Programação Orientada a Objetos com PHP

Método Construtor e Encapsulamento

Encapsulamento

O Encapsulamento é um recurso da Orientação a Objetos, um mecanismo que provê proteção de acesso aos membros internos de um objeto. O conceito é de que certas propriedades de uma classe só podem ser manipuladas por métodos dela mesma, como medida de garantir a responsabilidade sobre estas propriedades.

private	Membros declarados private só podem ser acessados dentro da classe em que foram declarados; devem ser acessados através de métodos. Símbolo UML \rightarrow (-)
protected	Podem ser acessados dentro da classe em que foram declarados e de classes descendentes; não podem ser acessados através de objetos (variáveis de instância) em aplicações que fazem uso da classe onde foi criado. Símbolo UML → (#)
public	Podem ser acessados livremente. Símbolo UML → (+)

Método Construtor:

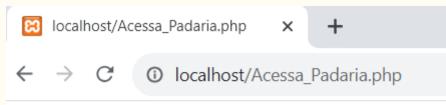
- Utilizado para definir o comportamento inicial de um objeto (no momento de sua criação);
- O método construtor é executado automaticamente no momento em que se faz uma instância de um objeto através de um operador *new*;
- Quando não é definido um método construtor, automaticamente as propriedades do objeto criado são inicializadas com valor NULL;
- Definição de um método construtor em uma determinada classe: __construct()

Método Destrutor (ou finalizador):

- Método executado automaticamente quando o objeto é desalocado da memória, ou quando lhe é atribuído o valor NULL, ou quando é utilizada a função unset() sobre o mesmo, ou quando o programa é finalizado.
- O método destrutor pode ser utilizado para finalizar conexões, apagar arquivos temporários criados durante o ciclo de vida do objeto, entre outras possibilidades.

```
C:\xampp\htdocs\Padaria.class.php - Notepad++
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Macro Executar Plugins Janela ?
                                    🛗 🛬 🔍 🥞 🖫 🖫
13 🚅 H 🖺 📳 📭 📥 l 🔏 📫 🛅 🗩 C l
님 Padaria.class.php 🛛 님 Acessa_Padaria.php 🖾 님 Dados_Complementares.class.php 🖾 님 I_Aluno.php 🖾 🔡 Manipula_Fazer
     □<?php
       class Padaria
  3
     □ {
  4
      private $padaria;
      private $pão doce;
  6
     function construct($padaria, $pão doce){
  8
           $this -> padaria = $padaria;
  9
           $this -> pão doce = $pão doce;
 10
 11
 12
     fipublic function set padaria($padaria) {
                                                         neste caso não é necessário o uso
 13
      -$this -> padaria = $padaria; }
                                                         dos métodos set( ), pois os
 14
                                                         atributos são inseridos através do
 15
      public function set pão doce($pão doce){
 16
     -$this -> pão doce = $pão doce; }
                                                         método construtor.
 17
 18
     bpublic function get padaria(){
 19
      return $this -> padaria;}
 20
     public function get pão doce(){
 22
      return $this -> pão doce;}
 23
 24
      L?>
 25
```





depois de fazer atribuições com método construtor Padaria Jardim Rádio Clube pão doce de frutas cristalizadas

Exemplo: uso de métodos Construtor e Destrutor.

```
class Pessoa
      private $código;
 4
      private $nome;
      private $altura;
      private $idade;
      private $nascimento;
 9
      private $escolaridade;
      private $salário;
10
11
12
      //método construtor
13
    function construct($código, $nome, $altura, $idade, $nascimento, $escolaridade, $salário){
      $this->código = $código;
14
15
      Sthis->nome = Snome:
16
      $this->altura = $altura;
17
      $this->idade = $idade;
18
      $this->nascimento = $nascimento;
19
      $this->escolaridade = $escolaridade;
20
      Sthis->salário = Ssalário:
21
22
    public function get código(){
23
     return $this->código;}
24
25
    public function get nome(){
26
     return Sthis->nome;}
27
28
    public function get altura(){
      return $this->altura;}
29
                                                                                                 (Dall'Oglio, 2014)
```

Exemplo: uso de métodos Construtor e Destrutor.

```
27
28
29
30
31
    public function get idade(){
32
     return $this->idade;}
33
34
    public function get nascimento() {
35
     return $this->nascimento;}
36
37
    public function get escolaridade(){
     return $this->escolaridade;}
38
39
    public function get salário() {
40
41
     return $this->salário;}
42
43
      //método destrutor
44
      function destruct()
     {echo "Objeto {$this->nome} finalizado...<br>\n";}
45
46
47
     L2>
```



nome: Patrick Willian código: 100304

idade: 19 altura: 1.75

nascimento: 03/05/1974

escolaridade: 1º ano faculdade

salário: 756.99

Objeto Patrick Willian finalizado...

Exercício proposto:

8a. Criar uma classe Dados_Complementares.class, com dados adicionais da classe Pessoa.class. Utilizar a aplicação Acessa_construtor para inserir e consultar os dados pessoais já existentes e complementares do Sr. Willian.

8b. Criar uma classe Conta_Bancária.class, com dados bancários relativos à classe Pessoa.class. Atualizar a aplicação Acessa_construtor para inserir e consultar dados da conta bancária e pessoais do Sr. Willian.

```
C:\xampp\htdocs\Dados_Complementares.class.php - Notepad++
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Macro Executar Plugins Janela ?
                                             🔍 🔍 | 🖫 🚟 | 🚍 👖 📜 🐷 💹 🐔 🗃 🕗 | 🗨 🔳 🕟 📦
 🖹 Aluno.php 🗵 📙 Dados_Complementares.class.php 🗵 📙 I_Aluno.php 🗵 📙 Acessa_construtor_l.php 🗵 📙 Usa_Interface.php 🗵 📙 Serviços_Reforma.php 🗵 🛗 Cliente_Reforma.php 🗵 🗎 aplica_
      ?php
        class Dados Complementares
      private $endereço;
           private $estado civil;
            private $grau instrução;
  8
            public function __construct($endereço, $estado civil, $grau instrução)
  9
 10
                $this->endereço = $endereço;
                $this->estado civil = $estado civil;
 11
                $this->grau instrução = $grau instrução;
 12
 13
 14
 15
            public function get endereço()
 16
 17
                return $this->endereço;
 18
 19
 20
            public function get estado civil()
 21
 22
                return $this->estado civil;
 23
 24
 25
            public function get grau instrução()
 26
 27
                return $this->grau instrução;
 28
 29
 30
 31
      L?>
 22
```

```
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Macro Executar Plugins Janela ?
🕞 🛂 💾 🖺 🥦 🧓 🦚 | 🚜 📭 🖺 | 🤝 et l 📾 🍆 | 🤏 et l 📠 🛬 | 🗨 🗨 📑 🗔 📑 🕇 🏗 📜 💹 🙉 🙉 🖟 🗀 🐵 | 🗨 🔳 🕟
📙 Aluno.php 🗵 📙 I_Aluno.php 🗵 📙 Acessa_construtor_l.php 🗵 📙 Usa_Interface.php 🗵 🛗 Serviços_Reforma.php 🗵 블 Cliente_Reforma.php 🗵 블 aplica_reforma.php 🗵 🛗 Pessoa.class.php
     ⊟<?php
      include once 'Pessoa.class.php';
      include once 'Dados Complementares.class.php';
  5
      //criação de objeto
      $Willian = new Pessoa(100304, "Patrick Willian", 1.75, 19, "03/05/1974", "1° ano faculdade", 756.99);
  6
      $complemento = new Dados Complementares ("Rua B n° 20, Jardim Itatins - Peruíbe/SP", "solteiro", "2° grau técnico");
  8
 9
      //manipulação do objeto
 10
      echo nl2br ("nome:"."\t".$Willian->get nome()."\n");
 11
      echo nl2br ("código:"."\t".$Willian->get código() ."\n");
      echo nl2br ("idade:"."\t".$Willian->get idade()."\n");
 12
      echo nl2br ("altura:"."\t".$Willian->get altura()."\n");
 13
 14
      echo nl2br ("nascimento:"."\t".$Willian->get nascimento()."\n");
 15
      echo nl2br ("escolaridade:"."\t".$Willian->get escolaridade()."\n");
      echo nl2br ("salário:"."\t".$Willian->get salário()."\n");
 16
      echo nl2br ("\nDADOS COMPLEMENTARES\n");
 17
      echo nl2br ("endereço:"."\t".$complemento->get endereço()."\n");
 18
 19
      echo nl2br ("estado civil:"."\t".$complemento->get estado civil()."\n");
      echo nl2br ("grau de instrução:"."\t".$complemento->get grau instrução()."\n\n");
 20
      -?>
 21
 22
```



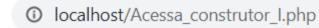












nome: Patrick Willian

código: 100304

idade: 19 altura: 1.75

nascimento: 03/05/1974

escolaridade: 1º ano faculdade

salário: 756.99

DADOS COMPLEMENTARES

endereço: Rua B nº 20, Jardim Itatins - Peruíbe/SP

estado civil: solteiro

grau de instrução: 2º grau técnico

Objeto Patrick Willian finalizado...

Referências Bibliograficas

- *php* Programando com Orientação a Objetos Pablo Dall'Oglio - editora Novatec - 2014

Desenvolvimento web com PHP e MySQL
 Evaldo Junior Bento – editora Casa do Código 2018