Modificadores de acesso

Com os modificadores de acesso determinamos a **visibilidade** de um método ou atributo pertencente a uma classe. Ou seja, definimos se ele pode ou não ser acessado fora da classe em que foi declarado.

Modificadores de acesso

Para modificar a visibilidade de um atributo ou método devemos preceder sua declaração de uma das palavras reservadas que representam o modificador, da seguinte forma:

```
1  modificador $atributo;
2
3  modificador function metodo() { }
```

Tipos

public

Este é o nível de acesso mais permissivo. Ele indica que o método ou atributo da classe é público, ou seja, pode ser acessado em qualquer outro ponto do código e por outras classes.

No código abaixo podemos ver um exemplo de uso do modificador de acesso public:

```php
class Exemplo {
    public $publico = 'Public';
    public function metodoPublico() { }
}
```

public

Este é o nível de acesso mais permissivo. Ele indica que o método ou atributo da classe é público, ou seja, pode ser acessado em qualquer outro ponto do código e por outras classes.

No código abaixo podemos ver um exemplo de uso do modificador de acesso public:

```php
class Exemplo {
    public $publico = 'Public';
    public function metodoPublico() { }
}
```

public

No código acima foi declarada uma classe denominada Exemplo que possui um atributo público chamado $publico e um método público intitulado metodoPublico(). Neste caso ambos podem ser acessados por qualquer parte do código, incluindo outras classes.

private

Este modificador é o mais restrito. Com ele definimos que somente a própria classe em que um atributo ou método foi declarado pode acessá-lo. Ou seja, nenhuma outra parte do código, nem mesmo as classes filhas, pode acessar esse atributo ou método.

No código abaixo podemos ver um exemplo de uso do modificador de acesso private:

```
1  class Exemplo{
2   private  $privado = 'Privado';
3   private function metodoPrivado() {}
4  }
```

protected

Esse modificador indica que somente a própria classe e as classes que herdam dela podem acessar o atributo ou método. Dessa forma, ao instanciar a classe os elementos protegidos (protected) não podem ser acessados diretamente, como ocorre com o public.

A seguir podemos ver um exemplo de uso do modificador de acesso protected:

```php
class Exemplo {
    protected $protegido = 'Protegido';
    protected function metodoProtegido() { }
}
```

protected

No código acima foi declarada uma classe denominada Exemplo que possui um atributo protegido chamado $protegido e um método protegido intitulado metodoProtegido(). Neste caso ambos só podem ser acessados pela própria classe e pelas suas classes filhas.

Exemplo – Conta Bancária

Crie uma pasta com o nome de: **conta_bancaria**

Crie o arquivo: **conta_bancaria.php** e com os seguintes comandos:

ContaBancaria.php

```php
<?php
// Inicia a sessão para armazenar o saldo entre requisições
session_start();

// Definição da classe ContaBancaria
class ContaBancaria {
    // Propriedade privada que armazena o saldo
    // Variável privada, só está visível para essa classe
    private $saldo;
```

```php
10
11      // Construtor para inicializar o saldo com um valor da sessão ou padrão (zero)
12      public function __construct() {
13          // Se existir um saldo na sessão, utiliza-o, caso contrário, inicializa com 0
14          $this->saldo = isset($_SESSION['saldo']) ? $_SESSION['saldo'] : 0;
15      }
16
17      // Método set para definir o saldo
18      public function setSaldo($saldo) {
19          $this->saldo = $saldo;
20          // Atualiza o saldo na sessão para manter a persistência
21          $_SESSION['saldo'] = $this->saldo;
22      }
23
24      // Método get para obter o saldo atual
25      public function getSaldo() {
26          return $this->saldo;
27      }
```

```php

    // Método para depositar um valor na conta
    public function depositar($quantia) {
        // Verifica se a quantia é maior que zero antes de adicionar ao saldo
        if($quantia > 0) {
            //Pega o valor do saldo atual através da variável da classe (this->saldo) e soma
           //E atualiza o setSaldo()
            $saldoNovo = $this->saldo + $quantia;
            $this->setSaldo($saldoNovo);
        }
    }

    // Método para sacar um valor da conta
    public function sacar($quantia) {
        // Verifica se a quantia é válida e se há saldo suficiente
        if($quantia > 0 && $quantia <= $this->getSaldo()) {
            $this->setSaldo($this->getSaldo() - $quantia);
        } else {
```

```
46                  echo "Saldo insuficiente para o saque.<br>";
47              }
48          }
49      }
50      ?>
51
```

Exemplo – Conta Bancária

Crie um novo arquivo: **index.php** e com os seguintes comandos:

index.php

C:\Users\edwar\Desktop\USBWebserver v10\root\conta_bancaria\index.php

```
<html lang="pt-PT">
<head>
    <meta charset="UTF-8">
    <title>Exemplo de Conta Bancária com Set e Get</title>
</head>
<body>
    <h2>Formulário da Conta Bancária</h2>

    <!-- Formulário que permite ao usuário depositar ou sacar dinheiro -->
    <form method="POST" action="">
        Depósito: <input type="number" name="deposito" step="0.01" placeholder="Valor a depositar"><br><br>
        Sacar: <input type="number" name="sacar" step="0.01" placeholder="Valor a sacar"><br><br>
        <input type="submit" value="Enviar">
    </form>

```

```php
<?php
// Inclui o ficheiro que contém a classe ContaBancaria
require_once 'ContaBancaria.php';

// Cria ou recupera a instância da conta bancária
$conta = new ContaBancaria();

// Verifica se o formulário foi submetido
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Verifica se foi informado um valor para depósito
    if (!empty($_POST['deposito'])) {
        // Deposita o valor informado na conta
        $conta->depositar(floatval($_POST['deposito']));
    }

    // Verifica se foi informado um valor para saque
    if (!empty($_POST['sacar'])) {
        // Tenta sacar o valor informado da conta
```

```
35              $conta->sacar(floatval($_POST['sacar']));
36          }
37      }
38
39      // Exibe o saldo atualizado da conta
40      echo "<h3>Saldo Atual</h3>";
41      echo "O saldo atual da conta é: R$" . number_format($conta->getSaldo(), 2, ',', '.') . "<br>";
42      ?>
43  </body>
44  </html>
45
```