

CLASSES

Prof^a Renata Morais

SINTAXE BÁSICA DE CRIAÇÃO DE UMA CLASSE

```
<modificador de acesso> class NomeDaClasse{  
    //declaração dos atributos  
    //declaração dos métodos  
}
```

Palavra reservada que indica a criação da classe

//Criação de uma classe em php

```
<?php
```

```
class Produto {  
  
    private $codigo;  
    private $nome;  
    private $preco;  
  
    public function getCodigo() {  
        return $this->codigo;  
    }  
    public function setCodigo($codigo) {  
        $this->codigo = $codigo;  
    }  
    public function getNome() {  
        return $this->nome;  
    }  
    public function setNome($nome) {  
        $this->nome = $nome;  
    }  
    public function getPreco() {  
        return $this->preco;  
    }  
    public function setPreco($preco) {  
        $this->preco = $preco;  
    }  
}
```

//Criação de uma classe em java

```
public class Produto {  
  
    private String nome;  
    private int quantidade;  
  
    public Produto() {  
        // Código do construtor  
    }  
  
    public void apresentarProduto() {  
        // Código do método  
    }  
}
```

//Criação de uma classe em python

```
class Pessoa:  
    def __init__(self, nome, idade):  
        self.nome = nome  
        self.idade = idade  
  
    def setNome(self, nome):  
        self.nome = nome  
  
    def setIdade(self, idade):  
        self.idade = idade  
  
    def getNome(self):  
        return self.nome  
  
    def getIdade(self):  
        return self.idade
```

CRIANDO UMA CLASSE NO PHP

- Usando a palavra chave **class**, seguida do nome da classe

```
<?php
    class MyClass
    {
        // As propriedades e métodos da Classe vem aqui
    }
?>
```

- Após criar a classe, ela pode ser instanciada e guardada em alguma variável usando a palavra chave **new**:


```
$obj = new MyClass;
```

COLOCANDO O CÓDIGO NA ESTRUTURA DO PHP

```
<?php
    class MinhaClasse
    {
        // As propriedades e métodos da Classe vem aqui
    }
    $obj = new MinhaClasse;
    var_dump($obj);
?>
```

Nota: var_dump - função que mostrará uma representação estruturada sobre uma ou mais expressões, incluindo o tipo e o valor.

COLOCANDO O CÓDIGO NA ESTRUTURA DO PHP

C: > xampp > htdocs > Aula >  PrimeiraClasse.php > ...

```
1  <?php
2  | class MinhaClasse
3  | {
4  |     public $prop1 = "Sou um atributo da classe!";
5  | }
6  $obj = new MinhaClasse;
7  echo $obj->prop1; // Mostra a saída/conteúdo da propriedade
8  ?>
9  
```

DEFININDO MÉTODOS DE UMA CLASSE

- Métodos são funções específicas das classes.
- São ações particulares que os objetos serão capazes de executar e são definidas dentro das classes.
- POO permite que os objetos referenciem-se usando \$this. Quando estiver dentro de um método, use \$this da mesma forma que você usaria o nome do objeto fora da classe.

```
<?php
class MinhaClasse
{
    public $prop1 = "Sou um propriedade de classe!";
    public function setProperty($newval) //Método
    {
        $this->prop1 = $newval;
    }
    public function getProperty() //Método
    {
        return $this->prop1 . "<br />";
    }
}
$obj = new MinhaClasse;
echo $obj->prop1;
?>
```

MÉTODOS MODIFICADORES(SET) E RECUPERADORES(GET)

- Sempre que um método tem a função única e exclusiva de alterar (modificar) o atributo de uma classe ele deve começar com o a palavra **set** seguido do nome do atributo que ele altera.
- Sempre que um método tem a função de fornecer ou recuperar o valor dos atributos de um objeto, devem começar com a palavra **get** seguido do nome do atributo que ele recupera.

```
<?php
class MinhaClasse {
    public $prop1 = "Sou uma propriedade de classe!";
    public function setProperty($newval) {
        $this->prop1 = $newval;
    }
    public function getProperty() {
        return $this->prop1 . "<br />";
    }
}

$obj = new MinhaClasse;
echo $obj->getProperty(); // Lê o valor da propriedade
$obj->setProperty("Sou um novo valor da propriedade!"); // Atribui um novo
valor
echo $obj->getProperty(); // Lê o valor novamente para mostrar a mudança

?>
```


MODIFICADORES DE ACESSO

- Os modificadores de acesso são padrões de visibilidade de acessos às classes, atributos e métodos.
- Esses modificadores são palavras-chaves reservadas pelo Java, ou seja, palavras reservadas não podem ser usadas como nome de métodos, classes ou atributos.
- São eles:
 - Public
 - Private
 - Protected
 - Package
 - Default

MODIFICADORES DE ACESSO

- public
 - pode ser acessada de qualquer lugar e por qualquer entidade que possa visualizar a classe a que ela pertence.
- private
 - determina que os atributos são privados à classe, ou seja, só podem ser acessados por métodos da própria classe. Esse modificador não se aplica às classes, somente para seus métodos e atributos. Esses atributos e métodos também não podem ser visualizados pelas classes herdadas.
- protected
 - torna o membro acessível às classes do mesmo pacote ou através de herança, seus membros herdados não são acessíveis a outras classes fora do pacote em que foram declarados.

MODIFICADORES DE ACESSO

- default (padrão):
 - a classe e/ou seus membros são acessíveis somente por classes do mesmo pacote, na sua declaração não é definido nenhum tipo de modificador, sendo este identificado pelo compilador.
- Package
 - nível em que a classe pode ser acessada apenas por outras classes pertencentes ao mesmo pacote.

CLASSE CLIENTE

c: > xampp > htdocs > Aula > ClasseCliente.php > ...

```
1  <?php
2
3  class cliente {
4
5      public $nome;
6      public $saldo;
7
8      public function confirmarrecebimento(){
9          echo "<br/>Confirmando Informação";
10     }
11
12     public function pagarconta($valor){
13         echo "<br/>Foi pago o valor de R$ ".$valor;
14     }
15
16 }
17
18 $tempCliente = new Cliente();
19
20 //atribuindo valores aos atributos
21 $tempCliente->nome = "Renata";
22 $tempCliente->saldo = 500;
23
24 //chamando os métodos da classe
25 $tempCliente->confirmarrecebimento();
26 $tempCliente->pagarconta(300);
27
28 echo "<br/>Nome do Cliente : ".$tempCliente->nome;
29 echo "<br/>Nome do Saldo : ".$tempCliente->saldo;
30
31 ?>
```