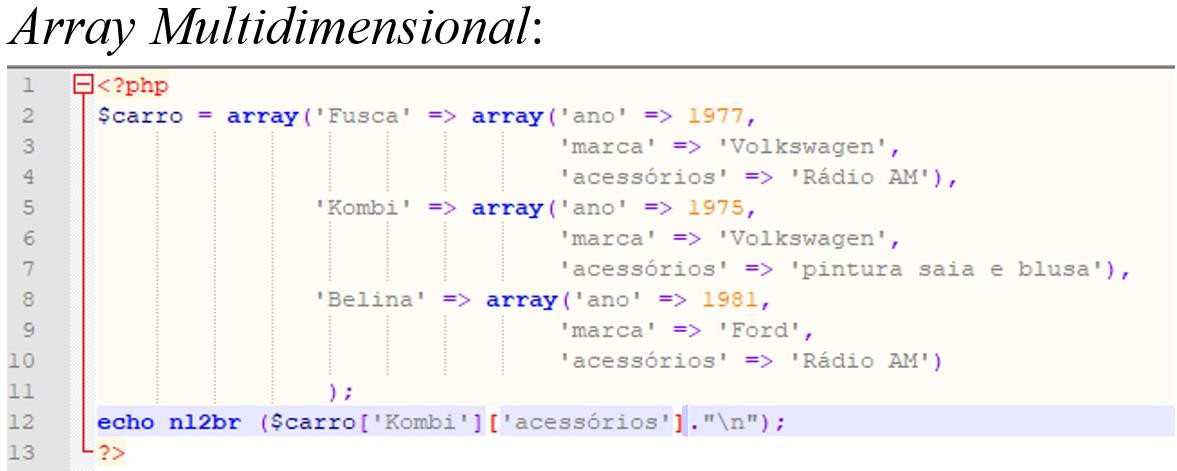
Trabalho 1 – Tópicos Especiais – PHP

# Componentes do grupo

Nome: matrícula Nome: matrícula Nome: matrícula Nome: matrícula

# Instruções

* data para entrega: 23/09 –Avaliação do 1º bimestre
* em grupos de até 4 alunos.
* considerar no trabalho o código fonte e o *print* das simulações – fazer os tópicos dos exercícios nos espaços indicados neste documento.
* nomear o arquivo como “Trabalho 1 de Tópicos Especiais + nome de um dos componentes do grupo”.
* entregar o trabalho em arquivo PDF no *email* do professor: [mauricio.mario@fatec.sp.gov.br](mailto:mauricio.mario@fatec.sp.gov.br)
* enviar o arquivo pelo *email* institucional da Fatec de um dos membros do grupo. Arranjo multidimensional – Cidade



**Exercício 1:** Seguindo a estrutura do arranjo $carro dada acima, criar um arranjo Cidade com atributos pertinentes, e mostrar no navegador os dados relativos a estes atributos.

# código fonte:

<?php

$cidade = array(

'Santos' => array(

'estado' => 'SP',

'país' => 'Brasil',

'região' => 'Sudeste'

),

'São Borja' => array(

'estado' => 'RS',

'país' => 'Brasil',

'região' => 'Sul'

)

);

print\_r ($cidade['Santos']['estado']. "\n");

print\_r ($cidade['Santos']['país']. "\n");

print\_r ($cidade['Santos']['região']. "\n");

?>

**resultado da execução no navegador:**

**Exercício 2:** Escrever código que represente a abstração de um condomínio, com atributos pertinentes e os respectivos métodos set( ) e get( ) para inserir e retornar os valores dos atributos.

# código fonte:

<?php

class Condominio {

private $endereco, $proprietario;

public function set\_atributos($endereco, $proprietario) {

$this->endereco = $endereco;

$this->proprietario = $proprietario;

}

public function get\_atributos() {

return [

'Endereço' => $this->endereco,

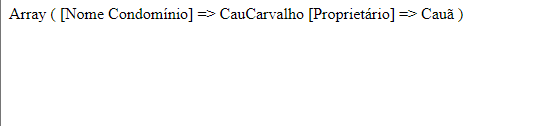
'Proprietário' => $this->proprietario

];

}

}

?>

**resultado da execução no navegador:**

**Exercício 3:** Implementar a abstração de uma padaria, também com atributos pertinentes e os métodos set ( ) e get ( ) de manipulação dos atributos.

# código fonte:

<?php

class Padaria {

private $endereco, $donoPadaria, $horarioAbertura, $enderecoNumero;

public function set\_atributos($endereco, $donoPadaria, $horarioAbertura, $enderecoNumero) {

$this->endereco = $endereco;

$this->donoPadaria = $donoPadaria;

$this->horarioAbertura = $horarioAbertura;

$this->enderecoNumero = $enderecoNumero;

}

public function get\_atributos() {

return [

'Endereço' => $this->endereco,

'Número da padaria' => $this->enderecoNumero,

'Dono da padaria' => $this->donoPadaria,

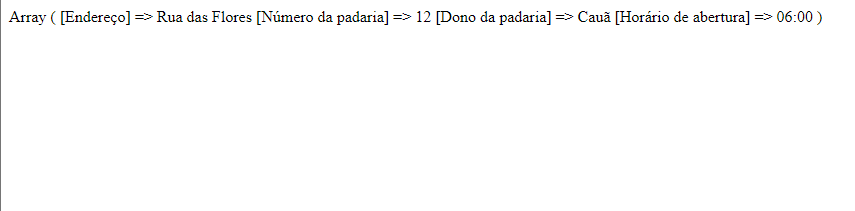
'Horário de abertura' => $this->horarioAbertura

];

}

}

?>

**resultado da execução no navegador:**

**Exercício 4:** Criar aplicação PHP que acesse através de variáveis de instância as classes Condomínio e Padaria, e que possa inserir valores aos atributos de ambas e mostrá-los no navegador.

***rofº Mauricio Conceição Mario***

# código fonte:

**resultado da execução no navegador:**

**Exercício 5:** Considerar a abstração de um automóvel, com os atributos:

* + marca;
  + cor;
  + modelo;
  + preço.

Utilizar o padrão de encapsulamento de Orientação a Objetos, ou seja, métodos públicos e atributos privados. Construir aplicação PHP para que os valores dos atributos possam ser inseridos através de um arquivo HTML, utilizando métodos set(S\_POST[ ]), e retornados através do navegador com métodos get( ).

# código fonte da abstração:

***p***

**Arquivo HTML:**

# código fonte da aplicação PHP:

**resultado da execução no navegador:**

## profº Mauricio Conceição Mario