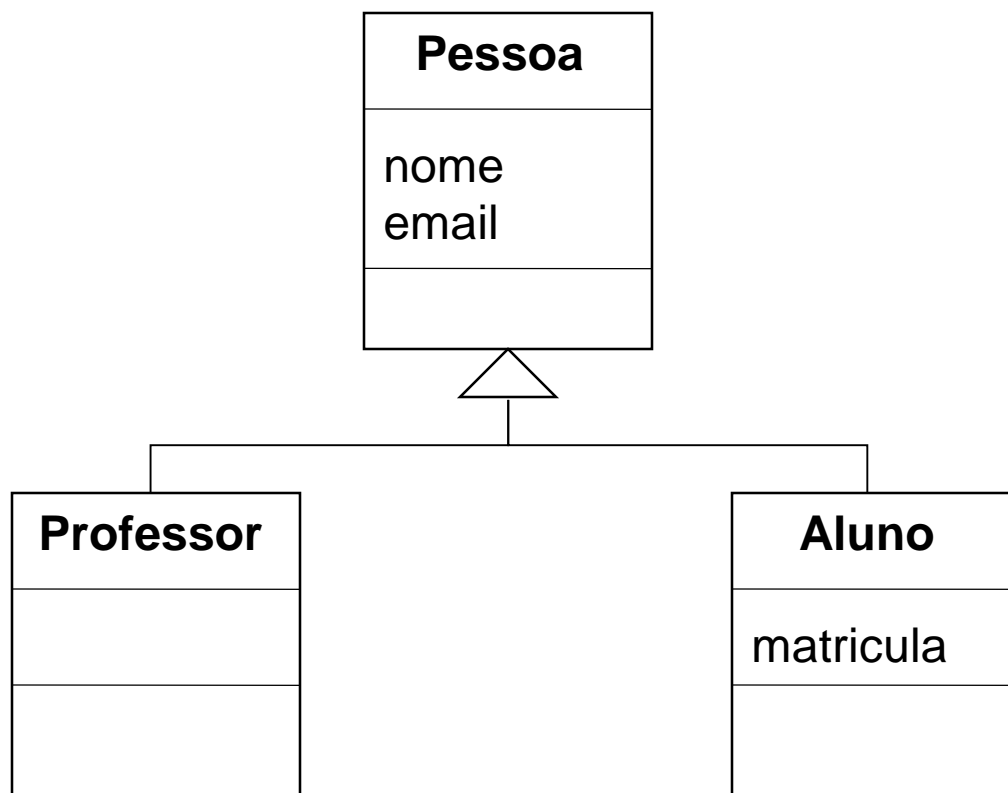


Diagrama de Classes

Diagrama de Classes

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

Exemplo de Diagrama



Outro Diagrama de Classes

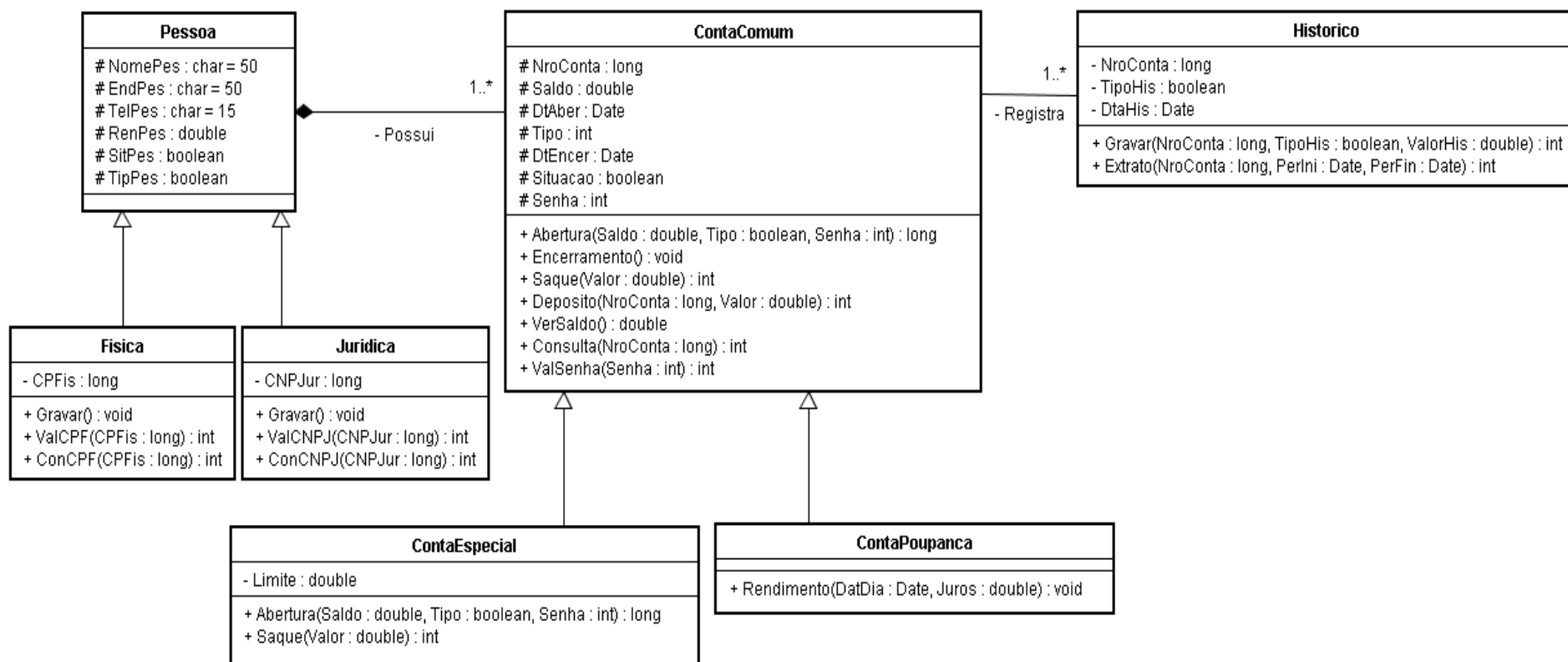


Diagrama de Classes

A estrutura do projeto

Diagrama de Classes

- O mais importante e o mais utilizado diagrama da UML
- Permite a visualização das classes que compõem o sistema
- Representa
 - Atributos e métodos de uma classe
 - Os relacionamento entre classes.

Diagrama de Classes

- Apresenta uma visão estática de como as classes estão organizadas
- Preocupação com a estrutura lógica

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
 - Diagrama de Seqüência
 - Diagrama de Atividades

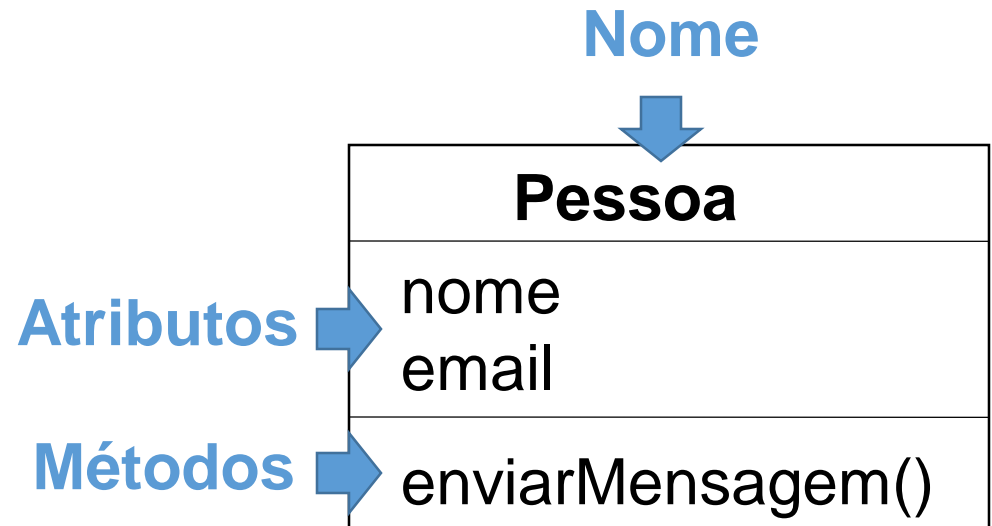
Representação de uma Classe

- Uma classe é representada por um retângulo com três divisões:
 - Nome da Classe
 - Atributos da Classe
 - Métodos da Classe

Pessoa
nome email
enviarMensagem()

Representação de uma Classe

- Uma classe é representada por um retângulo com três divisões:
 - Nome da Classe
 - Atributos da Classe
 - Métodos da Classe



Tipos de visibilidade

- Pública (+)
 - O atributo ou método pode ser utilizado por qualquer classe
- Protegida (#)
 - Somente a classe ou sub-classes terão acesso
- Privada (-)
 - Somente a classe terá acesso

Tipos de visibilidade

- Pública (+)
 - O atributo ou método pode ser utilizado por qualquer classe
- Protegida (#)
 - Somente a classe ou sub-classes terão acesso
- Privada (-)
 - Somente a classe terá acesso

Pessoa
nome - email
+ enviarMensagem()

Relacionamento

- Classes possuem relacionamentos entre elas
 - Compartilham informações
 - Colaboram umas com as outras
- Principais tipos de relacionamentos
 - Associação
 - Agregação / Composição
 - Herança
 - Dependência

Comunicação entre Objetos (I)

- Conceitualmente, objetos se comunicam através da troca de mensagens.
- Mensagens definem:
 - O nome do serviço requisitado
 - A informação necessária para a execução do serviço
 - O nome do requisitante.

Comunicação entre Objetos (II)

- Na prática, mensagens são implementadas como chamadas de métodos
 - Nome = o nome do método
 - Informação = a lista de parâmetros
 - Requisitante = o método que realizou a chamada

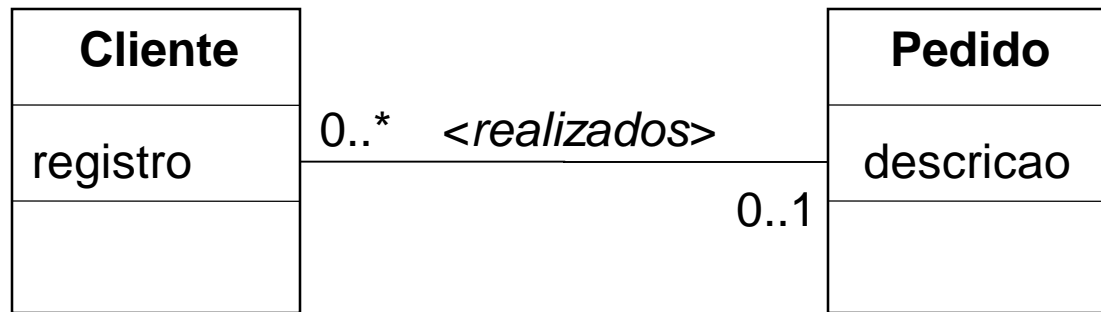
Associações

- 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

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.

Representação de Associação

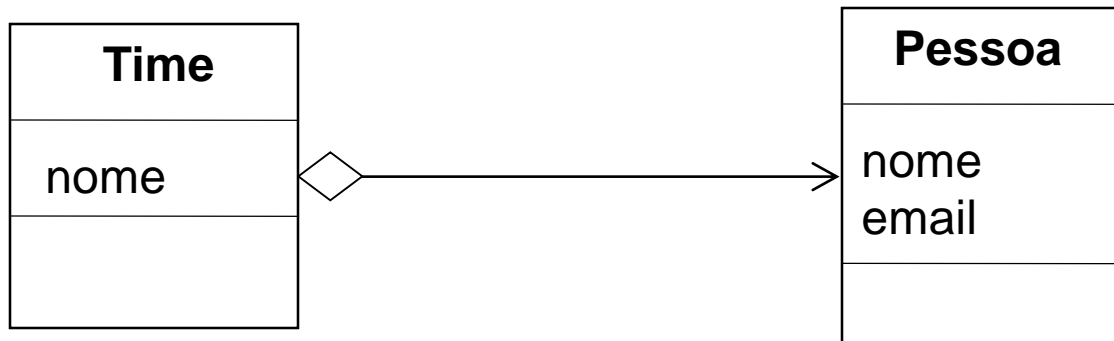


Agregação

- 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 de Agregação

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

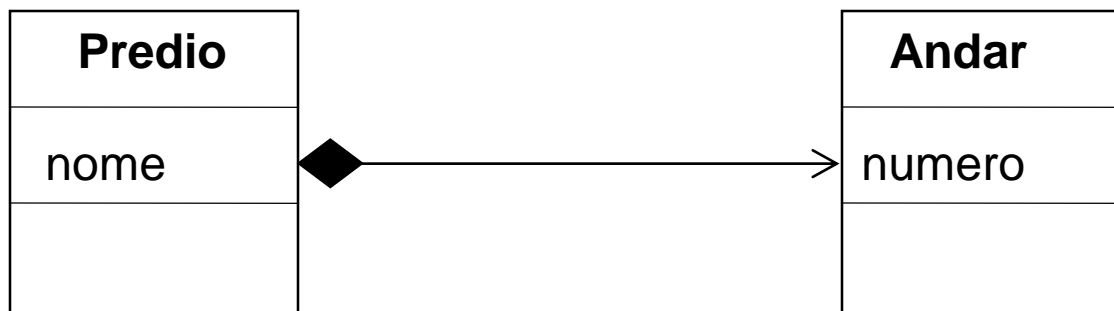


Composição

- Uma variação do tipo agregação
- Representa um vínculo mais forte entre 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

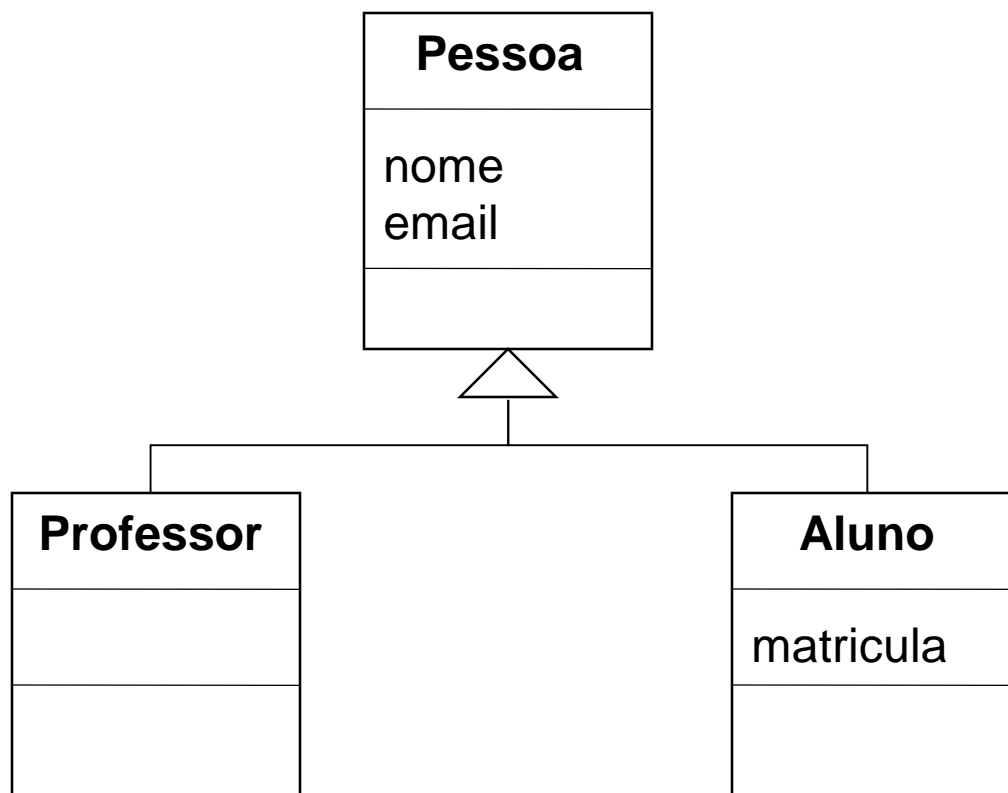
- Um losango preenchido, e da mesma forma que na Agregação, deve ficar ao lado do objeto-todo



Especialização / Generalização

- Identificar classes-mãe (gerais) e classes-filhas (especializadas)
- Atributos e métodos definidos na classe-mãe são **herdados** pelas classes-filhas

Especialização / Generalização

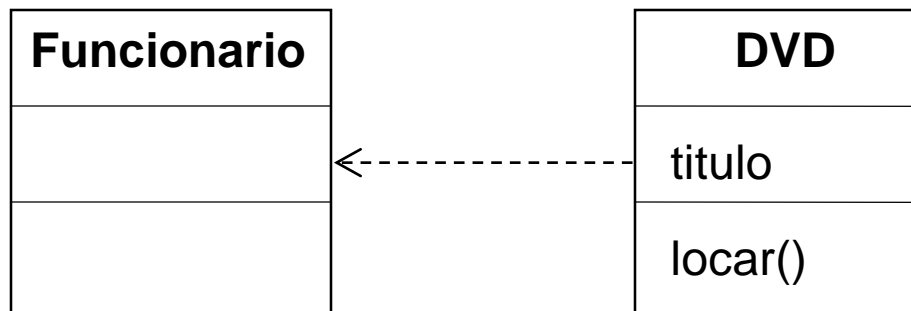


Dependência

- Tipo menos comum de relacionamento
- Identifica um baixo grau de dependência de uma classe em relação a outra

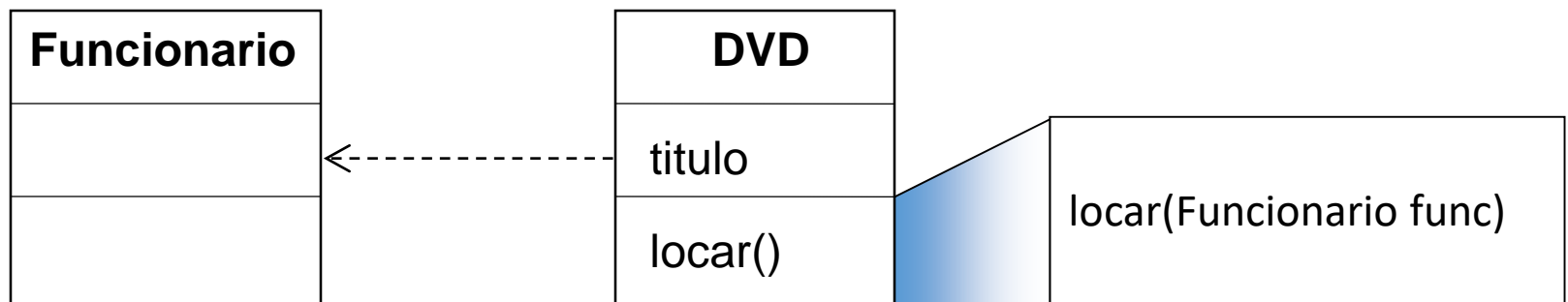
Dependência

- Representado por uma reta tracejada entre duas classes
- Uma seta na extremidade indica o dependente



Dependência

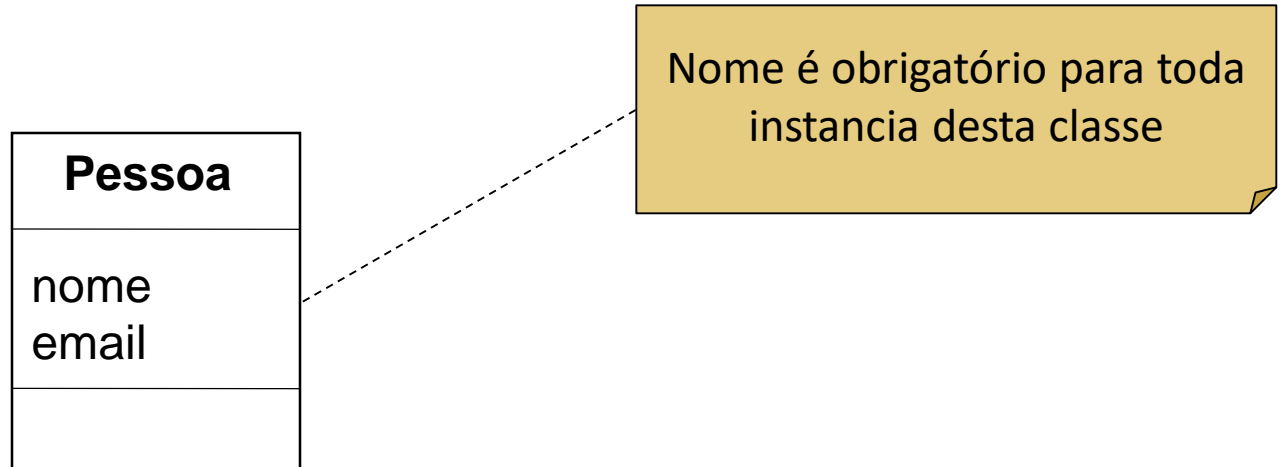
- Representado por uma reta tracejada entre duas classes
- Uma seta na extremidade indica o dependente



Notas

- Informativos
 - Algum comentário na classe, método ou atributo
 - Alguma restrição de funcionalidade
- Objetivo é informa como o objeto se comporta

Notas



Referências

- DEITEL, H. M.; DEITEL P. J. **Java: Como Programar**, 6a. Edição. Pearson, 2005.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus, 2000.