

Herança e Polimorfismo

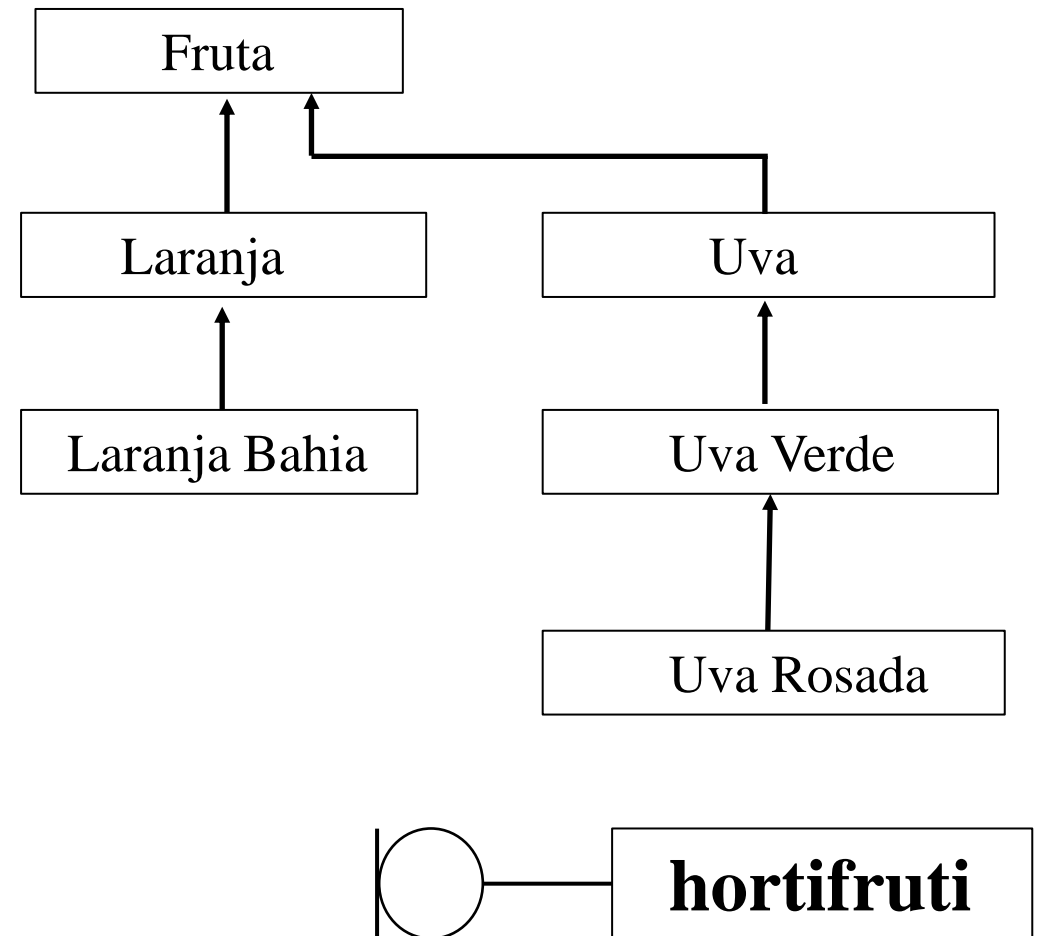
profº Mauricio Conceição Mario

Herança

- Herança é uma propriedade originada a partir de uma ligação hierárquica entre classes, onde classes que estão em camadas inferiores dessa ligação hierárquica podem herdar atributos e comportamentos das classes que estão em hierarquia superior. O uso de Herança permite, portanto, o reaproveitamento de propriedades de uma classe.
- A classe que se encontra no topo da hierarquia é denominada de *superclasse* ou *classe-mãe*, enquanto que as classes que estão na hierarquia inferior são chamadas de *subclasses* ou *classes-filhas*.

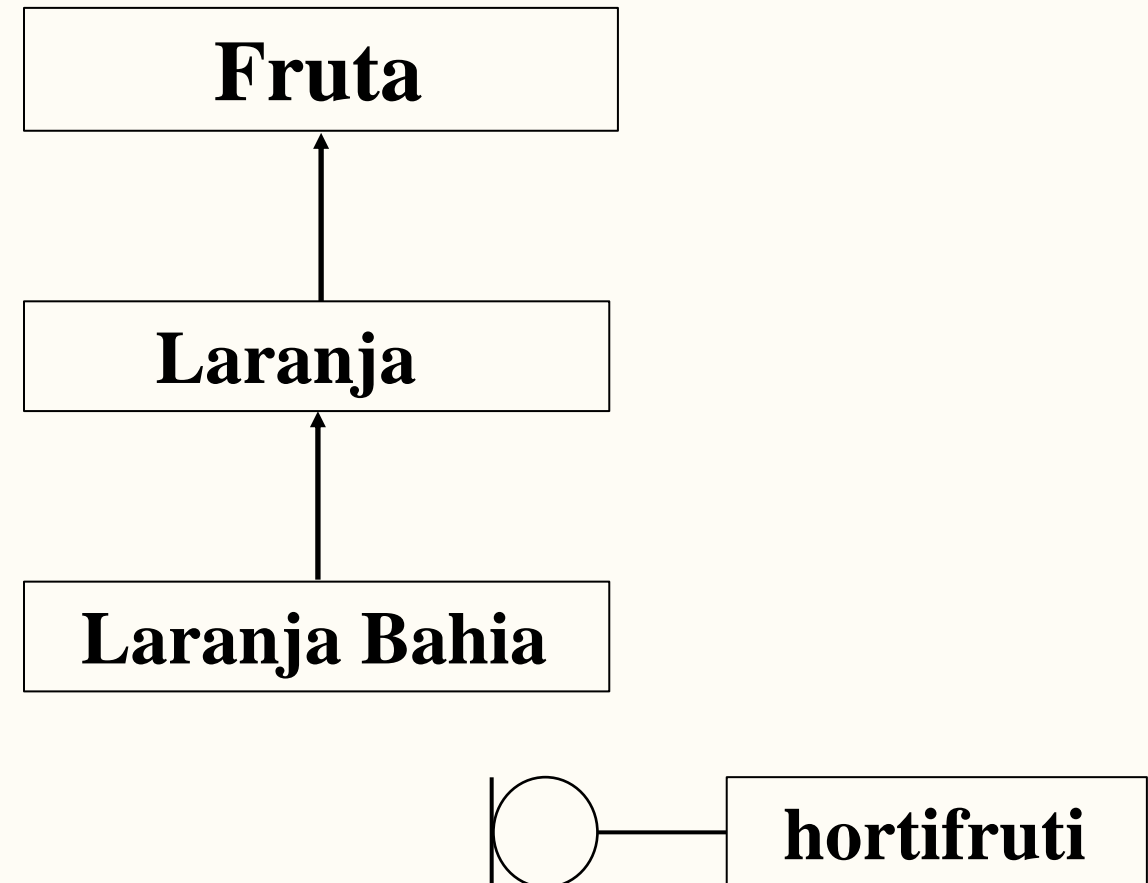
Exercício 10: De acordo com o diagrama de classes ao lado, inserir na relação de herança relativa à abstração das frutas, a classe “Uva” (no mesmo nível hierárquico da classe “Laranja”), e as classes-filhas “Uva Verde” e “Uva Rosada”.

Utilizar a aplicação “hortifruti” para inserir e retornar atributos pertinentes a estas novas classes da aplicação. Fazer a simulação no navegador.



Exemplo: superclasse ou classe-mãe → Fruta.class.php

```
1 <?php
2 class Fruta
3 {
4     protected $arvore;
5     protected $nome_fruta;
6     protected $preço;
7
8     public function __construct($detalhe){
9         print "detalhe da fruta: <br> {$detalhe}<br>\n";
10    }
11
12    public function set_arvore($arvore){
13        $this->arvore = $arvore;
14    }
15
16    public function get_arvore(){
17        return $this->arvore;
18    }
19
20    public function set_nome_fruta($nome_fruta){
21        $this->nome_fruta = $nome_fruta;
22    }
23
24    public function get_nome_fruta(){
25        return $this->nome_fruta;
26    }
27
28    public function set_preço($preço){
29        $this->preço = $preço;
30    }
31
32    public function get_preço(){
33        return $this->preço;
34    }
35 }
36 ?>
```



```

1 <?php
2 class Uva extends Fruta
3 {
4     protected $origem;
5
6     public function __construct($f) {
7         parent::__construct($f);
8     }
9
10    public function set_origem($origem) {
11        $this->origem = $origem;
12    }
13
14    public function get_origem() {
15        return $this->origem;
16    }
17 }
18 ?>

```

```

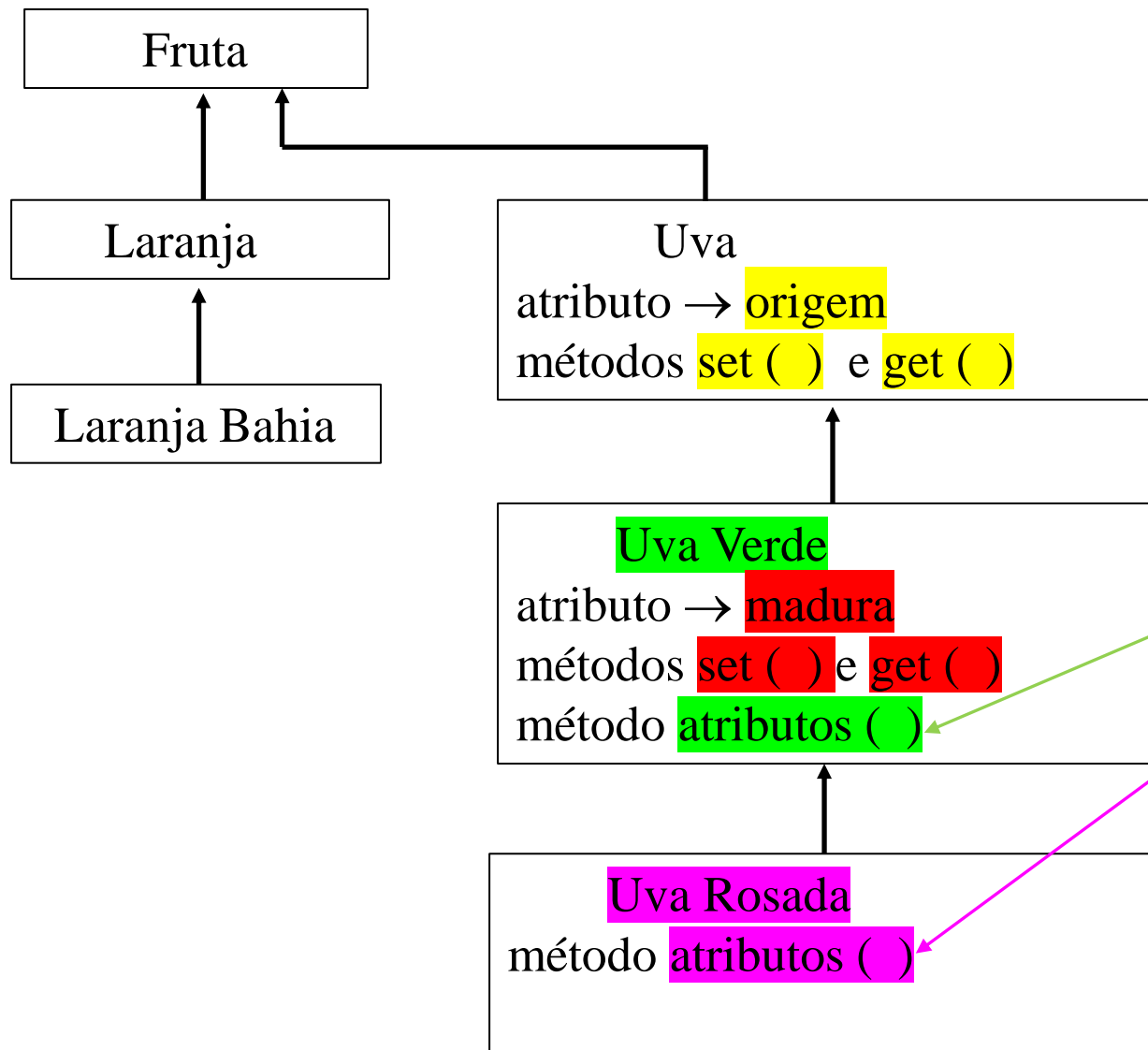
1 <?php
2 class Uva_Rosada extends Uva_Verde
3 {
4
5     public function __construct($e) {
6         parent::__construct($e);
7     }
8
9     public function atributos() {
10        echo nl2br("na hierarquia desta herança
11        a classe Uva_Rosada aparece como classe
12        filha de Uva_Verde e pode acessar os
13        atributos origem e madura\n");
14    }
15 }
16 ?>

```

```

1 <?php
2 class Uva_Verde extends Uva
3 {
4
5     protected $madura;
6     public function __construct($e) {
7         parent::__construct($e);
8     }
9
10    public function set_madura($madura) {
11        $this->madura = $madura;
12    }
13
14    public function get_madura() {
15        $this->madura;
16        if ($this->madura == True)
17            return "sim";
18        else return "não";
19    }
20
21    public function atributos() {
22        echo nl2br("na hierarquia desta herança
23        a classe Uva_Verde aparece no mesmo nível
24        da classe Laranja_Bahia e pode acessar o
25        atributo origem\n");
26    }
27 }
28 ?>
29

```



acesso ao método
`atributos ()` é feito
de acordo com a
instância, ou à classe
Uva Verde ou à classe
Uva Rosada.
→ caracteriza o conceito
de *Polimorfismo*: métodos
com a mesma assinatura
(identificação e forma)
porém com funcionalidades
diferentes.



```

1  <?php
2  require_once 'Fruta.class.php';
3  require_once 'Laranja.php';
4  require_once 'Laranja_Bahia.php';
5  require_once 'Uva.php';
6  require_once 'Uva_Verde.php';
7  require_once 'Uva_Rosada.php';
8
9  $f = new Laranja_Bahia("laranja bahia está bem doce");
10 $f->set_arvore("laranjeira");
11 $f->set_nome_fruta("laranja bahia");
12 $f->set_preço(8.22);
13 $f->set_cor("amarelo");
14 echo nl2br("dá na árvore: ".$f->get_arvore()."\n");
15 echo nl2br("fruta: ".$f->get_nome_fruta()."\n");
16 echo nl2br("preço = ".$f->get_preço()." a dúzia.\n");
17 echo nl2br("cor: ".$f->get_cor()."\n");
18 echo nl2br("\n");
19
20 $g = new Laranja("laranja lima está azeda");
21 $g->set_arvore("laranjeira");
22 $g->set_nome_fruta("laranja lima");
23 $g->set_preço(6.05);
24 $g->set_cor("amarelo-esverdeada");
25 echo nl2br("dá na árvore: ".$g->get_arvore()."\n");
26 echo nl2br("fruta: ".$g->get_nome_fruta()."\n");
27 echo nl2br("preço = ".$g->get_preço()." a dúzia.\n");
28 echo nl2br("cor: ".$g->get_cor()."\n");
29 echo nl2br("\n");
30
31 $h = new Fruta("maçã argentina está bem grande ");
32 $h->set_arvore("macieira");
33 $h->set_nome_fruta("maçã");
34 $h->set_preço(12.66);
35 echo nl2br("dá na árvore: ".$h->get_arvore()."\n");
36 echo nl2br("fruta: ".$h->get_nome_fruta()."\n");
37 echo nl2br("preço = ".$h->get_preço()." a dúzia.\n");
38 echo nl2br("\n");

```

```

39
40 $uv = new Uva_Verde("uva verde boa para salada de frutas");
41 $uv->atributos();
42 $uv->set_madura(True);
43 $uv->set_arvore("videira");
44 $uv->set_nome_fruta("uva verde");
45 $uv->set_preço(14.87);
46 $uv->set_origem("Vinhedo");
47 echo nl2br("dá na árvore: ".$uv->get_arvore()."\n");
48 echo nl2br("fruta: ".$uv->get_nome_fruta()."\n");
49 echo nl2br("está madura? ".$uv->get_madura()." "\n");
50 echo nl2br("preço = ".$uv->get_preço()." a dúzia.\n");
51 echo nl2br("cidade da uva: ".$uv->get_origem()."\n");
52 echo nl2br("\n");
53
54 $ur = new Uva_Rosada("uva rosada muito saborosa");
55 $ur->atributos();
56 $ur->set_madura(False);
57 $ur->set_arvore("videira");
58 $ur->set_nome_fruta("uva rosada");
59 $ur->set_preço(11.34);
60 $ur->set_origem("Jundiaí");
61 echo nl2br("dá na árvore: ".$ur->get_arvore()."\n");
62 echo nl2br("fruta: ".$ur->get_nome_fruta()."\n");
63 echo nl2br("está madura? ".$ur->get_madura()." "\n");
64 echo nl2br("preço = ".$ur->get_preço()." a dúzia.\n");
65 echo nl2br("cidade da uva: ".$ur->get_origem()."\n");
66 echo nl2br("\n");
67
68 ?>

```

detalhe da fruta:

laranja bahia está bem doce

dá na árvore: laranjeira

fruta: laranja bahia

preço = \$8.22 a dúzia

cor: amarelo

detalhe da fruta:

laranja lima está azeda

dá na árvore: laranjeira

fruta: laranja lima

preço = \$6.05 a dúzia

cor: amarelo-esverdeada

detalhe da fruta:

maçã argentina está bem grande

dá na árvore: macieira

fruta: maçã

preço = \$12.66 a dúzia

detalhe da fruta:

uva verde boa para salada de frutas

na hierarquia desta herança

a classe Uva_Verde aparece no mesmo nível

da classe Laranja_Bahia e pode acessar o

atributo origem

dá na árvore: videira

fruta: uva verde

está madura? sim

preço = \$14.87 a dúzia

cidade da uva: Vinhedo

detalhe da fruta:

uva rosada muito saborosa

na hierarquia desta herança

a classe Uva_Rosada aparece como classe

filha de Uva_Verde e pode acessar os

atributos origem e madura

dá na árvore: videira

fruta: uva rosada

está madura? não

preço = \$11.34 a dúzia

cidade da uva: São Roque

Exercícios:

11. Inserir na aplicação “hortifruti.php” uma variável do tipo “Uva”, e acessar todos os atributos e métodos que podem ser acessados por esta classe, mostrando os valores atribuídos no navegador.
12. Criar uma aplicação, utilizando as hierarquias de herança, para representar algumas espécies de animais.

Referências Bibliograficas

- *php* Programando com Orientação a Objetos
Pablo Dall'Oglio - editora Novatec - 2014
- Desenvolvimento web com PHP e MySQL
Evaldo Junior Bento – editora Casa do Código - 2018