

Programação Orientada a Objetos – Aula 13

Prof. Dr. Eduardo Takeo Ueda
eduardo.tueda@sp.senac.br

UML (Unified Modeling Language)

- UML é uma linguagem visual para modelagem
- Independe de linguagem de programação
- Independe de processo de desenvolvimento
- Não é uma técnica de modelagem

Modelos

- O que é um modelo?
 - Um modelo é uma simplificação (representação) da realidade
- O que modelamos?
 - Dados, funcionalidades, comportamentos

Por que preciso de UML?

- Compreender melhor o sistema que deve ser desenvolvido
- Discutir ideias e soluções com outras pessoas
- Especificar a estrutura e o comportamento de um sistema

Desenvolvimento de um sistema/software

- **Análise de requisitos**
 - Casos de uso
- **Projeto/Design**
 - Abstrações (classes e objetos), Arquitetura
- **Implementação**
 - Programação com uma linguagem como Java, C#, ...
- **Testes**
 - Testes de unidade, integração e sistema

Astah



Astah

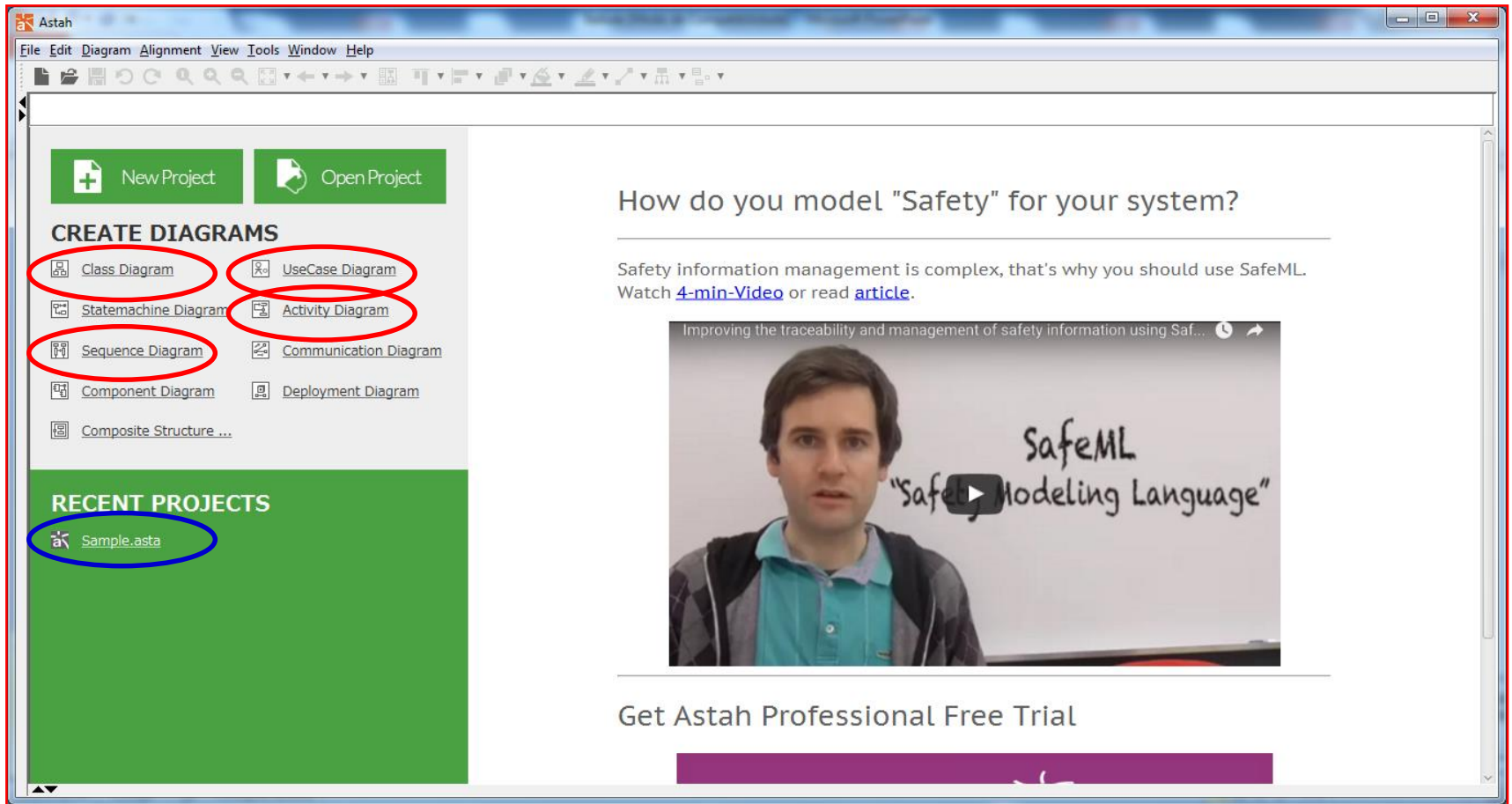
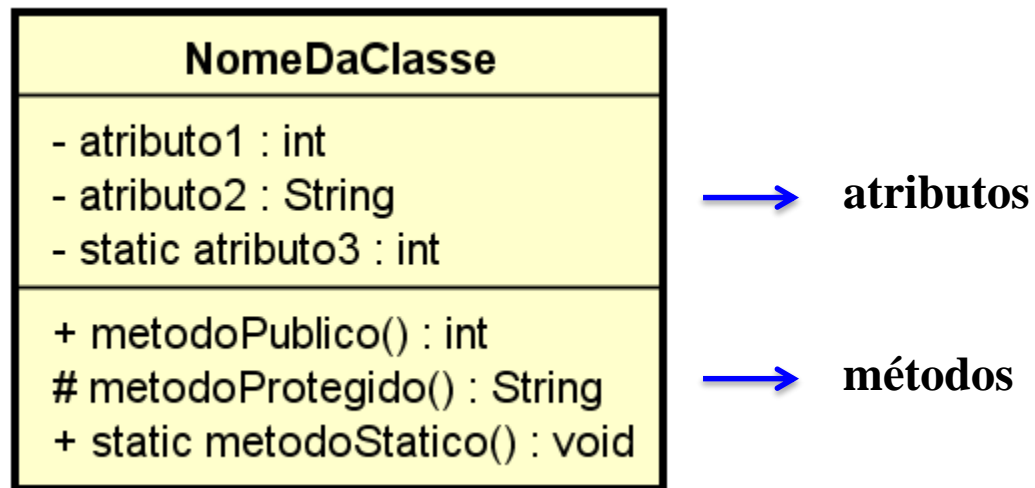


Diagrama de Classes

- Um diagrama de classes é utilizado para representar a **estrutura estática** de um projeto de software
- Visa permitir a visualização das **classes** que compõem o sistema, com os respectivos **atributos** e **métodos**, bem como mostrar como as classes se **relacionam**

Classes

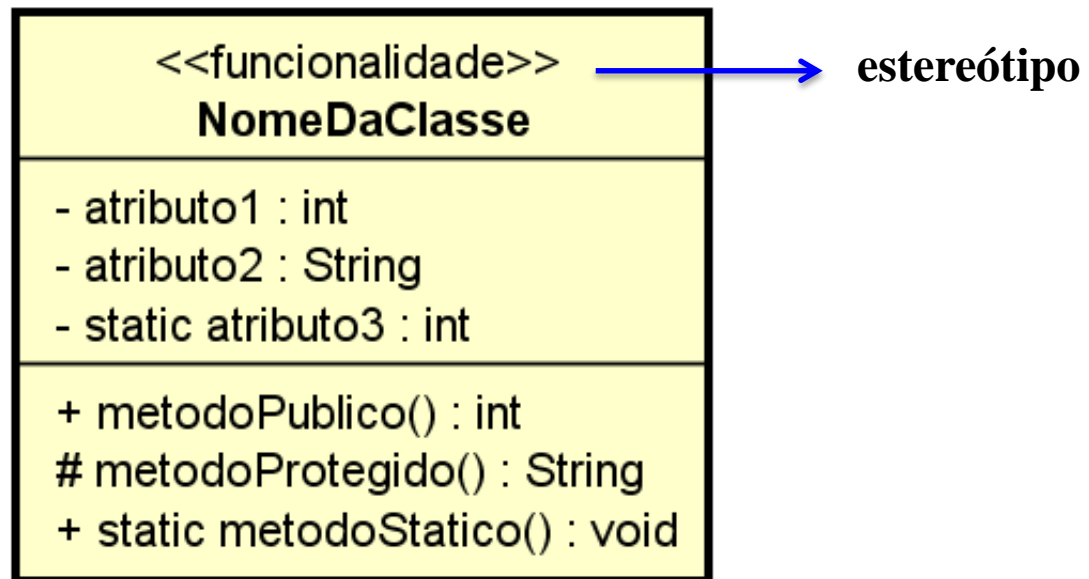
- Em UML uma classe é representada por um retângulo dividido em 3 partes



- (-) private, (#) protected, (+) public

Estereótipos

- Estereótipos podem ser usados para indicar funcionalidades da classe no contexto do projeto do software



Relacionamentos

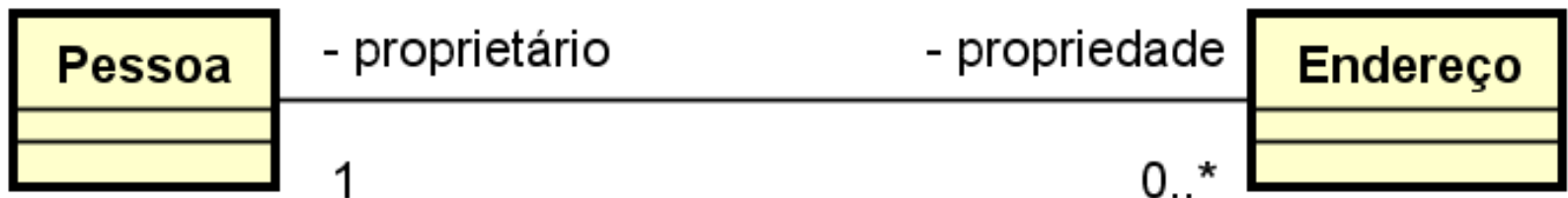
- Classes costumam ter relacionamentos entre si com o intuito de compartilhar informações e colaborarem umas com as outras para permitir a execução dos processos
- Tipos de relacionamento
 - Associação
 - Especialização/Generalização
 - Dependência

Associação

- Descreve um vínculo que ocorre normalmente entre duas classes (binária), entre uma classe com ela mesma (unária) e entre várias classes (ternária/N-ária)
- Determina que objetos de uma classe estão de alguma forma ligadas aos objetos de outra classe, podendo haver troca de informações e compartilhamento de métodos

Associação binária (ou normal)

- É o tipo de associação mais comum em projetos de software

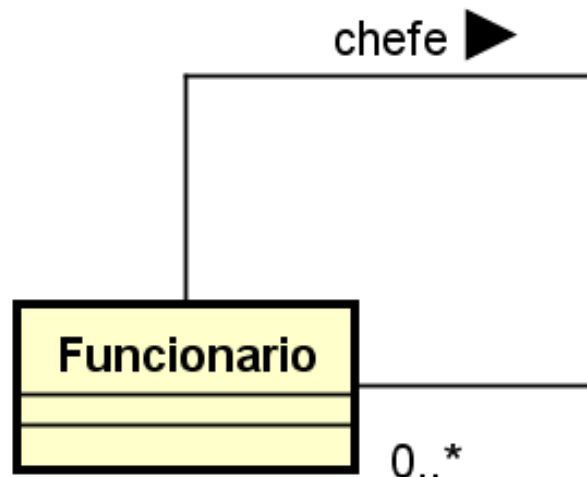


Multiplicidade em relacionamentos

Multiplicidade	Significado
0..1	No mínimo zero e no máximo um. Indica que os objetos das classes não precisam obrigatoriamente estar relacionados mas, se houver, apenas uma instância da classe se relaciona
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com a outra classe.
0..*	No mínimo nenhum e no máximo muitos. Indica que pode ou não haver instâncias da classe participante do relacionamento.
*	Muitos. Indica que muitos objetos da classe estão envolvidos
1..*	No mínimo um e no máximo muitos. Indica que há pelo menos um objeto envolvido no relacionamento, podendo haver muitos.
3..5	No mínimo três e no máximo cinco.

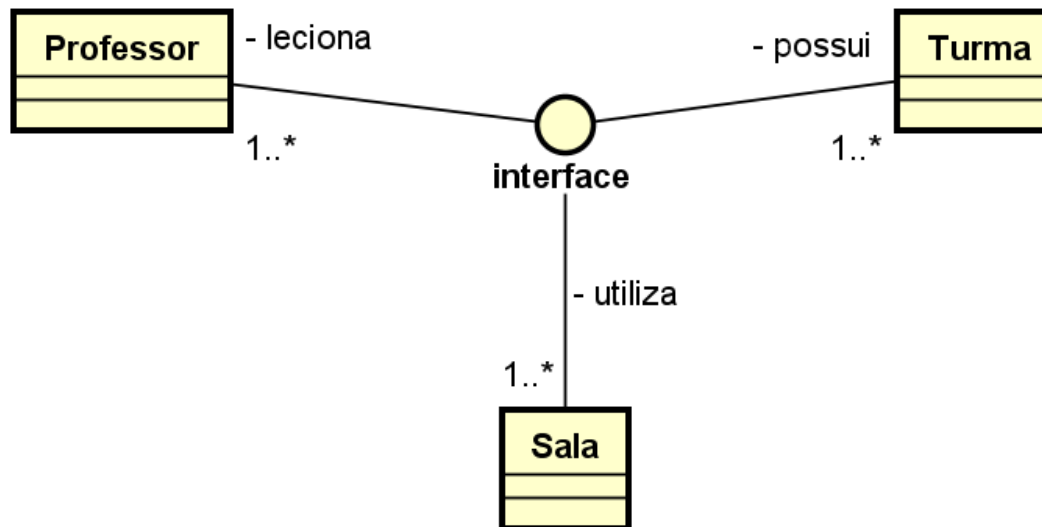
Associação unária (ou reflexiva)

- É possível conectar uma classe a ela mesma através de uma associação



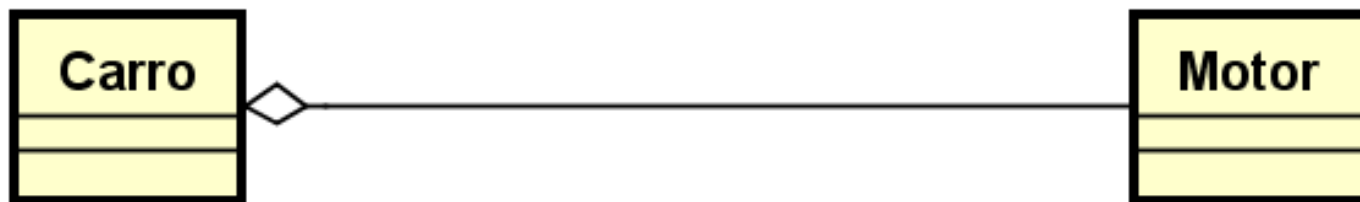
Associação ternária

- Mais de duas classes podem ser associadas entre si, a associação ternária associa três classes



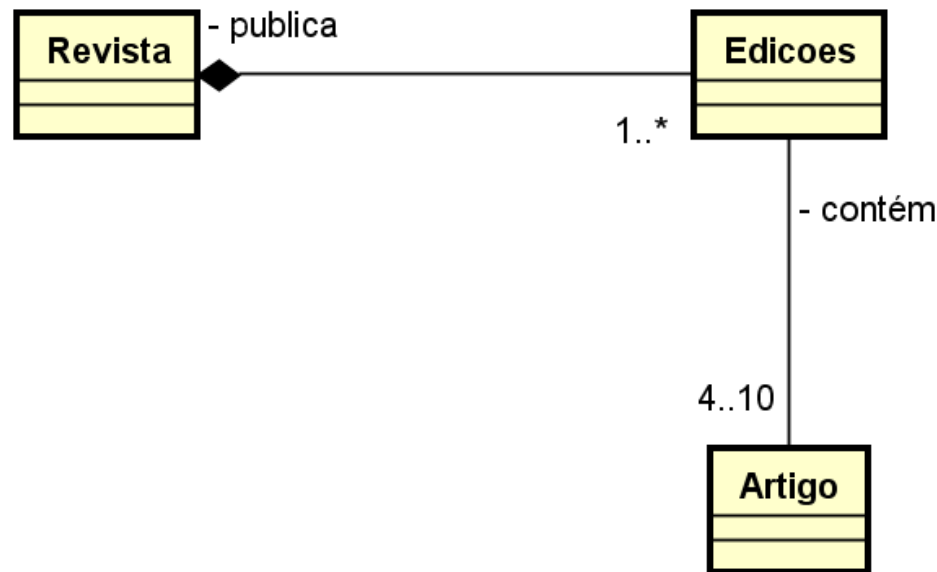
Agregação

- Agregação é um caso particular de associação, que indica que uma classe é parte ou está contida em outra classe



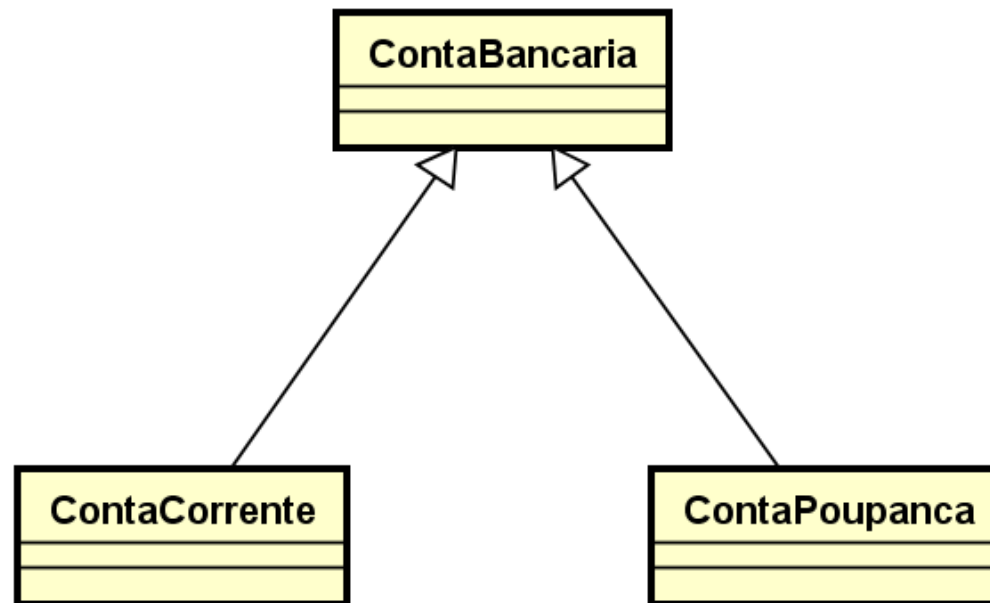
Composição

- Variação da agregação com vínculos mais fortes entre os objetos, i. é, um objeto existe apenas quando outros também existem



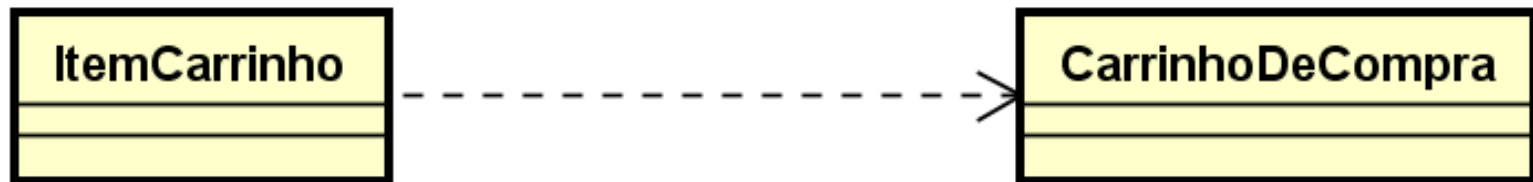
Especialização/Generalização

- Identifica superclasses (gerais) e subclasses (especializadas)



Dependência

- Indica que uma classe depende de outra classe para realizar alguma ação



Classe associativa

- Uma classe pode ser utilizada na associação de duas ou mais classes

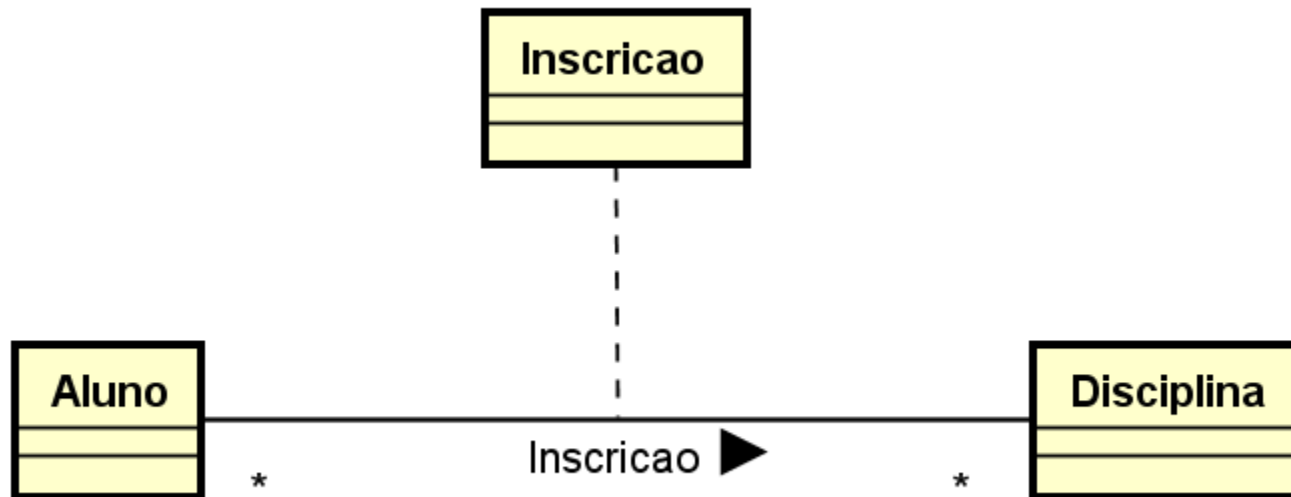


Diagrama de Classes

