

LINGUAGEM DE PROGRAMAÇÃO - PHP

EDL

RODRIGO JAHARA DE ALBUQUERQUE TEIXEIRA
github.com/rodrigojat/EDL

ORIGEM & INFLUÊNCIAS

- Criado em 1994 por Rasmus Lerdof, a primeira versão do PHP foi um conjunto de binários Common Gateway Interface (CGI) escrito na linguagem de programação C.
- Sua função era para acompanhamento de visitas do seu currículo online, nomeando o conjunto de scripts de “Personal Home Page Tools”, frequentemente referenciado como “PHP Tools”.
- Em 1995, Rasmus liberou o código fonte do PHP Tools para o público.
- O PHP é uma linguagem de desenvolvimento com uma sintaxe baseada no C/C# e no Java.

CLASSIFICAÇÃO

- PHP é uma linguagem interpretada, imperativa, funcional, orientada a objeto, procedural e de tipagem fraca e dinâmica.
- O PHP, apesar de ser classificada como linguagem de uso geral, é essencialmente adequado para o desenvolvimento web e pode ser embutido em documentos HTML, ou seja, é capaz de escrever páginas HTML completas, assim como pode receber informações de páginas HTML, como por ex: dados de um formulário.

Exemplos de uso

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $name = "Rodrigo";
      $age = 24;
      $faculdade = "UERJ";
      echo "Estudo na $faculdade";
    ?>
    <h1>Brincando com php</h1>
    <h3><?="Seu nome eh $name e sua idade eh $age" ?></h3>
  </body>
</html>
```

Estudo na UERJ

Brincando com php

Seu nome eh Rodrigo e sua idade eh 24

Exemplo acima mistura html com php sem problemas na execução.

Classes

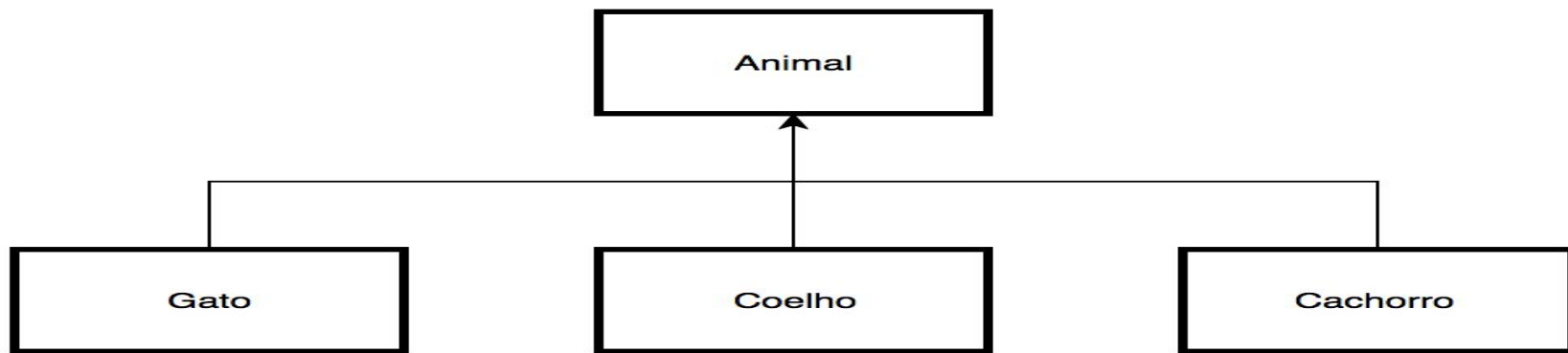
Em php , a declaração de uma classe é bem parecida com Java:

```
<?php
Class Aluno {
    public $nome;
    public $matricula;

    public function setNome ($nome){
        $this->nome = $nome;
    }
    public function GetNome(){
        return $this->nome;
    }
    public function setMatricula($matricula){
        $this->matricula = $matricula;
    }
    public function GetMatricula(){
        return $this->matricula;
    }
}

$AlunoNovo = new Aluno();
$AlunoNovo->setNome("Rodrigo");
echo $AlunoNovo->GetNome();
```

Rodrigo



Class Animal { // Código }

```
Class Gato extends Animal {
Animal{
```

```
// Código
}
```

```
Class Coelho extends Animal {
```

```
// Código
```

```
}
```

```
Class Cachorro extends
```

```
// Código
```

```
}
```

Herança é normalmente utilizada para se atingir o reuso de código. Mas, em essência, ela não é sobre reuso. É sobre definir uma hierarquia estrita de tipo. Reuso é a consequência de se usar herança.

Herança

```
<?php
```

```
class Pessoa
```

```
{  
    public $name;  
    public $age;  
    public $phone;  
  
    function PegaDados ($name,$age,$phone){  
        $this->name = $name;  
        $this->age = $age;  
        $this->phone = $phone;  
    }  
}
```

```
class Aluno extends Pessoa
```

```
{  
    public $matricula;  
    public $período;  
  
    function Matricula ($matricula){  
        $this->matricula = $matricula;  
    }  
}
```

```
$Aluno = new Aluno();  
$Aluno->PegaDados("Leonardo",18,00000000);  
echo $Aluno->name,"<br>",$Aluno->age,"<br>";  
$Aluno->Matricula = '0012010495';  
echo $Aluno->Matricula;
```

Leonardo
18
0012010495

Polimorfismo

- Polimorfismo simplesmente significa reescrever um método da superclasse na subclasse, e assim podendo determinar novos comportamentos desse método

```
<?php
// Superclasse
class Configura {
    public function ver () {
        | echo 'Sou a superclasse! <br>';
    }

    function __construct() {
        | $this->ver();
    }
}
```

Sou a superclasse!
Sou a subclasse!

```
// Subclasse
class Mostra extends Configura {
    public function ver () {
        | echo 'Sou a subclasse! <br>';
    }

    function __construct() {
        | $this->ver();
    }
}
```

```
// Objeto da superclasse
$superclasse = new Configura();
```

```
// Objeto da subclasse
$subclasse = new Mostra();
```


Avaliação comparativa com Java (Trait)

- O que é uma trait em php?
 - É um mecanismo para agregar características/comportamentos a um conjunto de classes de forma horizontal.
 - A diferença entre uma herança simples (com extends) para uma herança com trait é que a herança simples se dá de forma vertical, pois as linguagens tanto php como Java só permitem que uma classe herde diretamente de uma **única** classe. Por outro lado, é possível agregar várias traits a uma única classe, por isso dizemos que é herança horizontal.
- Resumindo : uma trait é um pacote de características/comportamentos semelhantes às classes abstratas, mas que podem ser acopladas a classes.
 - Uma classe pode ter inúmeras traits.

Usando Traits

- Usando como exemplo a classe Pessoa e Carro , que ambos têm em comum Velocidade.

```
<?php
trait Velocidade {
    public function velo() {
        echo "velocidade eh de ";
    }
}

class Pessoa {
    public $idade;
    public $peso;
    public $nome;
    use Velocidade;
    public function VeloPessoa () {
        echo "8";
    }
}

class Carro {
    public $marca;
    public $modelo;
    public $anodefabricacao;
    use Velocidade;
    public function VeloCarro () {
        echo "75";
    }
}

$obj = new Pessoa();
$obj2 = new Carro();
$obj2->velo();
$obj2->VeloCarro();
echo "<br>";
$obj->velo();
$obj->VeloPessoa();
?>
```

velocidade eh de 75
velocidade eh de 8

- Em Java, por se tratar de objetos totalmente diferentes que possuem algo em comum (Velocidade) e não poder ter heranças múltiplas, teríamos que criar duas classes também, entretanto haveria um maior custo de memória, pois teríamos que fazer mais funções de get e set para descobrir a velocidade e armazená-la em cada enquanto em php só preciso de uma função que retorna o valor.

```
class Carro{
    int vel;
    public void SetVel(int a){
        this.vel = a;
    }
    public int getVel(){
        return this.vel;
    }
    public void Velocidade (){
        System.out.println ("Carro com velocidade de: " + getVel());
    }
}

class Pessoa{
    int vel;
    public void SetVel(int a){
        this.vel = a;
    }
    public int getVel(){
        return this.vel;
    }
    public void Velocidade (){
        System.out.println ("Pessoa com velocidade de: " + getVel());
    }
}

public class Main{
    public static void main(String[] args) {
        Carro Carro1 = new Carro();
        Carro1.SetVel(75);
        Carro1.Velocidade();
        Pessoa Pessoa = new Pessoa();
        Pessoa.SetVel(5);
        Pessoa.Velocidade();
    }
}
```