PHPOO – Parte 2

DIOGO CEZAR TEIXEIRA BATISTA
http://inf.cp.utfpr.edu.br/diogo
diogo@diogocezar.com

HERANÇA EM PHP

- uma classe pode estender (extends) outra classe qualquer;
- todos os atributos e métodos estão disponíveis imediatamente, pela variável $this;
- o construtor da superclasse deve ser chamado explicitamente pelo construtor da subclasse;

```
parent::__construct();
```

- a definição da subclasse deve incluir a definição da superclasse;

EXEMPLO CONTA CORRENTE/ESPECIAL

ContaCorrente.php

```
<?php
class ContaCorrente{

        private $saldo;

        public function __construct($valor){
                $this->saldo = $valor;
        }

        public function saque($valor){
                if($this->saldo >= $valor){
                        $this->saldo -= $valor;
                }
        }

        public function deposito($valor){
                $this->saldo += $valor;
        }

        public function __toString(){
                ...
        }
}
?>
```

construtor

efetua o saque

imprime atributos

EXEMPLO CONTA CORRENTE/ESPECIAL

herda conta corrente

construtor da superclasse

imprime novos atributos

ContaEspecial.php

```
<?php
class ContaEspecial extends ContaCorrente{

        private $limite;

        function __construct($valor, $limite){
                parent::__construct($valor);
                $this->limite = $limite;
        }

        function saque($valor){
                if($this->saldo + $this->limite >= $valor){
                        $this->saldo  -= $valor;
                }
        }

        public function __toString(){
                ...
        }

}
?>
```

EXEMPLO CONTA CORRENTE/ESPECIAL

index.php

```php
<?php
include("class/ContaCorrente.php");
include("class/ContaEspecial.php");

$CC = new ContaCorrente(1000);
$CC->saque(500);
$CC->saque(500);
$CC->saque(10);
$CC->deposito(150);
echo "Conta Corrente: ";
echo $CC;

$CE = new ContaEspecial(1000, 500);
$CE->saque(500);
$CE->saque(500);
$CE->saque(10);
echo "Conta Especial: ";
echo $CE;
?>
```

incluindo classes

criando conta corrente

criando conta especial

Tratando Erros

```php
<?php
class TratamentoErros{
        private $nome;

        public function __construct($nome){
           try{
              if(empty($nome)){
                 throw new Exception("Nome não pode ser em branco.");
              }
              else{
                 $this->nome = $nome;
                 echo "Nome é ". $this->nome;
              }
           }
           catch(Exception $e){
              echo $e->getMessage();
           }
        }
}

$TE = new TratamentoErros("");
?>
```

nome em branco gera um erro.

CRIAR GETS e SETS?

- em php podemos usar um recurso para geração automática de GETs e SETs;
- a função __call($metodo, $parametros) é executada antes da chamada de qualquer método, e:
  - $metodo → nome do método a ser acessado;
  - $parametros → array com os parâmetros passados para o método.

CRIAR GETS e SETS?

- então, se tentarmos acessar esse método:

```
$objeto->setNome('Diogo')
```

```
os atributos de call serão:
function __call($metodo, $parametros){
   //$metodo -> setNome
   //$parametros -> array[0]:Diogo
}
```

CRIAR GETS e SETS?

- dessa forma, podemos fazer algumas verificações e criar um padrão para os atributos:
  - as 3 primeiras letras do nome do método são: get ou set?
    - em caso positivo, monta-se o nome para a atribuição ou recuperação do atributo:

```
setNome → nome;
getNome → nome;
setSobreNome → sobre_nome;
setDataNascimento → data_nascimento;
```

```
substr(strtolower(preg_replace('/([a-z])([A-Z])/', "$1_$2", $metodo)), 4);
```

CRIAR GETS e SETS?

- em setDataNascimento estamos tentanto atribuir um valor para $data_nascimento.
- precisamos transformar a string DataNascimento (camel caps) em data_nascimento (underscore) e depois atribuir o valor para ela.

```
...(preg_replace('/([a-z])([A-Z])/', "$1_$2", $metodo)), 4);
```

CRIAR GETS e SETS?

```
substr(strtolower(preg_replace('/([a-z])([A-Z])/',"$1_$2",$metodo)), 4);
```

```
public function __call($metodo, $parametros){
        if (substr($metodo, 0, 3) == 'set') {
                $var = [função acima]
                $this->$var = $parametros[0];
        }
        elseif (substr($metodo, 0, 3) == 'get'){
                $var = [função acima]
                return $this->$var;
        }
}
```

com essa função definida, pode-se chamar get e set para qualquer atributo da classe!