

Técnico em Desenvolvimento de Sistemas

Programação WEB 3

Linguagem PHP – Orientação a Objetos

Prof. Laércio Silva

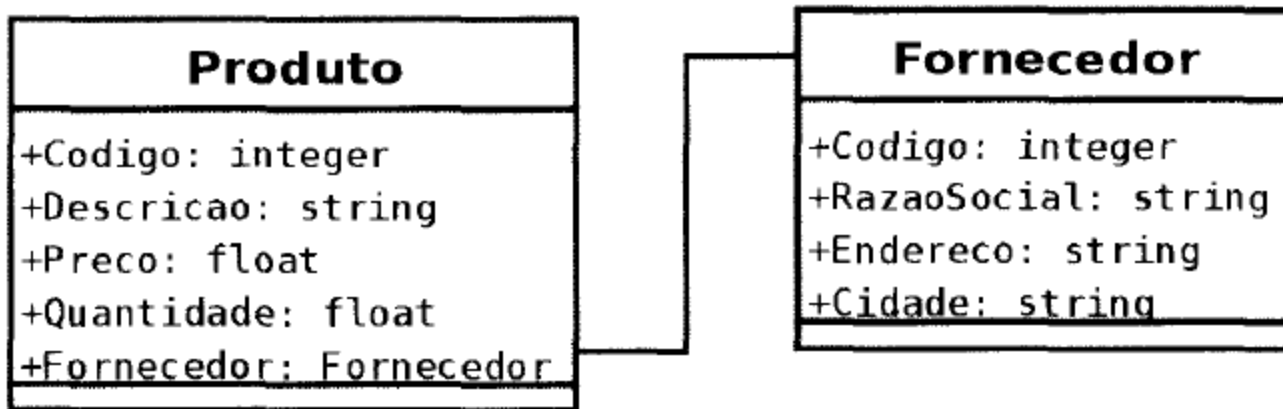
laercio.silva31@etec.sp.gov.br

Lndsilva@hotmail.com

Associação

- É a relação mais comum entre dois objetos, de modo que um possui uma referência à posição da memória onde o outro se encontra, podendo visualizar seus atributos ou mesmo acionar uma de suas funcionalidades (métodos).
- A forma mais comum de implementar uma associação é ter um objeto como atributo de outro.

Associação

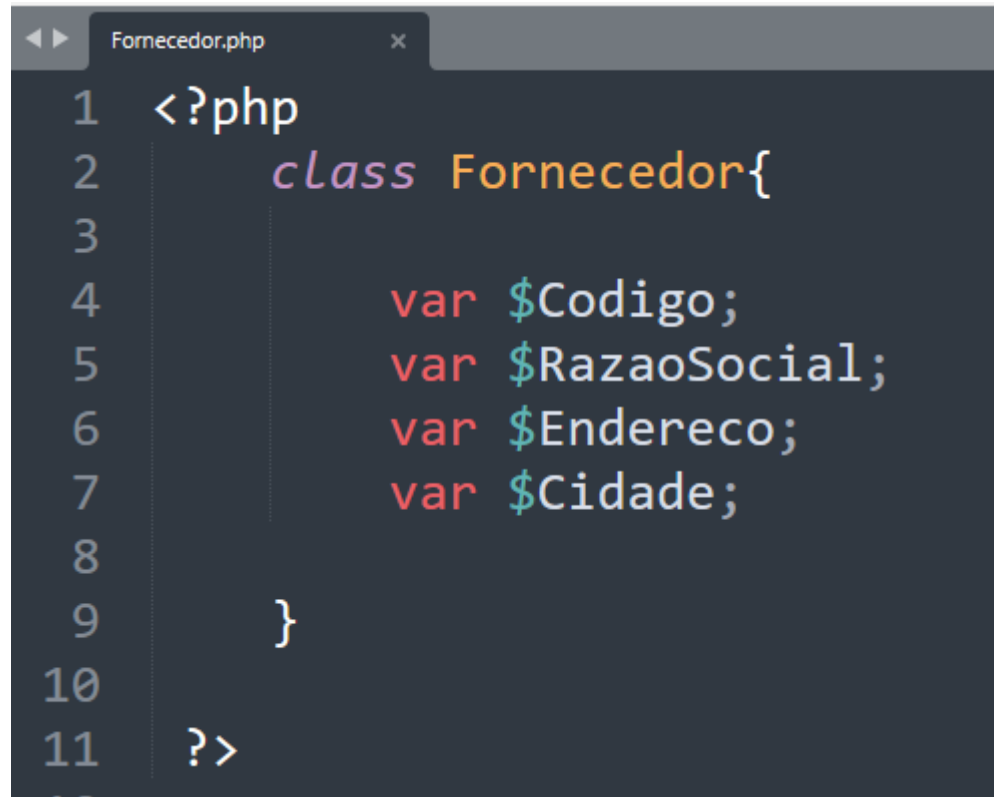


Programação a Objetos em PHP:

Associação

```
Produto.php
1  <?php
2
3      class Produto {
4          var $Codigo;
5          var $Descricao;
6          var $Preco;
7          var $Quantidade;
8
9
10         function ImprimeEtiqueta()
11         {
12             print 'Código: ' . $this->Codigo . "<br>";
13             print 'Descrição: ' . $this->Descricao . "<br><br>";
14         }
15     }
16
17  ?>
18
```

Associação



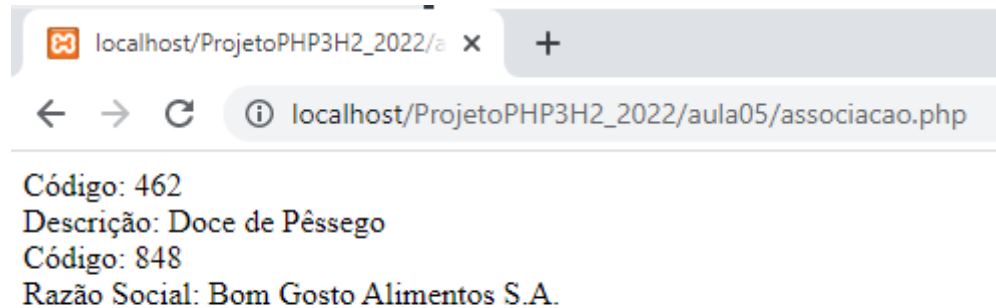
```
1 <?php
2     class Fornecedor{
3
4         var $Codigo;
5         var $RazaoSocial;
6         var $Endereco;
7         var $Cidade;
8
9     }
10
11 ?>
```

The image shows a code editor window titled 'Fornecedor.php'. The code defines a PHP class named 'Fornecedor'. The class has four public properties: '\$Codigo', '\$RazaoSocial', '\$Endereco', and '\$Cidade'. The code is written in a dark-themed editor with line numbers from 1 to 11.

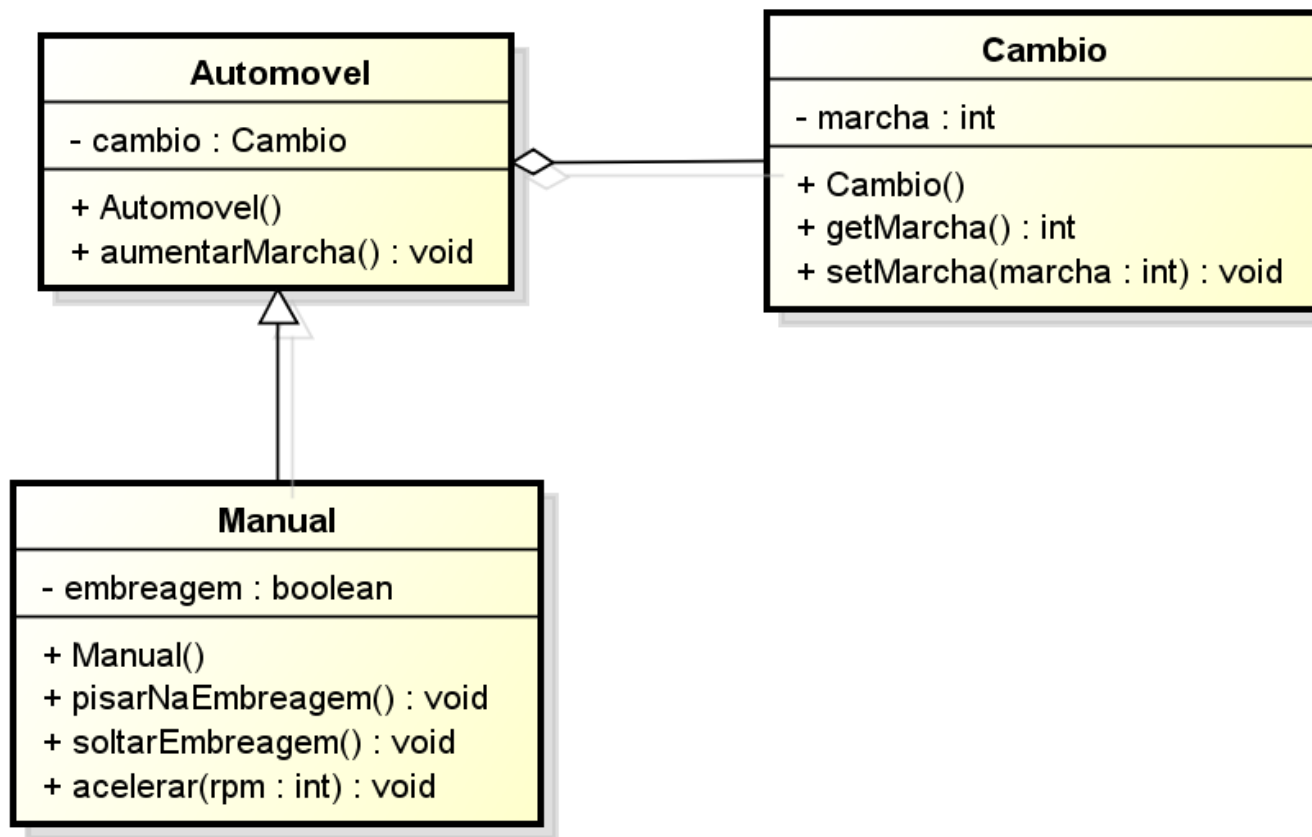
Associação

```
associacao.php x
1 <?php
2
3 include_once 'Fornecedor.php';
4 include_once 'Produto.php';
5
6 //instancia Fornecedor
7 $fornecedor= new Fornecedor();
8
9 $fornecedor->Codigo = 848;
10 $fornecedor->RazaoSocial= ' Bom Gosto Alimentos S.A.';
11 $fornecedor->Endereco = ' Rua Ipi ranga' ;
12 $fornecedor->Cidade = ' Poços de Caldas';
13
14 //instancia Produto
15 $produto= new Produto();
16
17 $produto->Codigo = 462;
18 $produto->Descricao = ' Doce de Pêssego';
19 $produto->Preco = 1.24;
20 $produto->Fornecedor = $fornecedor;
21
22 //imprime atributos
23 echo 'Código: ' . $produto->Codigo . "<br>";
24 echo 'Descrição: ' . $produto->Descricao . "<br>";
25 echo 'Código: ' . $produto->Fornecedor->Codigo . "<br>";
26 echo 'Razão Social: ' . $produto->Fornecedor->RazaoSocial . "<br>";
27
28 $produto->Codigo . "<br>";
29 $produto->Descricao . "<br>";
30 $produto->Fornecedor->Codigo . "<br>";
31 $produto->Fornecedor->RazaoSocial . "<br>";
32
33 ?>
```

Associação



Exercício - Associação



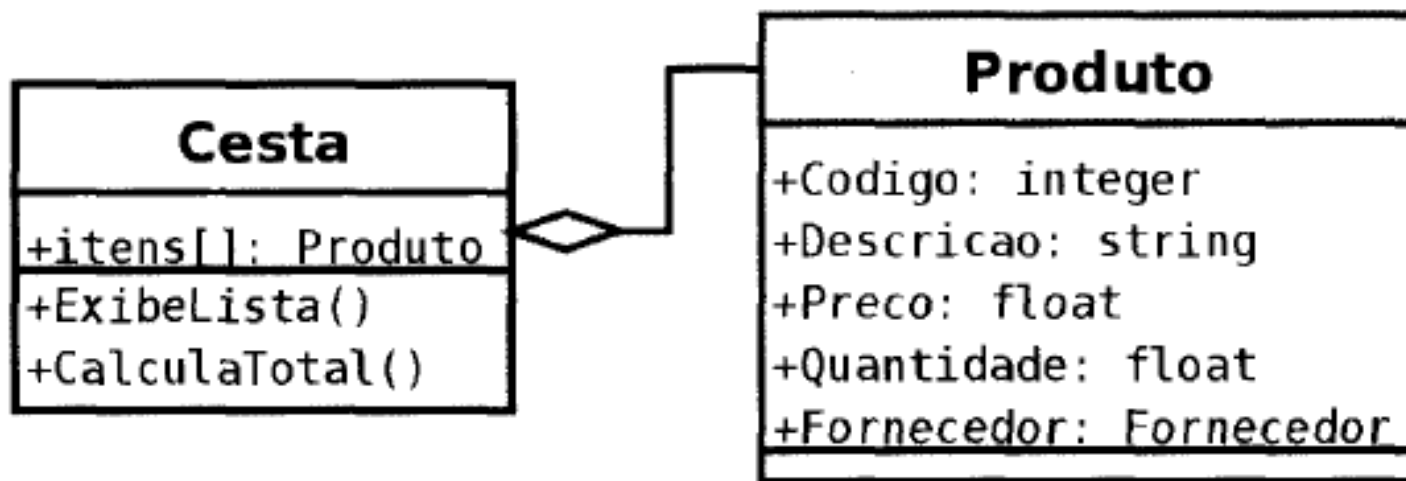
Agregação

- É o tipo de relação entre objetos conhecida como todo ou parte.
- Na agregação, um objeto agrega outro objeto, ou seja, torna um objeto externo parte de si mesmo pela utilização de um dos seus métodos.
- Assim, o objeto-pai poderá utilizar funcionalidades do objeto agregado.
- Nesta relação, um objeto poderá agregar uma ou muitas instâncias de um outro objeto. Para agregar muitas instâncias, a forma mais simples é utilizando arrays.
- Criamos um array como atributo da classe, sendo que o papel deste array é armazenar inúmeras instâncias de uma outra classe.

Agregação

- Utilizaremos o clássico exemplo de uma cesta de compras.
- A classe Cesta é o nosso todo e a classe Produto representa cada uma das partes.
- Uma Cesta é composta de inúmeros produtos (instâncias da classe Produto).
- Para agregar objetos do tipo Produto à Cesta, criamos o método `Adicionaltem()` na classe Cesta, que conta também com o método `ExibeLista()`, o qual, na verdade, chama o método `ImprimeEtiqueta()` de cada um dos produtos da Cesta e o método `CalculaTotal()`, que soma o preço de cada um dos produtos de uma Cesta.

Agregação



Programação a Objetos em PHP:

Agregação

```
1  <?php
2
3      class Produto {
4          var $Codigo;
5          var $Descricao;
6          var $Preco;
7          var $Quantidade;
8
9
10         function ImprimeEtiqueta()
11         {
12             print 'Código: ' . $this->Codigo . "<br>";
13             print 'Descrição: ' . $this->Descricao . "<br><br>";
14         }
15     }
16
17  ?>
```

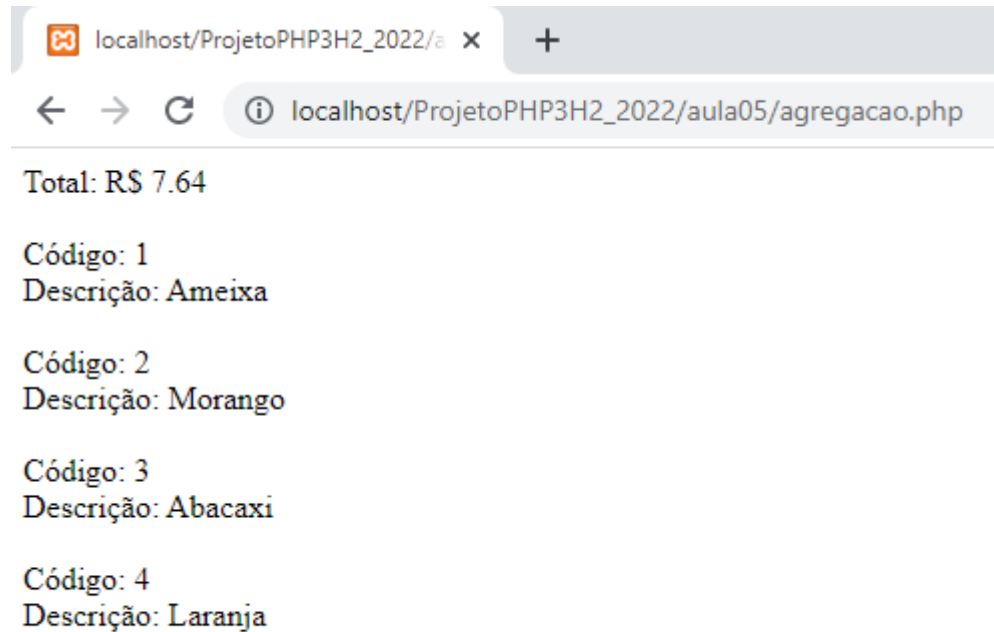
Agregação

```
Cesta.php x
1  <?php
2
3      class Cesta{
4
5
6          private $itens;
7
8          #adiciona itens na cesta
9          function AdicionaItem(Produto $item){
10             $this->itens[] = $item;
11         }
12
13         #exibe a lista de produtos
14         function ExibeLista(){
15             foreach ($this->itens as $item) {
16                 $item->ImprimeEtiqueta();
17             }
18         }
19
20         #calcula o valor total da cesta
21         function CalculaTotal(){
22             $total = 0;
23             foreach ($this->itens as $item) {
24
25                 $total += $item->Preco;
26
27             }
28
29             return 'Total: R$ ' . $total . '<br>';
30         }
31     }
32
33
34  ?>
```

Agregação

```
agregacao.php
1  <?php
2
3      include_once 'Cesta.php';
4      include_once 'Produto.php';
5
6      $produto1 = new Produto;
7      $produto2 = new Produto;
8      $produto3 = new Produto;
9      $produto4 = new Produto;
10
11     $produto1->Codigo = 1;
12     $produto1->Descricao = 'Ameixa';
13     $produto1->Preco = 1.40;
14
15     $produto2->Codigo = 2;
16     $produto2->Descricao = 'Morango';
17     $produto2->Preco = 2.24;
18
19     $produto3->Codigo = 3;
20     $produto3->Descricao = 'Abacaxi';
21     $produto3->Preco = 2.86;
22
23     $produto4->Codigo = 4;
24     $produto4->Descricao = 'Laranja';
25     $produto4->Preco = 1.14;
26
27     $cesta= new Cesta;
28
29     $cesta->AdicionaItem($produto1);
30     $cesta->AdicionaItem($produto2);
31     $cesta->AdicionaItem($produto3);
32     $cesta->AdicionaItem($produto4);
33
34     echo $cesta->CalculaTotal();
35     echo "<br>"; //quebra de linha
36     echo $cesta->ExibeLista();
37
38     ?>
```

Agregação



Agregação

- Talvez você já tenha percebido um possível problema neste tipo de abordagem.
- O que aconteceria se adicionássemos objetos que não são da classe Produto a uma Cesta?
- Se chamássemos métodos como `CalculaTotal()` e `ExibeLista()`, por exemplo, teríamos problemas, uma vez que eles confiam na existência do atributo `Preco` e do método `ImprimeEtiqueta()` da classe `produto`.

Agregação

- O PHP5 introduz o conceito de *TypeHinting*, ou seja, "*sugestão de tipo*".
- Nesse exemplo da classe cesta, o método Adicionaltem() recebe um produto chamado \$item.
- Veja que na frente do parâmetro indicamos a classe à qual ele deve pertencer.
- Caso o método Adicionaltem() seja executado passando um parâmetro que não é do tipo Produto, um erro fatal será lançado.

```
function Adicionaltem(Produto $item)
```

Programação a Objetos em PHP:

Agregação

```
agregacao2.php x
1  <?php
2
3      include_once 'Cesta.php';
4      include_once 'Fornecedor.php';
5      include_once 'Produto.php';
6
7      $fornecedor= new Fornecedor;
8      $fornecedor->RazaoSocial = 'Produtos Bom Gosto S.A. ';
9
10     $cesta= new Cesta();
11
12     $cesta->Adicionaltem($fornecedor);
13
14     $cesta->CalculaTotal();
15
16     ?>
17
```

Agregação

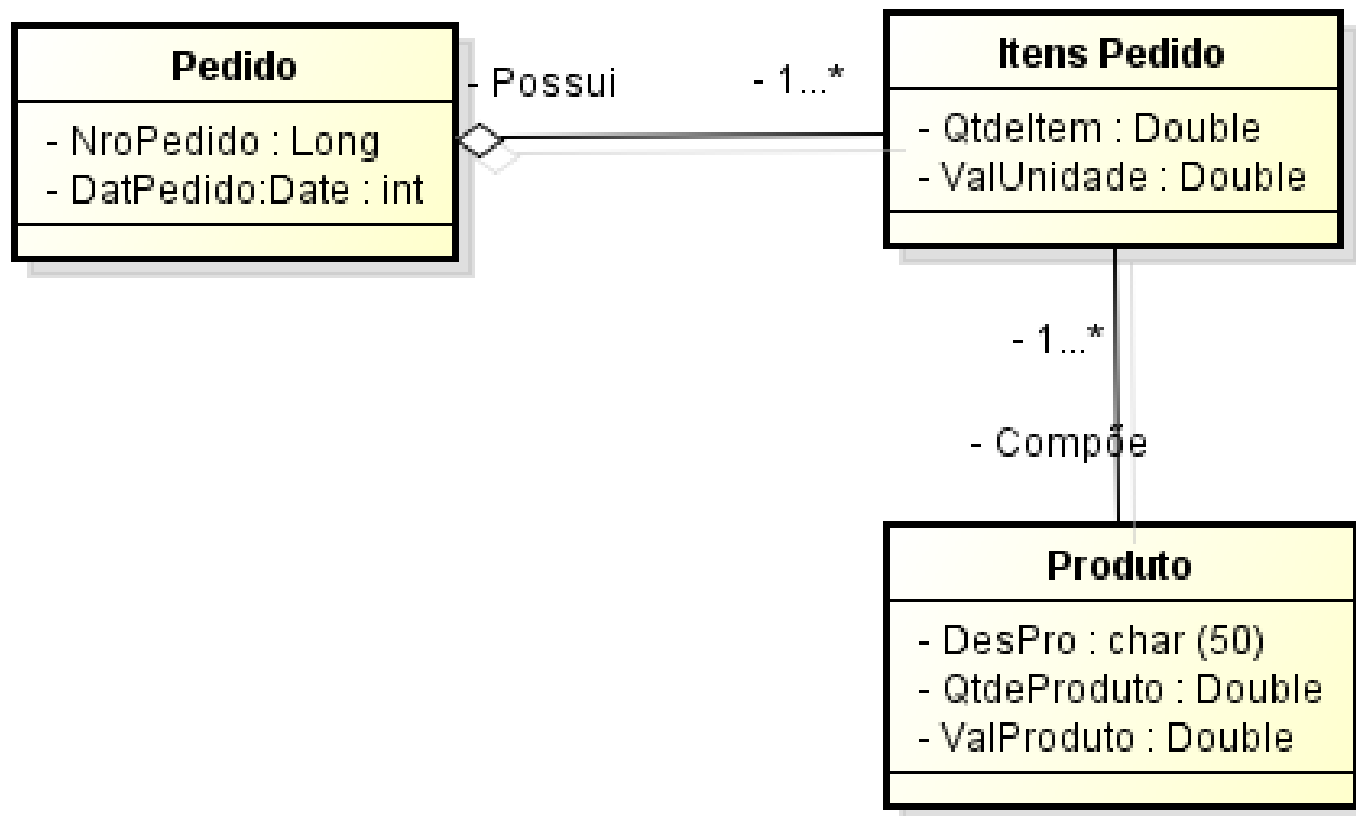


localhost/ProjetoPHP3H2_2022/aula05/agregacao2.php



Fatal error: Uncaught Error: Call to undefined method Cesta::Adicionaltem() in C:\xampp\htdocs\ProjetoPHP3H2_2022\aula05\agregacao2.php:12 Stack trace: #0 {main} thrown in C:\xampp\htdocs\ProjetoPHP3H2_2022\aula05\agregacao2.php on line 12

Exercício - Agregação



Muito Obrigado

Até a Próxima Aula

Prof. Laércio Silva
Email: Indsilva@hotmail.com