

# INF 112 - Programação II

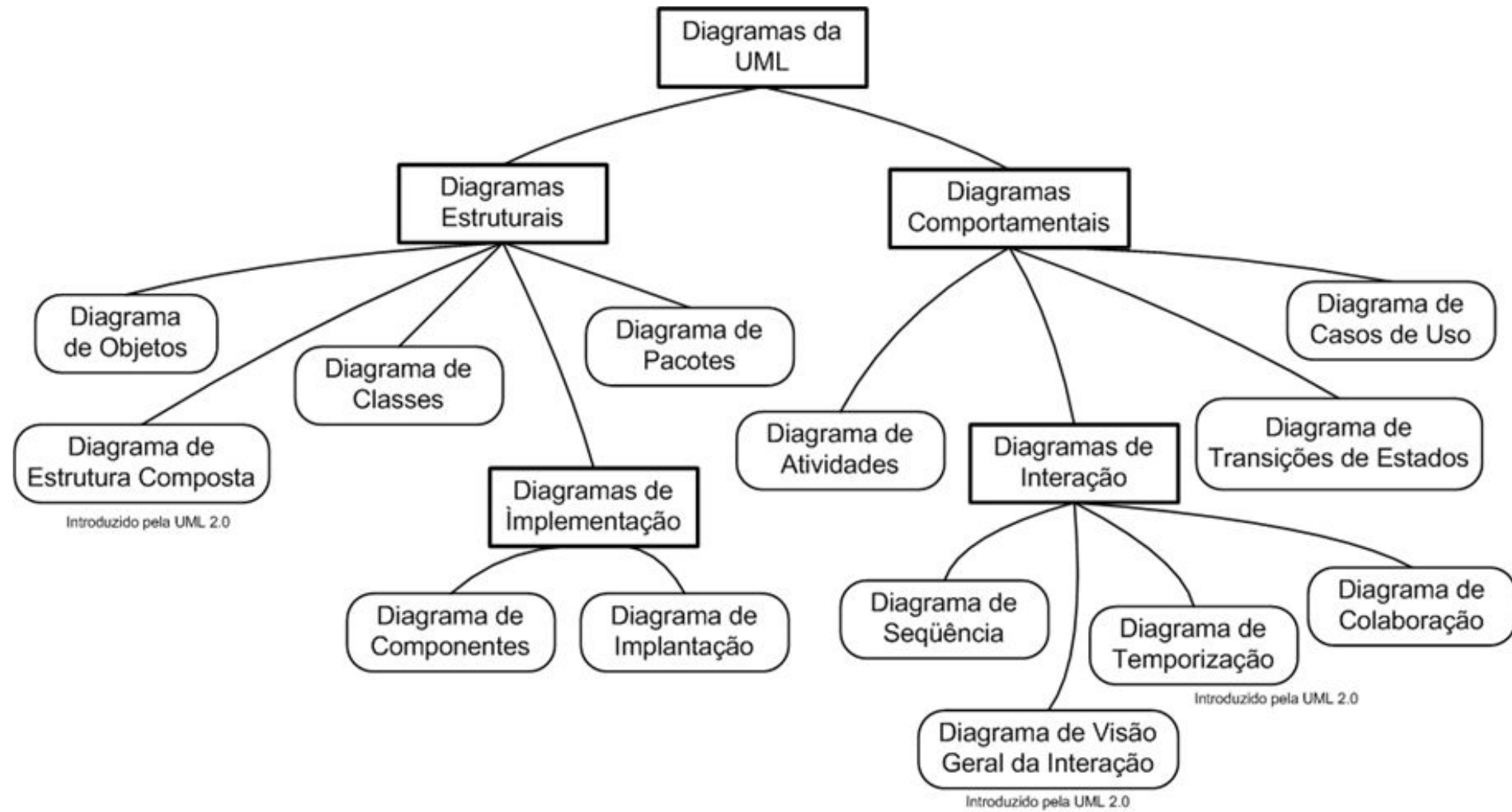
## Introdução a Diagramas de Classes

---

# UML (*Unified Modeling Language*)

- Linguagem-padrão (de modelagem) para representação de elementos de forma padronizada
- Pode ser empregada para:
  - visualização
  - especificação
  - construção e,
  - documentação de artefatos.

# Diagramas UML 2.0



# Diagramas de Classes

- Serve de apoio para a maioria dos outros diagramas
- Define a estrutura das classes do sistema
- Estabelece como as classes se relacionam
- É um dos mais importantes e utilizado diagrama da UML

# Diagramas de Classes

- Permite a visualização das classes que compõem o sistema
- Representa:
  - Atributos e métodos de uma classe
  - Os relacionamentos entre classes
- Apresenta uma visão estática de como as classes estão organizadas
- Preocupação com a estrutura lógica

# Diagramas de Classes

- Elementos Básicos de um Diagrama de Classes

- Classes

- Atributos

- Métodos

- Relacionamentos

- Associação

- Agregação

- Composição

- Herança (Generalização)

- etc

# Como representar as classes?



# Atributos

- Permite a identificação de cada objeto de uma classe
- Os valores dos atributos podem variar de instância para instância
- Atributos devem conter o tipo de dados a ser armazenado
  - Byte, boolean, int, double, char, String, etc.



# Métodos

- São apenas declarados neste diagrama
- Diagrama de Classes não define a implementação
- Outros diagramas permitem modelar o comportamento interno dos métodos (não é o nosso foco aqui!)
  - Diagrama de Sequência
  - Diagrama de Atividades

# Elementos do Diagrama de Classes

- Classes: São representadas graficamente como um retângulo subdividido em três partes distintas:

- Nome da classe

- Atributos

- Métodos (ou operações)

# Elementos do Diagrama de Classes

- Classes: São representadas graficamente como um retângulo subdividido em três partes distintas:

- Nome da classe

- Começa com letra maiúscula, deve estar no singular e estar coerente com o domínio do problema

- Atributos

- Métodos (ou operações)

# Elementos do Diagrama de Classes

- Classes: São representadas graficamente como um retângulo subdividido em três partes distintas:

- Nome da classe

- Atributos

- Representam alguma propriedade do item que está sendo modelado para todos, ou a maioria, dos objetos da classe.

- Métodos (ou operações)

# Elementos do Diagrama de Classes

- Classes: São representadas graficamente como um retângulo subdividido em três partes distintas:

- Nome da classe

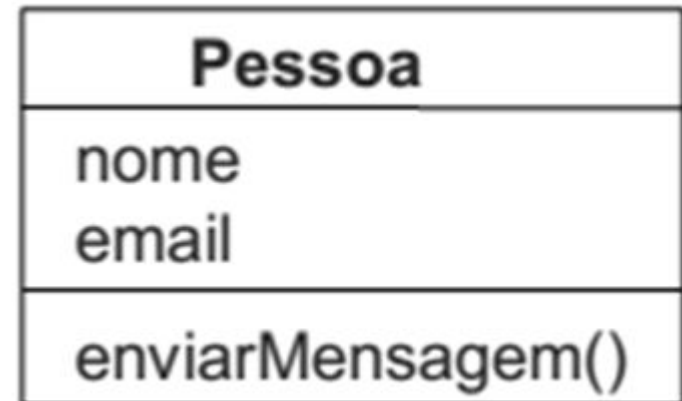
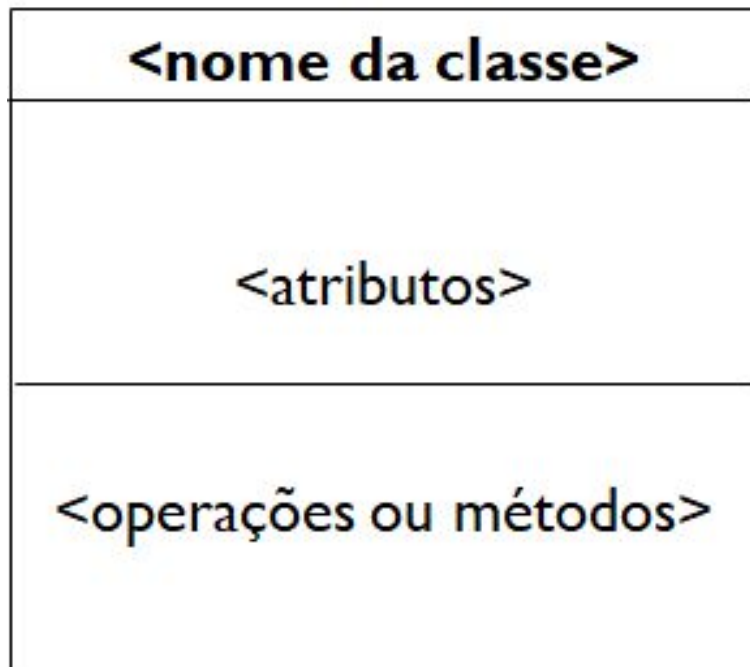
- Atributos

- Métodos (ou operações)

- São abstrações de algo que pode ser feito com um objeto da classe; pode ser um conjunto vazio.

# Elementos do Diagrama de Classes

Exemplo:



# Elementos do Diagrama de Classes

## Exemplo:

Aluno
Matrícula Nome Endereço DataVestibular ...
matricular() trancarMatrícula() formar() cancelarMatrícula() alterarDados()

Imóvel
Identificação Endereço Proprietário DataInícioAluguel DataFimAluguel
cadastrar() alugar() finalizarAluguel() ...

PedidoCompra
Número DataPedido DataAtendimento Cliente
incluir() cancelar() emitir() atualizar()

# Tipos de Visibilidade

- Pública (+)

- o atributo ou método pode ser acessado por qualquer classe;

- Protegida (#)

- o atributo só é passível de acesso pela própria classe ou por uma classe da hierarquia;

- Privada (–)

- o atributo só pode ser acessado pela própria classe.



# Exemplo (Classe ContaBancária)

ContaBancária
---------------

ContaBancária
número
saldo
dataAbertura

ContaBancária
criar()
bloquear()
desbloquear()
creditar()
debitar()

ContaBancária
número
saldo
dataAbertura
criar()
bloquear()
desbloquear()
creditar()
debitar()

ContaBancária
-número : String
-saldo : Quantia
-dataAbertura : Date
+criar()
+bloquear()
+desbloquear()
+creditar(in valor : Quantia)
+debitar(in valor : Quantia)

# Identificando as Classes

## Identificação de Atributos e Associações

Atributos: descrevem propriedades dos objetos de uma classe.

Exemplos:

Livro
titulo
numeroPaginas

Cliente
nome
endereco
email
RG
CPF
dataNascimento

Pedido
numeroPedido
dataPedido
status

Editora
nome
endereco

# Representação de uma Interface

- Uma interface é semelhante a uma classe, “mas não tem atributos”

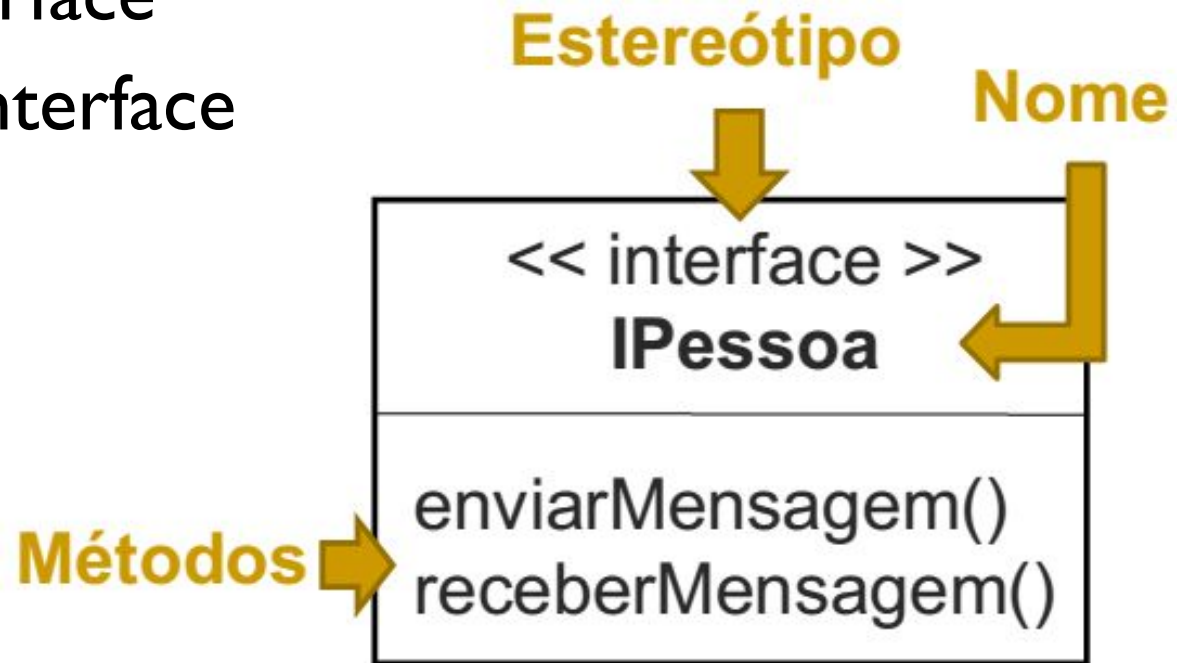
- Uma interface possui:

- Nome da interface

- Métodos da interface

- Estereótipo

- Interface



# Relacionamento

- Classes possuem relacionamentos entre elas (para comunicação)

- Compartilham informações

- Colaboram umas com as outras

- Principais tipos de relacionamentos:

- Associação

- Agregação/Composição

- Herança

# Multiplicidade

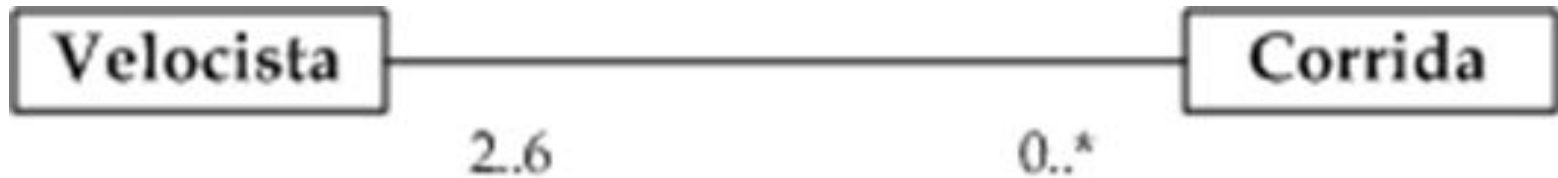
0..1	No máximo um. Indica que os Objetos da classe associada não precisam obrigatoriamente estar relacionados.
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com os objetos da outra classe.
0..*	Muitos. Indica que podem haver muitos objetos da classe envolvidos no relacionamento
1..*	Um ou muitos. Indica que há pelo menos um objeto envolvido no relacionamento.
3..5	Valores específicos.

# Multiplicidade - Exemplos



- Pode haver um cliente que esteja associado a vários pedidos.
- Pode haver um cliente que não esteja associado a pedido algum.
- Um pedido está associado a um, e somente um, cliente.

# Multiplicidade - Exemplos



- Uma corrida está associada a, no mínimo, dois velocistas
- Uma corrida está associada a, no máximo, seis velocistas.
- Um velocista pode estar associado a nenhuma ou várias corridas.

# Associações

- Descreve um vínculo entre duas classes

- Chamado Associação Binária

- Determina que as instâncias de uma classe estão de alguma forma ligadas às instâncias da outra classe



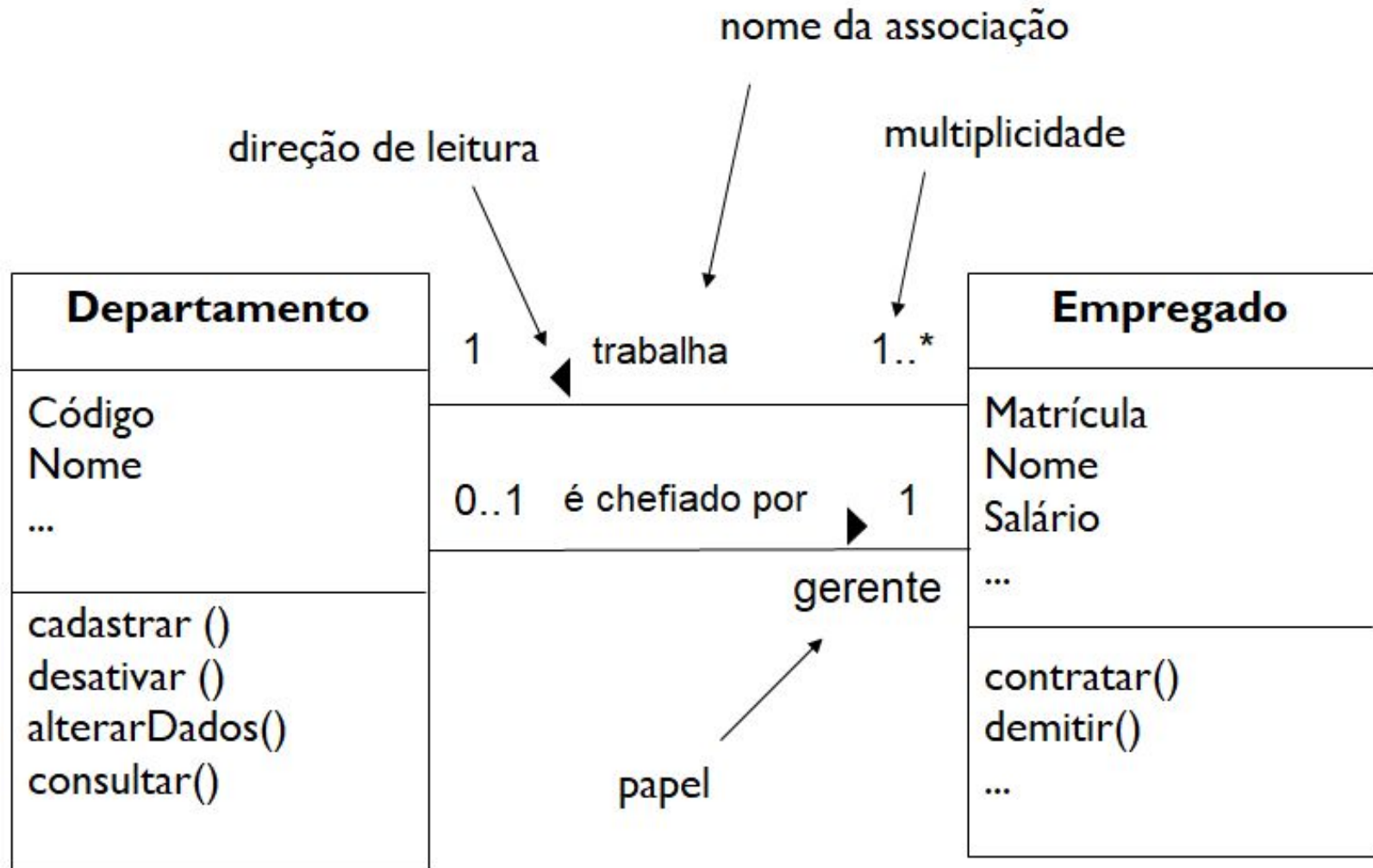
# Notação para Associações

Na UML associações são representadas por uma linha que liga as classes cujos objetos se relacionam.

Exemplos:

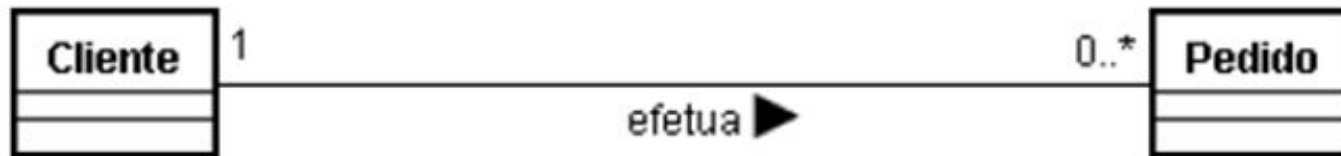


# Notação para Associações

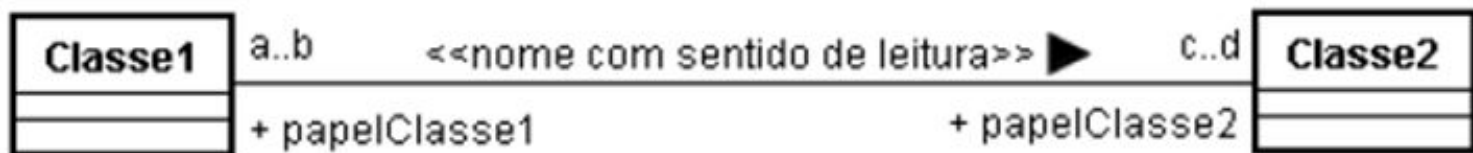


# Notação para Associações

*Exemplo:* Cliente efetua Pedido



*Notação:*



# Agregação

- Tipo especial de associação

- Demonstra que as informações de um objeto precisam ser complementadas por um objeto de outra classe

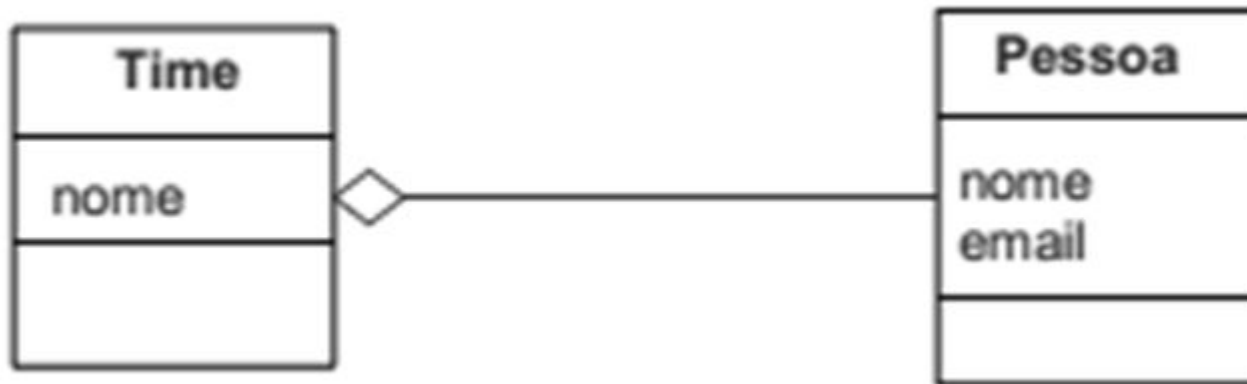
- Associação Todo-Parte

- objeto-todo

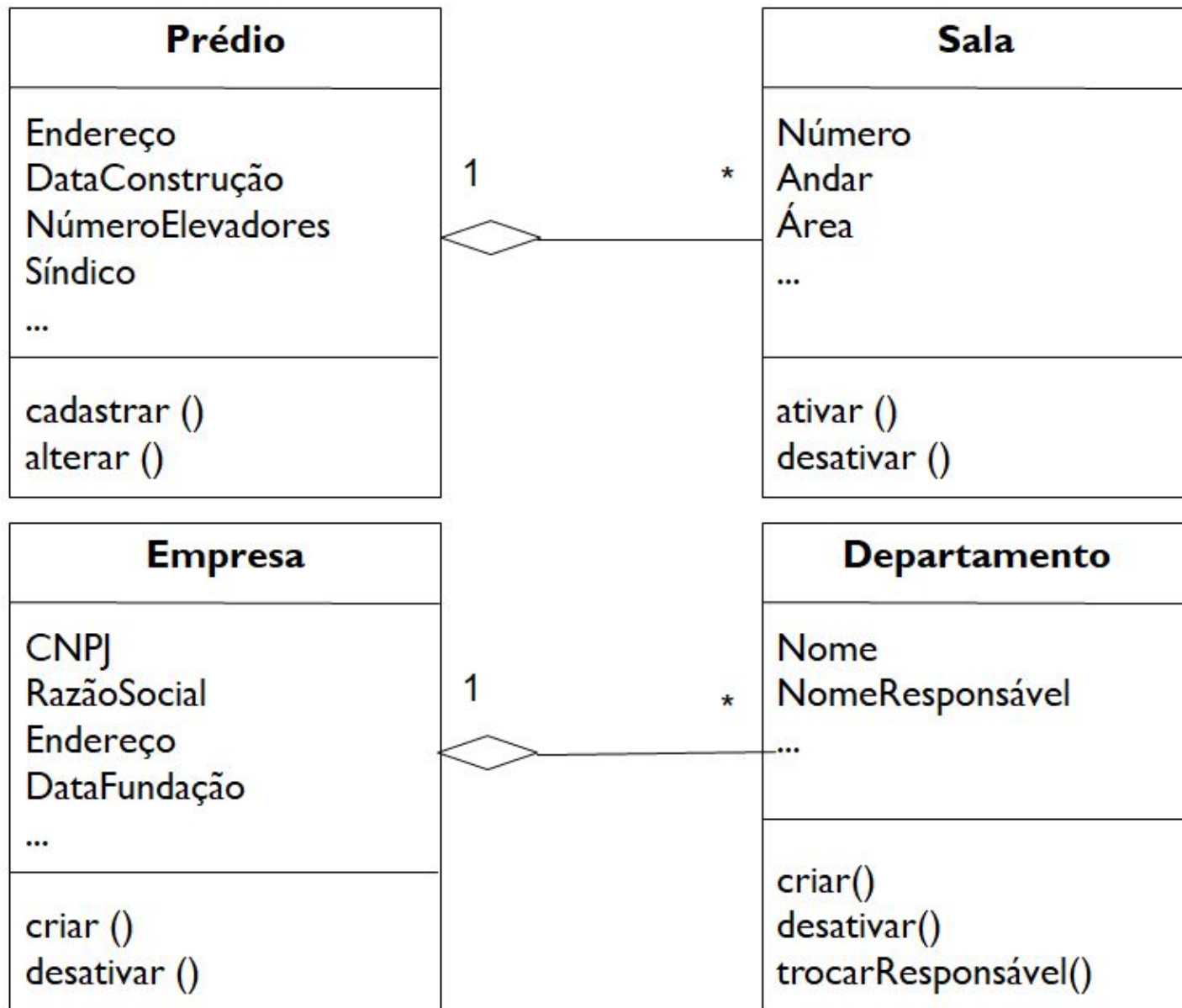
- objeto-parte

# Representação

- Um losango na extremidade da classe que contém os objetos-todo



# Exemplos



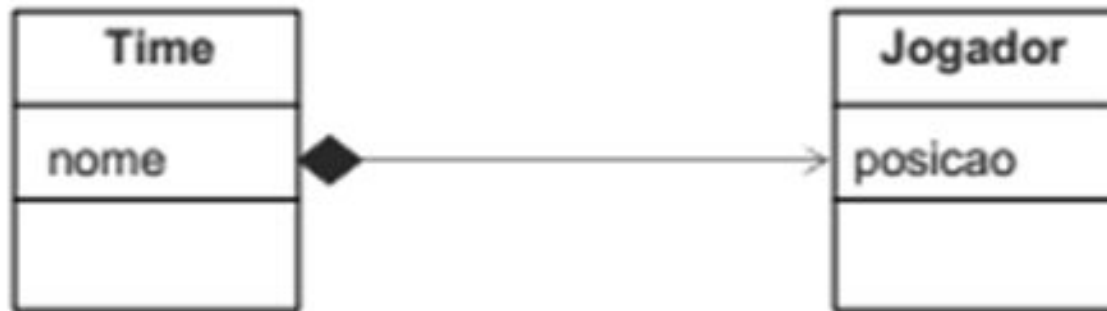
# Composição

- Uma variação do agregação
- Representa um vínculo mais forte entre os objetos-todo e objetos-parte
- Objetos-parte **têm** que pertencer ao objeto todo
  - O todo não existe (ou não faz sentido) sem a parte

# Representação da Composição

- Um losango preenchido

- Da mesma forma que na Agregação, deve ficar ao lado do objeto-todo





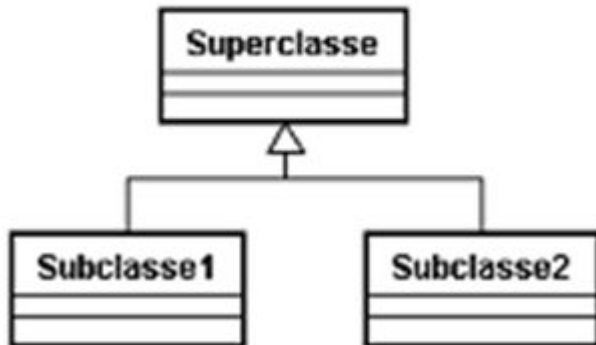
# Especialização/Generalização

- Identificar superclasse (geral) e subclasses (especializadas)
  - Semântica “é-um”
  - Tudo o que a classe geral pode fazer, as classes específicas também podem
- Atributos e métodos definidos na superclasse são herdados pelas subclasses

# Especialização/Generalização

Atributos descrevem propriedades dos objetos de uma classe.

*Notação:*



*Exemplo:*



# Especialização/Generalização

