

INSTITUIÇÃO: SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL - SENAI

UNIDADE: CIMATEC

DISCIPLINA: Programação Web com Software Livre - **TURMA:** 2016.2 - **DOCENTE:** GIL JADER

CURSO: TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - **CH:** 60h

LISTA DE EXERCÍCIOS – Introdução a O.O. em PHP (Lista3)

O PHP5 introduziu o conceito de interceptação (Métodos Mágicos) em operações realizadas por objetos por meio de métodos inicializados por `__` (*underline duplo*). O PHP reserva todas as funções com nomes iniciadas com `__` como mágicas. É recomendado que não se utilize funções com nomes com `__` no PHP.

1. O método `__set()` intercepta a atribuição de valores a propriedades do objeto. Sempre que for atribuído um valor a uma propriedade do objeto, automaticamente esta atribuição passa pelo método `__set()`, o qual recebe o nome da propriedade e o valor a ser atribuído, pode atribuí-lo ou não.

Crie uma classe denominada Cachorro (salve como Cachorro.class.php) que possua apenas um único atributo privado (Nascimento). A classe deverá utilizar o método mágico `__set()` que deverá interceptar as atribuições para o atributo Nascimento, validando o valor (uma data) atribuído a partir da verificação da existência da existência de três parte separadas por “/”.

- a. Caso o valor informado atenda a validação, deverá ser exibido o valor que foi atribuído e a mensagem informando que o valor foi atribuído.
- b. Caso o valor informado não atenda a validação, deverá ser exibido o valor que foi atribuído e uma mensagem informando que o valor **não** foi atribuído.
- c. Criar um script nomeado de set.php para criar instâncias da classe Cachorro e visualizar os resultados.

2. O método `__get()` intercepta requisições de propriedades. Sempre que for requisitada uma propriedade, automaticamente essa requisição passará pelo método `__get()`, o qual recebe o nome da propriedade requisitada, podendo retorná-la ou não.

Crie uma classe denominada ProdutoGet (salve como ProdutoGet.class.php) que possua os atributos públicos: (Codigo, Descricao e Quantidade) e apenas um único atributo privado (Preco). A classe deverá ter uma constante nomeada MARGEM com valor inicial de 10. A classe deverá utilizar o método mágico `__get()` que deverá interceptar as requisições para a propriedade Preco, retornando o preço com o valor ajustado com a margem de lucro. Utilize a fórmula: $\text{propriedade} * (1 + (\text{MARGEM}/100))$.

- a. Criar um script nomeado de get.php para criar instâncias da classe ProdutoGet e visualizar os resultados. Passe como parâmetros para testar um objeto criado os valores:
 - i. Código = 1
 - ii. Descrição = 'Pen Drive 512MB'
 - iii. Quantidade = 1
 - iv. Preço = 345.67

3. O método `__call()` intercepta a chamada de métodos. Sempre que for executado um método que não existir no objeto, a execução será direcionada para ele, que recebe dois parâmetros, o nome do método requisitado e o parâmetro recebido, podendo decidir o procedimento a realizar.

Crie uma classe denominada ProdutoCall (salve como ProdutoCall.class.php) que possua os atributos públicos: (Código, Descrição e Quantidade) e apenas um único atributo privado (Preço). A classe deverá ter uma constante nomeada MARGEM com valor inicial de 10. A classe deverá utilizar o método mágico `__call()` que deverá interceptar as instâncias criadas dessa classe.

- a. Criar um script nomeado de call.php para criar instâncias da classe ProdutoCall e visualizar os resultados. Tente executar um método denominado **Vender(10)** (quem não existe na classe) e verifique o resultado.

4. Quando imprimimos objetos na tela, por meio de comandos como `echo` e `print`, o PHP exibe no console o identificador interno do objeto, por exemplo:
 - Object id #1
 - Object id #2

Para alterar esse comportamento, podemos definir o método `__toString()` para cada classe. Caso o método `__toString()` exista, no momento em que mandamos exibir um objeto no console, o PHP irá imprimir o retorno dessa função.

Crie uma classe denominada CachorroToString (salve como CachorroToString.class.php). A classe deverá utilizar o método mágico `__toString()` que deverá ser executado sempre que um objeto criado desta classe for impresso. O método deverá retornar a propriedade \$nome que deve ser definida no método construtor.

- a. Criar um script nomeado de toString.php para criar instâncias da classe CachorroToString e em seguida visualize os resultados imprimindo **diretamente** o os objetos criados por meio do comando **echo**.

5. O comportamento padrão do PHP quando atribuímos um objeto a outro é criar uma referência entre objetos. Dessa forma, teremos duas variáveis apontando para a mesma região da memória. Então, quando precisamos duplicar um objeto na memória, podemos utilizar o método `__clone`, responsável por definir o comportamento da ação de clonagem, atuando diretamente nas propriedades do objeto resultante dessa ação.

Crie uma classe denominada CachorroClone (salve como CachorroClone.class.php) que possua os atributos públicos: (coleira, nome, idade e raca). A classe deverá utilizar o método mágico `__clone()` que deverá interceptar as instâncias criadas dessa classe.

- O método clone quando acionado deverá incrementar em 1 o código da coleira, zerar a idade e acrescentar o nome Júnior no final do nome.
- Criar um script nomeado de clone.php para criar instâncias da classe CachorroClone e visualizar os resultados.
- No script clone.php, crie um objeto da classe CachorroClone passando os parâmetros (100, 'Totó', 10, 'Fox Terrier') e em seguida clone o objeto criado. Exiba os valores do objeto criado e do objeto clonado.

6. Sempre que se instancia um objeto em PHP, é necessário ter a declaração da classe na memória e podemos fazer isso, por exemplo, utilizando o `include_once`. Podemos realizar tal operação no início da aplicação introduzindo todas as classes que poderão ser necessárias durante a execução da aplicação.

Para simplificar tal procedimento, o PHP introduziu a função `__autoload()`, ou “carga automática”. Com ela, a carga da classe é realizada de forma dinâmica sempre que um objeto for instanciado.

Crie um script nomeado autoload.php que deverá a função de carga automática `__autoload()` para incluir automaticamente as classes que serão instanciadas. Em seguida crie um objeto da classe Produto e exiba os valores de suas propriedades.

7. Pesquise sobre o método `toXml()` e implemente um exemplo.