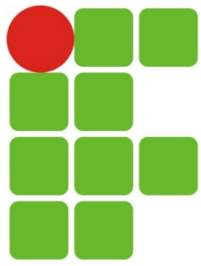
Prof. Leandro Martins Dallanora

leandro.dallanora@iffarroupilha.edu.br



Introdução

- Herança é um dos pilares da Programação Orientada a Objetos (POO).
- No contexto da POO, a herança permite que uma classe (chamada **subclasse** ou **classe filha**) herde atributos e métodos de outra classe (chamada **superclasse** ou **classe pai**).
- Isso facilita o reaproveitamento de código e promove uma estrutura mais organizada.



Conceitos

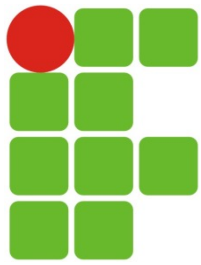
- **Superclasse (Classe Pai):**

- É a classe que contém métodos e atributos que serão herdadas por outra classe.
- As classes que herdam de uma superclasse podem reutilizar os métodos e atributos dessa classe, sem a necessidade de reescrever o código.

- **Exemplo:**

```
<?php
class Animal {
    public $nome;

    public function dormir() {
        echo "O animal está dormindo";
    }
}
```



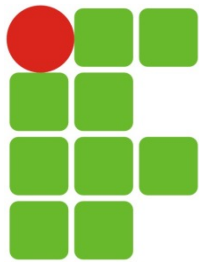
Conceitos

- **Subclasse (Classe Filha):**

- A subclasse é a classe que herda os atributos e métodos da superclasse.
- Além de herdar as funcionalidades, a subclasse pode adicionar novos métodos ou até sobrescrever métodos da superclasse.

- **Exemplo:**

```
<?php
class Cachorro extends Animal {
    public function latir() {
        echo "O cachorro está latindo";
    }
}
```



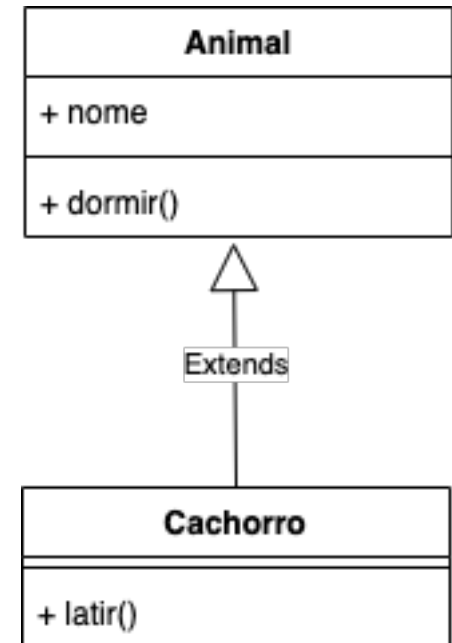
Testando

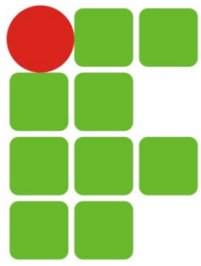
```
<?php
class Animal {
    public $nome;

    public function dormir() {
        echo "O animal está dormindo";
    }
}

class Cachorro extends Animal {
    public function latir() {
        echo "O cachorro está latindo";
    }
}
```

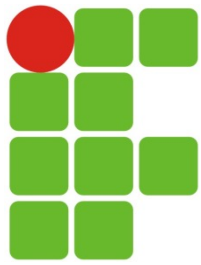
```
// Exemplo de uso
$dog = new Cachorro();
$dog->nome = "Rex";
$dog->dormir(); // O animal está dormindo
$dog->latir(); // O cachorro está latindo
```





Extends

- Em PHP, a herança é implementada usando a palavra-chave **extends**.
- A subclasse usa **extends** para indicar que está herdando de uma superclasse.
- Isso promove a reutilização do código em vez de repetir o código comum em várias classes.
- Assim podemos definir o comportamento compartilhado em uma superclasse e permitir que outras classes o herdem.

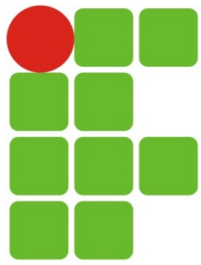


Construtor

- Se a superclasse tiver um construtor, a subclasse pode herdar esse comportamento ou criar seu próprio construtor. Para chamar o construtor da classe pai usamos **parent::__construct()**. Exemplo:

```
<?php
class Animal {
    public function __construct($nome) {
        $this->nome = $nome;
    }
}

class Cachorro extends Animal {
    public function __construct($nome) {
        parent::__construct($nome);
    }
}
```



Benefícios e Limitações

- **Reutilização de Código:** Classes filhas podem herdar código já escrito na classe pai, economizando esforço e evitando redundâncias.
- **Organização e Manutenção:** Torna o código mais modular e mais fácil de manter. Ao modificar a superclasse, todas as subclasses automaticamente refletem essas mudanças.
- **Extensibilidade:** É possível criar novas classes com base em classes existentes, adicionando ou modificando funcionalidades.
- **Herança Múltipla:** Diferente de outras linguagens, PHP não suporta herança múltipla direta (uma classe não pode herdar de mais de uma classe ao mesmo tempo).