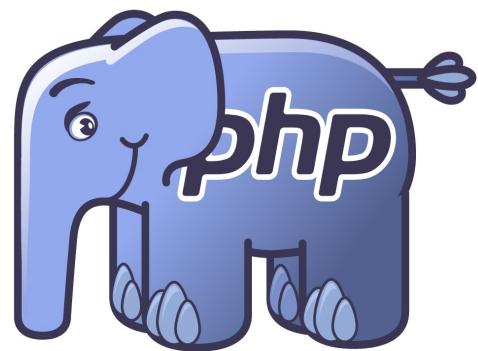


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIÁS
Câmpus Luziânia

Orientação a Objetos



Agenda

- Programação estruturada
- Orientação a objetos
 - Classe
 - Objeto
 - Construtores e destrutores
 - Herança
 - Polimorfismo
 - Abstração
 - Encapsulamento

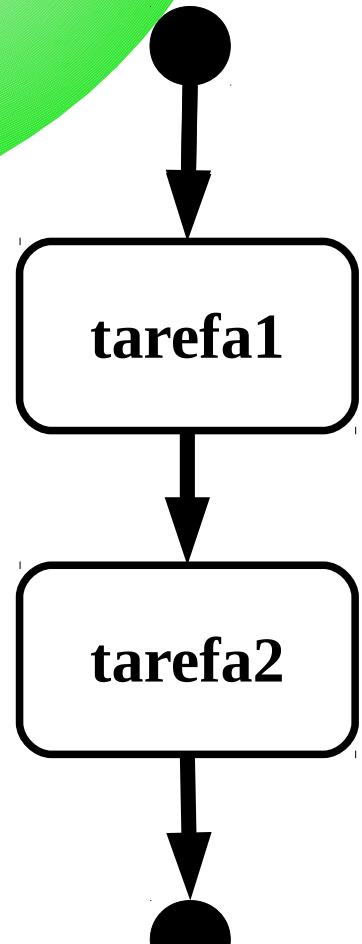
Programação estruturada

- É um paradigma de programação;
- É baseada na modularização;
- O programa é dividido em unidades menores chamadas procedimentos ou funções;
- Essas unidades menores desempenham tarefas bem específicas e podem ser executadas várias vezes;
- É possível executar as funções sob diferentes circunstâncias.

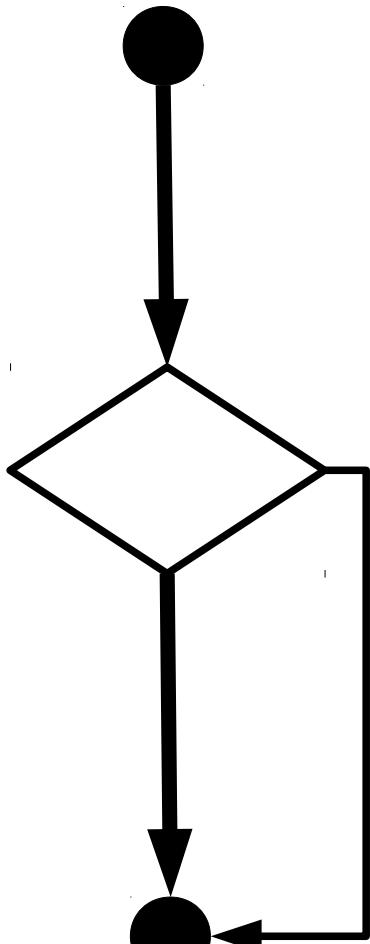
Programação estruturada

- É um paradigma de programação;
- É baseada na modularização;
- O programa é dividido em unidades menores chamadas procedimentos ou funções;
- Essas unidades menores desempenham tarefas bem específicas e podem ser executadas várias vezes;
- É possível executar as funções sob diferentes circunstâncias.

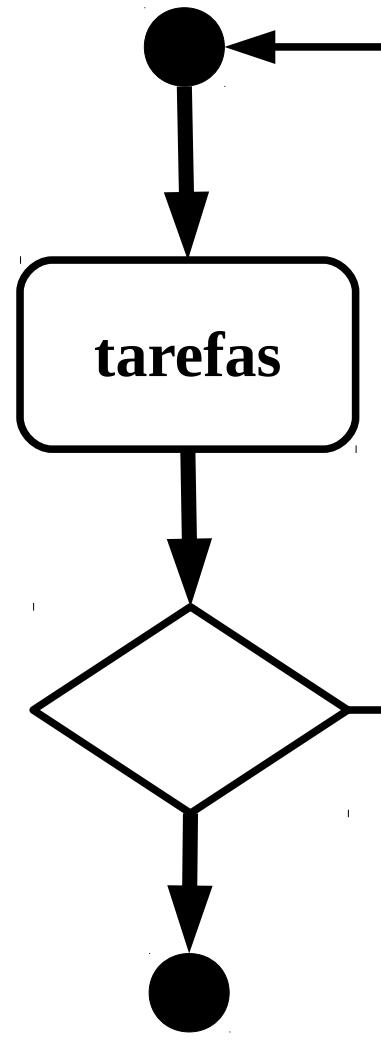
Programação estruturada



SEQUÊNCIA



DECISÃO



ITERAÇÃO

Programação estruturada

- A sequência representa os passos para executar um programa em função de suas tarefas desempenhadas;
- A decisão permite selecionar um determinado fluxo de processamento baseado em expressões lógicas (SE, SENÃO);
- A iteração permite a execução repetitiva de um determinado bloco de comandos do programa.

Orientação a Objetos

- É fundamental entender o conceito de classes e objetos;
- Classe é uma estrutura que define um tipo de dados, podendo conter atributos (variáveis) e também funções (métodos) para manipular estes atributos.

Orientação a Objetos



A screenshot of a code editor window titled "Produto1.class.php". The code defines a class named "Produto" with four public properties: \$Codigo, \$Descricao, \$Preco, and \$Quantidade. The code is written in PHP syntax, starting with a PHP tag, followed by the class definition, and ending with a closing tag.

```
<?php  
class Produto  
{  
    public $Codigo;  
    public $Descricao;  
    public $Preco;  
    public $Quantidade;  
}  
?>
```

Orientação a Objetos

- Um objeto contém exatamente a mesma estrutura e as propriedades de uma classe;
- Mas sua estrutura é dinâmica, seus atributos podem mudar de valor durante a execução do programa;
- Podemos declarar diversos objetos vindos de uma mesma classe.

Orientação a Objetos

```
 Produto1.class.php x  objeto1.php x

<?php
//insere a Classe
include_once 'classes/Produto1.class.php';

//cria um objeto
$produto = new Produto;
|
//atribui valores
$produto->Codigo = 4001;
$produto->Descricao = 'CD - The Best of Eric Clapton';

var_dump($produto);
?>
```

Orientação a Objetos

```
 Produto1.class.php x  objeto1.php x  Produto2.class.php x
<?php
class Produto
{
    public $Codigo;
    public $Descricao;
    public $Preco;
    public $Quantidade;

    function imprimeEtiqueta()
    {
        print 'Código: '.$this->Codigo."<br>\n";
        print 'Descrição: '.$this->Descricao."<br>\n";
    }
}
?>
```

Orientação a Objetos

```
<?php
//insere a Classe
include_once 'classes/Produto1.class.php';

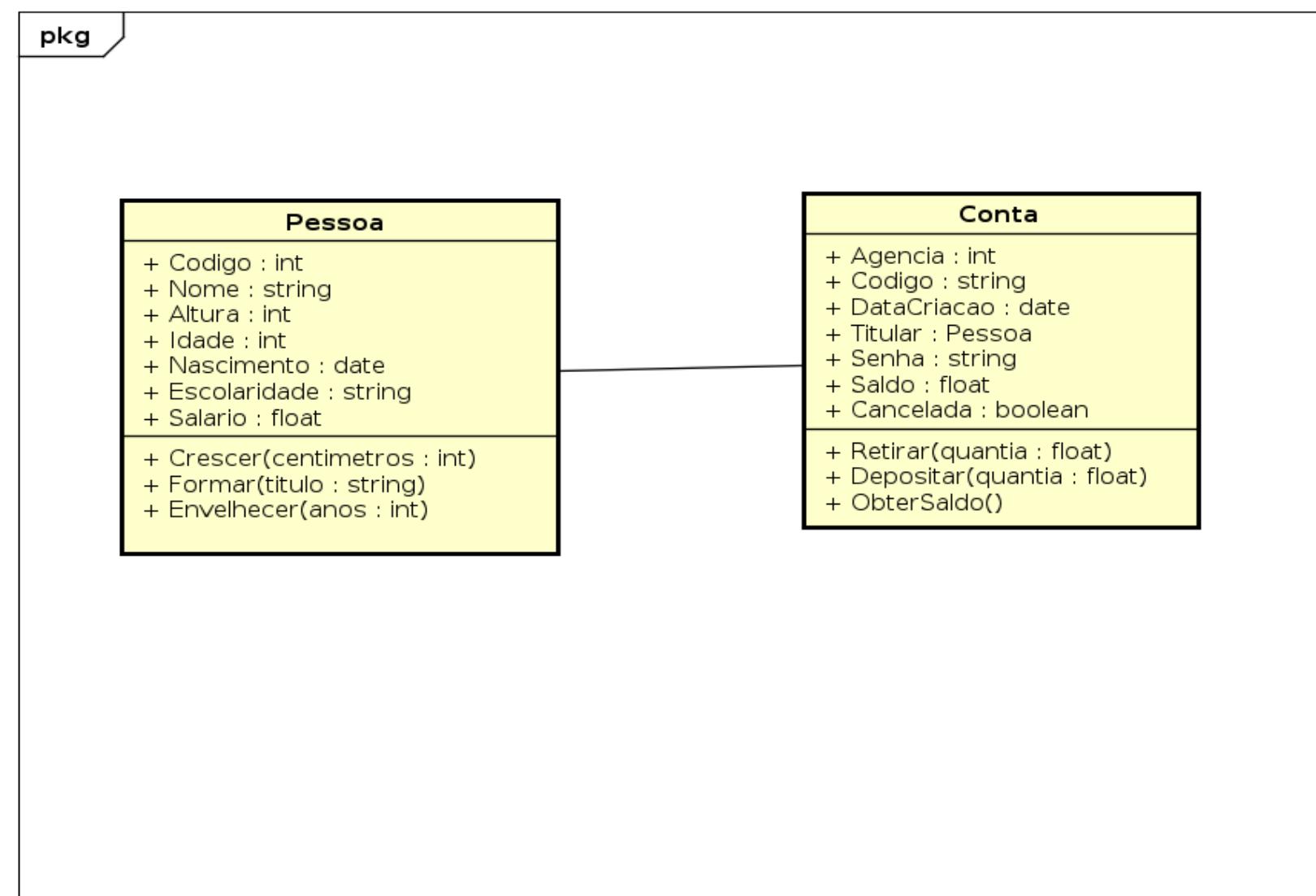
//cria um objeto
$produto1 = new Produto;
$produto2 = new Produto;

//atribui valores
$produto1->Codigo = 4001;
$produto1->Descricao = 'CD - The Best of Eric Clapton';

$produto2->Codigo = 4002;
$produto2->Descricao = 'CD - The Eagles Hotel California';

//imprime informações de etiqueta
$produto1->imprimeEtiqueta();
$produto2->imprimeEtiqueta();
|
?>
```

Classe



Classe

- É uma estrutura estática para descrever objetos mediante atributos (propriedades) e métodos (funcionalidades);
- A classe é um modelo ou template para criação de objetos;
- Podem ser classes: entidades de negócio da aplicação (pessoa, conta, cliente, fornecedor), entidades de interface (janela, botão, painel, frame, barra), etc.

Classe

- Os membros de uma classe são declarados na ordem:
 - Primeiro, as propriedades (usando var);
 - Segundo, os métodos (usando function).

Objeto

- É uma estrutura dinâmica originada com base em uma classe;
- É uma instância de uma classe, pois existe durante um dado instante de tempo – da sua criação até a sua destruição;
- Para instanciar um objeto de uma determinada classe, declaramos uma variável e lhe atribuimos o operador **new** seguido do nome da classe.

Construtores e Destrutores

- Construtor é um método especial para definir o comportamento inicial de um objeto;
- O método construtor é executado automaticamente no momento em que instanciamos um objeto por meio do operador new;
- O método construtor retorna por definição o próprio objeto que está sendo instanciado.

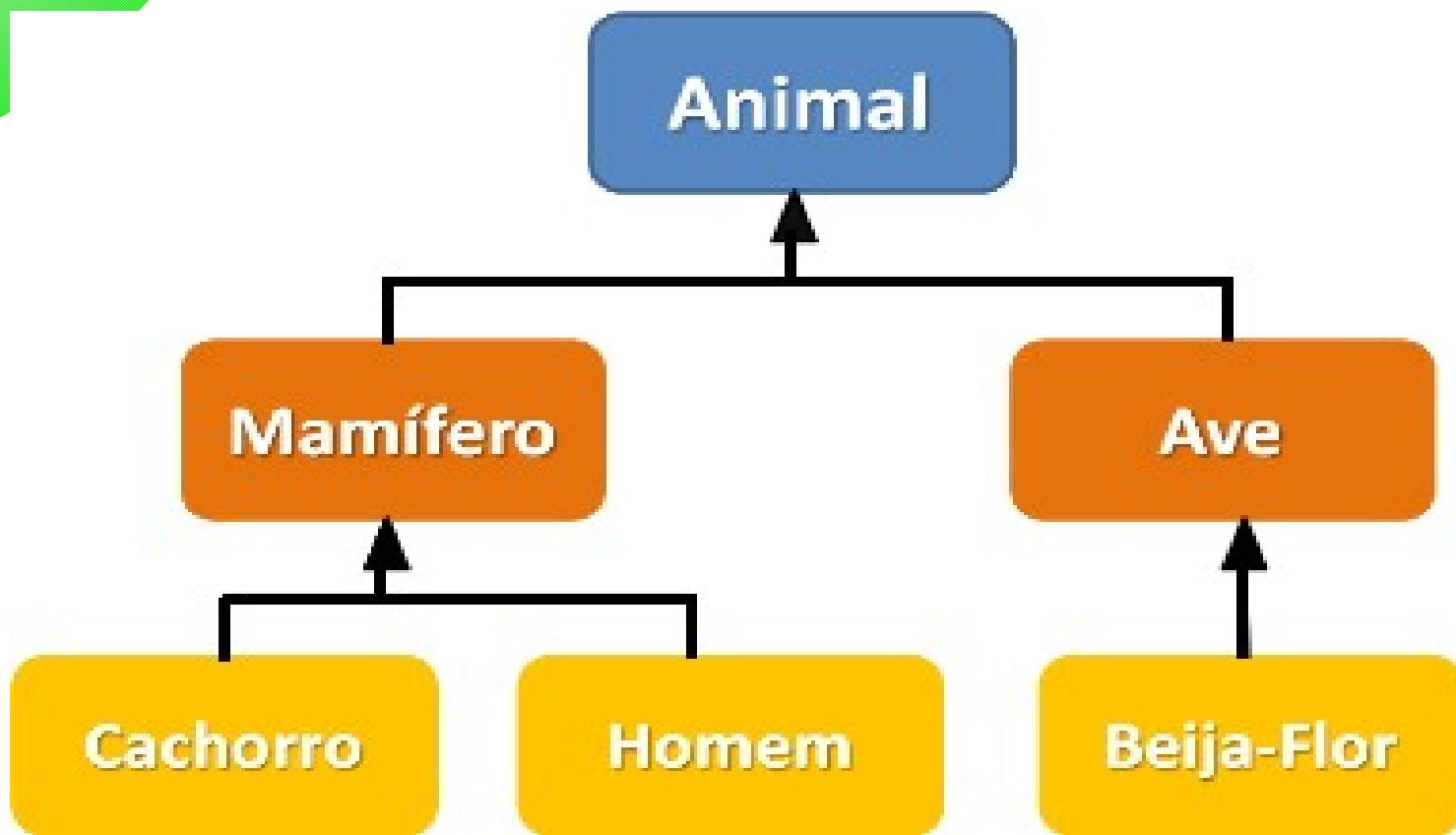
Construtores e Destrutores

- Um destrutor ou finalizador é um método especial executado automaticamente quando:
 - O objeto é deslocado da memória;
 - Quando atribuímos o valor NULL ao objeto;
 - Quando utilizamos a função unset();
 - Quando o programa é finalizado.

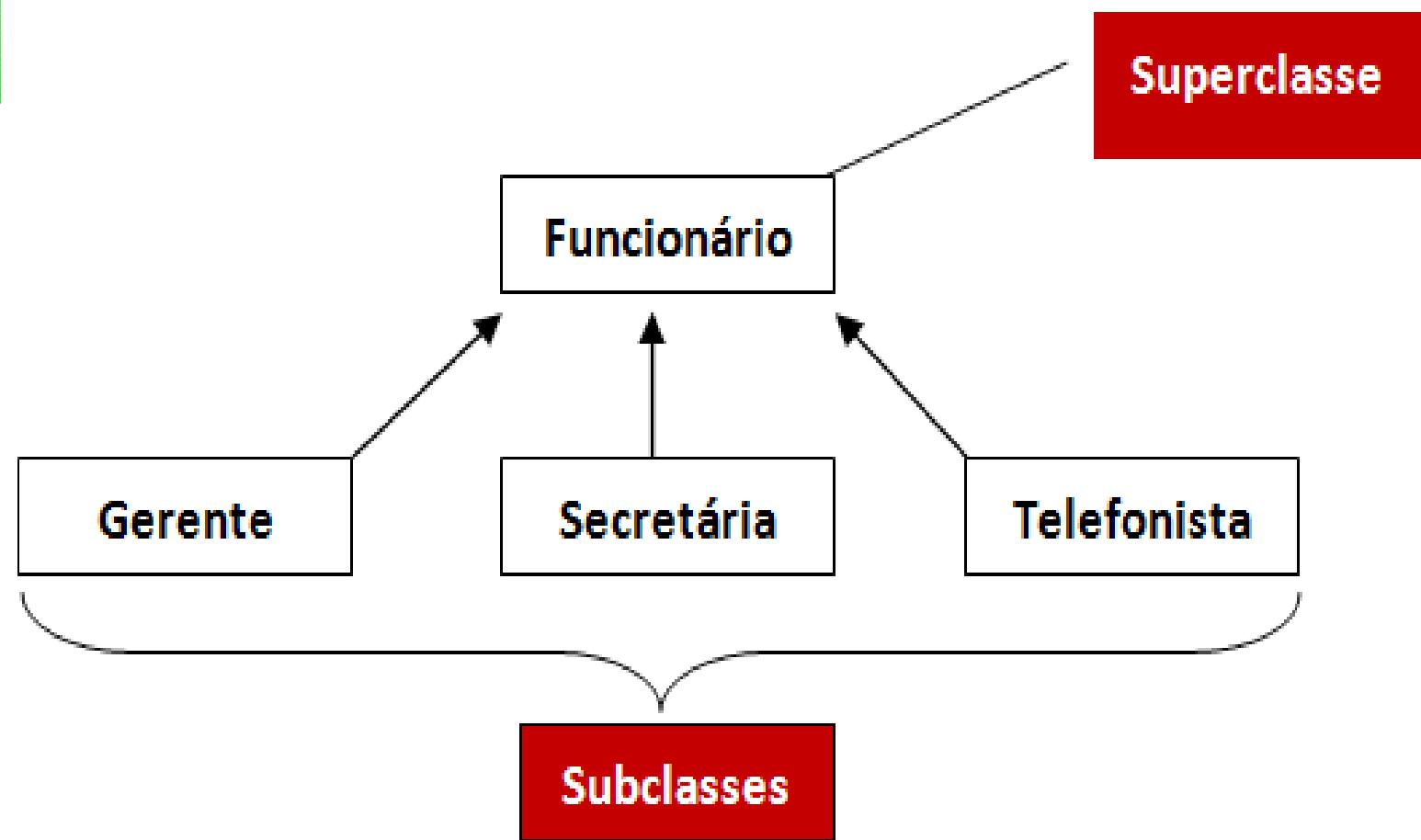
Herança

- Reúso;
- Possibilidade de reutilizar partes do código;
- Elimina duplicações e reescritas de código;
- Compartilhamento de atributos e comportamentos entre as classes de uma mesma hierarquia.

Herança



Herança



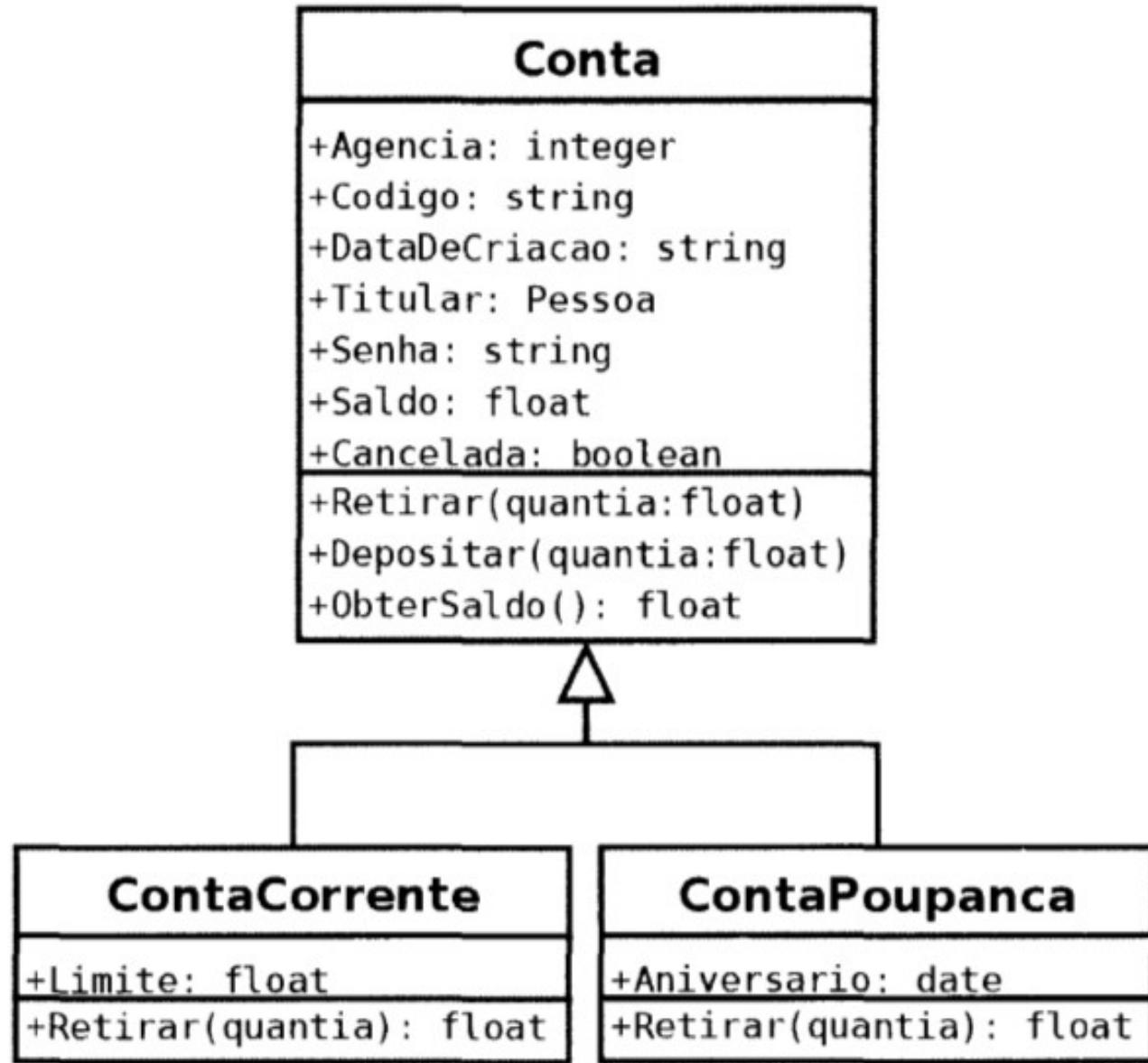
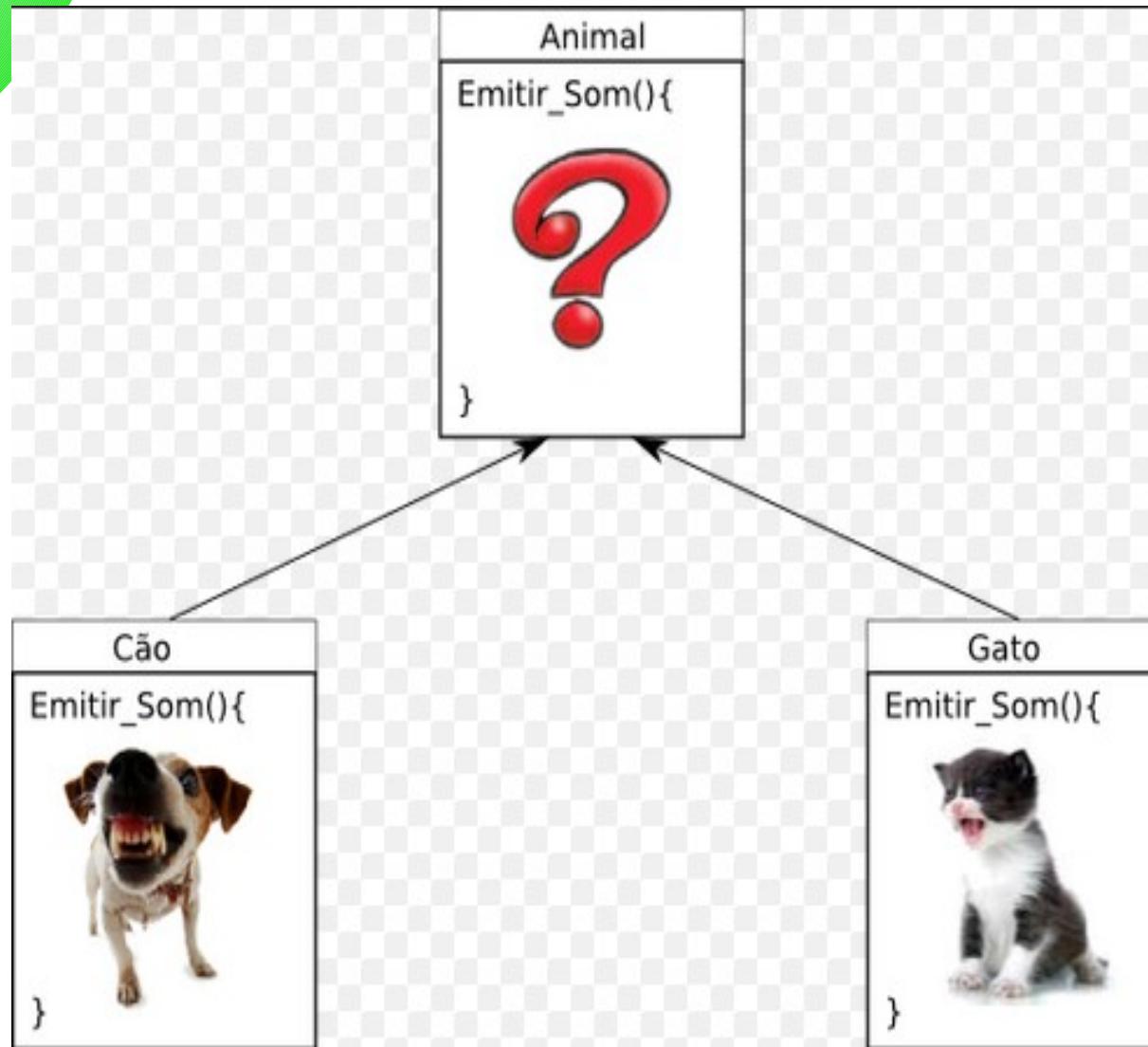


Figura 2.5 – Herança de classes.

Polimorfismo

- A palavra significa 'muitas formas';
- Na O.O. significa que :
 - Classes derivadas de uma Superclasse tenham métodos iguais, mas comportamentos diferentes.

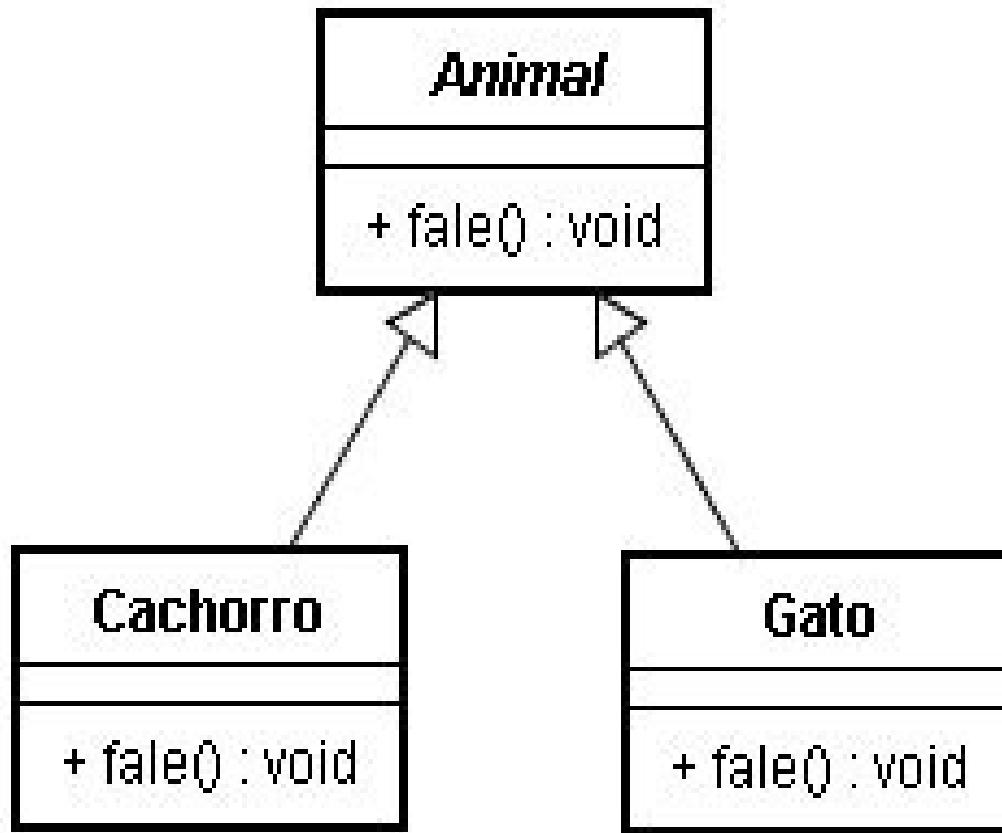
Polimorfismo



Polimorfismo



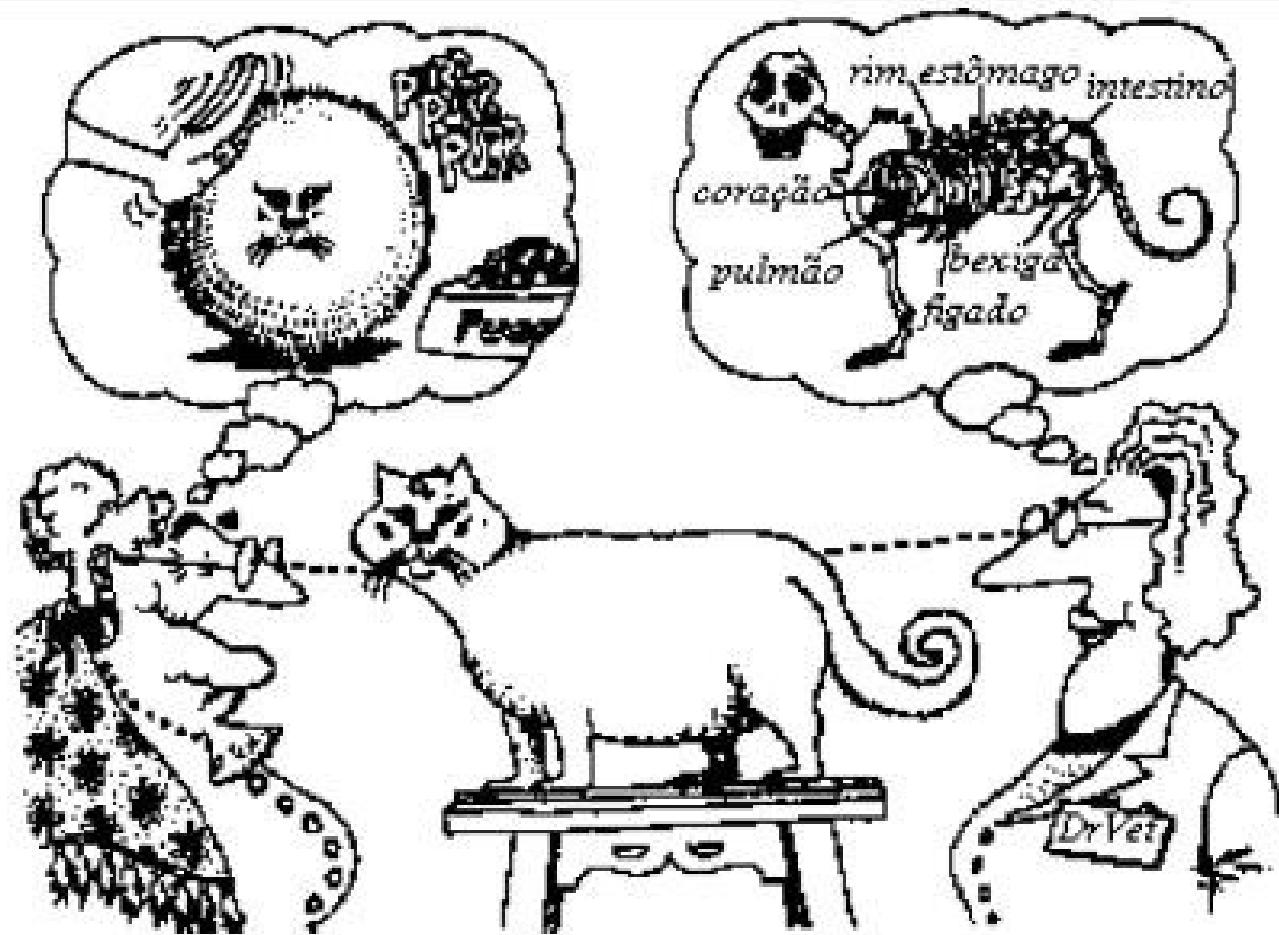
Polimorfismo



Abstração

- Nos primeiros computadores, os programas consistiam em um único bloco, pois a divisão em blocos consumia mais recursos;
- Dividir para conquistar;
- A modularização resolve muitos problemas complexos e facilita o reúso do código;
- Abstração é um conceito fundamental para conseguir uma boa modularização.

Abstração

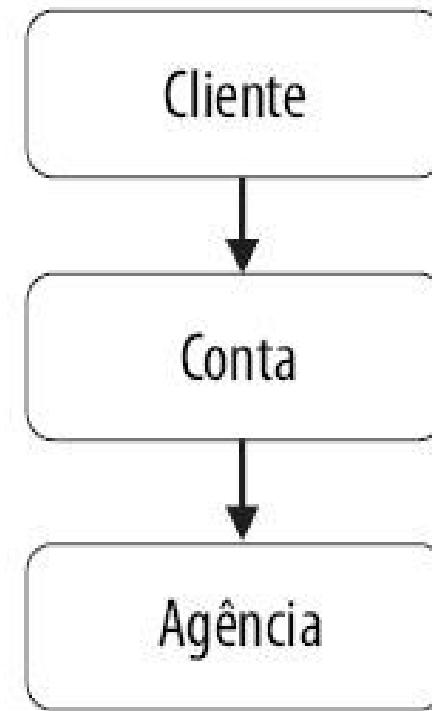


Abstração

Objetos reais



Representação dos objetos



Classes Abstratas

- Servem de base para outras classes;
- São classes que nunca serão instanciadas na forma de objetos, somente suas filhas serão;
- É importante marcar estas classes como sendo classes abstradas na linguagem de programação;
- Assim elas não serão instanciadas durante a execução do programa.

Classes Finais

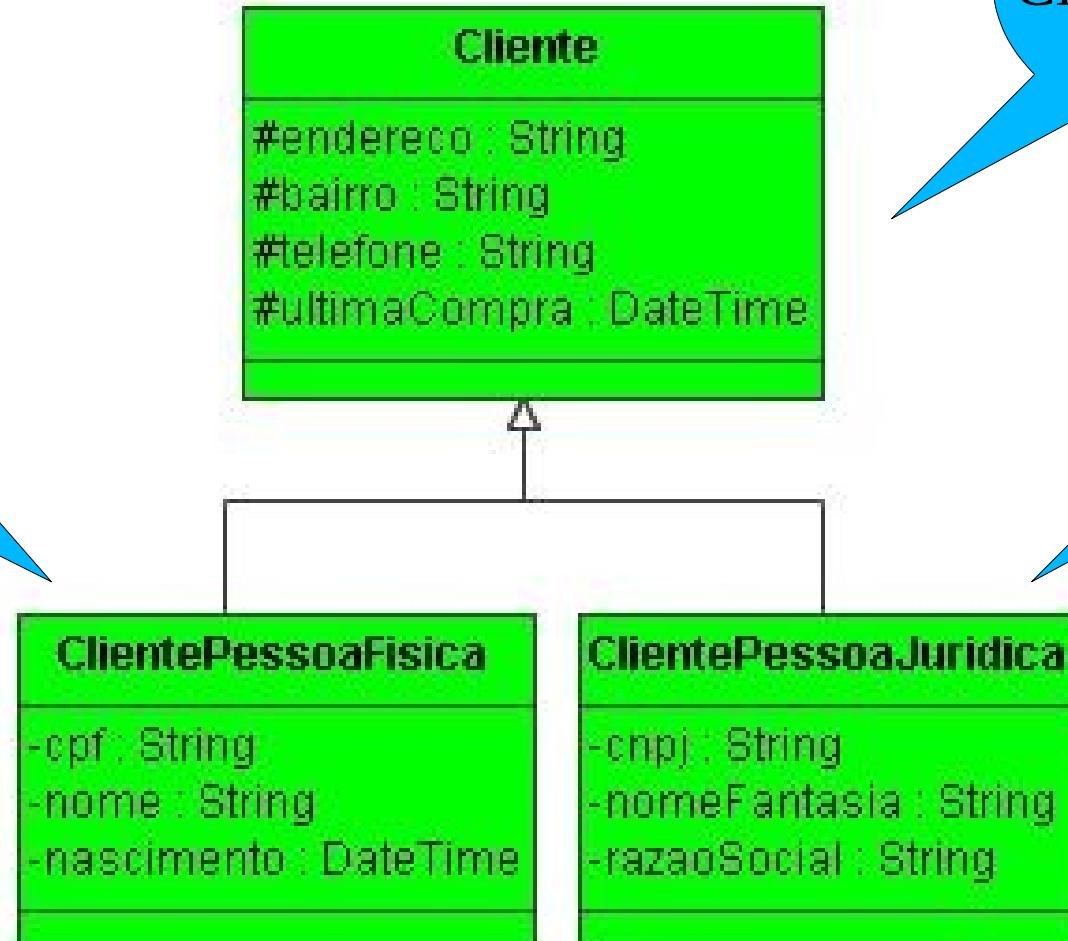
- Ela não pode ser uma superclasse;
- Ela não poderá ser base em uma estrutura de herança;
- Se marcarmos esta classe como FINAL, ela não poderá mais ser especializadas.

Exemplo

Classe Final

Classe Abstrata

Classe Final



Referências

- DALL`OGLIO, Pablo. **PHP : Programando com orientação a objetos, 2^a edição.** São Paulo : Novatec Editora, 2009.