

# Programação Orientada a Objetos com PHP

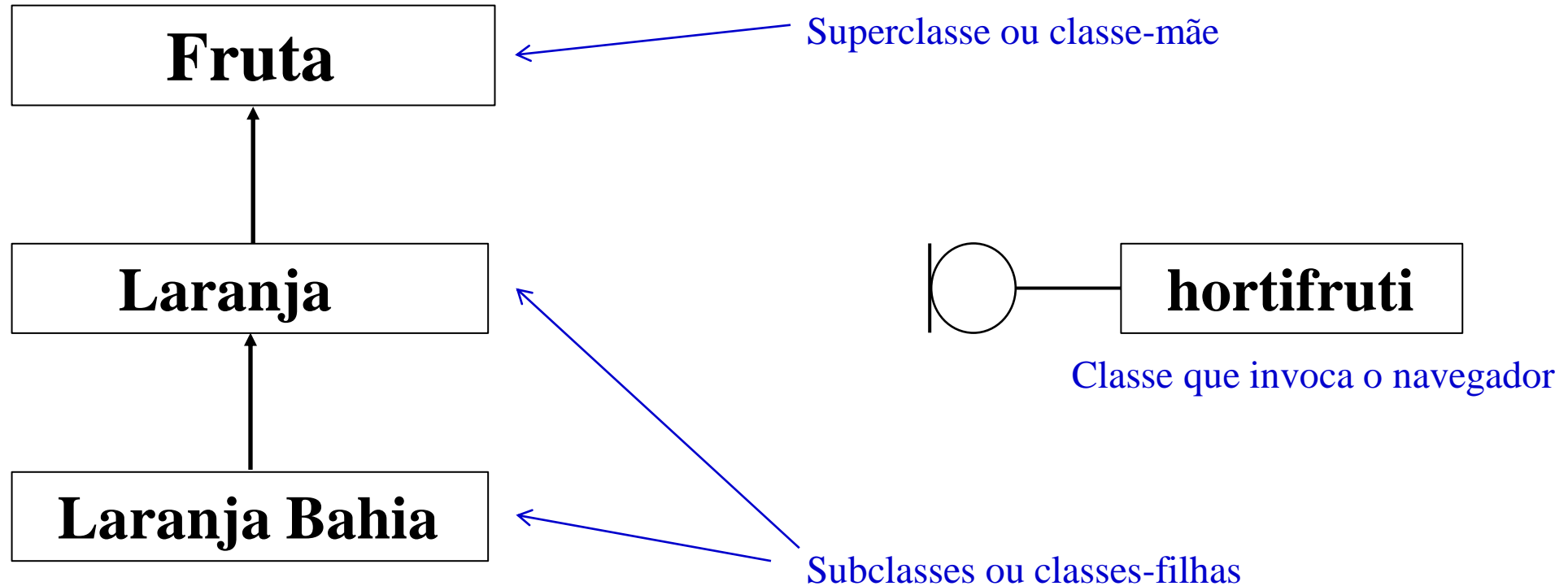
## **Herança**

*profº Mauricio Conceição Mario*

# Herança

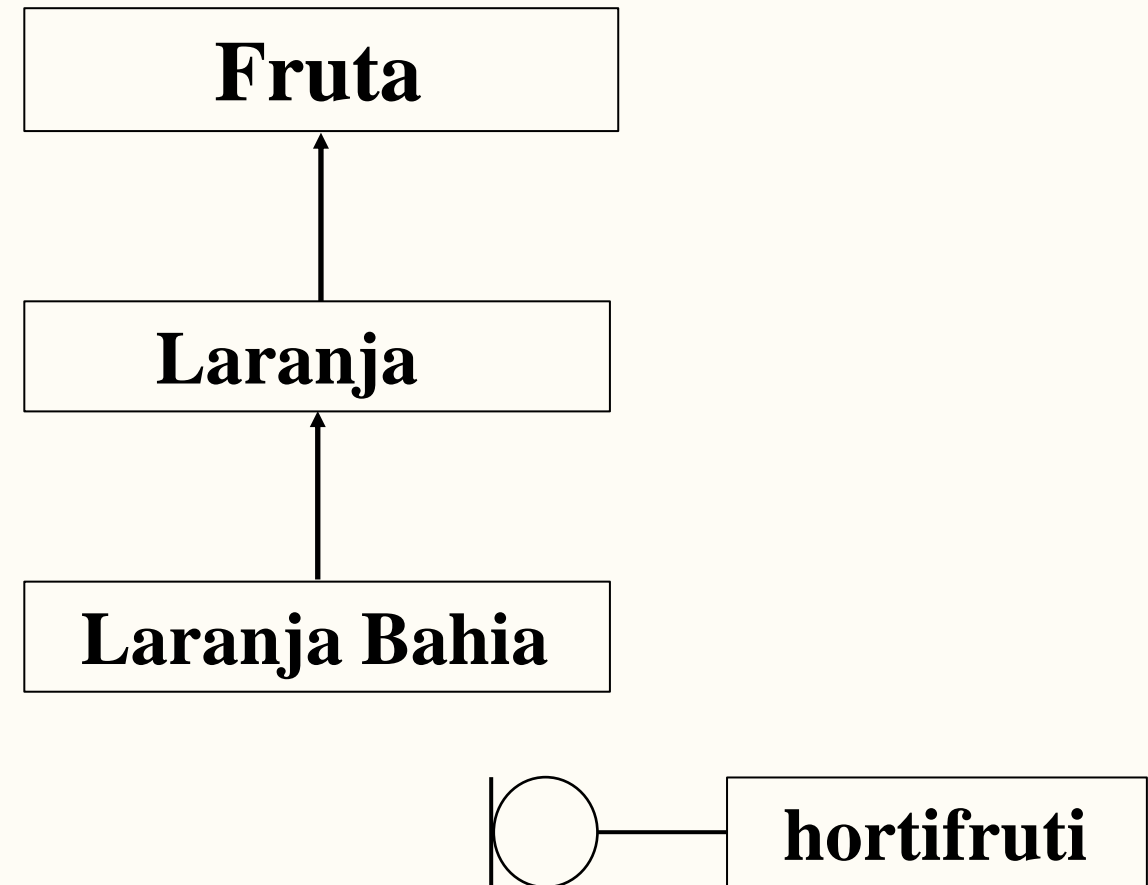
- Herança é uma propriedade originada a partir de uma ligação hierárquica entre classes, onde classes que estão em camadas inferiores dessa ligação hierárquica podem herdar atributos e comportamentos das classes que estão em hierarquia superior. O uso de Herança permite, portanto, o reaproveitamento 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*.

# Exemplo de hierarquia de classes em Herança, representando a abstração relativa às frutas:



## 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 ?>
```



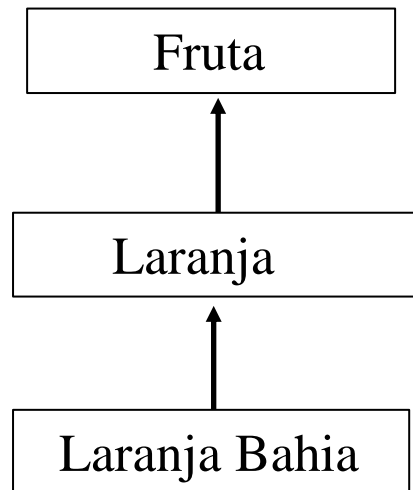
Subclasse ou classe-filha: **Laranja.php**

```

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

```

**parent::\_\_construct (...) → acesso ao método construtor da superclasse Fruta**

Subclasse ou classe-filha: **Laranja\_Bahia.php**

**parent::\_\_construct (...) → acesso ao método construtor da classe Laranja**

```

Fruta.class.php x Laranja.php x Laranja_Bahia.php x hortifruti.php x
1 <?php
2 class Laranja_Bahia extends Laranja
3 {
4     public function __construct($e) {
5         parent::__construct($e);
6     }
7
8
9
10
11
12
13
14 }
15 ?>

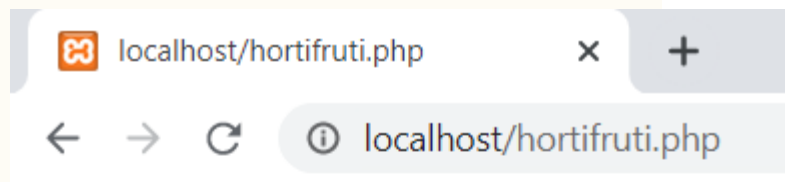
```

# aplicação: hortifruti.php

```

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

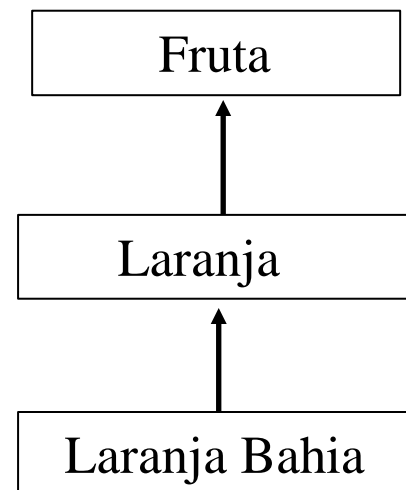
```



detalhe da fruta:  
 laranja bahia está bem doce  
 dá na árvore: laranjeira  
 fruta: laranja bahia  
 preço = \$8.22 a dúzia  
 cor: amarelo

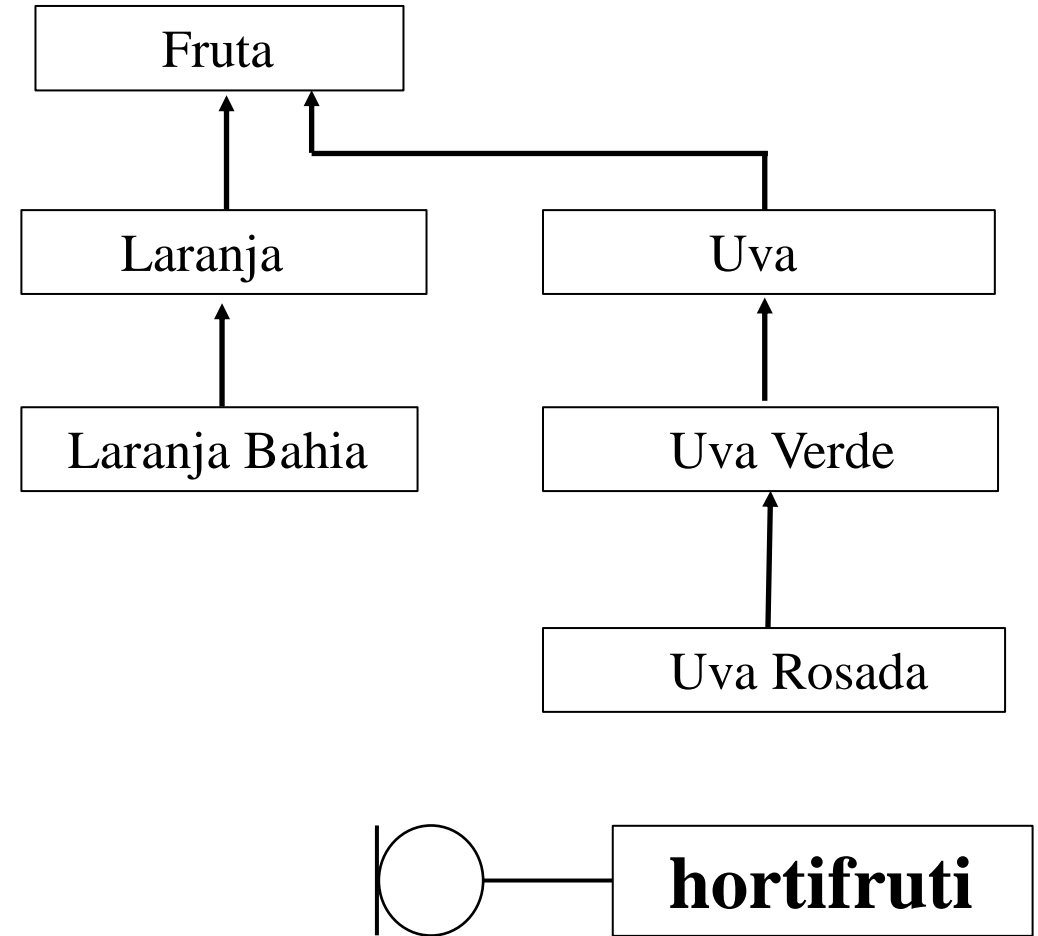
detalhe da fruta:  
 laranja lima está azeda  
 dá na árvore: laranjeira  
 fruta: laranja lima  
 preço = \$6.05 a dúzia  
 cor: amarelo-esverdeada

detalhe da fruta:  
 maçã argentina está bem grande  
 dá na árvore: macieira  
 fruta: maçã  
 preço = \$12.66 a dúzia



**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.



## 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