

Aula 01 - Retomando Classes

Prof. Caio Saraiva Coneglian - caio.coneglian@gmail.com

Disciplina: Engenharia de Software II



Objetivo

- Apresentar o diagrama de classes e as relações.

Diagrama de Classes



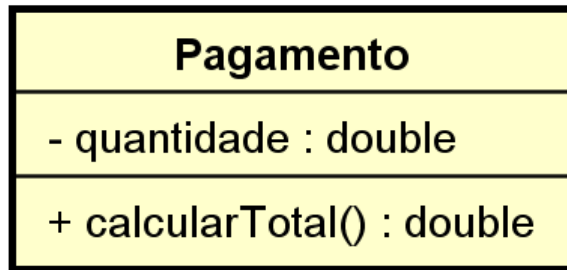
Diagramas de Classe

- UML;
- Representação de estrutura e relações das classes:
- Estes diagramas são usados como modelo para os objetos.

Classe



Classe no Diagrama de Classes



Relacionamentos



Relacionamento entre as classes

- As classes possuem relacionamentos entre elas;
 - Compartilham informações;
 - Colaboram com as classes;
- Principais tipos de relacionamentos:
 - Associação;
 - Agregação/Composição;
 - Herança;
 - Dependência.

Associação



Associação

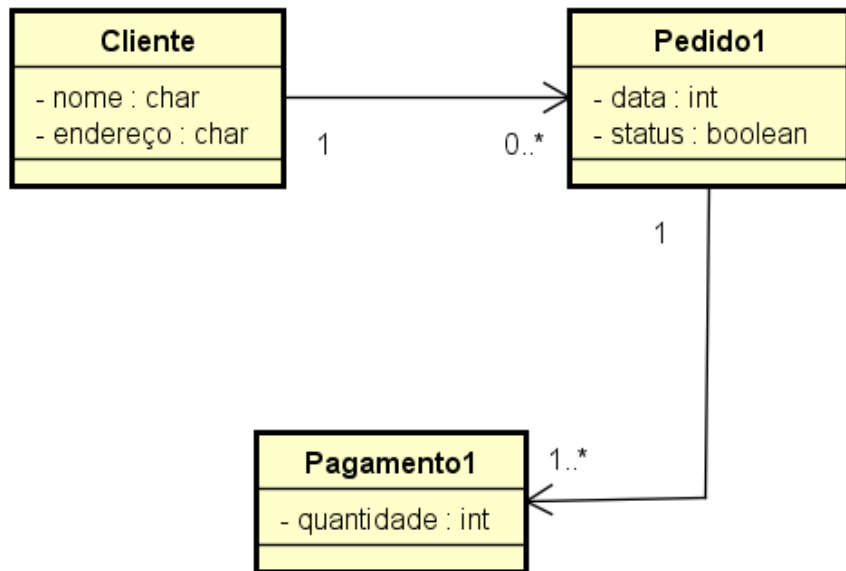
- Chamado de:
 - Associação Binária;
- Descreve um vínculo entre duas classes;

As instâncias de uma classe estão ligadas às instâncias de outra classe por meio das associações.

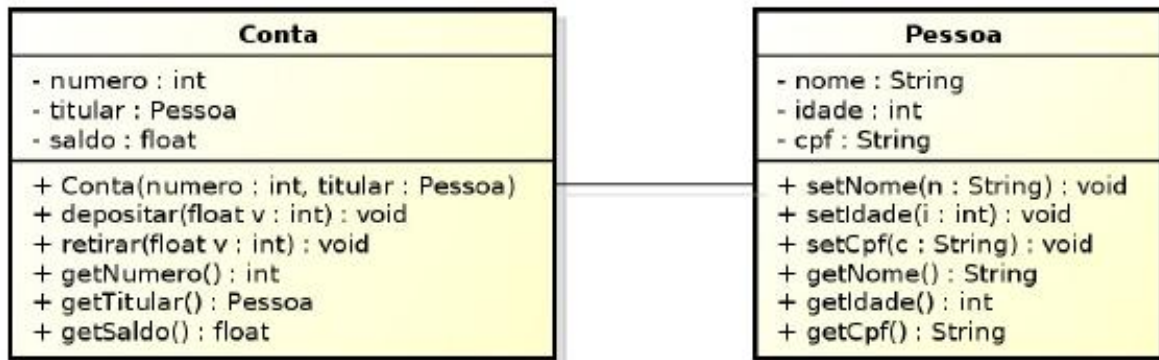
Multiplicidade

Relacionamento	Comentário
0 .. 1	0 ou 1 instância (no máximo 1). Indica que não é necessário estar obrigatoriamente vinculadas.
1 .. 1	1 e somente 1. Apenas uma instância se relaciona com as instâncias da outra classe.
0 .. *	Muitos. Podem haver muitas instâncias (objetos) envolvidos no relacionamento
1 .. *	1 ou muitos. Indica que há pelo menos 1 instância envolvidos no relacionamento.
2 .. 4	Valores específicos.

Associação



Exemplo



Agregação

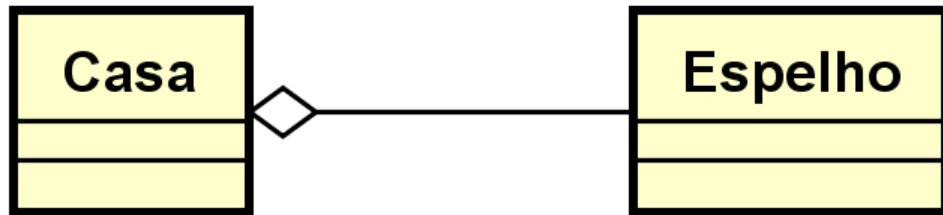


Agregação

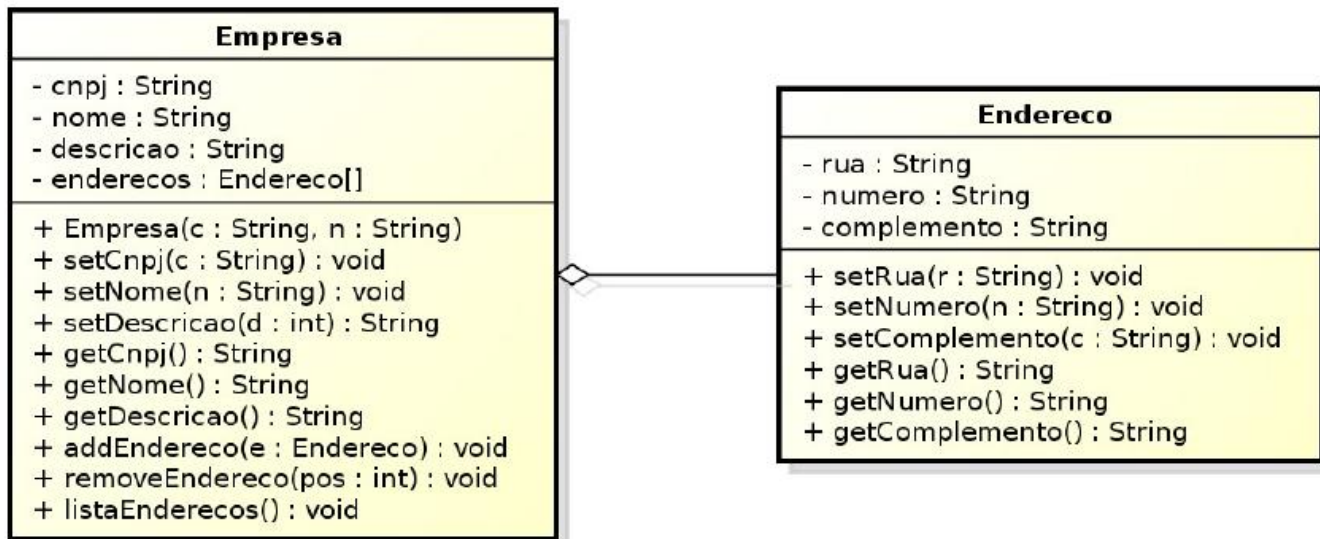
- Tipo de Associação;
- Esse relacionamento:
 - Demonstra que as informações de um objeto precisam ser complementadas por um objeto de outra parte;
- Associação Todo-Parte;
 - objeto-todo;
 - objeto-parte.

Agregação

- Um TODO é relacionado com suas PARTES.



Exemplo



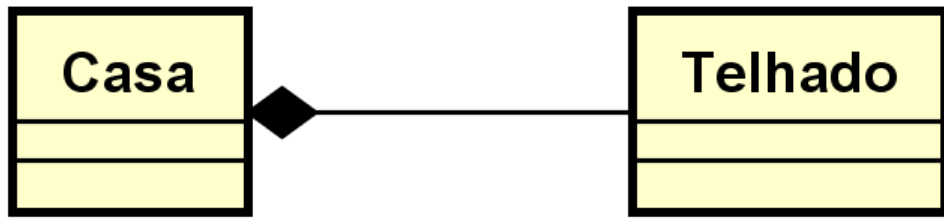
Composição



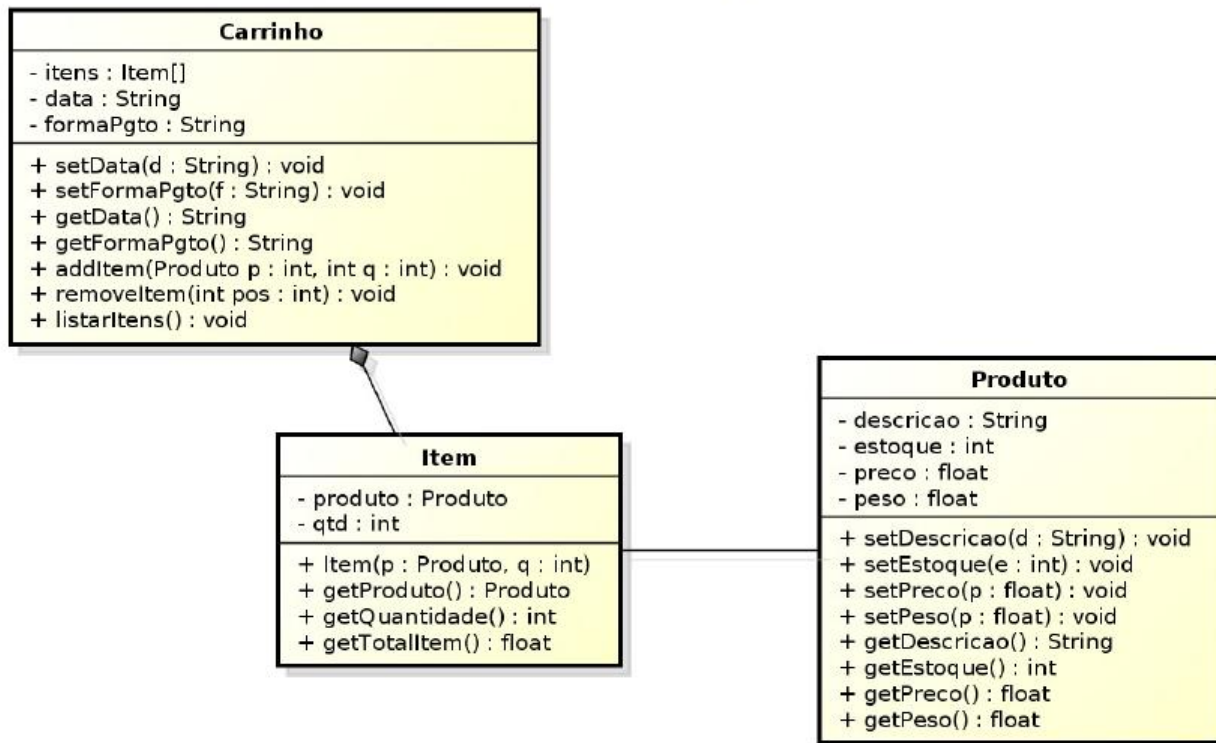
Composição

- Um tipo de agregação;
- Uma composição é:
 - um vínculo mais forte entre o objeto-todo e o objeto-parte;
- Objetos-parte **TÊM** que pertencer ao objeto-todo.
 - O todo não existe (ou não faz sentido) sem as partes;
 - Ou, as partes não existem sem o todo.

Composição



Exemplo

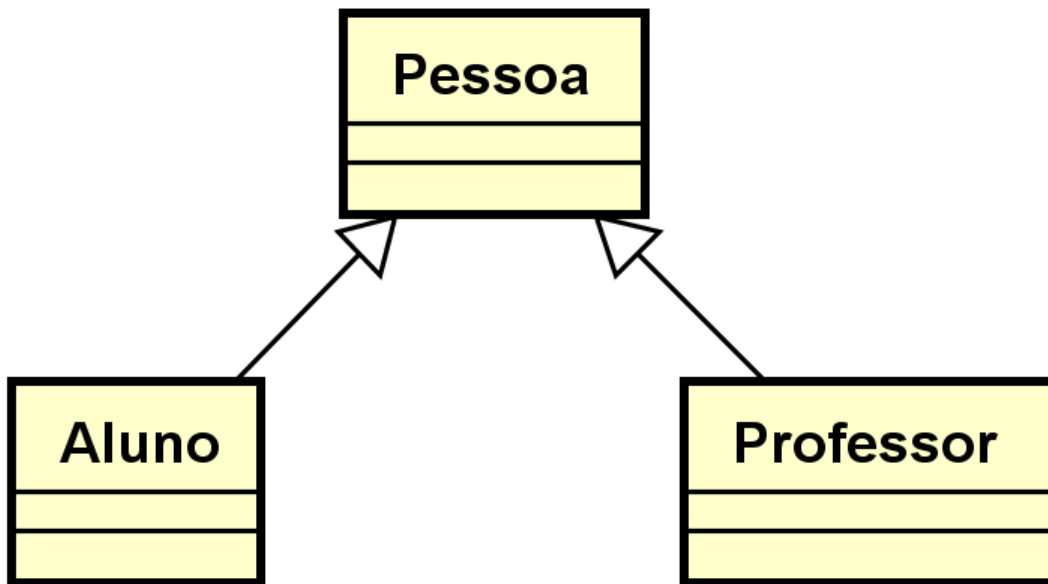


Especialização/Generalização



Especialização e Generalização

- Super-classe (geral) e sub-classe (especializada):
 - Herança;
 - O que uma classe de nível superior faz, pode ser feito por uma mais específica;
- Atributos e métodos da classe mãe são herdados pelas classes filhas.



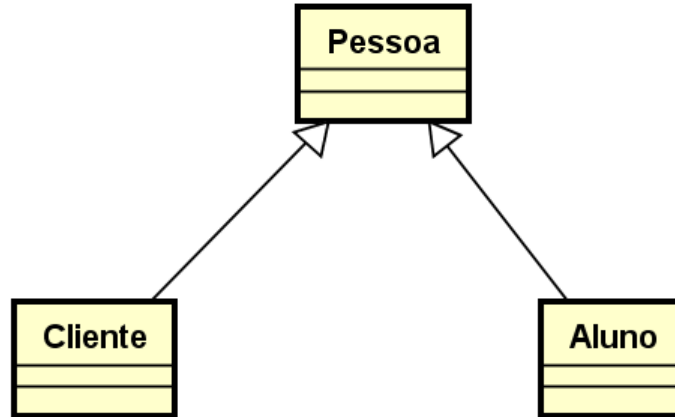
Vantagens Herança

- Herança:
 - Fonte de conhecimento sobre o domínio do sistema;
 - Mecanismo de abstração:
 - Classificar entidades;
 - Mecanismo de reuso em vários níveis:
 - Projeto e programação

Desvantagens Herança

- Herança:
 - Classes de objetos não são auto-contidas:
 - Não podem ser compreendidas sem referência às suas super-classes;
 - Reusar gráficos da fase de análise pode não funcionar:
 - Os gráficos de herança podem mudar nas fases de análise, projeto e implementação:
 - Pois tem diferentes funções (devem ser refinados).

Exemplo



Dependência



Dependência

- Tipo menos comum de relacionamento;
- Identifica uma relação FRACA entre os objetos de duas classes.
- Classe cliente é dependente de alguns serviços da classe fornecedora, mas não tem uma dependência estrutural interna com esse fornecedor.

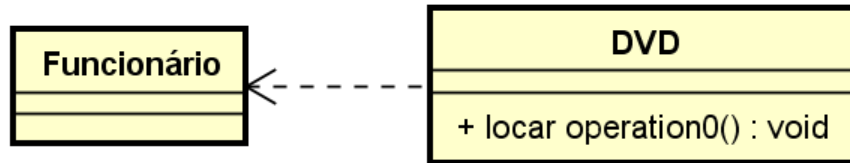
Dependência

Na prática, quando uma classe recebe como parâmetro outra classe, essa classe acessa o objeto global da outra.

Nessa caso, há dependência.

Dependência

Seta pontilhada



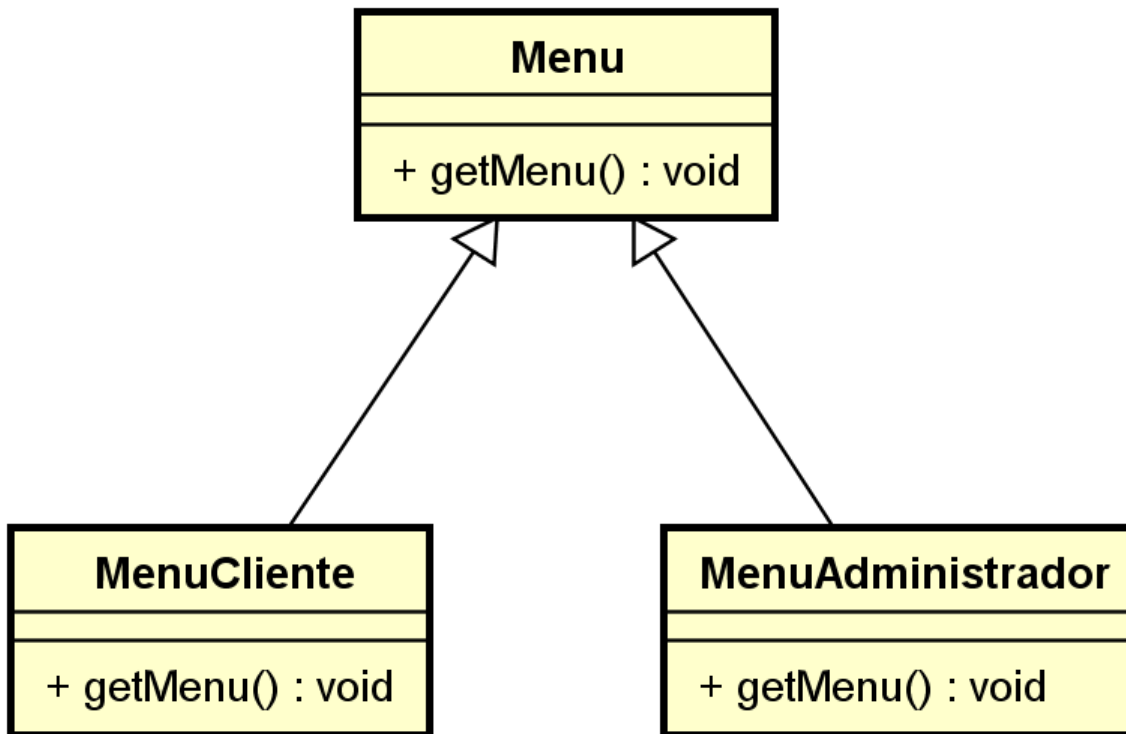
Polimorfismo



Polimorfismo

- Polimorfismo é a característica de linguagens orientadas a objetos que permite que diferentes objetos respondam a mesma mensagem cada um a sua maneira.
- O polimorfismo representa a capacidade de uma única referência invocar métodos diferentes.

Polimorfismo



Pacotes de Análise



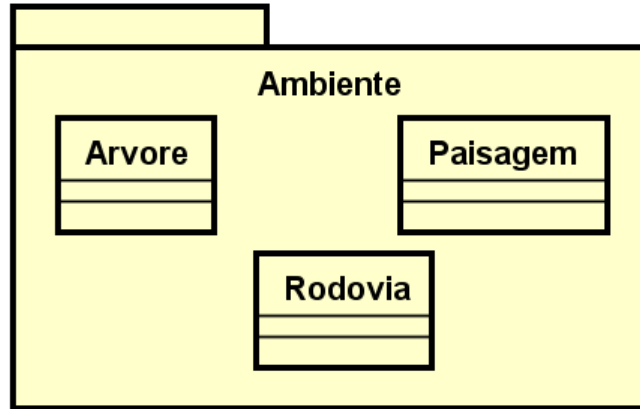
Pacotes de Análises

- Categorizar as classes é importante:
 - Visar empacotar como um agrupamento (pacote de análise);
- Exemplo:
 - Videogame:
 - Ambiente de Jogo:
 - Árvore, Paisagem, Rodovia, Parede, Ponte;
 - Personagem do Jogo:
 - Protagonista, Antagonista e PapéisApoio;
 - Regras do Jogo:
 - RegrasDeMovimentação e RestriçõesNaAção

Informação Importante

- Símbolos das classes:
- +: Visibilidade Pública
- -: Oculto dos demais pacotes
- #: Acessível apenas para pacotes contidos em determinado pacote.

Pacotes de Análises



Prática



Exercício 1

Um projeto de software pode empregar várias linguagens de programação.

Exercício 2

Um país é composto por várias cidades e uma delas é a sua capital

Exercício 3

Um projeto envolve várias pessoas e uma dessas pessoas é o gerente de projetos.

Exercício 4

Uma curva pode ser definida como uma coleção de pontos ordenados.

Exercício 5

Uma janela gráfica (interface com o usuário) compreende vários elementos de interação. Por exemplo, botões, menus e barras de rolagem.

Exercício 6

Um arquivo possui permissões de acesso. Cada permissão está associada a um grupo (de usuários).

Exercício 7

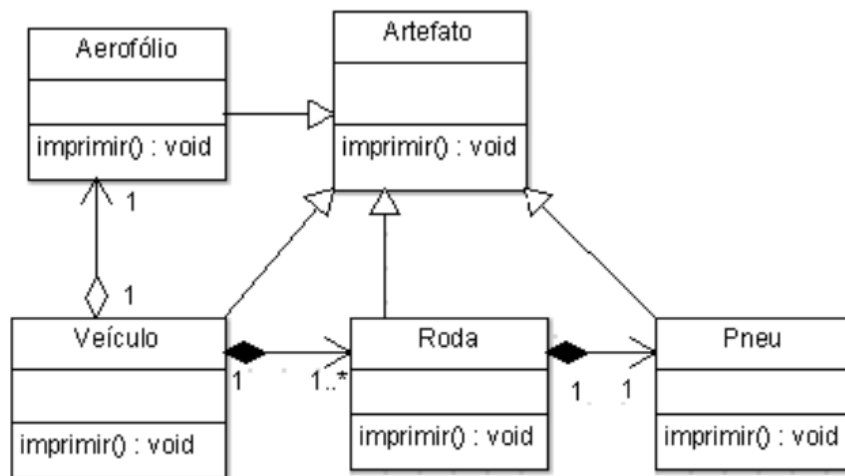
Um texto é uma combinação de parágrafos que, por sua vez, são combinações de sentenças.

Exercício 8

Uma graduação possui disciplinas. Cada disciplina é organizada em conteúdos. Alunos estão vinculados a uma disciplina. Professor está vinculado a uma disciplina. Coordenador é um professor. Cada curso possui um coordenador.

Exercício 9 - Poscomp 2018

- 28 - De acordo com o diagrama de classes UML a seguir, assinale a alternativa correta.



Exercício 9 - Poscompo 2018

- A) O relacionamento entre “Veículo” e “Roda” se relaciona diretamente com polimorfismo.
- B) O relacionamento entre “Roda” e “Pneu” se relaciona diretamente com polimorfismo.
- C) O relacionamento entre “Veículo” e “Pneu” se relaciona diretamente com polimorfismo.
- D) O relacionamento entre “Artefato” e “Veículo” se relaciona diretamente com polimorfismo.
- E) O relacionamento entre “Veículo” e “Aerofólio” se relaciona diretamente com polimorfismo.