



Java Orientado a Objetos

RECODE
pro



Módulo 03 – Java Orientado a Objetos Aula 01



O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML

A UML- Unified Modeling Language ou Linguagem de Modelagem Unificada - é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. A UML é independente tanto de linguagem de programação quanto de processos de desenvolvimento. Isso quer dizer que a UML pode ser utilizada para a modelagem de sistemas, não importando qual a linguagem de programação a ser utilizada, ou qual a forma de desenvolvimento adotada.

Módulo 03 – Java Orientado a Objetos Aula 01



O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML

A UML possui diversos diagramas conforme lista abaixo, porém nesse tópico vamos explorar o diagrama de classes, pois é o assunto principal para nosso contexto da orientação a objetos.

1 - Diagramas estruturais

- Diagramas de classes
- Diagramas de objetos
- Diagramas de componentes
- Diagramas de implementação ou instalação
- Diagramas de pacotes
- Diagramas de estrutura composta
- Diagrama de perfil

2 - Diagrama comportamental

- Diagrama de caso de uso
- Diagrama de sequência
- Diagrama de colaboração
- Diagrama de transição de estados
- Diagrama de atividade

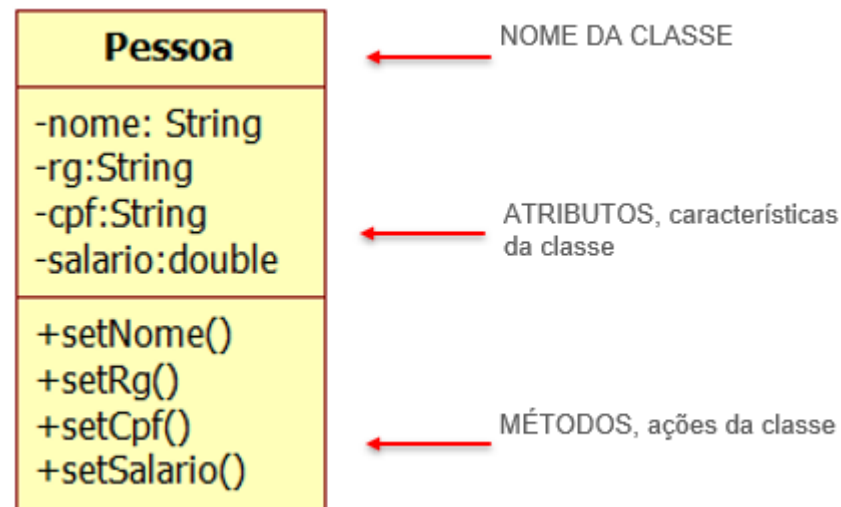


Módulo 03 – Java Orientado a Objetos Aula 01



Diagramas de classes

O Diagrama de Classes é utilizado para fazer a representação de estruturas de classes de negócio, interfaces e outros sistemas e classes de controle. Além disso, o diagrama de classes é considerado o mais importante para a UML, pois serve de apoio para a maioria dos demais diagramas. Em resumo um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos.



Módulo 03 – Java Orientado a Objetos Aula 01



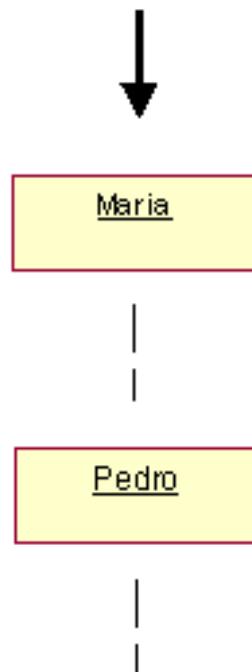
O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML

Classe



Objetos



Objeto:

É uma instância (Cópia) de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Exemplo de objetos da classe Pessoa: João, José, Maria, Antônia.

Módulo 03 – Java Orientado a Objetos Aula 01



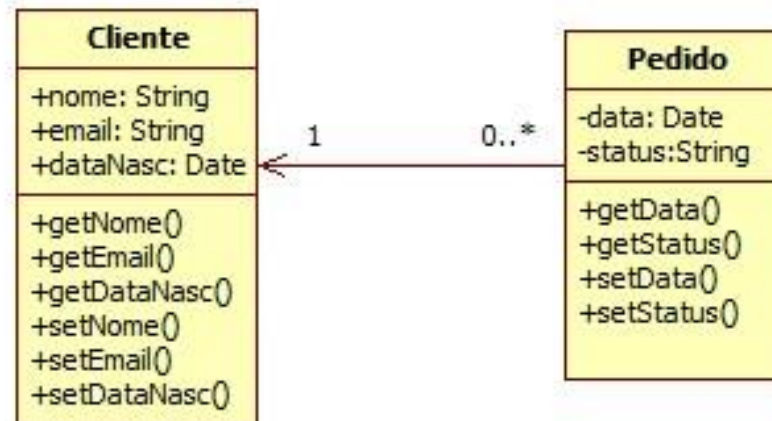
O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML

Associação Binária

Associações binárias ocorrem quando são identificados relacionamentos entre objetos de duas classes distintas. Esse tipo de associação é, em geral, a mais comum encontrada.

Exemplo de associação com a classe **cliente** e **pedido**, nesse modelo um objeto da classe cliente pode se relacionar com nenhum ou vários objetos da classe pedido (0..*) e um objeto da classe pedido pode se relacionar com apenas um objeto da classe cliente (1).



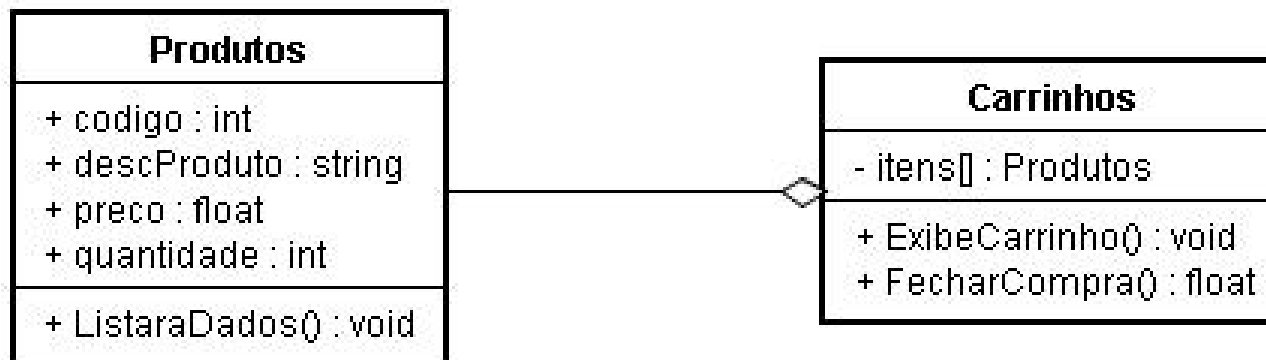
Módulo 03 – Java Orientado a Objetos Aula 01



Uma visão de Classes com diagramas UML

Agregação

É um tipo especial de associação onde tenta-se demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementados pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte); conhecemos como todo/parte. O objeto-pai poderá usar as informações do objeto agregado. Nesta relação, um objeto poderá agregar uma ou mais instâncias de um outro objeto. Para agregar muitas instâncias, a forma mais simples é utilizando arrays. Criamos um array como atributo da classe, sendo que o papel deste array é armazenar inúmeras instâncias de uma outra classe"



Módulo 03 – Java Orientado a Objetos Aula 01

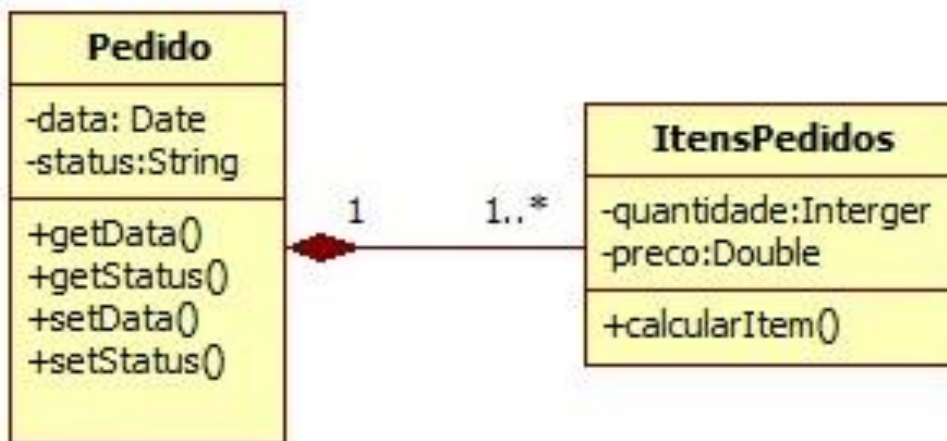


Uma visão de Classes com diagramas UML

Composição

Uma associação do tipo composição constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de estar associados a um único objeto-todo. Em uma composição os objetos-parte não podem ser destruídos por um objeto diferente do objeto-todo ao qual estão relacionados. O símbolo de composição diferencia-se graficamente do símbolo de agregação por utilizar um losango preenchido.

Modelo UML: um ou vários itens (1..*) compõe um pedido e um item do pedido (1) só pode estar naquele pedido.



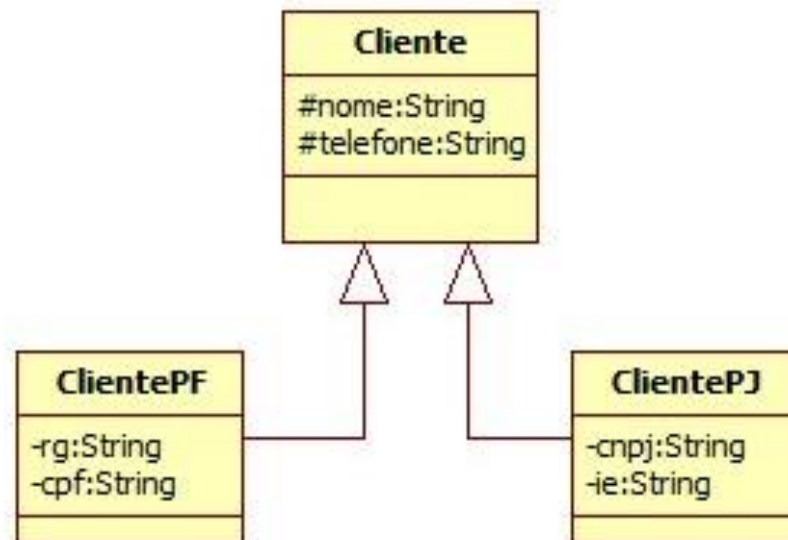
Módulo 03 – Java Orientado a Objetos Aula 01



Uma visão de Classes com diagramas UML

Herança

Herança (ou generalização) é o mecanismo pelo qual uma classe (subclasse) pode estender outra classe (superclasse), aproveitando seus comportamentos (métodos) e variáveis possíveis (atributos). Um exemplo de herança: Mamífero é superclasse de Humano. Ou seja, um Humano é um mamífero



Módulo 03 – Java Orientado a Objetos Aula 01



Uma visão de Classes com diagramas UML

A **multiplicidade** nada mais é do que os limites inferior e superior da quantidade de objetos aos quais outro objeto está associado. Esses limites podem receber os valores: apenas um; zero ou muitos; um ou muitos; zero ou um; ou até um intervalo específico

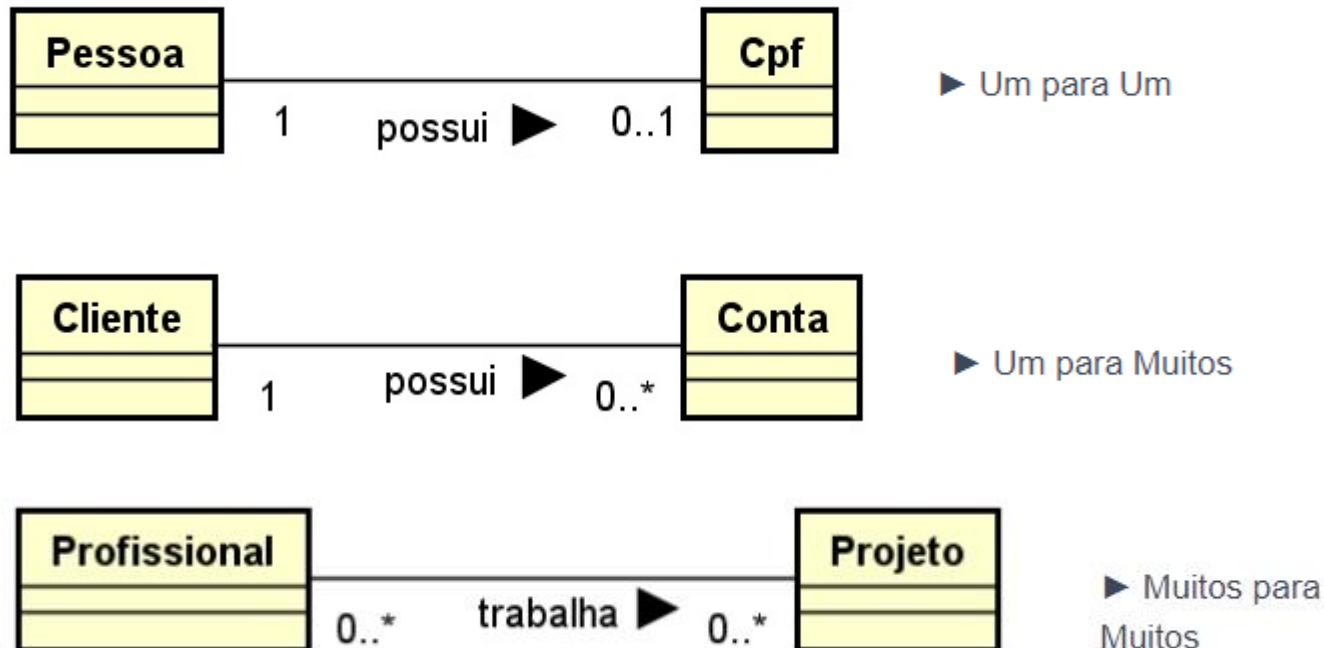
<i>Tipos</i>	<i>Significa</i>
0..1	Zero ou uma instância. A notação n..m indica n para m instâncias.
0..* ou *	Não existe limite para o número de instâncias.
1	Exatamente uma instância.
1..*	Ao menos uma instância.

Módulo 03 – Java Orientado a Objetos Aula 01



Uma visão de Classes com diagramas UML

Multiplicidade

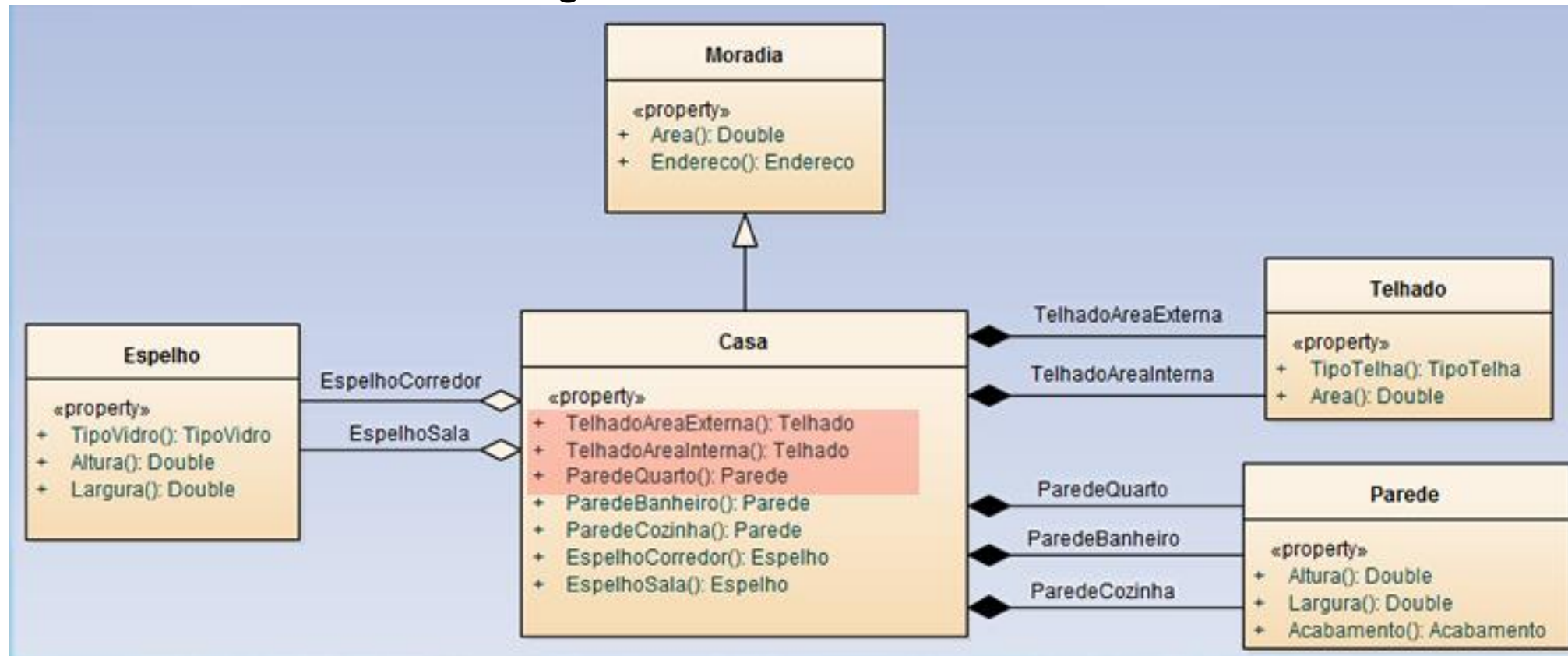


Módulo 03 – Java Orientado a Objetos Aula 01



O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML



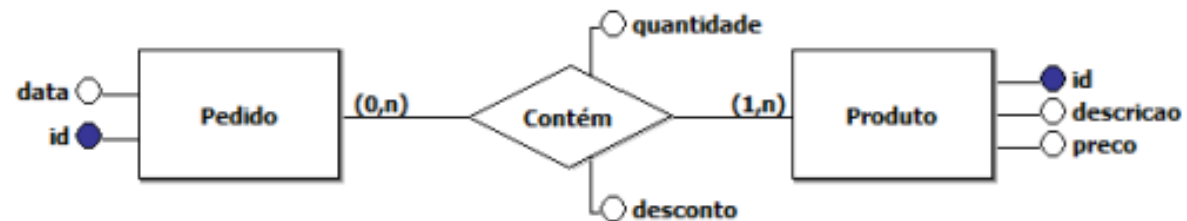
Módulo 03 – Java Orientado a Objetos Aula 01



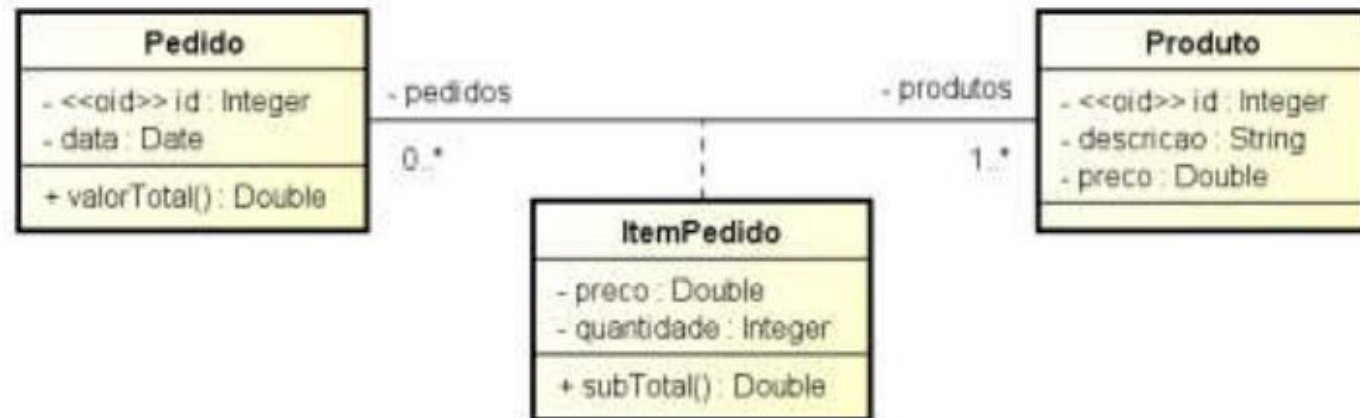
O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML

Móveis Angular					
Avenida Vale do Silício, 751, Bairro Estudante Uberlândia-MG					
NOTA FISCAL					
Pedido n°:	1001				
Data	25/06/2017				
Detalhes do pedido:					
Produto	Descrição	Preço unitário	Quantidade	Desconto	Subtotal
8021	Cadeira simples	400.00	4	0%	1600.00
8055	Mesa retangular	1500.00	1	10%	1350.00
8014	Estante	2000.00	1	5%	1900.00
Total do pedido:					4850.00



OU:

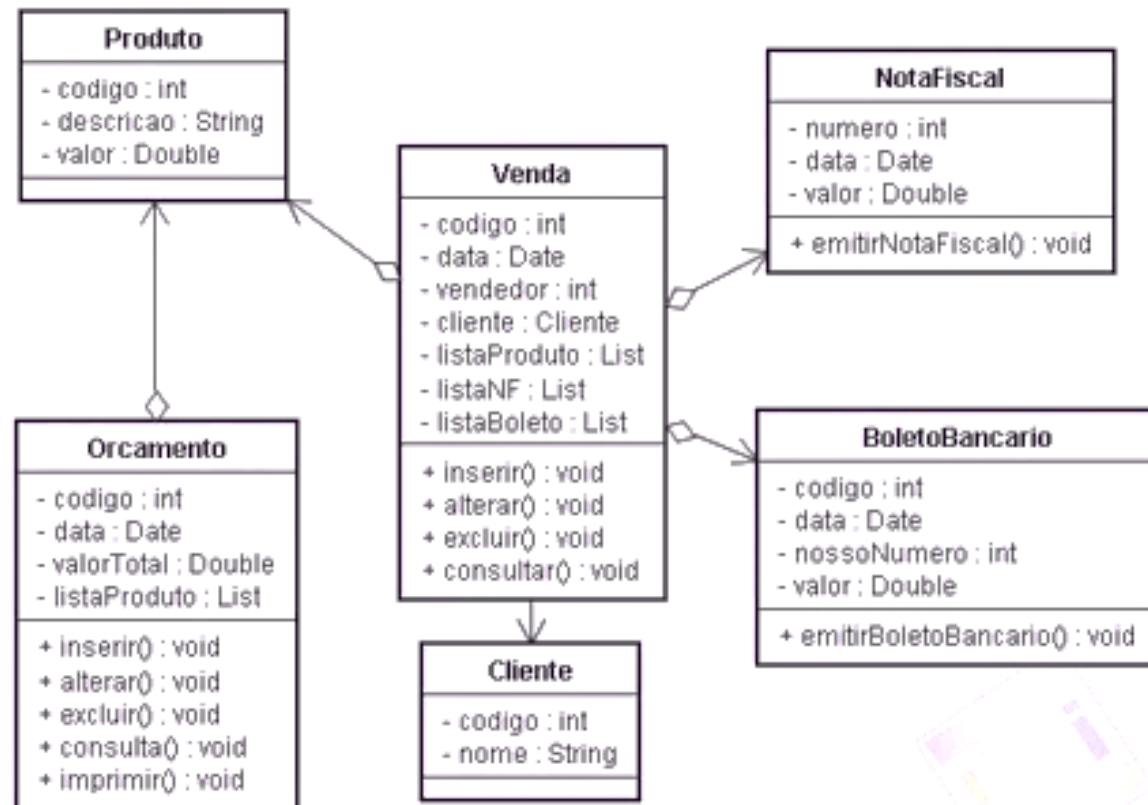


Módulo 03 – Java Orientado a Objetos Aula 01



O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML

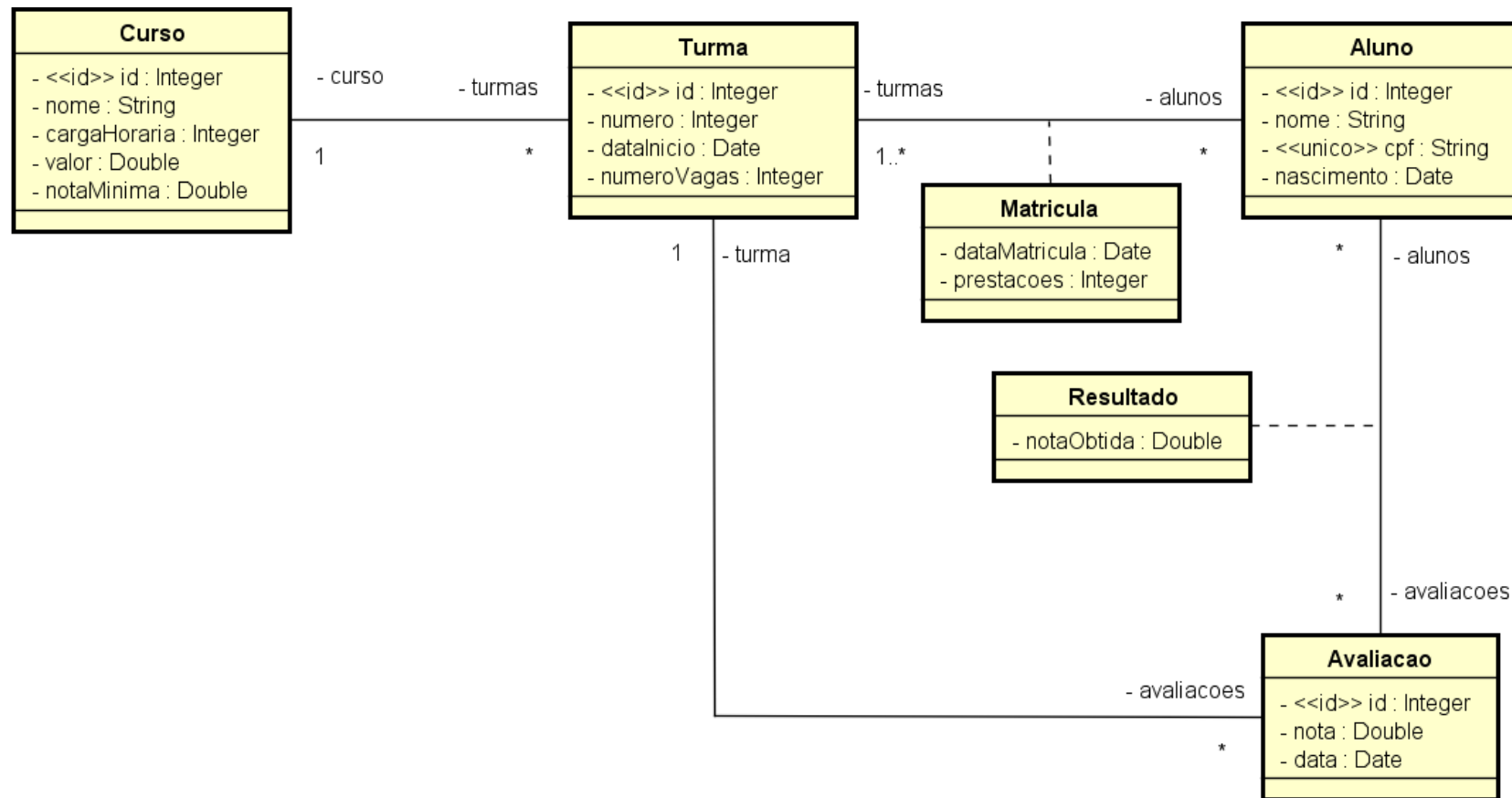


Módulo 03 – Java Orientado a Objetos Aula 01



O paradigma da Orientação a Objeto

Uma visão de Classes com diagramas UML



Módulo 03 – Java Orientado a Objetos Aula 01



Exercício 01

Desenvolver o seguinte diagrama de classe: Ler os dados de um produto em estoque (nome, preço e quantidade no estoque). Em seguida:

- Mostrar os dados do produto (nome, preço, quantidade no estoque, valor total no estoque)
- Realizar uma entrada no estoque e mostrar novamente os dados do produto
- Realizar uma saída no estoque e mostrar novamente os dados do produto.

Módulo 03 – Java Orientado a Objetos Aula 01



Resolução possível

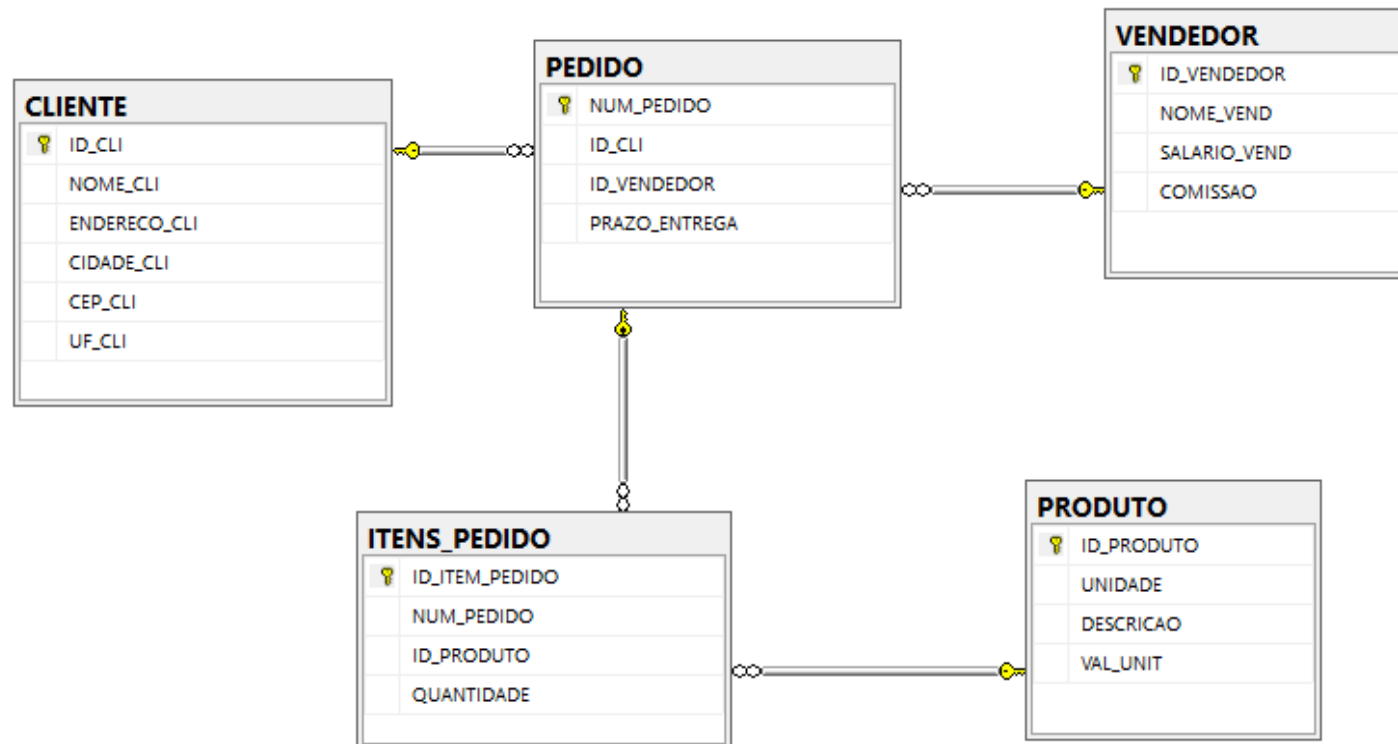
Product
<ul style="list-style-type: none">- Name : string- Price : double- Quantity : int
<ul style="list-style-type: none">+ TotalValueInStock() : double+ AddProducts(quantity : int) : void+ RemoveProducts(quantity : int) : void

Módulo 03 – Java Orientado a Objetos Aula 01



Exercício 02

Desenvolver o diagrama de classe para o modelo de entidade e relacionamento abaixo, criar os métodos que achar necessário e suas associações.



Módulo 03 – Java Orientado a Objetos Aula 01



Exercício 03 – Criar o diagrama de classe para o problema abaixo.

Uma biblioteca deseja fazer o registro de seus empréstimos de livros. Quando um usuário pega um livro emprestado, deve ser registrada a data de empréstimo. Por padrão, o prazo de empréstimo é de dois dias, considerando atraso se o livro for devolvido depois deste tempo. Cada livro possui um título, gênero, editora e número de páginas. Um livro pode participar de uma coleção. Cada livro também possui um valor diário de multa, caso o usuário devolva o livro com atraso em relação à data prevista de devolução.



RECODE



www.recode.org.br

RECODE
pro

recodepro.org.br

Institucional



[/rederecode](#)



[/recoderede](#)