



## Atividades de Fixação - Yasmin - TI30

### Atividade 1 – Conceitos

Explique com suas palavras:

1. O que é uma classe abstrata?

É uma classe que serve como modelo para outras classes. Ela não pode ser instanciada diretamente, mas pode conter métodos implementados ou métodos abstratos (sem implementação), que devem ser implementados pelas subclasses.

2. Qual é a diferença entre uma classe abstrata e uma classe final?

A *classe abstrata* serve como modelo base e pode ser herdada, mas não pode ser instanciada. E a *classe final* pode ser instanciada, mas não pode ser herdada. Ou seja, nenhuma outra classe pode estender uma classe final.

### Atividade 2 – Interpretação de Código

Explique o que ocorre ao executar:

```
$p = new Pessoa();
```

- Ocorre um erro, pois *Pessoa* é uma classe abstrata, e classes abstratas não podem ser instanciadas diretamente.

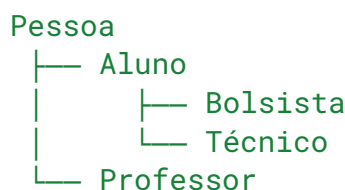
E por que o comando abaixo funciona normalmente:

```
$v = new Visitante();
```

- Funciona porque *Visitante* é uma classe concreta, ou seja, não é abstrata, e portanto pode ser instanciada normalmente.

### Atividade 3 – Identificação Hierárquica

Na estrutura a seguir:



Responda:

- a) Quem é a **superclasse** de *Aluno*? *Pessoa*
- b) Quem é a **subclasse** de *Aluno*? *Bolsista* e *Técnico*
- c) Quem é **ancestral** de *Bolsista*? *Aluno* e *Pessoa* (ancestral direto e indireto)

## Atividade 4 – Complete o Quadro

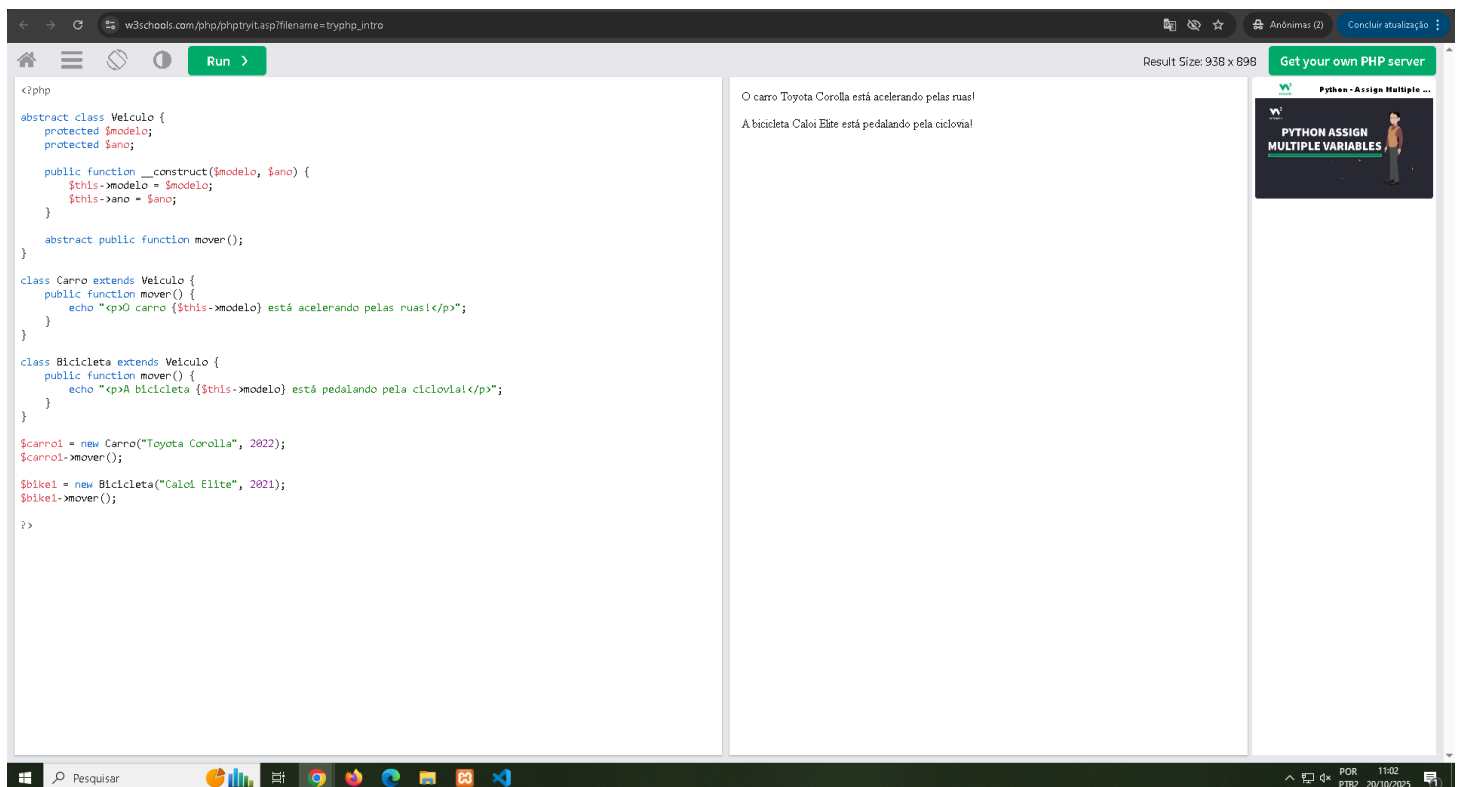
Situação	Tipo correto
Serve apenas como modelo base	<i>Classe abstrata</i>
Não pode ser herdada	<i>Classe final</i>
Não pode ser instanciada	<i>Classe abstrata</i>
Método que deve ser implementado nas filhas	<i>Método abstrato</i>
Método que não pode ser sobrescrito	<i>Método final</i>

## Atividade 5 – Prática em PHP

Crie uma classe abstrata chamada **Veiculo** com os atributos **modelo**, **ano** e um método abstrato **mover()**.

Crie duas subclasses (**Carro** e **Bicicleta**) que implementem o método **mover()** com mensagens diferentes.

No final, instancie ambas as classes e chame o método **mover()**.



```
<?php

abstract class Veiculo {
    protected $modelo;
    protected $ano;

    public function __construct($modelo, $ano) {
        $this->modelo = $modelo;
        $this->ano = $ano;
    }

    abstract public function mover();
}

class Carro extends Veiculo {
    public function mover() {
        echo "<p>O carro {$this->modelo} está acelerando pelas ruas!</p>";
    }
}

class Bicicleta extends Veiculo {
    public function mover() {
        echo "<p>A bicicleta {$this->modelo} está pedalando pela ciclovia!</p>";
    }
}

$carro1 = new Carro("Toyota Corolla", 2022);
$carro1->mover();

$bike1 = new Bicicleta("Caloi Elite", 2021);
$bike1->mover();

?>
```

O carro Toyota Corolla está acelerando pelas ruas!

A bicicleta Caloi Elite está pedalando pela ciclovia!