

## 2ª Nota de Aula

### Programação com Orientação a Objetos em PHP

#### Classes

Uma classe define um novo tipo de dados e é a estrutura fundamental da orientação a objetos. Pode conter variáveis e funções ou procedimentos declarados dentro dela os quais são chamados, respectivamente, de atributos e métodos. A princípio, para definir atributos em classes, utilizamos a palavra-chave **var**.

```
<?php
class Produto
{
    var $codigo;
    var $descricao;
    var $preco;
    var $quantidade;
}
?>
```

#### Objetos

Objetos são instâncias de classes, ou seja, são “variáveis do tipo da classe”. Possuem os mesmos atributos e métodos da classe à qual pertencem e tais atributos podem ter seus valores alterados ao longo do tempo. O seguinte trecho de código mostra um objeto da classe Produto sendo instanciado e seus atributos sendo acessados.

```
<?php
Produto $p = new Produto;

$p->codigo = 1;
$p->descricao = "lapis";
$p->preco = 1.00;
$p->quantidade = 100;

echo $p;
?>
```

Note que o operador para acesso a membros é o -> e que este trecho de código não funcionará porque tentamos imprimir o objeto \$p através de um echo; isso não pode ser feito porque o echo não sabe o formato de um objeto Produto. Isso pode ser corrigido, por enquanto, através da adição de um método à classe que permita a impressão dos atributos de interesse. Os códigos a seguir mostram essa alternativa.

```
<?php
class Produto
{
```

```
        var $codigo;  
        var $descricao;  
        var $preco;  
        var $quantidade;  
  
        function imprimeEtiqueta()  
        {  
            echo "Produto: ". $this->descricao . "<BR>";  
            echo "Código: ". $this->codigo . "<BR>";  
            echo "Preço: ". $this->preco . "<BR>";  
            echo "Quantidade: ". $this->quantidade . "<BR>";  
        }  
    ?>  
  
<?php  
    Produto $p = new Produto;  
  
    $p->codigo = 1;  
    $p->descricao = "lapis";  
    $p->preco = 1.00;  
    $p->quantidade = 100;  
  
    $p->imprimeEtiqueta();  
?>
```

### *Construtores e Destrutores*

Construtores são métodos especiais que são utilizados para definir o comportamento do objeto no momento de sua criação. Um método construtor é chamado automaticamente na criação de um objeto quando usamos o operador **new**. Caso não seja implementado um construtor dentro da classe, as propriedades de um objeto serão nulas (inicializadas com o valor NULL) em sua criação. Para definir um método construtor, utilizamos o método `__construct()`.

Um destrutor é um método executado automaticamente quando um objeto é desalocado da memória, quando atribuímos o valor NULL a ele, quando usamos a função **unset()** ou, ainda, quando o programa é finalizado. Para definir um destrutor, utilizamos o método `__destruct()`. O destrutor, geralmente, é utilizado para finalizar conexões, apagar arquivos temporários, etc.

O código seguinte mostra o exemplo de uma classe Pessoa com construtor e destrutor.

```
<?php  
class Pessoa  
{  
    var $codigo;  
    var $nome;
```

```
var $altura;
var $idade;
var $escolaridade;

function Crescer($cent)
{
    if($cent>0)
    {
        $this -> altura += $cent;
    }
}
function Formar($titulacao)
{
    $this -> escolaridade = $titulacao;
}
function Envelhecer()
{
    $this->idade++;
}
function __construct($codigo, $nome, $altura, $idade, $escolaridade)
{
    $this->codigo = $codigo;
    $this->nome = $nome;
    $this->altura = $altura;
    $this->idade = $idade;
    $this->escolaridade = $escolaridade;
}
function __destruct()
{
    echo"Objeto{$this -> nome} finalizado <br>";
}
}
?>
```