

Atividade Prática

Programação Web II

Quase todo aplicativo ou blog lida com usuários. Quer seja o processo de registro, o login e logout, o envio de lembretes aos usuários que perderam suas senhas ou a alteração das senhas sob demanda, todo o código que trata os usuários deve ser agrupado em uma classe de usuário. Em nosso exemplo, chamamos a classe que trata os usuários de `Usuario`, de acordo com a convenção de nomenclatura vigente.

Vamos escrever uma classe de usuário com os conhecimentos passados em aula. Esta classe conterá o nome e o sobrenome de cada usuário e poderá dizer um “olá mundo!” a qualquer pessoa que use nossa aplicação.

1. O primeiro passo é determinar o nome da classe, os nomes das propriedades para o nome e o sobrenome e o nome do método que retorna “olá”.

Nome da classe: **“Usuario”**

Propriedades da classe: **“\$nome”** e **“\$sobrenome”**

Método da classe: **“dialogo”**, **“registrar”** e **“email”**

2. Escreva a classe `Usuario` e adicione as propriedades.
3. Adicione um método que diga “olá”:
4. Crie a primeira instância e a chame de `$usuario01`. Use a palavra-chave `new` para criar um objeto de uma classe.
5. Defina os valores para o nome e o sobrenome de `$usuario01`.
6. Obtenha o nome e o sobrenome do usuário e imprima-o na tela com o comando `echo`.
7. Use o método `ola()` com as variáveis de nome e sobrenome para dizer olá ao usuário.
8. Adicione outro objeto, chame-o de `$usuario02`, dê a ele o nome ‘Johnny’ e o sobrenome ‘Walker’, e diga olá para o usuário.
9. Adicione ao método `ola()` a capacidade de utilizar a propriedade `$primeiroNome`, de forma que o método `ola()` possa retornar a string “ola, \$primeiroNome”.
10. Crie um novo objeto `$usuarioRambo` com o nome “Sylvester” e o sobrenome “Stallone”.
11. Utilize o comando `echo` no método `ola()` para o objeto `$usuarioRambo` e veja o resultado.
12. Adicione um método `registrar()` à classe que retorne a string “>> registrado”.
13. Adicione um método `email()` à classe que retorne a string “>> email enviado”.
14. Adicione `return $this` ao método `ola()` para que ele possa ser encadeado a qualquer outro método na classe.
15. Adicione `return $this` ao método `registrar()` para que ele também possa ser encadeado.
16. Crie um novo objeto `$usuarioJair` com o primeiro nome de “Jair”. Para este objeto, encadeie os métodos na seguinte ordem: `ola()->registrar()->email()`

Observe que cada método que queremos encadear deve retornar a palavra-chave `$this` para não quebrar a cadeia. Portanto, os métodos `ola()` e `registrar()` devem retornar a palavra-chave `$this`, mas não há necessidade de retornar `$this` do método `email()`, pois ele termina a cadeia.

Código:

```
<?php

class Usuario {

    private $nome;
    private $sobrenome;

    function __construct($n, $s) {
        $this->nome = $n;
        $this->sobrenome = $s;
    }

    function dialogo() {
        echo "Ola, $this->nome $this->sobrenome";

        return $this;
    }

    function registrar() {
        echo ">> registrado";

        return $this;
    }

    function email() {
        echo ">> email enviado";

        return $this;
    }
}

$usuario01 = New Usuario("Leonardo", "Oliveira");
$usuario01->dialogo();

echo "<br>";
echo "<br>";

$usuario02 = New Usuario("Johnny", "Walker");
$usuario02->dialogo();
```

```
echo "<br>";  
echo "<br>";  
  
$usuarioRambo = New Usuario("Sylvester", "Stallone");  
$usuarioRambo->dialogo();  
  
$usuarioJair = New Usuario("Jair", "");  
  
echo "<br>";  
echo "<br>";  
  
$usuarioJair->dialogo()->registrar()->email();  
  
?>
```