Programação
Avançada

Orientação a Objetos

Chauã Queirolo

Sumário

- Classes
- Objetos
- Herança
- Polimorfismo

Paradigmas de Programação

- Visão do programador em relação aos programas
  - Estruturação
  - Execução

- Principais paradigmas:
  - Declarativo
  - Funcional
  - Imperialista
  - Orientado a Objetos

Paradigmas de Programação

- **Paradigma Declarativo**
  - Baseado em axiomas e lógica de predicados
  - Foco na descrição do problema
  - Exemplo: Prolog
- **Paradigma Funcional**
  - Baseado em funções
  - Cada função resolve um problema específico
  - Exemplo: Lisp, ML
- **Paradigma Imperialista ou Procedural**
  - Conjunto de instruções executadas sequencialmente
  - Exemplos: Fortran, Cobol, C, Basic

Paradigmas de Programação

- **Paradigma Orientado a Objetos**
  - Descreve o sistema com elementos do mundo real
  - Considera que todas as componentes são objetos
  - Cada objeto possui sua estrutura e desempenha ações específicas
  - Objetos são classificados de acordo com suas características
  - Exemplo: Java, C++, C#, Python

- Vantagens:
  - Abstração
  - Modularização
  - Extensibilidade
  - Reaproveitamento de código

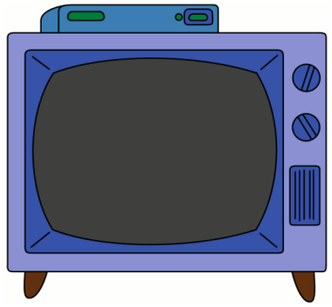
Objetos

- Tudo que está em volta são objetos
    - O universo é formado por objetos
- Cada objeto possui características e desempenha funções
- Exemplos de objetos:

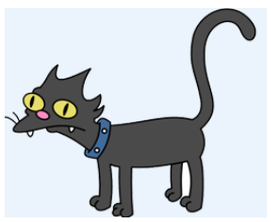
Bart

Homer

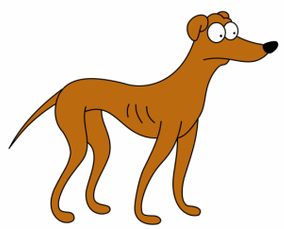
Duff Beer



Televisão



Bola de Neve

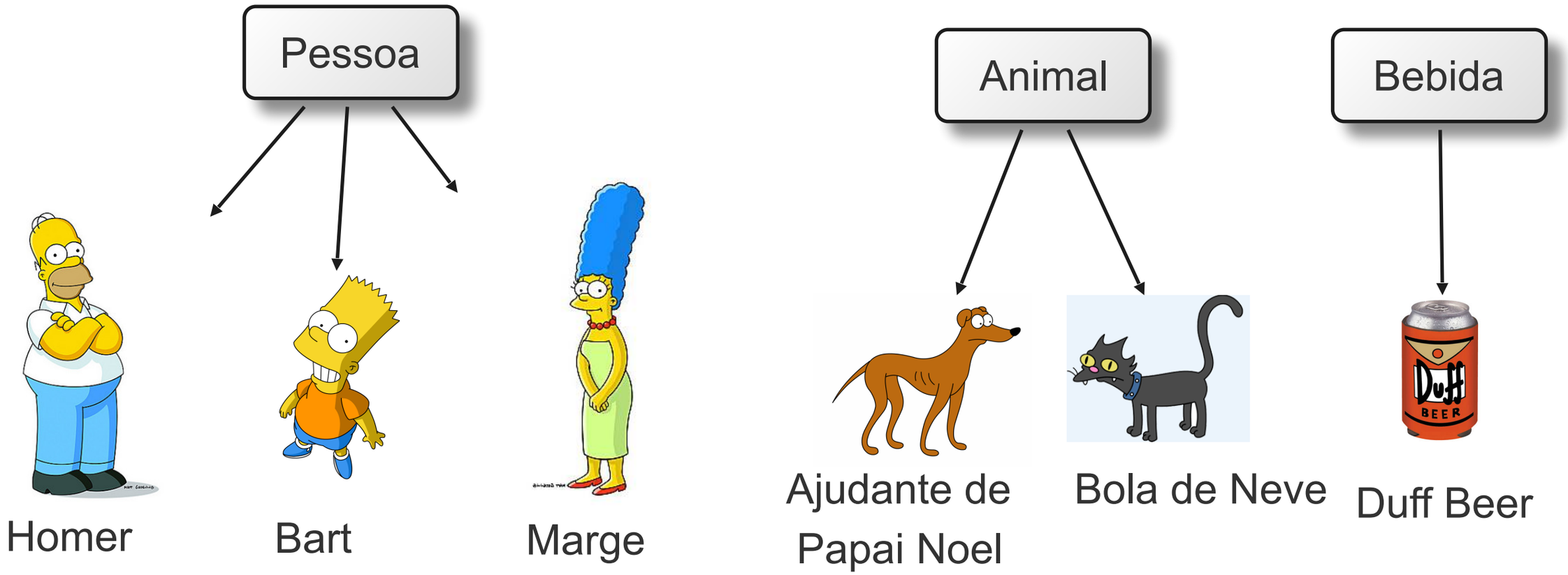


Ajudante de Papai Noel

Marge

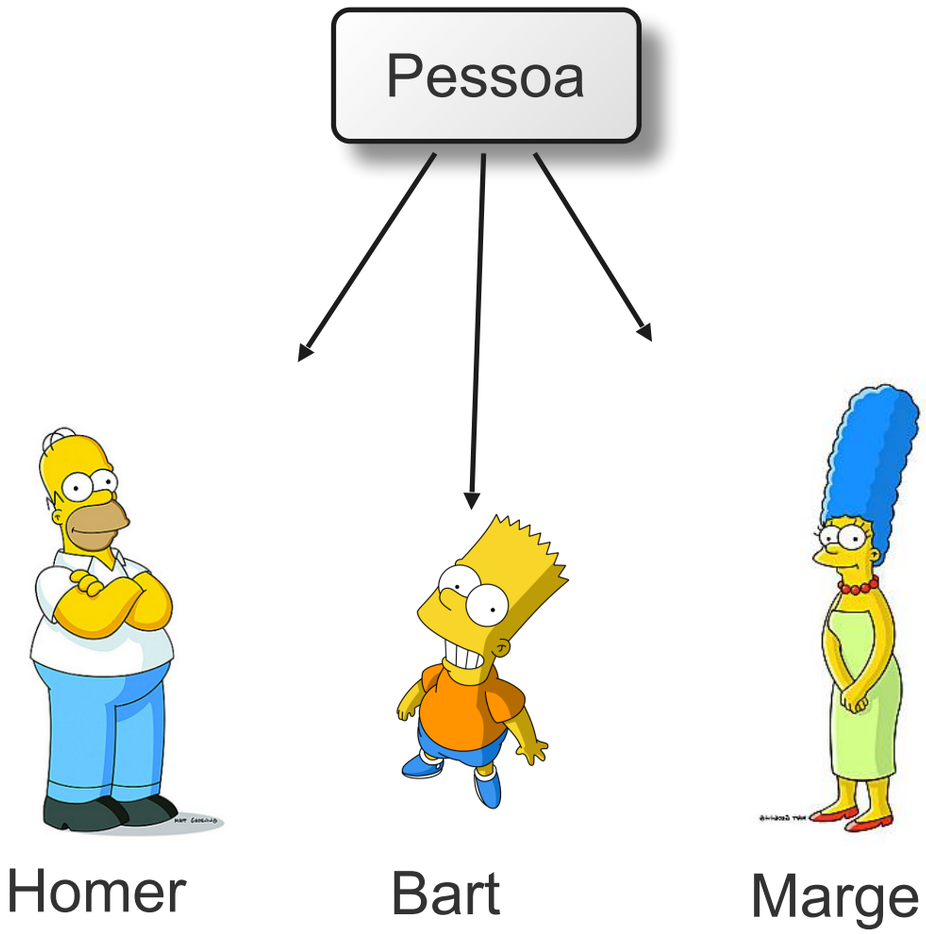
Classes

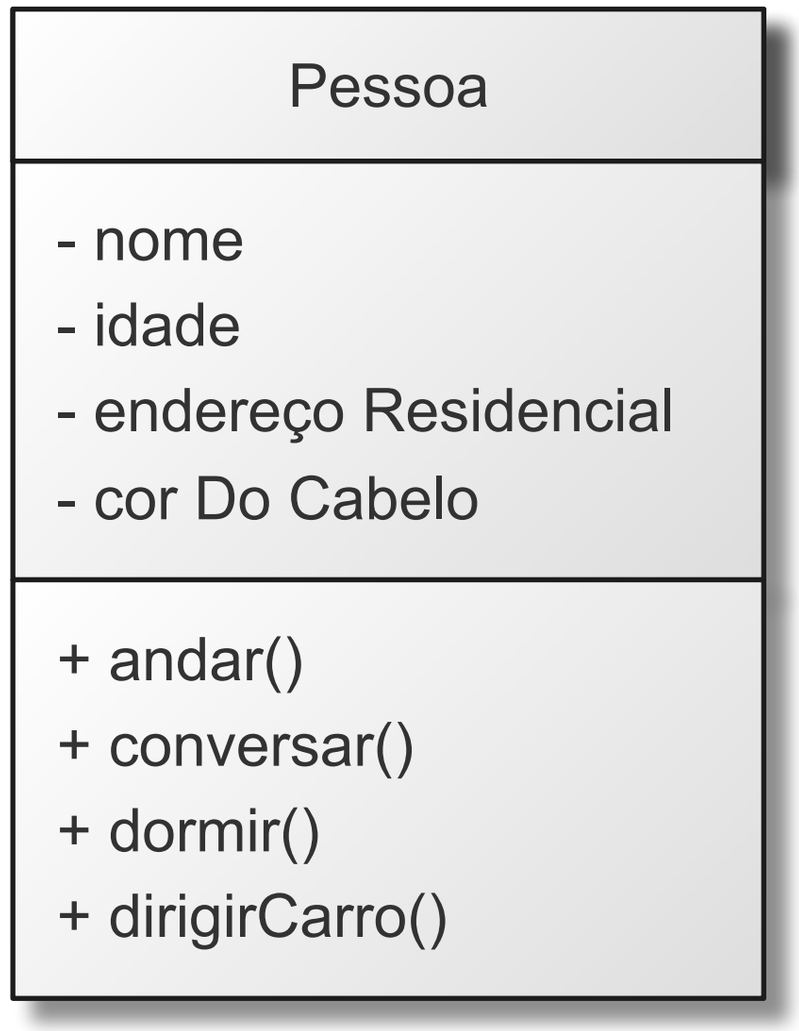
- Classificar objetos com
  - Características semelhantes
  - Funcionalidades semelhantes
- Classe é um agrupamento de objetos semelhantes entre si
- Define uma estrutura



Classes

- **Atributos**: características, propriedades
- **Métodos**: funcionalidades, ações, procedimentos

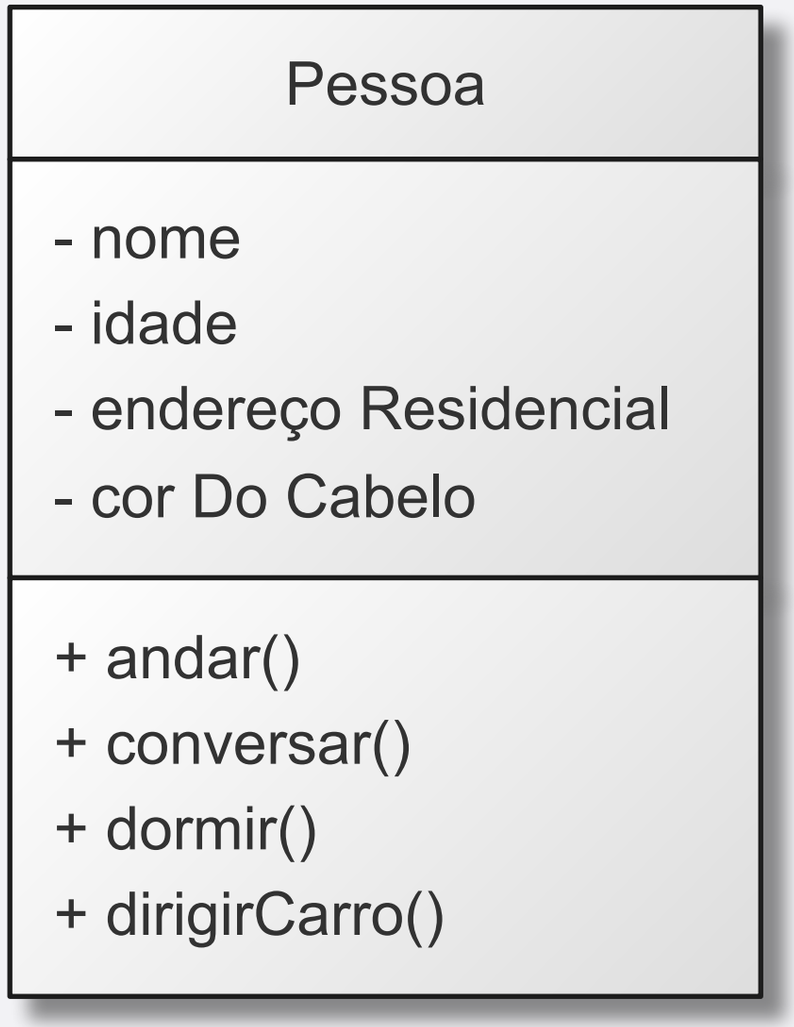




Relacionamento entre classes

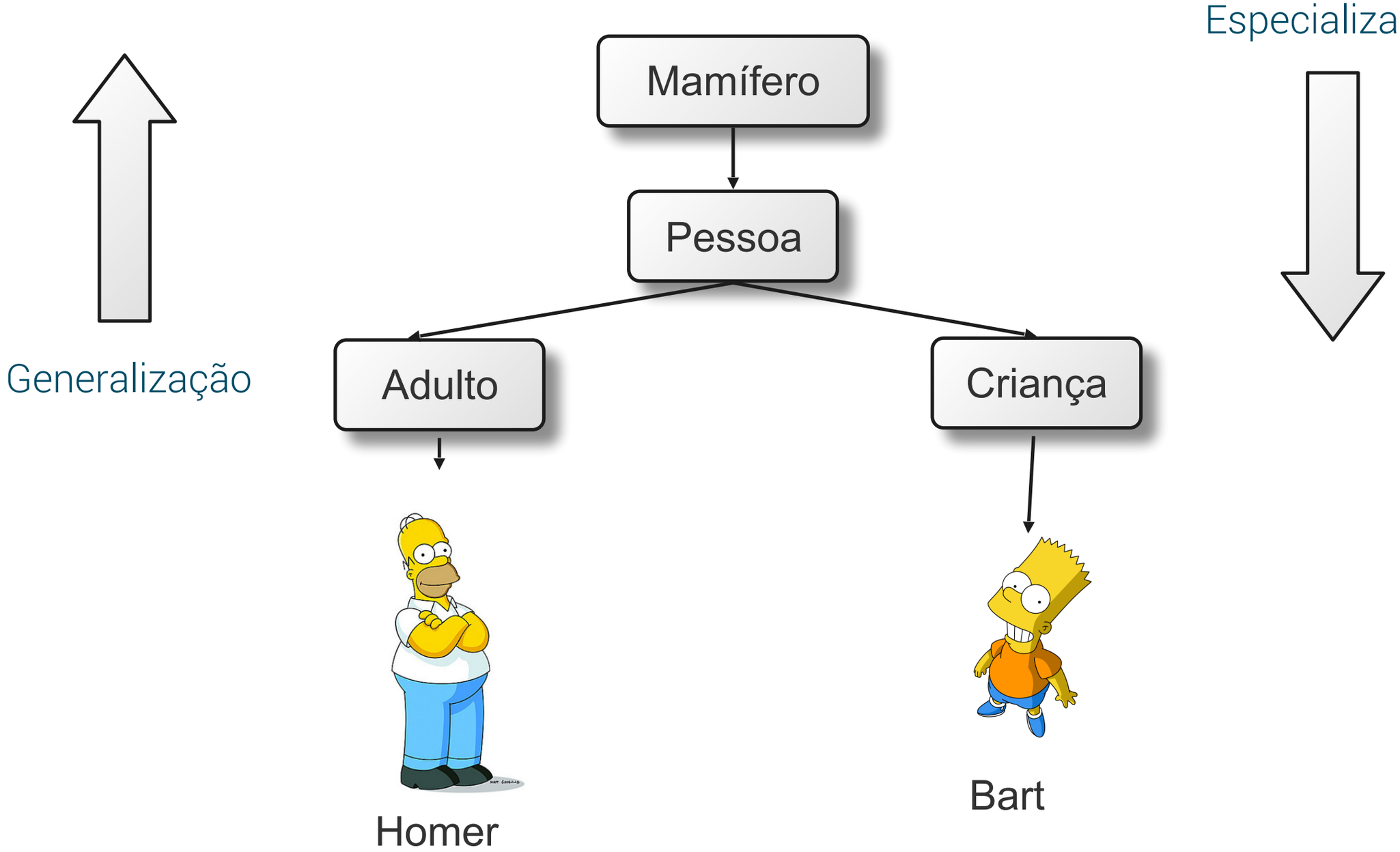
- Associação
- Agregação
- Composição
- Herança

- Exemplo: Um endereço pode ser composto em:
  - Nome da rua
  - Cidade
  - CEP
  - etc...



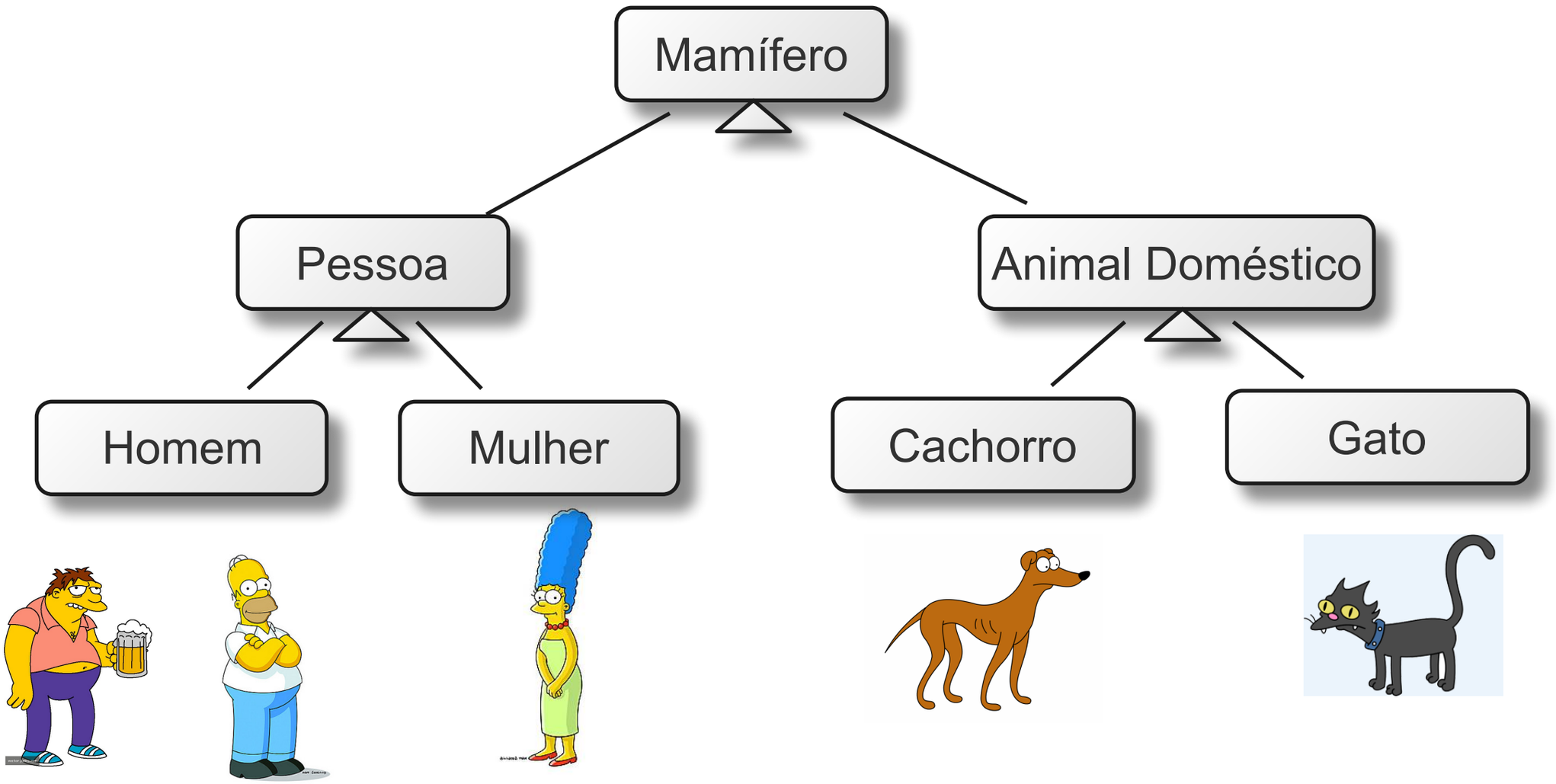
Herança

- Generalização e Especialização



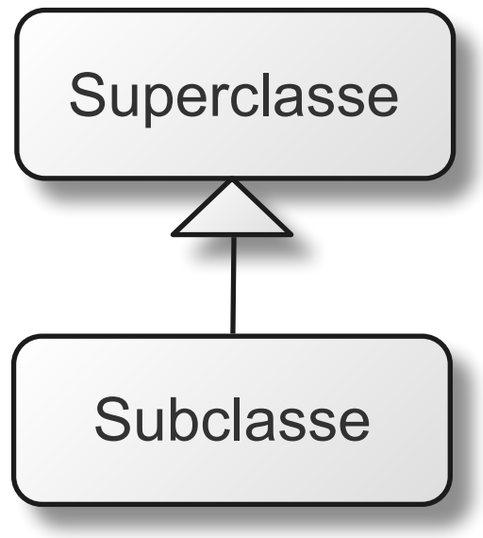
Herança

- Estabelece a condição “é um” entre duas classes



Herança

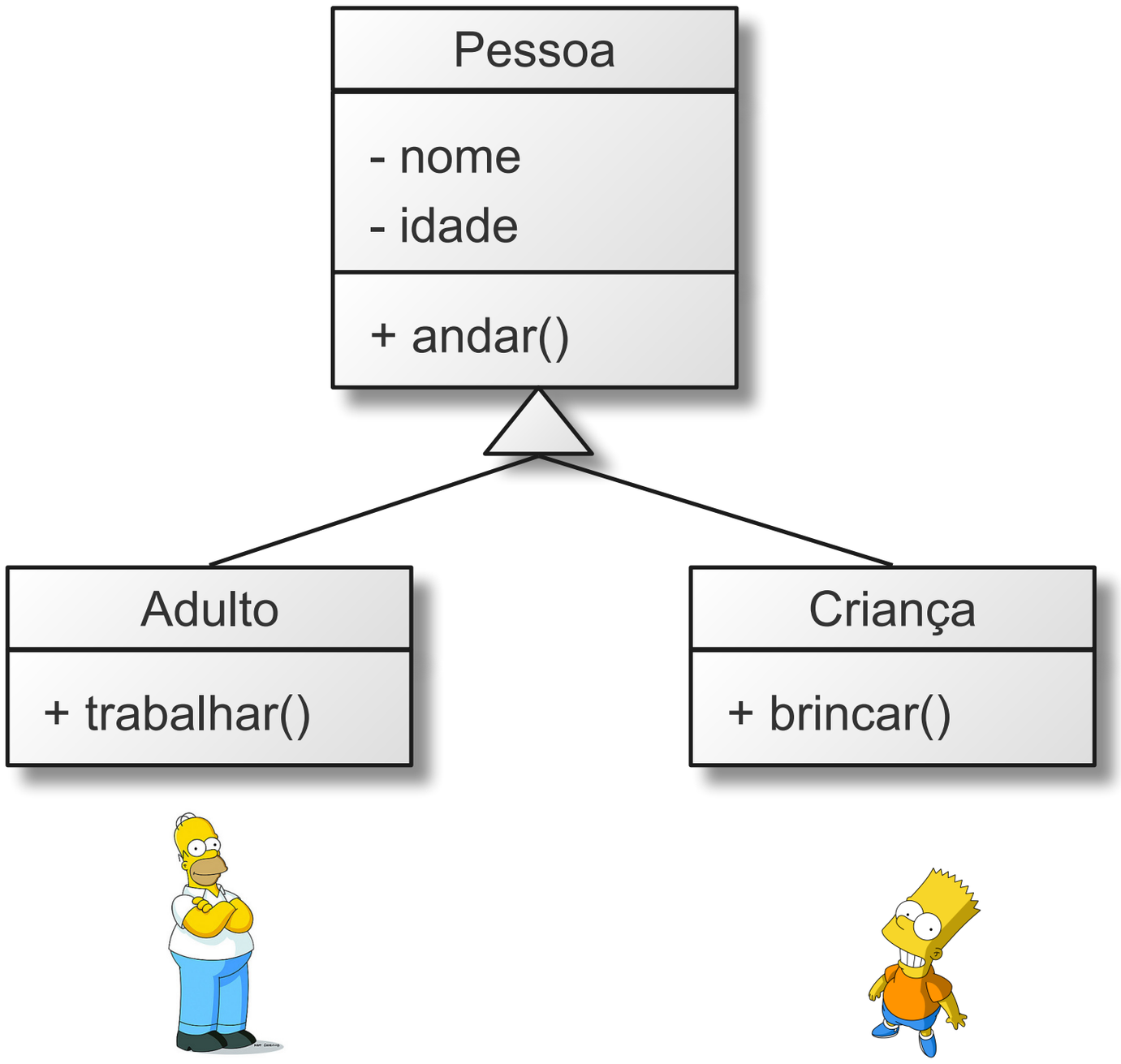
- Superclasse: classe pai, ou classe base
- Subclasse: classe filha, ou derivada



- Todos os atributos e métodos da classe base são herdados pela classe derivada
- Java não permite herança múltipla!

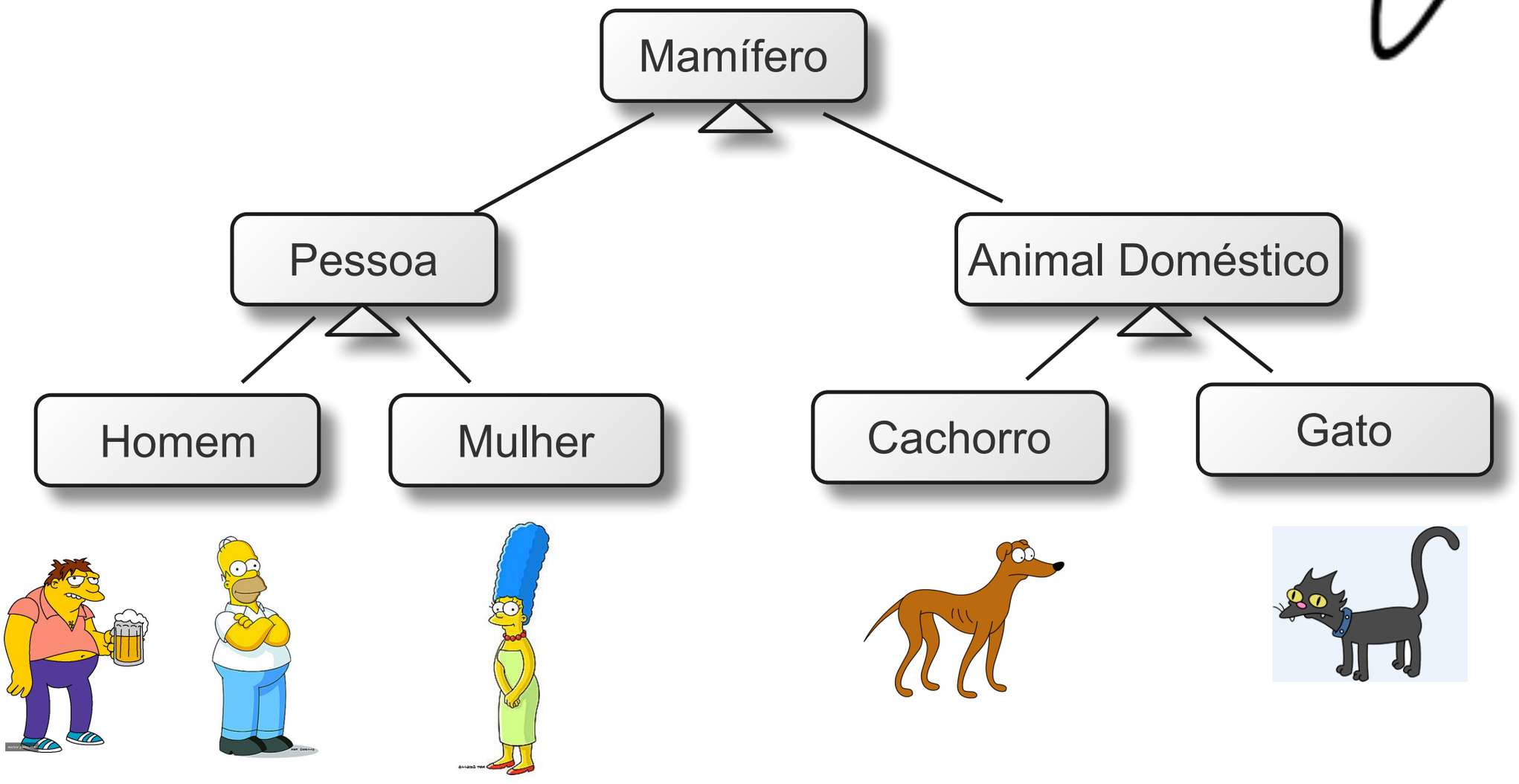
Herança

- Exemplo:



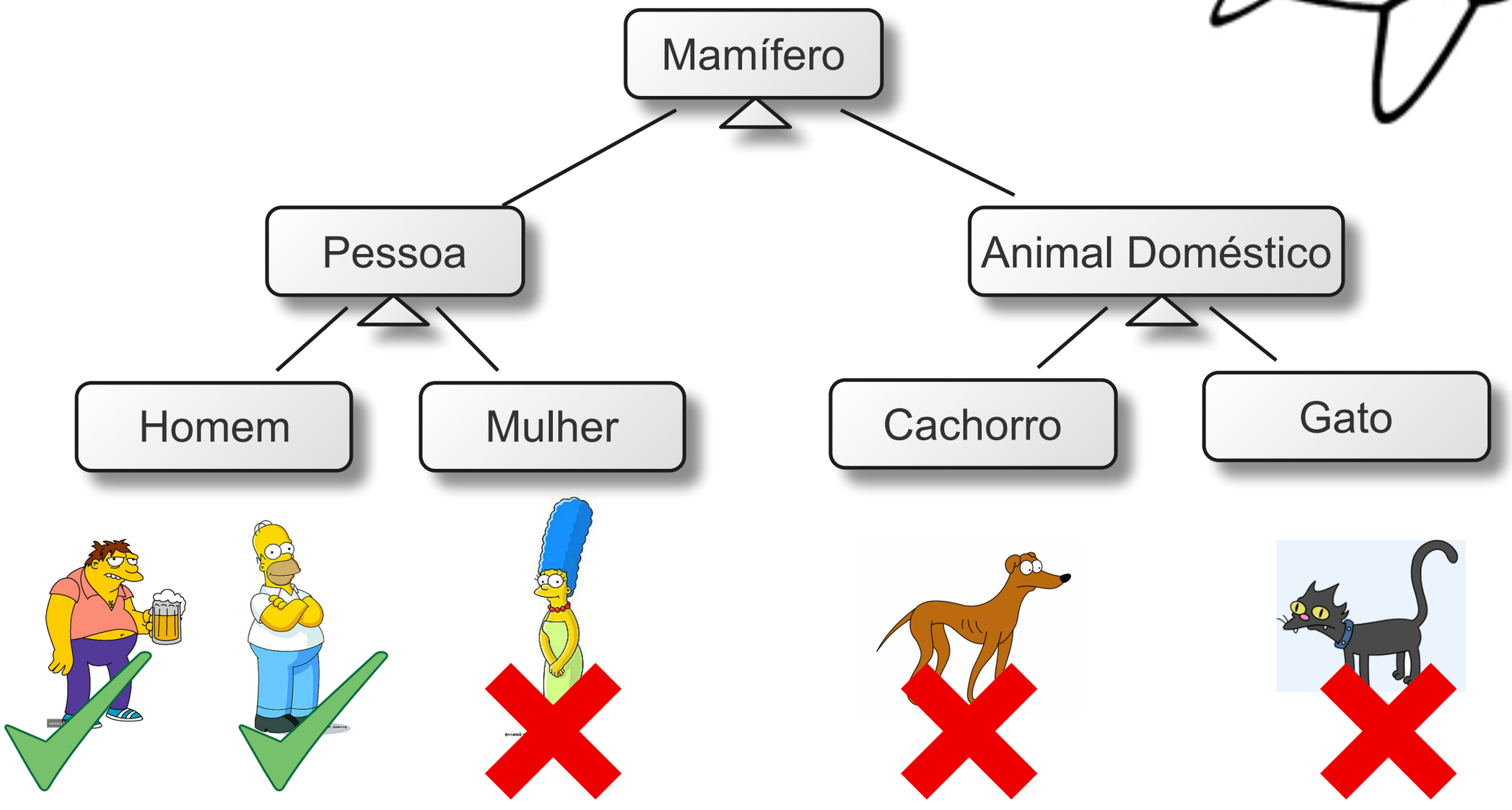
Generalização e Especialização

- Um avião vai sair do aeroporto de viagem
- Existem restrições de quem pode ou não viajar



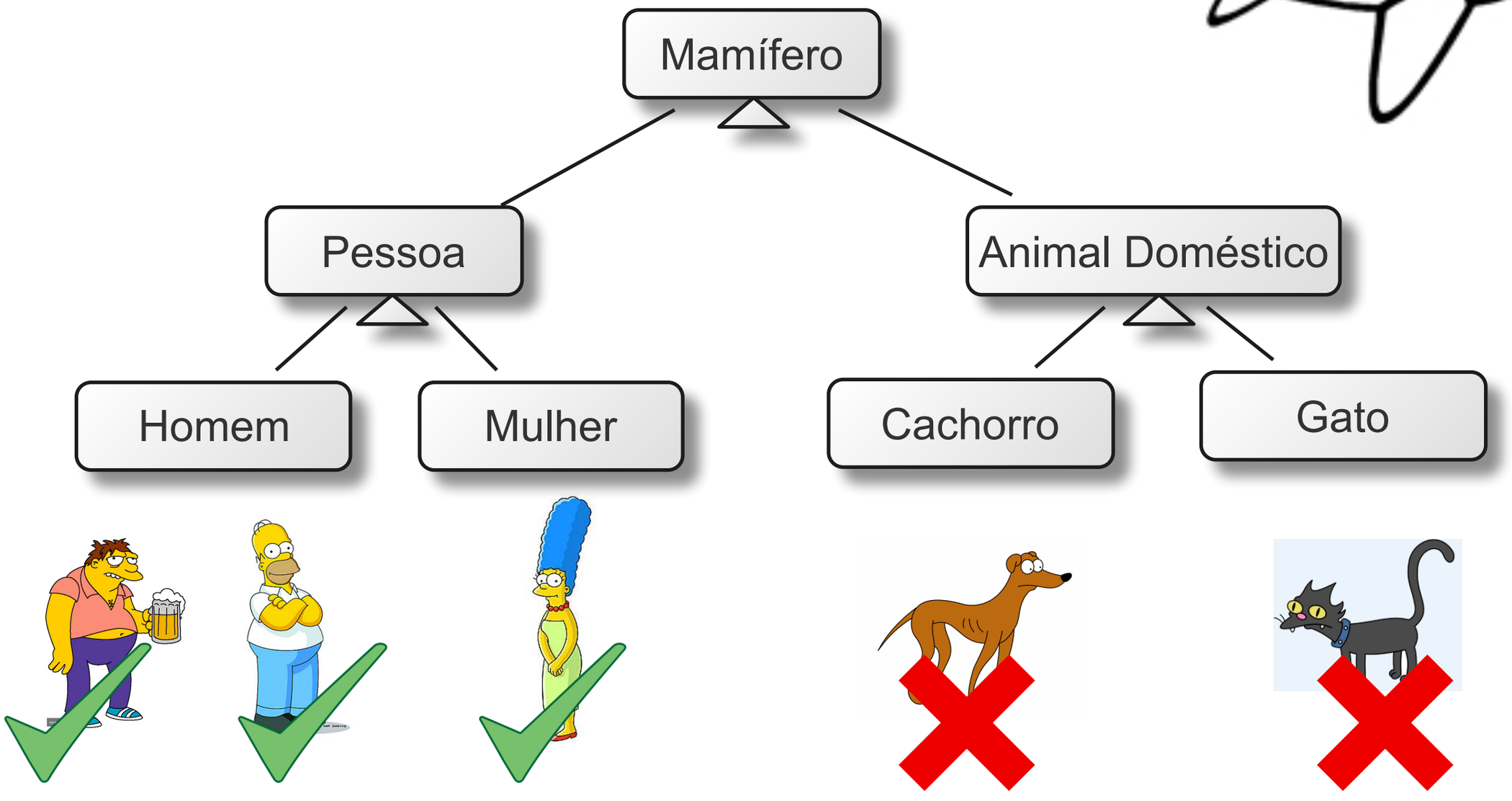
Generalização e Especialização

- Neste avião só podem viajar homens



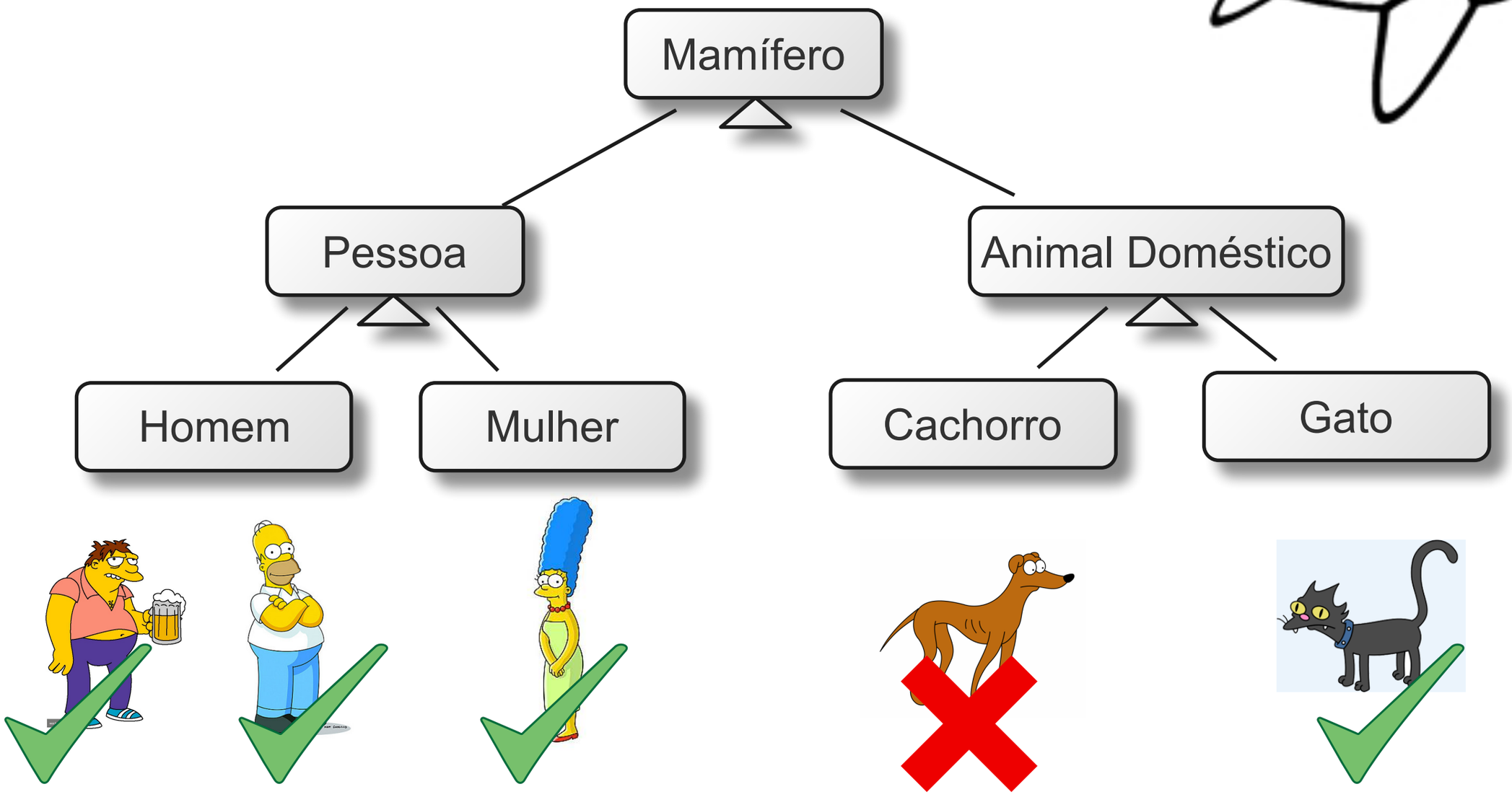
Generalização e Especialização

- Neste avião só podem viajar pessoas



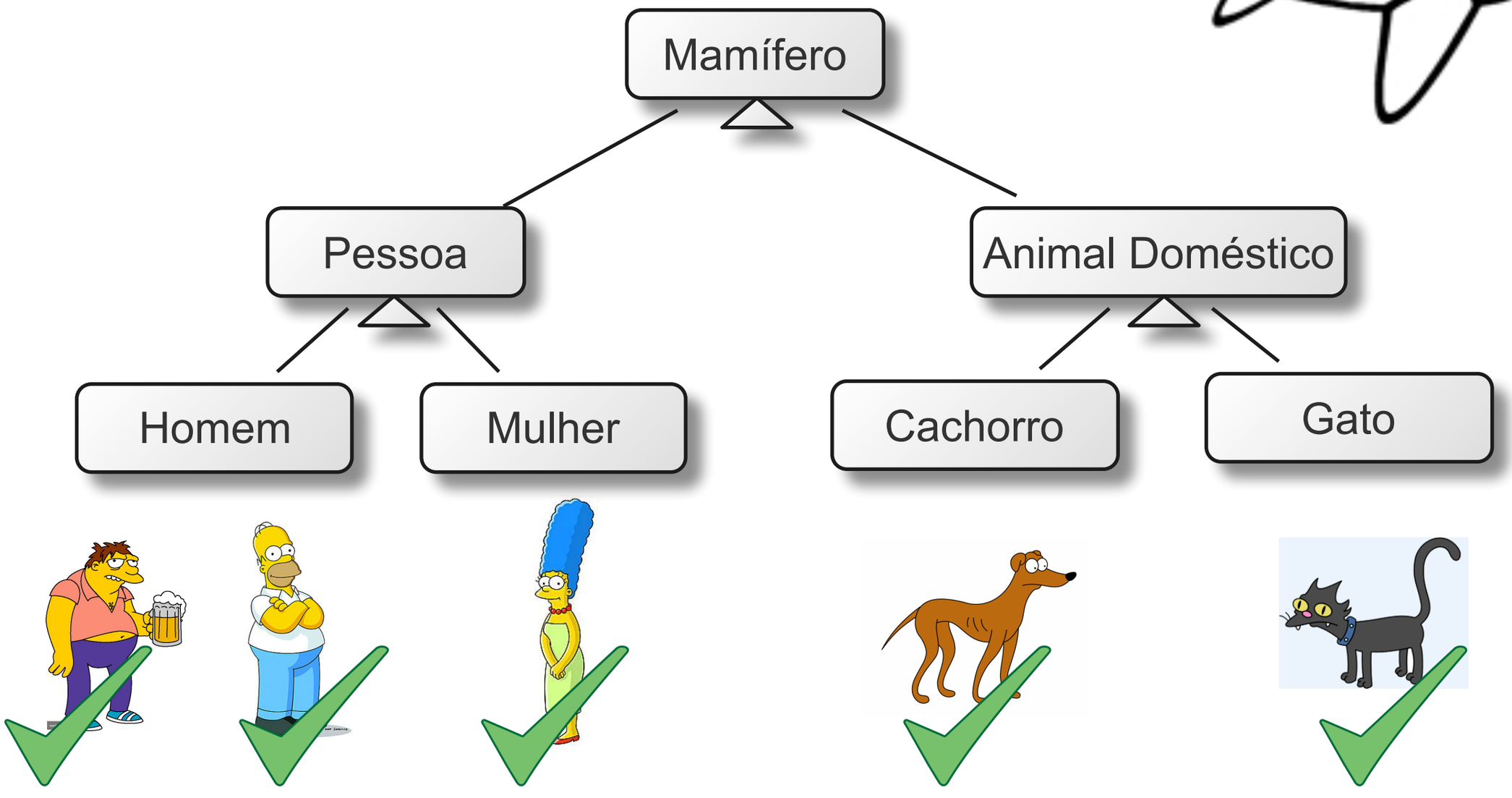
Generalização e Especialização

- Neste avião só podem viajar pessoas e gatos



Generalização e Especialização

- Neste avião só podem viajar mamíferos



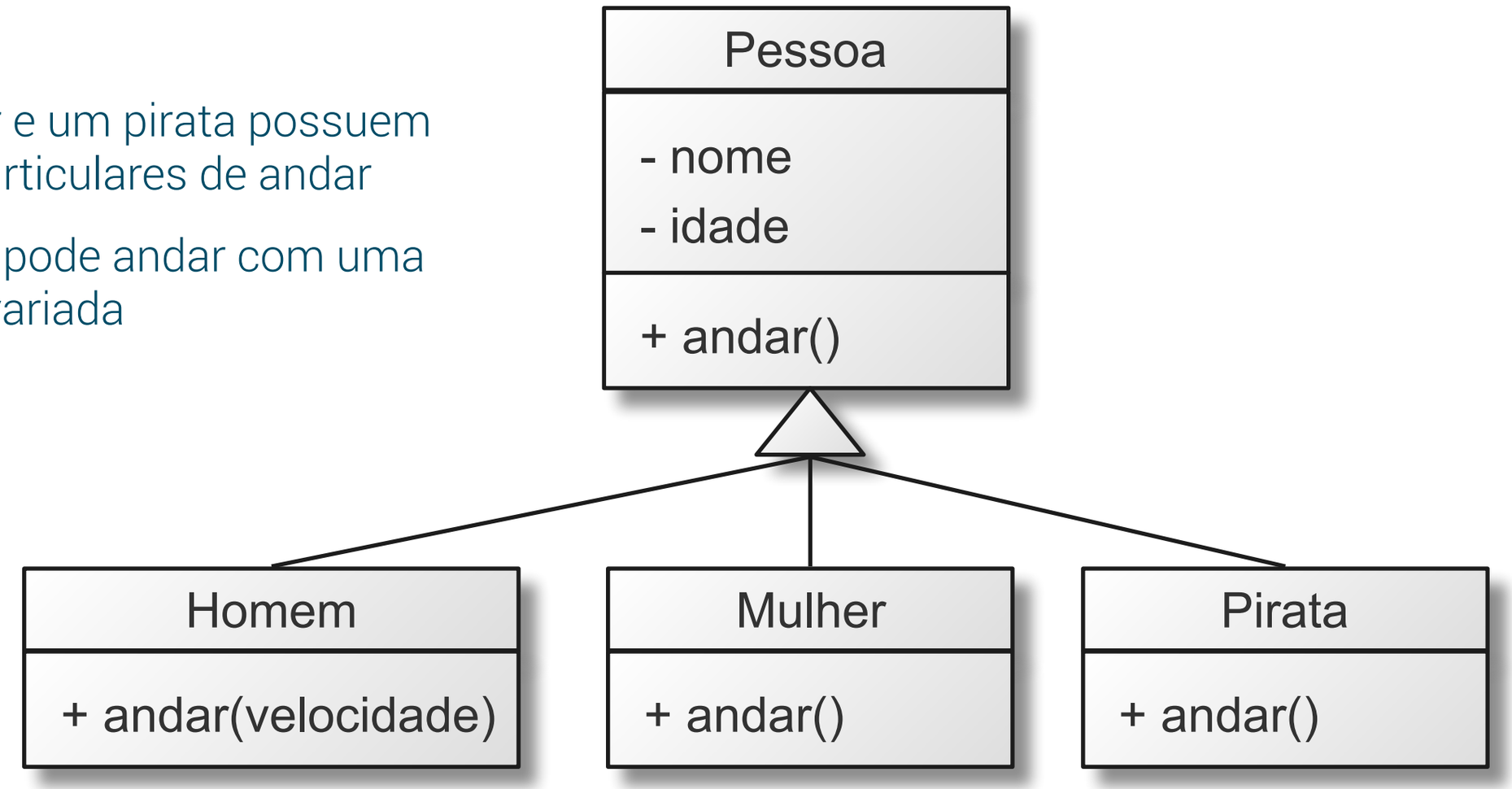
Generalização e Especialização

- Neste avião só podem viajar mamíferos

Polimorfismo

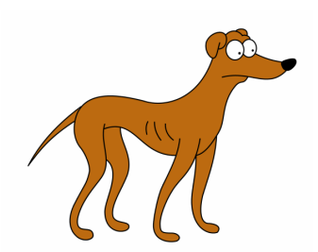
- Métodos com a mesma assinatura
  - Desempenham funções diferentes

- Exemplo:
  - Uma mulher e um pirata possuem maneiras particulares de andar
  - Um homem pode andar com uma velocidade variada



Mensagem

- Maneira com objetos se comunicam
  - Chamada dos métodos de um objeto



```
// Cria e manda mensagens para o
// cachorro
Cachorro ajudante = new Cachorro();
ajudante.latir();

Cachorro cachorro = new Cachorro();
cachorro.latir();

// Um mamífero sabe latir?
Mamifero bidu = new Cachorro();
bidu.latir();
```

Atividades

- Considerar uma classe que descreva uma Bicicleta
  - Quais atributos de uma bicicleta?
  - Quais ações que uma bicicleta desempenha?
  - Quais são três possíveis subclasses?
  - Qual uma possível superclasse?
  - Pode ser definida alguma interface?

Dúvidas