Resumo de Capítulos segundo Ebook disponibilizado no portal

Acadêmico: João Victor Surdi de Melo

Capítulo 2:

O capítulo começa nos mostrando que a POO (Programação Orientada à Objetos) é um paradigma central no desenvolvimento de softwares modernos, fala também, que a linguagem PHP é uma grande aliada, já que, desde sua versão mais recentes, incorpora essa metodologia com robustez.

Logo após, o capítulo nos apresenta o conceito de Classe e Objeto, onde Classe é um molde e Objeto o que será moldado pela classe:

*“onde uma classe funciona como um molde para a criação de objetos, que são instâncias dessa classe.”*

Um exemplo de código abaixo, onde Classe é definida pela palavra-chave “class”:

Class Carro {

public $cor;

public $modelo;

public function \_construct($cor, $modelo) {

$this->cor = $cor;

$this->modelo = $modelo;

}

public function detalhesCarro(){

return “Modelo: ”. $this->modelo . “Cor: ”. $this->cor;

}

}

Objetos são criados com o operador “new”, como mostra o exemplo abaixo:

$meuCarro = new Carro(“vermelho”, “ferrari”);

echo $meuCarro->detalhesCarro();

A saída dessa operação será:

Modelo: ferrari Cor: vermelho

É necessário que entendamos a necessidade desse tipo de metodologia no mercado atual...

O uso adequado dessa metodologia faz com que o desenvolvedor se encontre em um código mais limpo e organizado, tornando mais fácil sua manutenção e expansão, sendo mais fácil e prático projetar e modelar processos reais dentro do software.

Além de tornar o código mais limpo e modular, também nos deixa preparados para trabalhar com conceitos mais avançados da POO, como herança e polimorfismo que serão vistos mais à frente neste documento.

Outro conceito fundamental também na POO é o conceito de encapsulamento, que mantém os dados de dentro da classe protegidos, através dos níveis de segurança e proteção (public, private ou protected), este método, como o próprio nome sugere, encapsula todos os dados dentro da classe para não ser possível o acesso de terceiros, apenas sendo possível acessar os métodos públicos.

A compreensão desses fundamentos que citei acima, são necessários não só para fortalecer a base dos programas que são construídos, mas também agiliza na manutenção do sistema e reutilização de código.

Definições básicas:

**- Classe:** É o “Molde” para o objeto, ele irá definir os atributos relacionados ao objeto.

**- Objeto:** Instâncias da Classe, objetos seguirão e manterão os estado e comportamentos definidos por sua classe.

**- Encapsulamento:** Protege os dados de dentro da classe, apresentando apenas o que é necessário por meio de métodos públicos

**- Construtores:** São principalmente úteis para criar novas instâncias de objetos e realizar quaisquer configurações necessárias

Partindo para o capítulo 3 do ebook, vemos um pouco mais sobre métodos e atributos, explicando inicialmente o que são atributos.

Atributos, ou comumente chamadas propriedades, são, como o próprio nome sugere, os atributos dentro da classe, vamos resgatar o exemplo de cima, onde “$cor” e “$modelo” são atributos da classe “Carro”.

“

Class Carro {

public $cor;

public $modelo;

public function \_construct($cor, $modelo) {

$this->cor = $cor;

$this->modelo = $modelo;

}

public function detalhesCarro(){

return “Modelo: ”. $this->modelo . “Cor: ”. $this->cor;

}

}

“

Como podem ver, atributos têm níveis de segurança e acessibilidade, esses podem ser: “public”, “private” ou “protected”.

Atributo público (“public”): Podem ser acessados de qualquer lugar.

Atributo protegido (“protected”): Pode ser acessado somente dentro da própria classe e/ou subclasse.

Atributo privado (“private”): pode ser acessado somente de sua classe de onde foi criado.

Métodos são parecidos com os atributos em muitos aspectos, assim como eles, os métodos possuem os níveis de acessibilidade (“public”, “protected” e “private”).

Métodos são, de forma simplificada, operações que ocorrem entre os objetos e/ou nos dados da classe.