

```
<?php
```

```
//MÉTODOS ACESSORES:
```

```
class Forma {  
    private string $sabor; //Perceba que nossos atributos estão privados...  
    private string $formato;  
  
    public function setSabor($sabor):string { //Temos aqui setters para atribuição de valores, por convenção  
        return $this->sabor = $sabor;          //escrevemos "set" no método...  
    }  
  
    public function setFormato($formato):string {  
        return $this->formato = $formato;  
    }  
  
    public function getSabor():string { //Temos aqui getters para visualizar os atributos, por convenção escrevemos  
        return $this->sabor;          //get no método...  
    }  
  
    public function getFormato():string {  
        return $this->formato;  
    }  
  
}
```

```
//Note que nas 2 instâncias abaixo nós precisamos de atribuir valores aos atributos por através de métodos acessores...
```

```
$chocolate = new Forma();  
$chocolate->setSabor('chocolate');  
$chocolate->setFormato('quadrado');  
echo $chocolate->getSabor() . PHP_EOL; //Só conseguimos ver os valores se usarmos um get...  
echo PHP_EOL;  
echo PHP_EOL;
```

```
$morango = new Forma();  
$morango->setSabor('morango');
```

```
$morango->setFormato('redondo');  
var_dump($morango);  
// $morango->sabor; //Veja que quando tentamos visualizar o valor de um atributo privado, teremos um erro...
```

[Running] php "c:\Users\Almoxarifado\Documents\php\arquivos_das_aulas\30-metodos_acessores.php"

Chocolate

```
object(Forma)#2 (2) {  
    ["sabor":"Forma":private]=>  
    string(7) "morango"  
    ["formato":"Forma":private]=>  
    string(7) "redondo"  
}
```

```
class Animal {  
  
    private $nome;  
    private $idade;  
  
    public function __construct(string $nome, string $idade){  
        $this->nome = $nome;  
        $this->idade = $idade;  
    }  
  
    protected function getNome(){ //Perceba que deixamos os nossos métodos protegidos...  
        return $this->nome;  
    }  
  
    protected function getIdade(){ //Perceba que deixamos os nossos métodos protegidos...  
        return $this->idade;  
    }  
  
}  
  
class Cachorro extends Animal {
```

```

private $falar;

public function __construct(string $nome, string $idade, string $falar){ //Na classe filha nós empregamos todos
                                                                    //os atributos de ambas as classes
    parent::__construct($nome, $idade); //Mas note que para referenciar //mãe e filhas...
    $this->falar = $falar; //os atributos da classe mãe nós
                                                                    //usamos a palavra reservada "parent"
}                                                                    //para referenciar os atributos da
                                                                    //classe mãe...

public function getFalar():string
{
    return $this->falar;
}

public function getDados():string
{
    return $this->getNome() . PHP_EOL . $this->getIdade() . ' ano(s)'; //Note que métodos protegidos só podem
                                                                    //ser usados pelas classes filhas se
                                                                    //chamarmos o método dentro de outro
                                                                    //método e usando o "$this" para chamar
                                                                    //o método, existe outra forma de chamá-lo
                                                                    //usando o "parent", veja na classe "Gato"
}

class Gato extends Animal {

    private $falar;

    public function __construct(string $nome, string $idade, string $falar){

        parent::__construct($nome, $idade);
        $this->falar = $falar;

    }
}

```

```

    public function getFalar():string
    {
        return $this->falar;
    }

    public function getDados():string
    {
        return parent::getNome() . PHP_EOL . parent::getIdade() . ' ano(s)'; //Essa é outra forma de referenciar
                                                                                //métodos privados usando o "parent"
    }

}

$pluto = new Cachorro('Pluto', '7', 'Au! Au!');
echo $pluto->getDados();echo PHP_EOL;
echo $pluto->getFalar();echo PHP_EOL;
echo PHP_EOL;

$gato_de_botas = new Gato('Gato de Botas', '5', 'Meow!');
echo $gato_de_botas->getDados();echo PHP_EOL;
echo $gato_de_botas->getFalar();echo PHP_EOL;
echo PHP_EOL;

var_dump($pluto);echo PHP_EOL;
var_dump($gato_de_botas);

```

?>

[Running] php "c:\Users\Almoxarifado\Documents\php\arquivos_das_aulas\30-metodos_acessores.php"

Pluto

7 ano(s)

Au! Au!

Gato de Botas

5 ano(s)

Meow!

```
object(Cachorro)#3 (3) {  
  ["nome":"Animal":private]=>  
  string(5) "Pluto"  
  ["idade":"Animal":private]=>  
  string(1) "7"  
  ["falar":"Cachorro":private]=>  
  string(7) "Au! Au!"  
}
```

```
object(Gato)#4 (3) {  
  ["nome":"Animal":private]=>  
  string(13) "Gato de Botas"  
  ["idade":"Animal":private]=>  
  string(1) "5"  
  ["falar":"Gato":private]=>  
  string(5) "Meow!"  
}
```

[Done] exited with code=0 in 0.088 seconds