



Universidade Federal do Ceará
Centro de Ciências/Departamento de Computação
Código da Disciplina: CK0084 Ano: 2021
Professor: Ismayle de Sousa Santos

Aula
14

Sistemas de Informações e Banco de Dados

Introdução à Diagramas de Classe



rfbrkh3



ismaylesantos@great.ufc.br



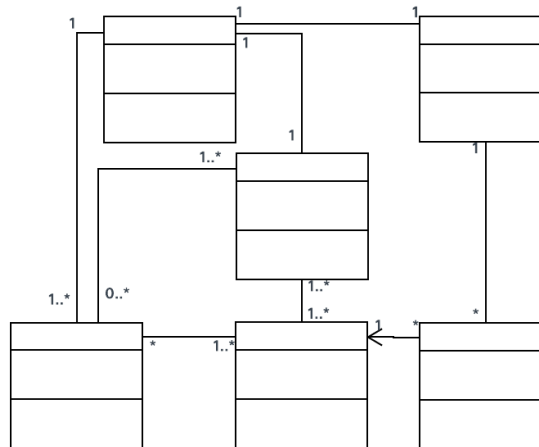
@IsmayleSantos

Hoje aprenderemos sobre ...

- **Introdução à Diagrama de classes**
 - **O que é um Diagrama de Classe?**
 - **Entendendo o Diagrama de Classe**
 - **Como Criar um Diagrama de Classe?**
 - **Relacionamentos no Diagrama de Classe**
-

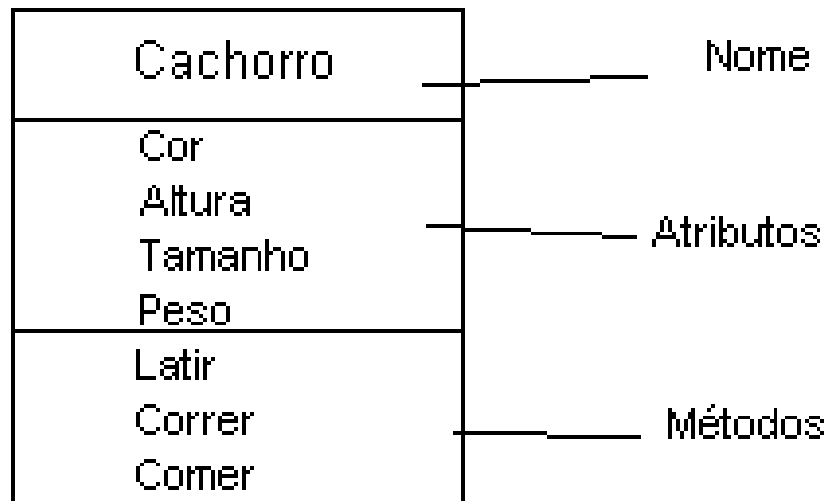
O que é um Diagrama de Classe?

- Em programação, um diagrama de classes é uma representação da estrutura e relações das classes
 - Eles são um tipo de diagrama da estrutura que modela suas **classes**, seus **atributos**, **operações** e relações entre objeto



Entendendo o Diagrama de Classe

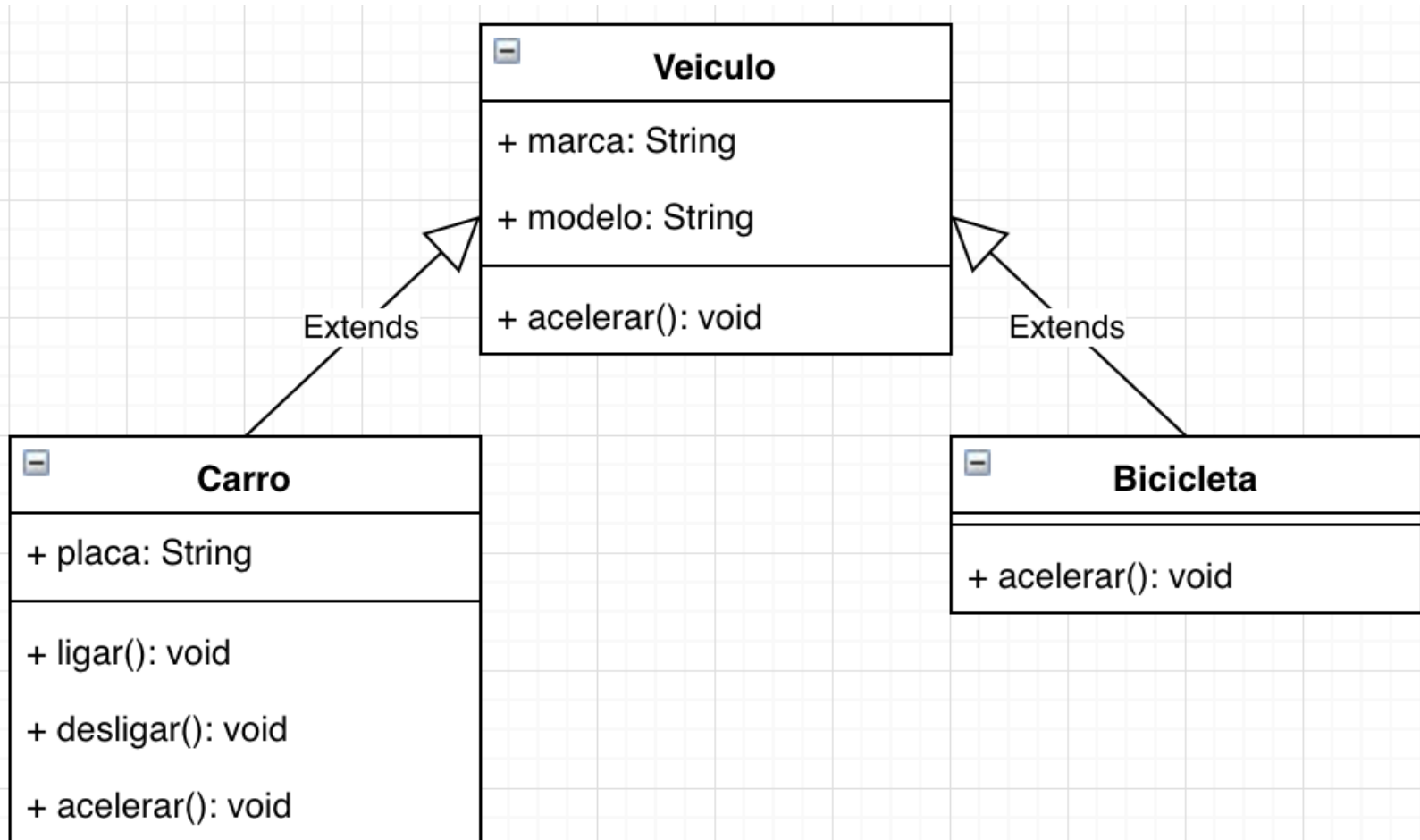
- Os diversos componentes em um diagrama de classes podem representar as classes que serão realmente programadas, os principais objetos ou as interações entre classes e objetos



Entendendo o Diagrama de Classe

- A forma de classe em si consiste em um retângulo com três linhas
 - A linha superior contém o nome da classe, a linha do meio, os atributos da classe e a linha inferior expressa os métodos ou operações que a classe pode utilizar
 - Classes e subclasses são agrupadas juntas para mostrar a relação estática entre cada objeto
-

Exemplo de Diagrama de Classe

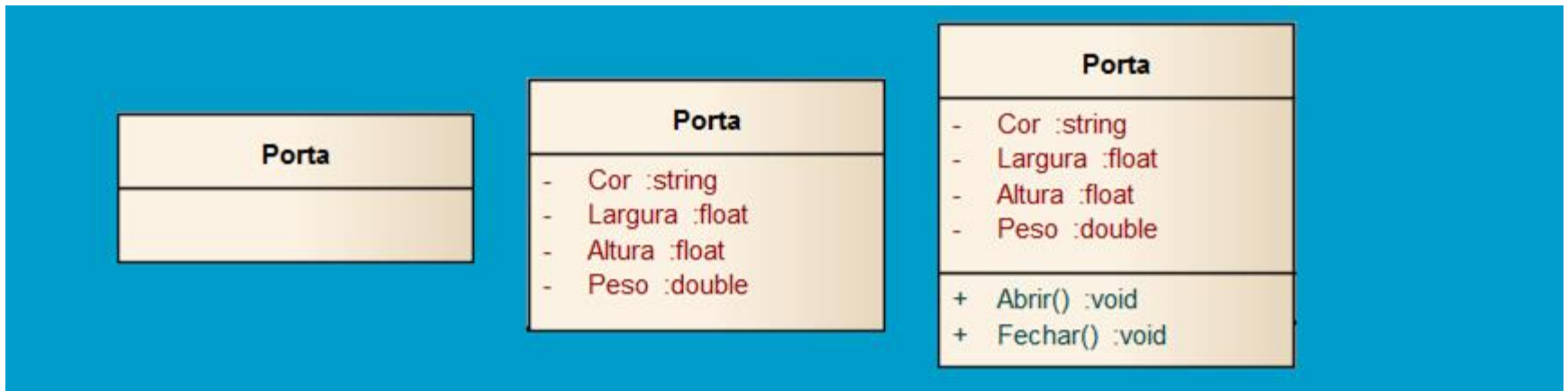


Benefícios do Diagrama de Classe

- Demonstra a estrutura estática das classes de um sistema
 - Expressam visualmente as necessidades e informações específicas de um sistema
 - Fornecem gráficos e uma descrição independente de implementação de tipos utilizados em um sistema
-

Exemplo de Criação de uma Classe no Diagrama

- O diagrama de classes padrão é composto de três partes:
 - Parte superior: contém o nome da classe
 - Parte do meio: contém os **atributos** da classe
 - Parte inferior: inclui as operações da classe (**métodos**)



Atributos em um Diagrama de Classe

- Os atributos representam o conjunto de características (estado) dos objetos daquela classe
 - Cada atributo possui uma **visibilidade**:
 - + público: visível em qualquer classe de qualquer pacote
 - # protegido: visível para classes do mesmo pacote
 - - privado: visível somente para classe
 - Também possui um **nome** que demonstra as características dos objetos e **tipo de dados** que são os mesmos tipos usados em Java: String, boolean, int, float, double, Date, etc...
-

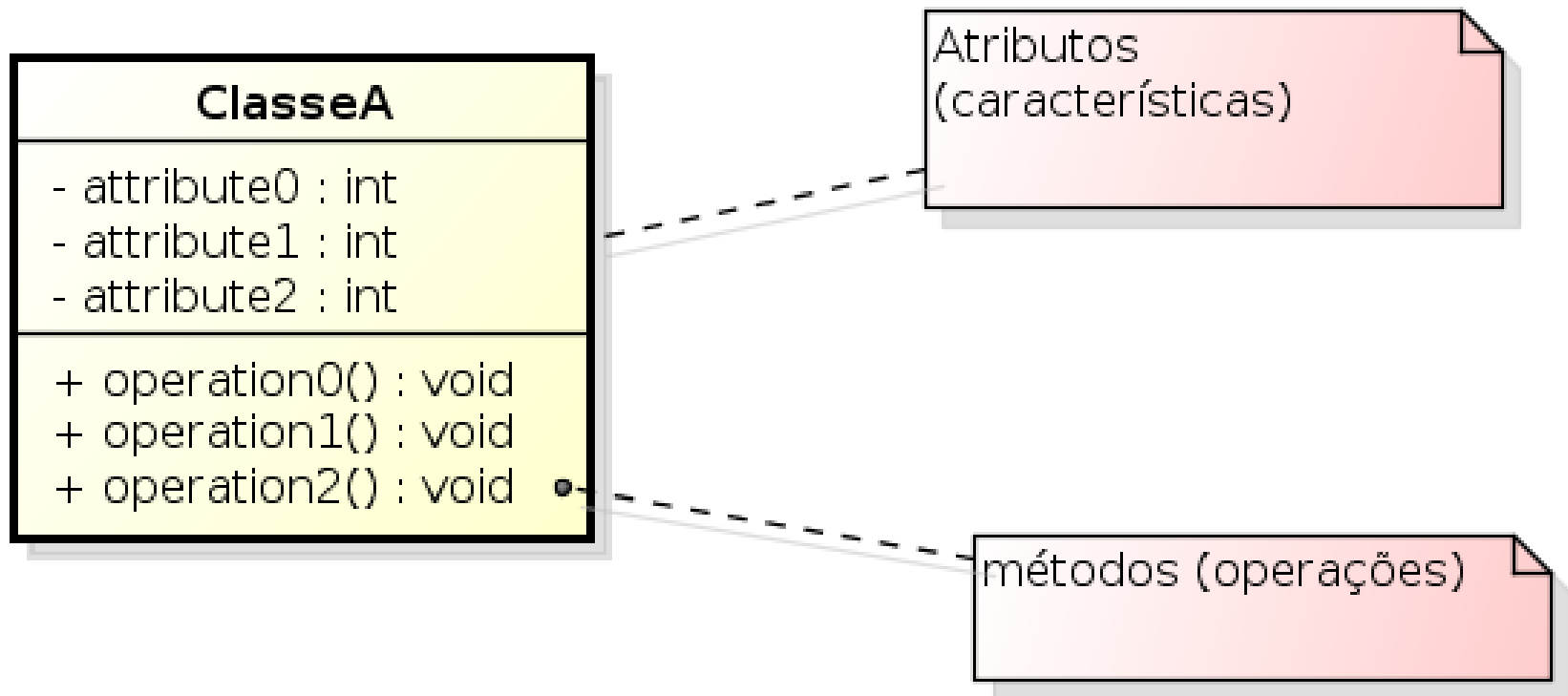
Métodos em um Diagrama de Classe

- Os métodos representam o conjunto de operações (comportamento) que a classe fornece
 - É exibido em formato de lista, cada operação ocupa sua própria linha
 - As operações descrevem como uma classe interage com dados
 - Cada método também possui sua **visibilidade** (+ público, # protegido e - privado)
 - Um **nome** do método deve expressar a ação que realiza, por exemplo incluirAluno() que não deve possuir espaços e nem começar com dígitos
-

Métodos em um Diagrama de Classe

- Também deve conter uma **lista de parâmetros** que deverá vir entre parênteses e separados por vírgula
 - E por fim, um **tipo de retorno** que informa que tipo de dado o método deverá retornar após a sua execução
 - Se o método não retornar nada, deverá ser usada a palavra “void” no tipo de retorno
-

Exemplo de Atributos e Métodos em um Diagrama de Classe



Como várias classes se associam?

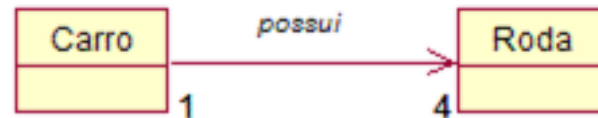


Os diversas relacionamentos e ligações que podem existir em diagramas de classes e objetos são chamadas de **Interações**

Relacionamentos no Diagrama de Classe

- Geralmente as classes não estão sós e se relacionam entre si
 - Possuem relacionamentos entre elas (para comunicação)
 - Compartilham informações
 - Colaboram umas com as outras

Descreve o relacionamento.



Relacionamentos no Diagrama de Classe

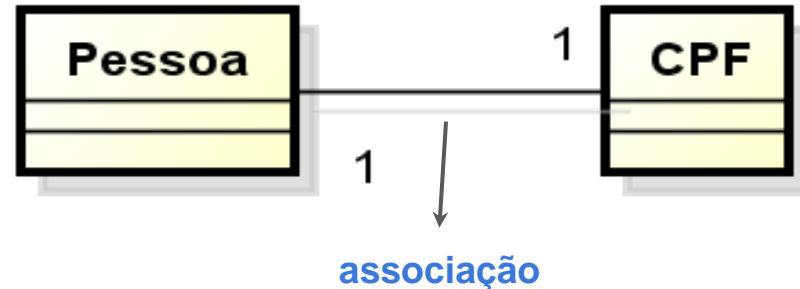
- Os relacionamentos possuem:
 - **Nome**: descrição dada ao relacionamento (faz, tem, possui,...)
 - **Sentido de leitura**
 - **Navegabilidade**: indicada por uma seta no fim do relacionamento
 - **Multiplicidade**: 0..1, 0..*, 1, 1..*, 2, 3..7
 - **Tipo**: associação (agregação, composição), generalização e dependência
 - **Papéis**: desempenhados por classes em um relacionamento
-

Relacionamentos no Diagrama de Classe

- Os relacionamentos ou interações no diagrama de classes são:
 - **Associação** (Agregação e Composição)
 - Define um relacionamento entre duas entidades conceituais do sistema
 - Ocorre quando uma classe possui atributos do tipo de outra classe
 - **Generalização**
 - Dependência
-

Associação

- Uma associação é um relacionamento estrutural que indica que os objetos de uma classe estão vinculados a objetos de outra classe
 - Descreve um vínculo entre duas classes
 - Determina que as instâncias de uma classe estão de alguma forma ligadas às instâncias da outra classe



Associação

- Cada associação possui indicadores de multiplicidade:

Multiplicidade	Significado
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
m..n	Valores específicos

Representação da Associação

- Uma associação é representada por uma linha sólida conectando duas classes
- Deve conter **multiplicidades** uma em cada extremo da linha de associação e **navegabilidade** ou direção de leitura que indica como a associação deve ser lida

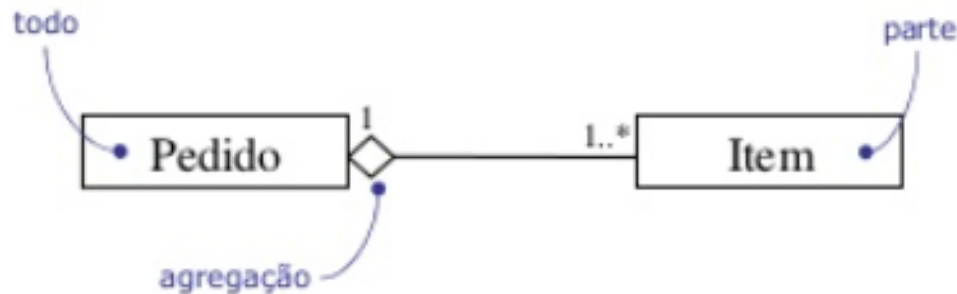


Agregação

- Agregação é uma associação em que **um objeto é parte de outro, de tal forma que a parte pode existir sem o todo**
 - Um objeto poderá agregar uma ou mais instâncias de um outro objeto
 - Uma agregação consiste de um objeto contendo referências para outros objetos, de tal forma que o primeiro seja o todo, e que os objetos referenciados sejam as partes do todo
-

Representação da Agregação

- Uma agregação é representada por losango na extremidade da classe que contém os objetos-todo
 - Um objeto “parte” pode fazer parte de vários objetos “todo”



Composição

- Uma composição é uma variação da agregação que tenta representar também uma relação todo - parte
 - Em uma composição um mesmo objeto-parte não pode se associar a mais de um objeto-pai
 - **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
-

Representação da Composição

- Uma composição é representada por losango preenchido na extremidade da classe que contém os objetos-todo

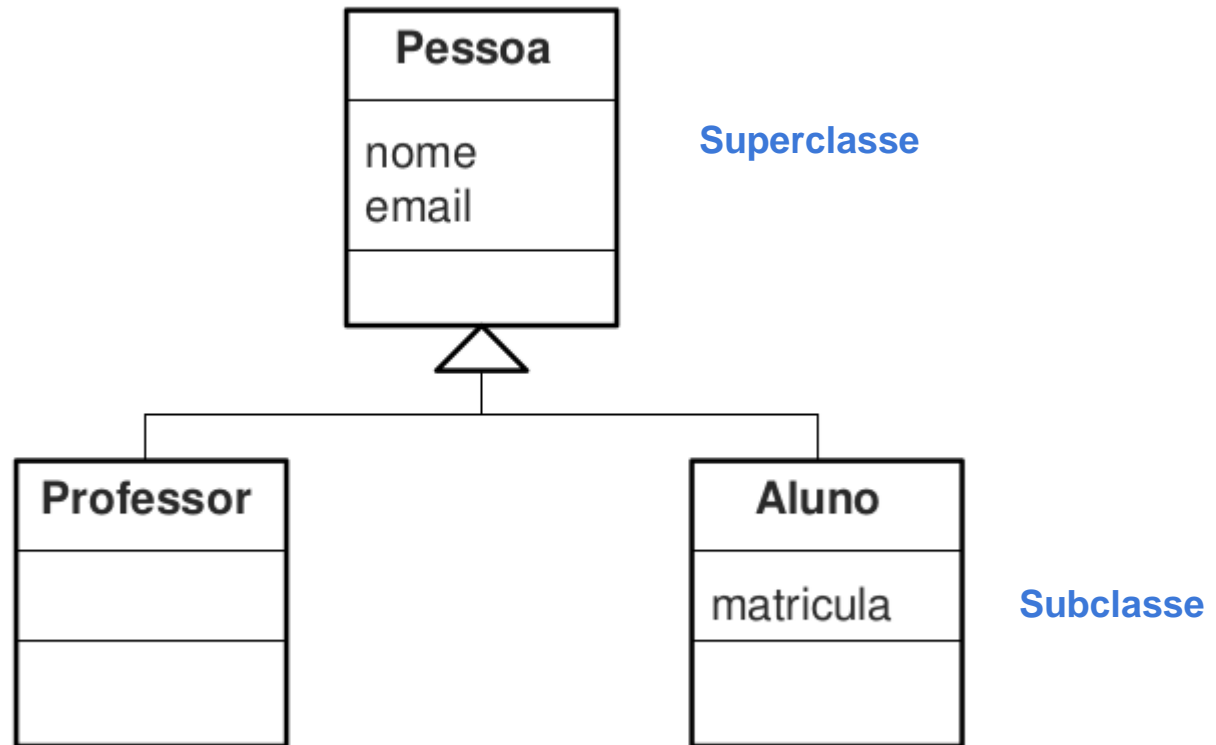


Generalização

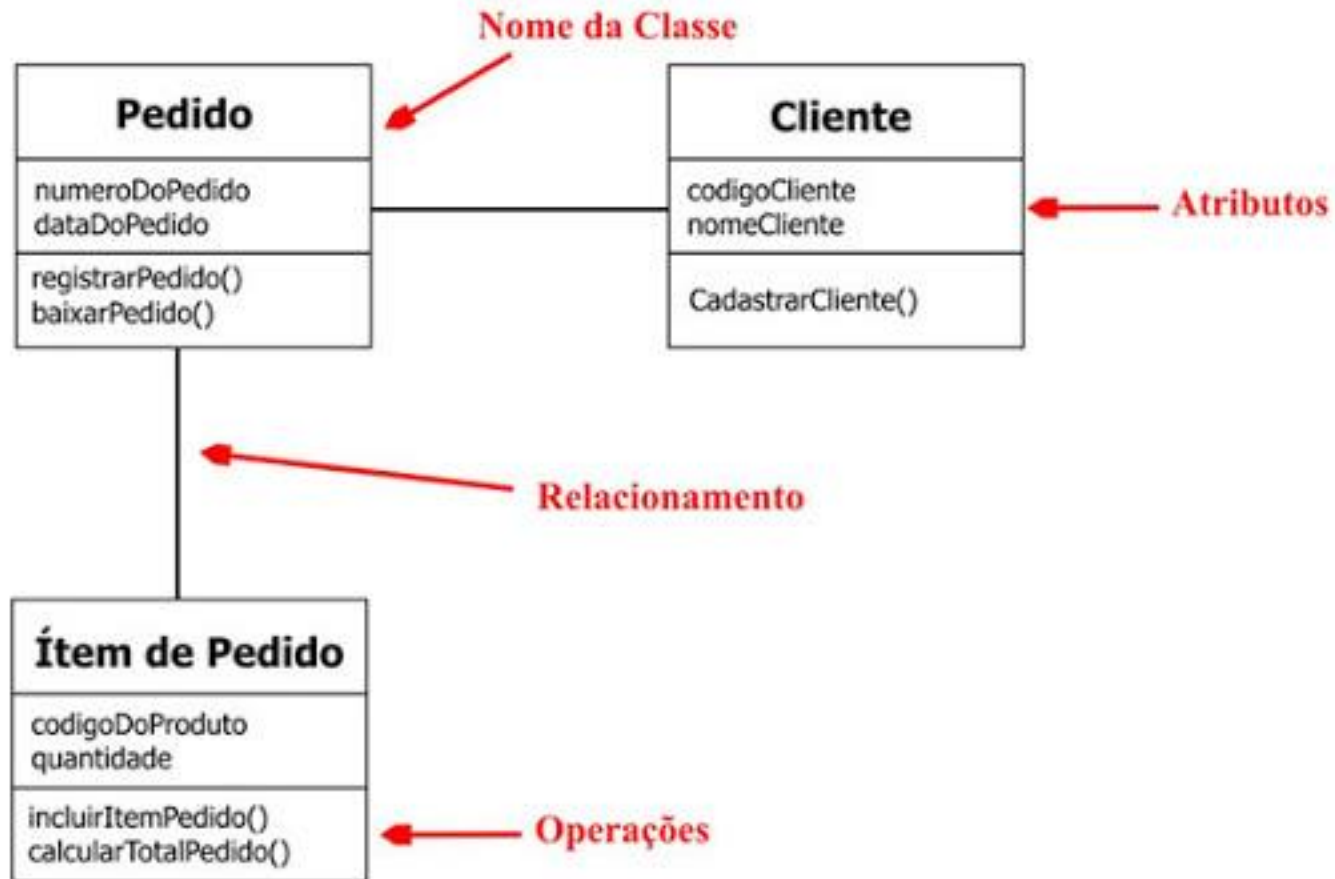
- É um relacionamento entre itens gerais (superclasses) e itens mais específicos (subclasses) “é um” “é um tipo de” superclasse
 - A generalização acontece quando existe a necessidade de criar uma classe **que herde as propriedades de outra classe**, isto é, os atributos, métodos e também relacionamentos (associações) da classe superior.
-

Representação da Generalização

- Uma generalização é representada por uma relação de herança



Visão Geral do Diagrama de Classe



Revisão dos Conceitos de POO

- Poo vs Estruturada
 - Classes, Objetos
 - Java
 - Tipos primitivos, comentários, operadores, array
 - Encapsulamento
 - Herança
 - Interface
 - Hierarquia de classes
 - Classes Abstratas
 - Sobrecarga
 - Polimorfismo
 - Exceções
 - Associação
-

Obrigado!

Por hoje é só pessoal...

Dúvidas?



rfbrkh3



ismaylesantos@great.ufc.br



@IsmayleSantos
