
Noções de UML

Prof. Gabriel M. Alves
IFSP - 2023

Sumário

- Visão geral de UML (*Unified Modeling Language*)
- Diagrama de Classes
- Relacionamento entre classes

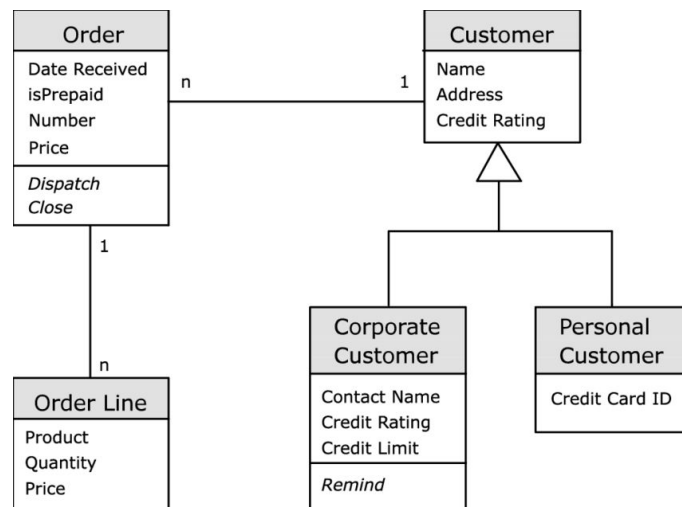


Visão geral de UML

- UML (Linguagem de Modelagem Unificada) **não é** uma linguagem de programação;
- Linguagem visual de modelagem utilizada no projeto de softwares para representar os seguintes aspectos:
 - Requisitos
 - Comportamentos
 - Estrutura lógica
 - Dinâmica de processos
 - Comunicação

Visão geral de UML

- Fornece **várias visões** do software que se deseja modelar representadas por diferentes diagramas UML;
- Diagramas estruturais
 - **Diagrama de classes** ←
 - Diagrama de objetos
- Diagramas comportamentais
 - Diagrama de Casos de Uso
 - Diagrama de Sequências
 - Diagrama de Colaboração
 - Diagrama de Estado
 - Diagrama de Atividades



Fonte: https://www.researchgate.net/publication/225686440_On_the_Nature_of_Design

Diagrama de Classes

- É um dos mais importantes e mais utilizados diagramas de UML;
- Representa as classes do sistema bem como seus atributos e métodos;
- Representa o relacionamento entre as classes;
- Fornece uma **visão estática** do sistema.

Nome da classe



Atributos



Métodos

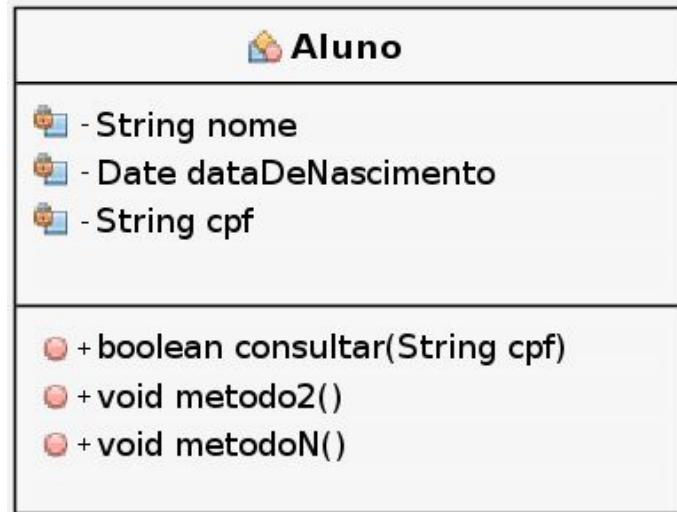
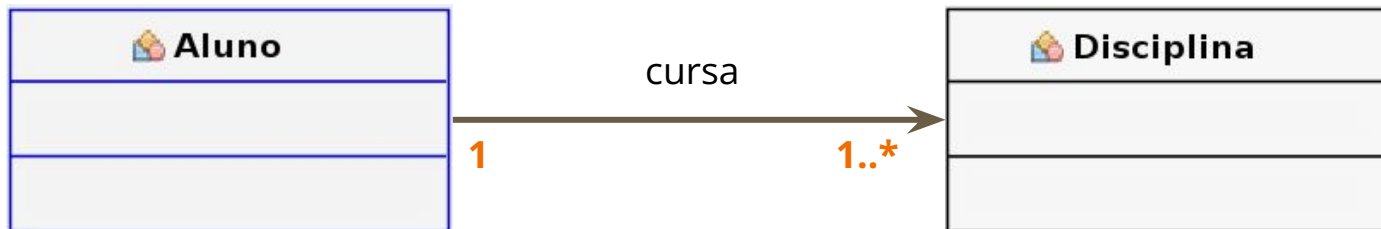


Diagrama de classes

- **visibilidade** nome : **tipo** = **valor inicial** {propriedades}
- **Visibilidade:** public (+); private (-); protected (#); package (~)
- **Tipo:** int, double, String, Date, etc
- **Valor inicial:** definido no momento da criação do objeto
- **Propriedades:** final, static, etc
- Exemplo: - **PI : double = 3.141589 {final}**

Relacionamento entre classes

- **Nome:** descreve o relacionamento (geralmente um verbo)
- **Navegabilidade:** indicada por uma seta no fim do relacionamento
- **Multiplicidade:** quantidade de elementos que cada relacionamento pode assumir (número de instâncias)



Relacionamento entre classes

- Multiplicidade:
 - Muitos: *
 - Exatamente um: 1
 - Zero ou mais: 0..*
 - Um ou mais: 1..*
 - Zero ou um: 0..1
 - Faixa especificada: 10..40

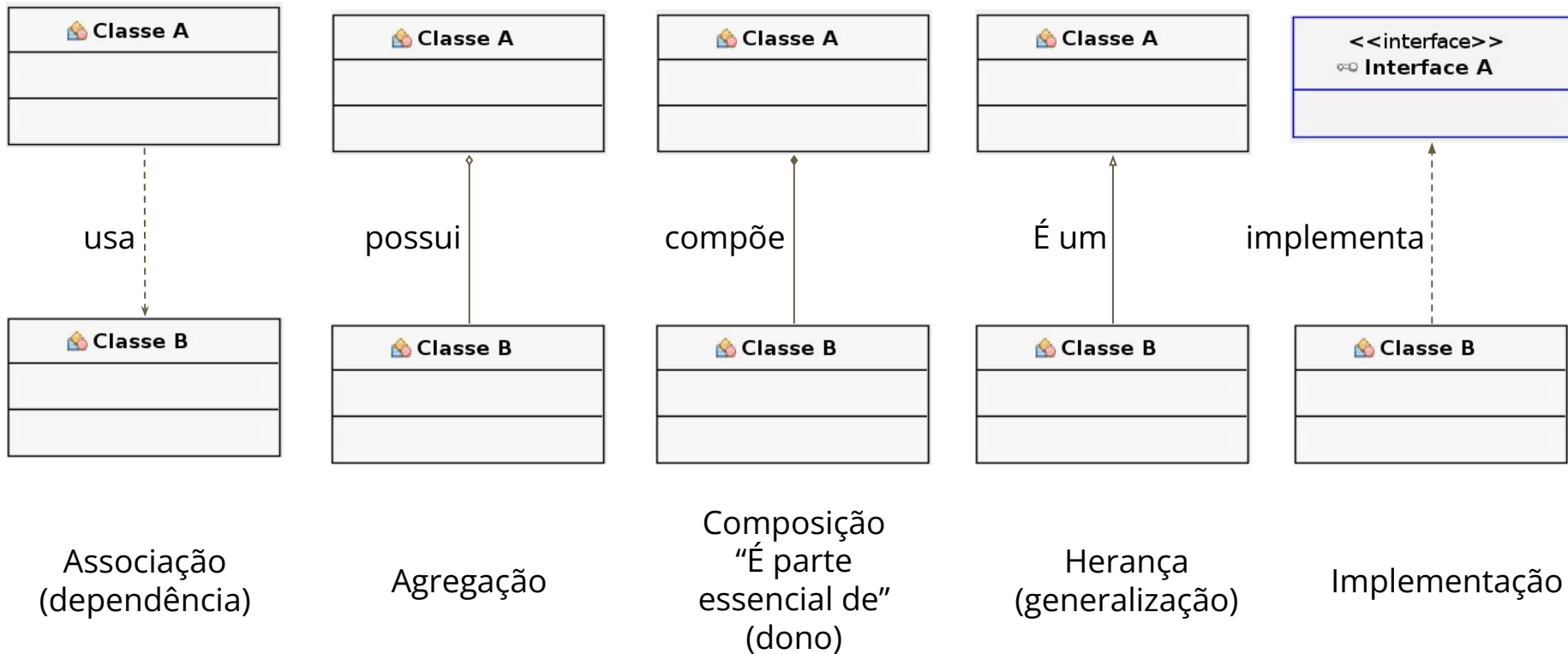


Fonte: <http://eastcliffcreatives.com/multiplicity/>



Fonte: <https://wavepho.weebly.com/multiplicity.html>

Relacionamento entre classes



Relacionamento simples - Associação

- Associação é a forma mais fraca de relacionamento entre classes
- As classes são **independentes**
- Representada por uma linha
- Pode conter um nome e uma seta
- Representa relacionamentos “usa um”
- Associação simples também é chamada de **dependência**

Relacionamento simples - Agregação e Composição

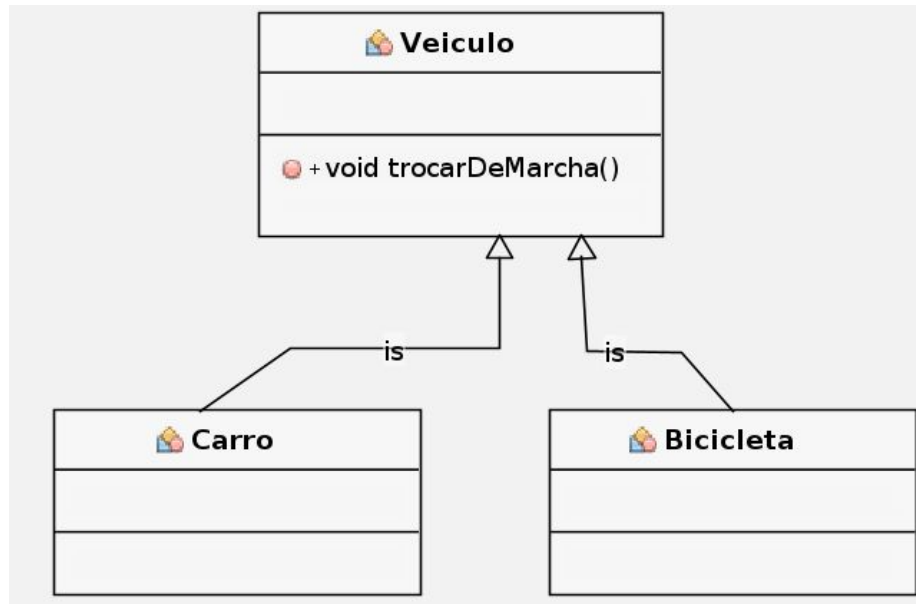
- São tipos de associação que representam relacionamentos “**tem um**”
- Uma classe **é formada** ou **contém** objetos de outras classes
- Exemplos:
 - Carro e roda
 - Prédio e apartamento
 - Computador e componentes

Relacionamento simples - Agregação e Composição

- Associação “todo-parte”
- Classe “**todo**” é o resultado da agregação/composição
- As instâncias (objetos) da classe “**parte**” formam a agregação/composição
- Agregação é a forma mais branda da composição
- **Composição**: as partes **não** podem existir independentes do todo
- **Agregação**: as partes podem existir independentes do todo

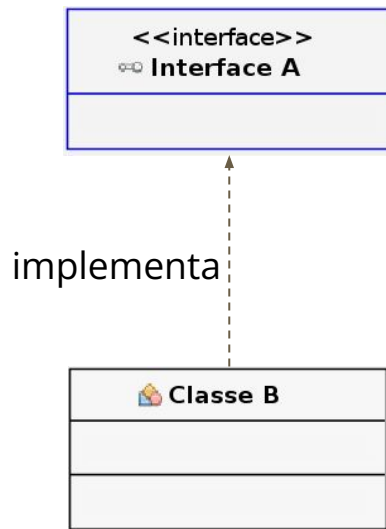
Relacionamento entre classes - Herança

- Herança representa relacionamento do tipo “é um”
- Permite organizar as classes hierarquicamente
- Generalização e Especialização
 - A partir da classe Geral (Base) deriva-se a classe Específica (Derivada ou sub-classes)
 - As classes especializadas precisam ter pelo menos uma propriedade que as diferencie



Relacionamento entre classes - Implementação

- Interface define **contrato** entre os objetos
- Não são classes comuns, não podem ser instanciadas
- Em classes é possível utilizar herança
- Em interfaces utiliza-se a implementação
- Notação semelhante a herança (apontar diferenças)
- Os métodos da interface precisam ser escritos pela classe que os implementa



Sugestão de leitura complementar

