

Diagrama de Classes

Prof. Fernando e Profa. Thayani

Diagramas de Classes UML

Um diagrama de classes é usado para descrever a estrutura estática de classes no sistema, permitindo definir os **atributos**, operações (**métodos**) e os **relacionamentos** entre as classes.

Apresenta uma visão estática da organização das classes, definindo sua estrutura lógica.

É um dos diagramas mais populares, e serve como base para a construção de outros diagramas UML.

Basicamente, descrevem o que deve estar presente no sistema modelado.

Classes, Atributos e Métodos

Uma **classe** é uma representação de um item do mundo real, físico ou abstrato, na forma de um tipo de dados personalizado.

As classes possuem estruturas internas chamadas de Atributos e de Métodos

Atributos são usados para armazenar os dados dos objetos de uma classe.

Métodos são as operações, ou funções que a instância pode executar.

Uma instância de classe é chamada de **Objeto**.

Classes, Atributos e Métodos - Exemplo

Classe: **Pessoa**

Atributos: **Nome, Altura, Idade, Peso**

Métodos: **Andar, Comer, Falar, Estudar, Dormir, Trabalhar**

Objeto da Classe (Instância):

Atributos:

- ▶ Nome: **Pedro**
- ▶ Altura: **1,60m**
- ▶ Idade: **42**
- ▶ Peso: **89kg**

Representação de uma Classe

Representamos uma classe usando um diagrama dividido em três compartimentos:

- ▶ Nome: Inclui o nome e o estereótipo da classe (informação sobre a classe)
- ▶ Atributos: Lista de atributos da classe no formato **nome:tipo** ou **nome:tipo=valor**
- ▶ Operações: Lista de métodos da classe no formato **método(parâmetros): tipo_retorno**

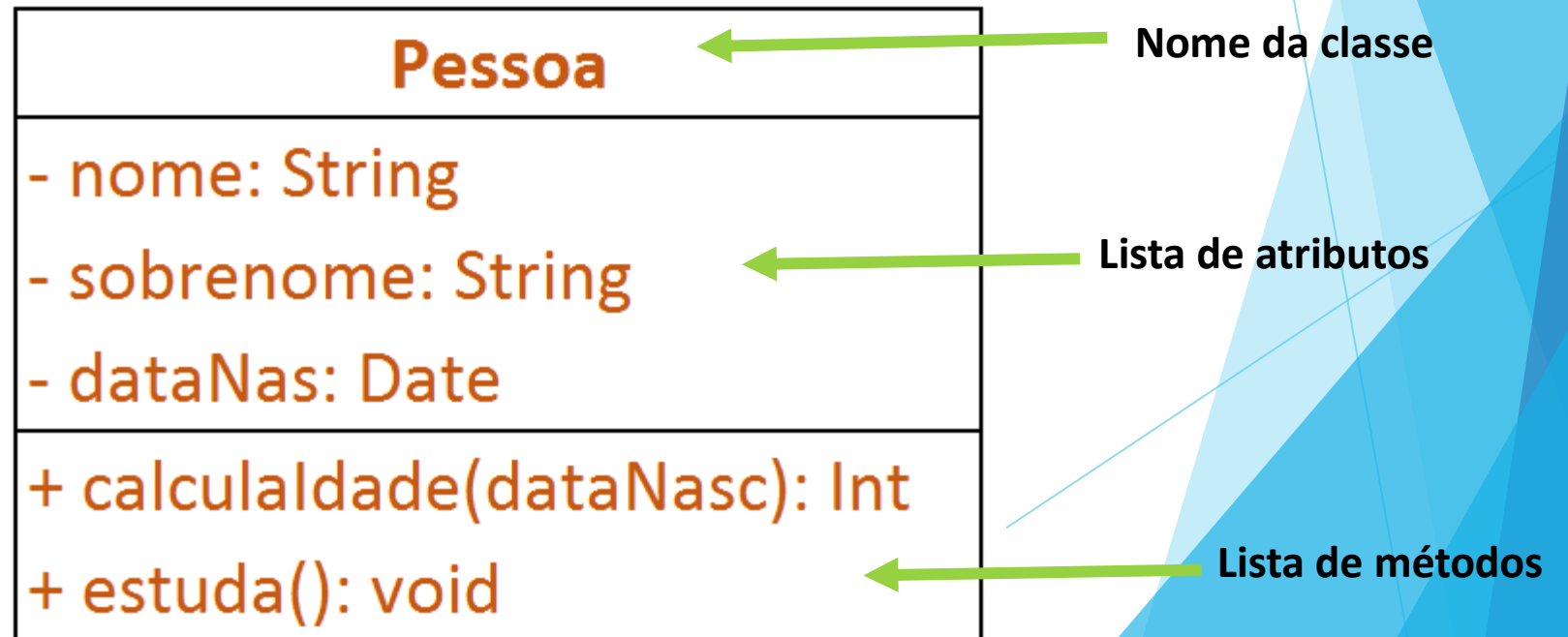
Visibilidade dos Membros (Atributos / Métodos)

Representamos a visibilidade dos atributos e das operações usando os modificadores de acesso a seguir:

- ▶ + Público
- ▶ #Protegido
- ▶ - Privado
- ▶ ~Pacote
- ▶ / Derivado

Representação de uma Classe

Exemplo: Representando uma classe **Pessoa**, que contém os atributos **nome**, **sobrenome** e **dataNasc**, além do método **calculaldade**:



Relacionamentos entre Classes

Um Relacionamento é uma conexão entre itens. Existem vários tipos de relacionamentos possíveis entre classes:

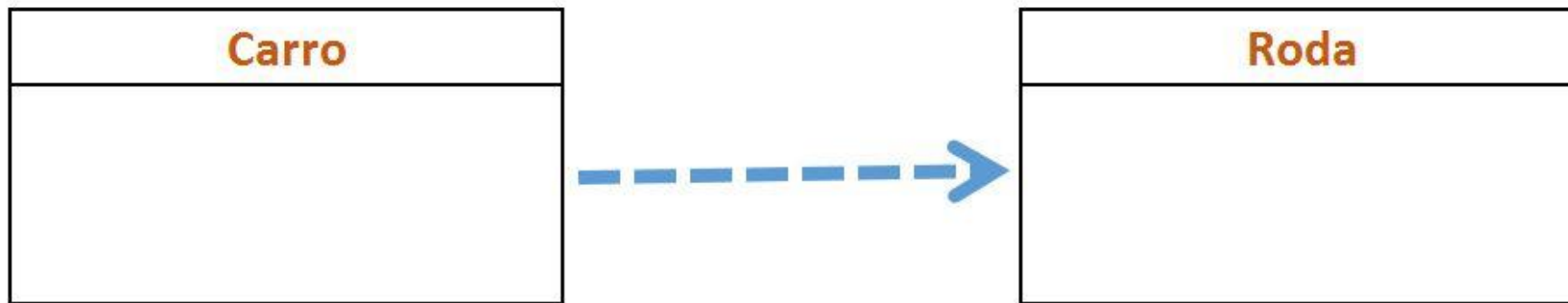
- ▶ Dependência
- ▶ Associação
- ▶ Agregação
- ▶ Composição
- ▶ Generalização

Cada uma desses relacionamentos possui uma representação gráfica específica.

Relacionamento de Dependência

Dependência fraca, usualmente transiente, que ilustra que uma classe usa informações e serviços de outra classe em algum momento, dependendo dela.

Do tipo "Classe A *depende* da Classe B".



Multiplicidade

A multiplicidade é usada para determinar o número mínimo e o número máximo de objetos envolvidos na associação, de cada lado, e também pode especificar o nível de dependência entre os objetos.

Multiplicidade	Significado
0 .. 1	No mínimo zero e no máximo um. Indica não-obrigatoriedade do relacionamento.
1 .. 1	Um e somente um. UM objeto da classe se relaciona com um objeto de outra.
0 .. *	Mínimo nenhum e no máximo muitos.
1 .. *	Mínimo um e no máximo muitos.
*	Muitos.
2..7	Mínimo 2 e no máximo 7. Ou seja, no mínimo duas e no máximo sete instância envolvidas no relacionamento

Relacionamento de Associação

Relacionamento mais forte do que a dependência, indica que a classe mantém uma referência a outra classe ao longo do tempo. As associações podem conectar mais de duas classes.

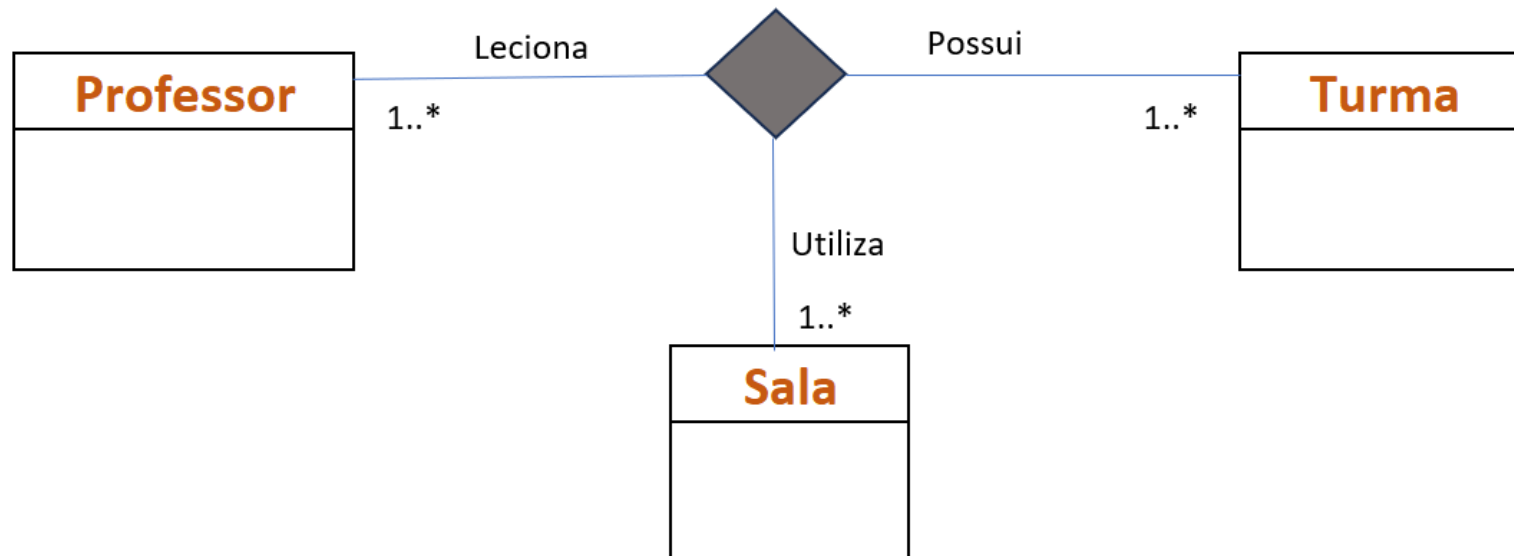
Do tipo "Classe A *tem uma* Classe B"

- A seta representa a Navegabilidade



Associação Ternária

Associação que conecta objetos de três classes. Um losango indica o ponto de convergência (conexão) das classes envolvidas.



Relacionamento de Agregação

Relacionamento mais específico do que a associação, indica que uma classe é um contêiner ou uma coleção de outras classes. As classes contidas não dependem do contêiner - assim, quando o contêiner é destruído, as classes continuam existindo.

Do tipo "Classe A *possui* uma Classe B"



Relacionamento de Composição

Varição mais específico da agregação, este relacionamento indica uma dependência de ciclo de vida forte entre as classes, de modo que quando um contêiner é destruído, seu conteúdo também o é.

Do tipo " Classe A é *parte* da classe B"








Relacionamento de Generalização / Especialização

Relacionamento entre itens gerais (superclasses / classes-mãe) e tipos mais específicos desses itens (subclasses / classes-filha). Representa a Herança entre as classes.

Do tipo “Classe A *é um tipo de* Classe B”.



Resumo da notação de Relacionamentos

- ▶ Associação 
- ▶ Agregação 
- ▶ Composição 
- ▶ Herança 
- ▶ Dependência 

Dúvidas?