



Formação Desenvolvedor Moderno Módulo: Banco de Dados

Capítulo: Modelo conceitual - nível de análise

<https://devsuperior.com.br>

1

Sinônimos

- Modelo de dados/domínio em nível conceitual
- Modelo de dados/domínio em nível de análise
- Modelo conceitual

2

Modelo Conceitual

Conceitos e atributos

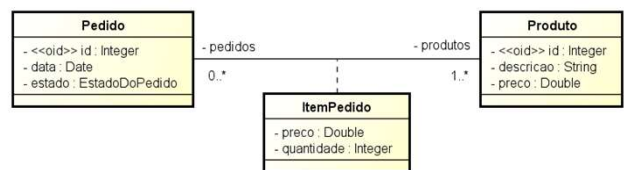
3

Agenda

- Definição de modelo conceitual
- Conceitos
- Atributos
- Representação UML de conceitos e atributos

4

Modelo Conceitual



- **Definição 1:** é um modelo que descreve a estrutura das informações que o sistema vai gerenciar (Wazlawick)

- **Definição 2:** é o Modelo de Domínio em nível de Análise:

- Pertence ao escopo do problema e não ao escopo da solução
- Independente de paradigma
- Independente de tecnologia

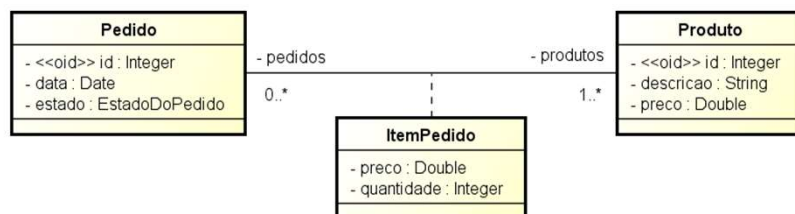
*Modelo de domínio:
modelo que descreve as
entidades do domínio,
bem como as inter-
relações entre elas.*

- Para representar o Modelo Conceitual, vamos utilizar a ferramenta:
 - Diagrama de Classes da UML

5

O Modelo Conceitual descreve:

- Conceitos
- Atributos
- Associações



6

Conceitos

- Um conceito pode ser qualquer entidade que tenha um **significado** para o sistema e que tenha uma necessidade de **armazenamento de dados**.
 - Exemplos: cliente, pedido, produto, fornecedor, etc.
- Um conceito deve ser uma **unidade coesa**.

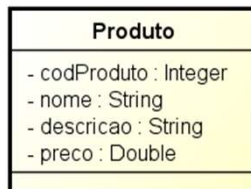
Não se deve misturar informações de várias coisas distintas em um mesmo conceito



Atributos

- Informações alfanuméricas simples, como números, textos, datas, etc. contidas em cada conceito.
 - Produto: descrição, preço
 - Cliente: nome, email, telefone, CPF, dataNascimento
- Notas (1FN):
 - Não pode ser multivalorado
 - RUIM: telefones ("3736-3938, 9988-3346, 3210-3939")
 - Não pode ser composto
 - RUIM: endereço ("Rua Floriano Peixoto, n° 250, apto 302, Bairro Copacabana, CEP 38410-384")
 - BOM: logradouro, numero, complemento, bairro, cep

Usando diagrama de classes da UML para representar conceitos e atributos



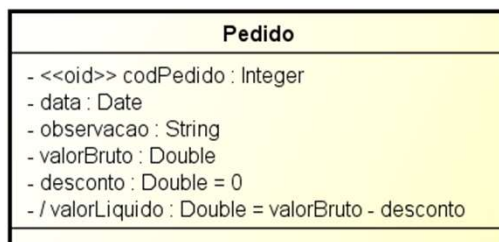
Embora usamos os tipos da linguagem Java (por causa da ferramenta CASE usada), modelagem conceitual é uma modelagem essencial

REGRAS BÁSICAS:

- Um conceito é representado por um retângulo dividido em três seções
- A primeira seção contém o nome do conceito
- A segunda seção contém os atributos
- Cada atributo é representado por:
 - nome : tipo
- O tipo é opcional no MC
- A terceira seção não é usada no MC

9

Usando diagrama de classes da UML para representar conceitos e atributos



OUTRAS REGRAS:

- Atributo identificador: estereótipo <<oid>>
- Pode haver valor inicial
- Pode haver atributos derivados (read only)

10

Resumo da aula

- O que é modelo conceitual
 - Modelo que descreve a estrutura das informações gerenciadas pelo sistema
 - Modelo de domínio em nível de análise
 - Pertence ao escopo do problema
- Conceitos
 - algo que tenha significado para o negócio e necessidade de armazenamento
 - unidade coesa do negócio
- Atributos
 - informações alfanuméricas simples
 - não pode ser multivalorado (1FN)
 - não pode ser composto (1FN)
- Representação de conceitos e atributos com diagrama de classes da UML
 - nome : tipo
 - Atributo identificador
 - Valor inicial
 - Atributo derivado

Pedido
- <<oid>> codPedido : Integer
- data : Date
- observacao : String
- valorBruto : Double
- desconto : Double = 0
- / valorLiquido : Double = valorBruto - desconto

Como identificar conceitos

Agenda

- Onde buscar informações
- Orientações para identificação de conceitos
- Exemplos

13

Onde buscar informações

Analista de
sistemas



negócio
+
requisitos do
sistema

DEVE-SE BUSCAR INFORMAÇÕES EM:

- Documentos produzidos pela análise de requisitos do sistema:
 - Visão geral do sistema
 - Casos de uso
- Processos de negócio
- Regulamentos / normas / leis
- Documentos de registro
- Papéis e atribuições
- Práticas e ritos estabelecidos
- Conhecimento tácito (entrevistas)
- Outros



14

Visão geral do sistema: documento de formato livre que especifica, em linhas gerais, os requisitos do sistema.

Sistema de Controle Escolar

Deseja-se construir um sistema acadêmico. Para isso, são registrados os cursos disponíveis, onde cada um possui um nome, carga horária e valor. Quando um curso vai ser oferecido, é registrada uma turma, informando os seguintes dados: número da turma, data de início e número de vagas. Uma matrícula de um aluno em uma turma consiste na data de matrícula e no número de prestações em que o aluno vai pagar o curso. Para cada aluno, é necessário cadastrar seu nome, cpf, e data de nascimento.

Cada aluno passa por várias avaliações durante o desenrolar do curso que está cursando. Uma avaliação possui nota e data. Depois que a avaliação ocorre, é registrado resultado de cada aluno da turma (a nota que ele tirou). Um aluno é aprovado em um curso se sua nota total for pelo menos 70% da nota prevista do curso.

É importante saber a porcentagem de aprovação por turma e por curso (considerando somente as turmas que já finalizadas). Deseja-se saber também a nota final de um aluno em um curso que ele cursou, e se ele foi aprovado ou não no curso. Além disso, o sistema deve ser capaz de saber os alunos aprovados e reprovados em uma turma, bem como o aluno com melhor desempenho da turma (pode haver empates).

15

Caso de uso: documento estruturado que especifica uma funcionalidade do sistema por meio da **troca de informações** entre usuários (atores de sistema) e o sistema.

Comprar Livros

Cenário principal

1. [IN] O **comprador** informa sua identificação.
2. [OUT] O **sistema** informa os livros disponíveis para venda (título, capa e preço) e o conteúdo atual do carrinho de compras.
3. [IN] O **comprador** seleciona os livros que deseja comprar.
4. O **comprador** decide se finaliza a compra ou se guarda o carrinho:
 - 4.1 Variante: Finalizar a compra.
 - 4.2 Variante: Guardar carrinho.

Variante 4.1: Finalizar a compra

- 4.1.1. [OUT] O **sistema** informa o valor total dos livros e apresenta as opções de endereço cadastradas.
- 4.1.2. [IN] O **comprador** seleciona um endereço para entrega.
- 4.1.3. [OUT] O **sistema** informa o valor de frete e total geral, bem como a lista de cartões de crédito já cadastrados para pagamento.
- 4.1.4. [IN] O **comprador** seleciona um cartão de crédito.
- 4.1.5. [OUT] O **sistema** envia os dados do cartão e valor da venda para a operadora.

- 4.1.6. [IN] A **operadora** informa o código de autorização.
- 4.1.7. [OUT] O **sistema** informa o prazo de entrega.

Variante 4.1: Guardar carrinho

- 4.2.1. [OUT] O **sistema** informa o prazo (dias) em que o carrinho será mantido.

Exceção 1a: Comprador não cadastrado

- 1a.1 [IN] O **comprador** informa seu CPF, nome, endereço e telefone.

Exceção 4.1.2a: Endereço consta como inválido

- 4.1.2a.1 [IN] O **comprador** atualiza o endereço.
Vai para 4.1.2.

Exceção 4.1.6a: A operadora não autoriza a venda

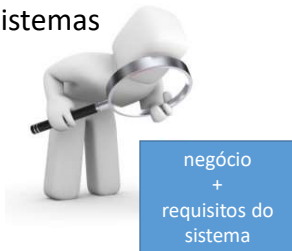
- 4.1.6a.1 [OUT] O **sistema** apresenta outras opções de cartão ao comprador.
- 4.1.6a.2 [IN] O **comprador** seleciona outro cartão.
Vai para 4.1.5.

(Wazlawick, 2011)

16

Orientações para identificação de conceitos

Analista de
sistemas



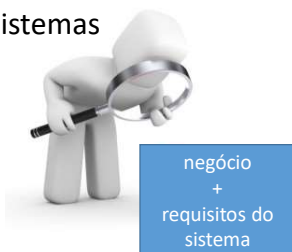
Com base nas informações, identificar conceitos relevantes para o negócio e com necessidade de armazenamento.

- Documentos (ordem de serviço, orçamento)
- Pessoas (cliente, fornecedor)
- Estruturas organizacionais (departamento)
- Eventos (venda, reserva, atendimento)

17

Orientações para identificação de conceitos

Analista de
sistemas



Atenção! Procure por:

Substantivos (pessoa, compra, produto, pagamento)

Expressões que denotem substantivos (autorização de pagamento)

Verbos que indiquem um possível conceito (comprar, pagar)

18

Exemplo 1 (especificação estilo "visão geral do sistema")

Deseja-se fazer um sistema para manter um cadastro dos funcionários de uma empresa. Deseja-se poder consultar o email e salário dos funcionários, bem como o telefone de seu departamento.

Funcionario
- <<oid>> codFuncionario : Integer
- nome : String
- email : String
- salario : Double
- departamento : String
- telefoneDepartamento : String

ERRADO

Funcionario
- <<oid>> codFuncionario : Integer
- nome : String
- email : String
- salario : Double

Departamento
- <<oid>> codDepartamento : Integer
- nome : String
- telefone : String

CORRETO

19

Exemplo 2 ("caso de uso")

Comprar Livros

Cenário principal

1. [IN] O **comprador** informa sua identificação.
2. [OUT] O **sistema** informa os livros disponíveis para venda (título, capa e preço) e o conteúdo atual do carrinho de compras.
3. [IN] O **comprador** seleciona os livros que deseja comprar.
4. O **comprador** decide se finaliza a compra ou se guarda o carrinho:
 - 4.1 Variante: Finalizar a compra.
 - 4.2 Variante: Guardar carrinho.

Variante 4.1: Finalizar a compra

- 4.1.1. [OUT] O **sistema** informa o valor total dos livros e apresenta as opções de endereço cadastradas.
- 4.1.2. [IN] O **comprador** seleciona um endereço para entrega.
- 4.1.3. [OUT] O **sistema** informa o valor de frete e total geral, bem como a lista de cartões de crédito já cadastrados para pagamento.
- 4.1.4. [IN] O **comprador** seleciona um cartão de crédito.
- 4.1.5. [OUT] O **sistema** envia os dados do cartão e valor da venda para a operadora.

- 4.1.6. [IN] A **operadora** informa o código de autorização.
- 4.1.7. [OUT] O **sistema** informa o prazo de entrega.

Variante 4.1: Guardar carrinho

- 4.2.1. [OUT] O **sistema** informa o prazo (dias) em que o carrinho será mantido.

Exceção 1a: Comprador não cadastrado

- 1a.1 [IN] O **comprador** informa seu CPF, nome, endereço e telefone.

Exceção 4.1.2a: Endereço consta como inválido

- 4.1.2a.1 [IN] O **comprador** atualiza o endereço.
- Vai para 4.1.2.

Exceção 4.1.6a: A operadora não autoriza a venda

- 4.1.6a.1 [OUT] O **sistema** apresenta outras opções de cartão ao comprador.
 - 4.1.6a.2 [IN] O **comprador** seleciona outro cartão.
- Vai para 4.1.5.

20

Exemplo 2 ("caso de uso")

Comprar Livros

Cenário principal

1. [IN] O **comprador** informa sua identificação.
2. [OUT] O **sistema** informa os livros disponíveis para venda (título, capa e preço) e o conteúdo atual do carrinho de compras.
3. [IN] O **comprador** seleciona os livros que deseja comprar.
4. O **comprador** decide se finaliza a compra ou se guarda o carrinho:
 - 4.1 Variante: Finalizar a compra.
 - 4.2 Variante: Guardar carrinho.

Variante 4.1: Finalizar a compra

- 4.1.1. [OUT] O **sistema** informa o valor total dos livros e apresenta as opções de endereço cadastradas.
- 4.1.2. [IN] O **comprador** seleciona um endereço para entrega.
- 4.1.3. [OUT] O **sistema** informa o valor de frete e total geral, bem como a lista de cartões de crédito já cadastrados para pagamento.
- 4.1.4. [IN] O **comprador** seleciona um cartão de crédito.
- 4.1.5. [OUT] O **sistema** envia os dados do cartão e valor da venda para a operadora.

4.1.6. [IN] A **operadora** informa o código de autorização.

4.1.7. [OUT] O **sistema** informa o prazo de entrega.

Variante 4.1: Guardar carrinho

4.2.1. [OUT] O **sistema** informa o prazo (dias) em que o carrinho será mantido.

Exceção 1a: Comprador não cadastrado

1a.1 [IN] O **comprador** informa seu CPF, nome, endereço e telefone.

Exceção 4.1.2a: Endereço consta como inválido

4.1.2a.1 [IN] O **comprador** atualiza o endereço.

Vai para 4.1.2.

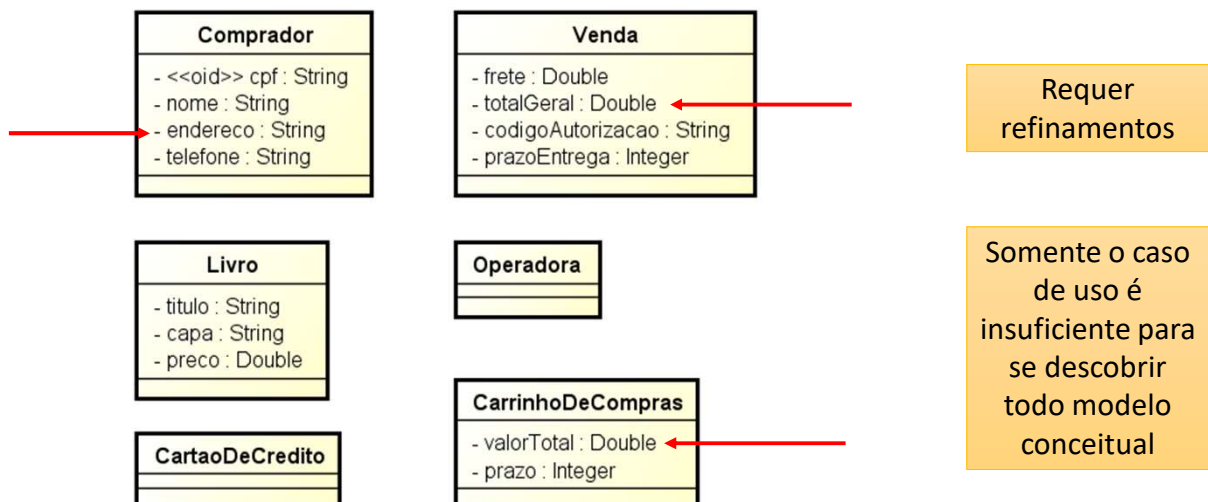
Exceção 4.1.6a: A operadora não autoriza a venda

4.1.6a.1 [OUT] O **sistema** apresenta outras opções de cartão ao comprador.

4.1.6a.2 [IN] O **comprador** seleciona outro cartão.

Vai para 4.1.5.

21



22

Resumo da aula

- Onde buscar informações
 - Documentos da análise de requisitos
 - Visão geral do sistema
 - Casos de uso
 - Processos, regulamentos, entrevistas, etc.
- Orientação:
 - Procurar conceitos com necessidade de armazenamento
 - Conceitos são substantivos
- Exemplos

23

Associações

24

Agenda

- Instâncias
- O que são associações
- Exemplo inicial

25

Instâncias

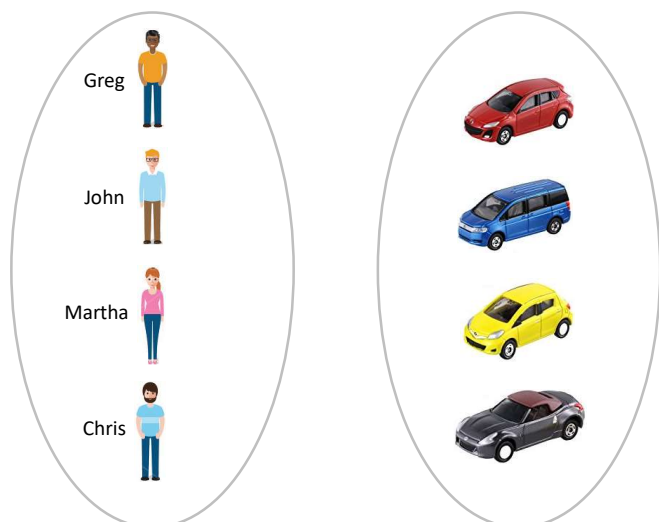
Exemplo:

Desejo criar um sistema para armazenar informações de pessoas e carros.

Conceitos:

- Pessoa
- Carro

Cada ocorrência dos meus conceitos recebe o nome de INSTÂNCIA ou OBJETO



26

O que são associações

Associação é um relacionamento estático entre dois conceitos.

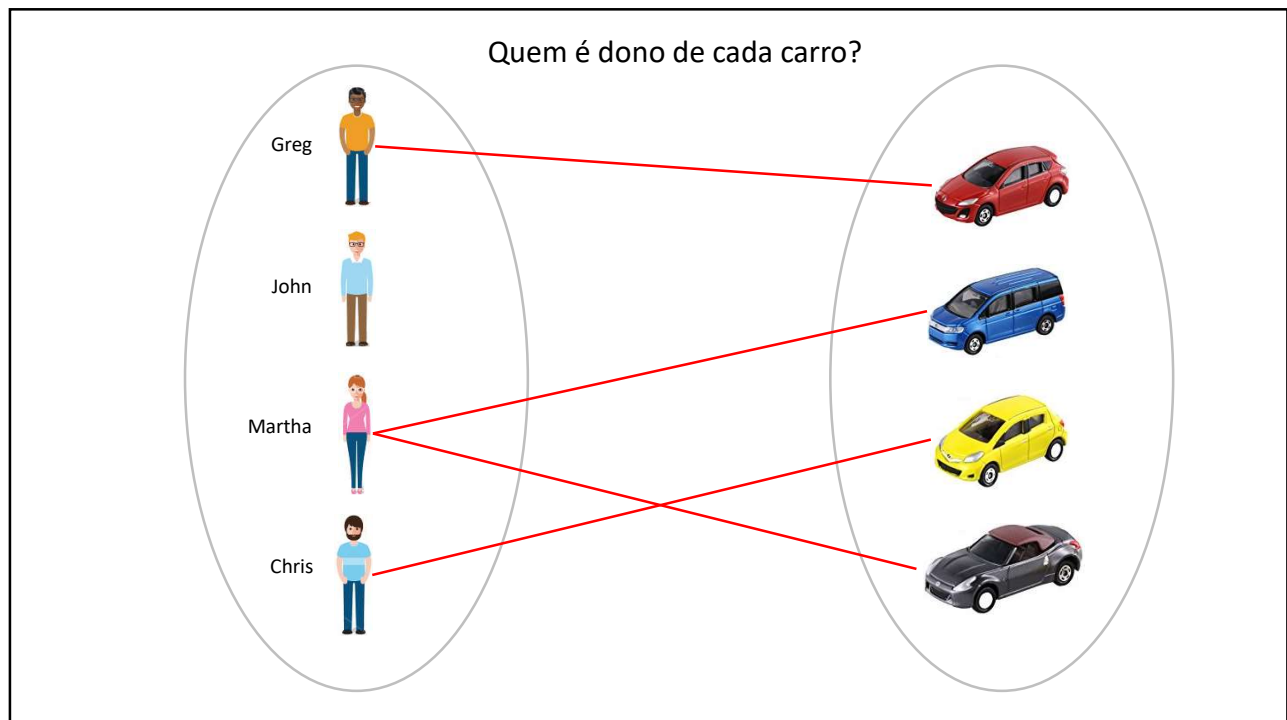
Exemplo:

Desejo criar um sistema para armazenar informações de pessoas e carros.

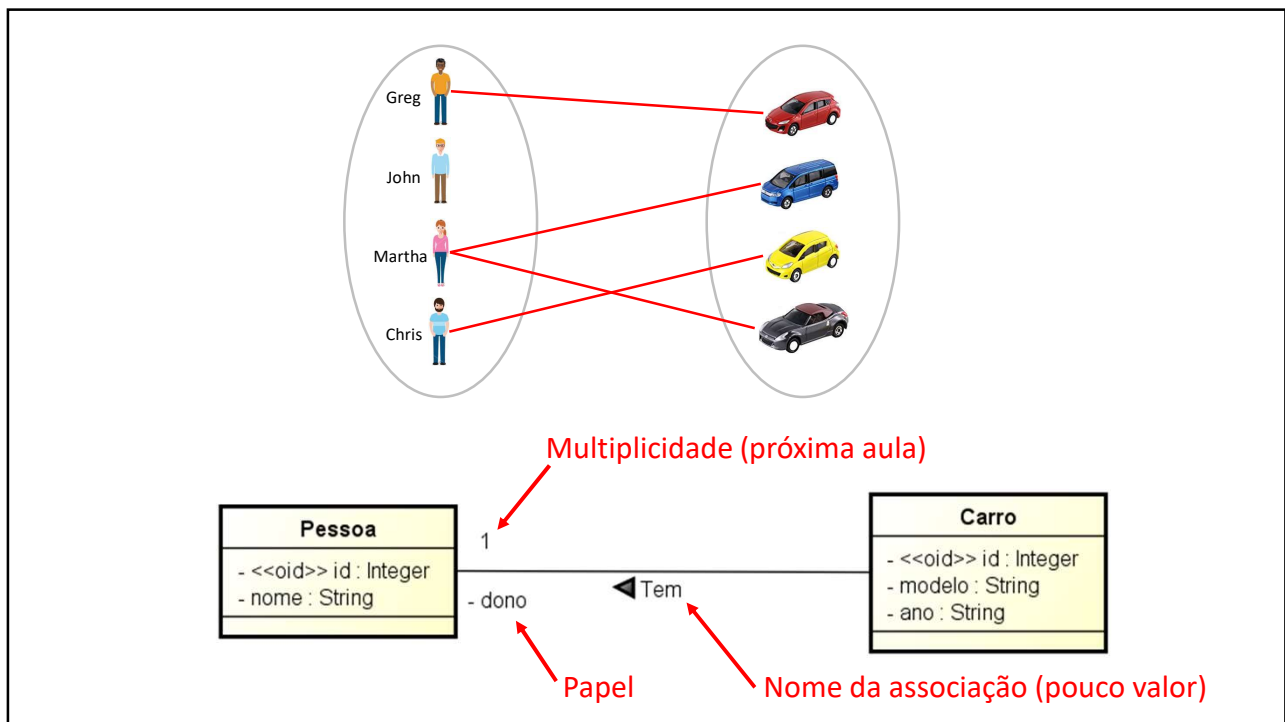
Mas eu não tenho simplesmente a necessidade de saber quais são as pessoas e quais são os carros:

Também desejo saber quem é o dono de cada carro!

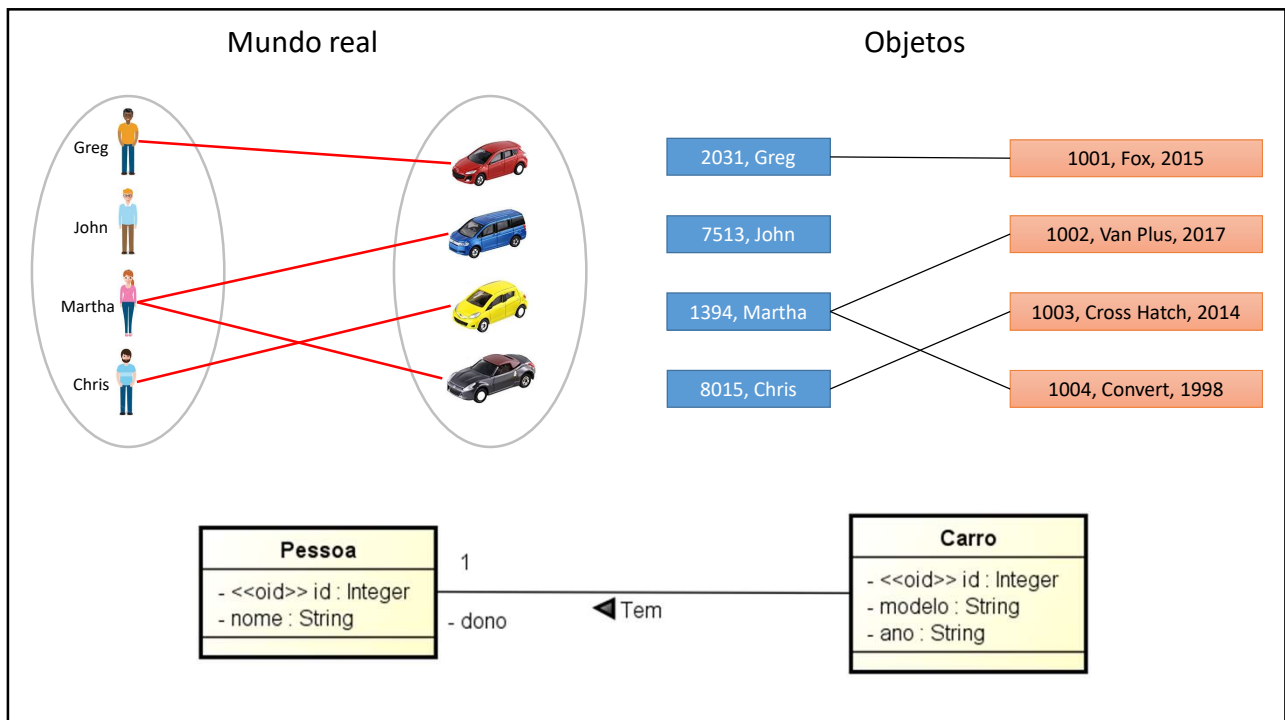
27



28



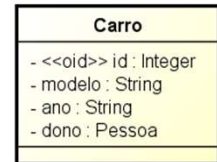
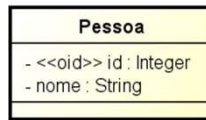
29



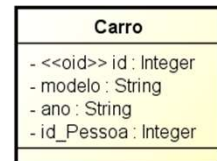
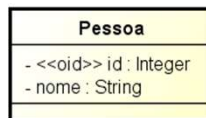
30

Atenção!

ERRADO



MUITO ERRADO



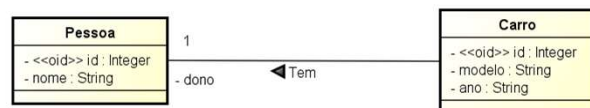
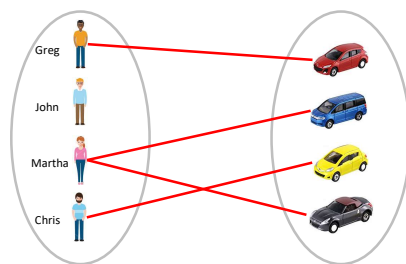
CORRETO



31

Resumo da aula

- Associação é um relacionamento estático entre dois conceitos



- Nome da associação (pouco valor)
- Nome de papel
- Multiplicidade (próxima aula)

- Não confunda com Modelo Relacional

32

Multiplicidade

33

Agenda

- O que é multiplicidade?
- Como encontrar as multiplicidades?
- Associações comuns
 - Um para muitos
 - Um para um
 - Muitos para muitos

34

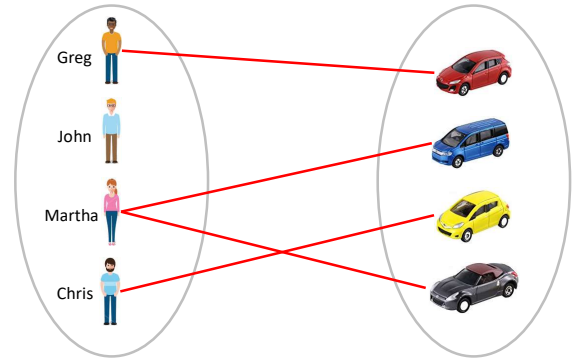
O que é multiplicidade?

É a quantidade mínima e máxima de objetos que uma associação permite em cada um de seus papéis.

Exemplo: um carro pode ter quantos donos?

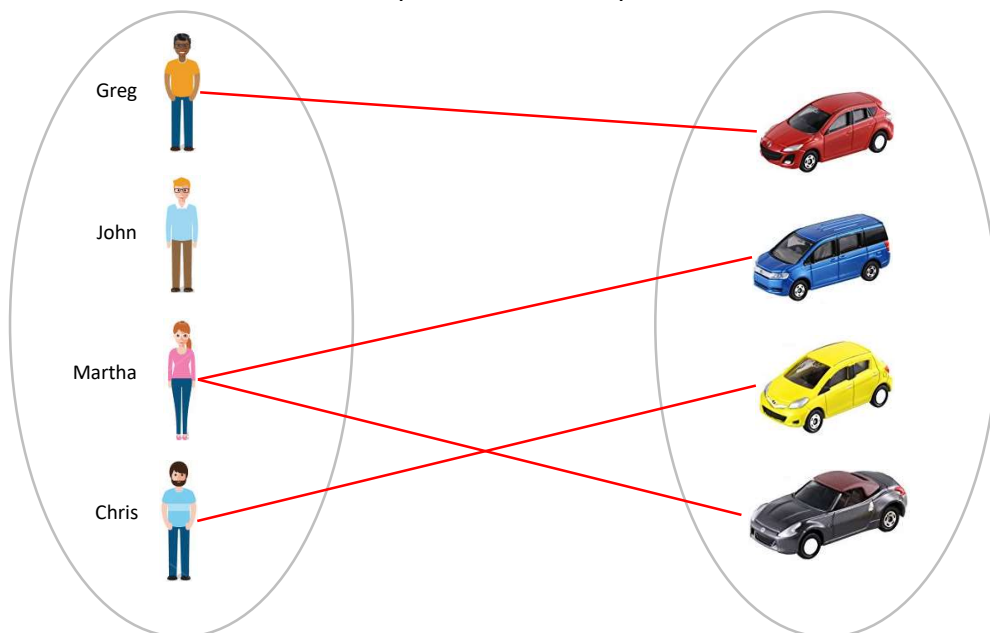
Mínimo: 1

Máximo: 1

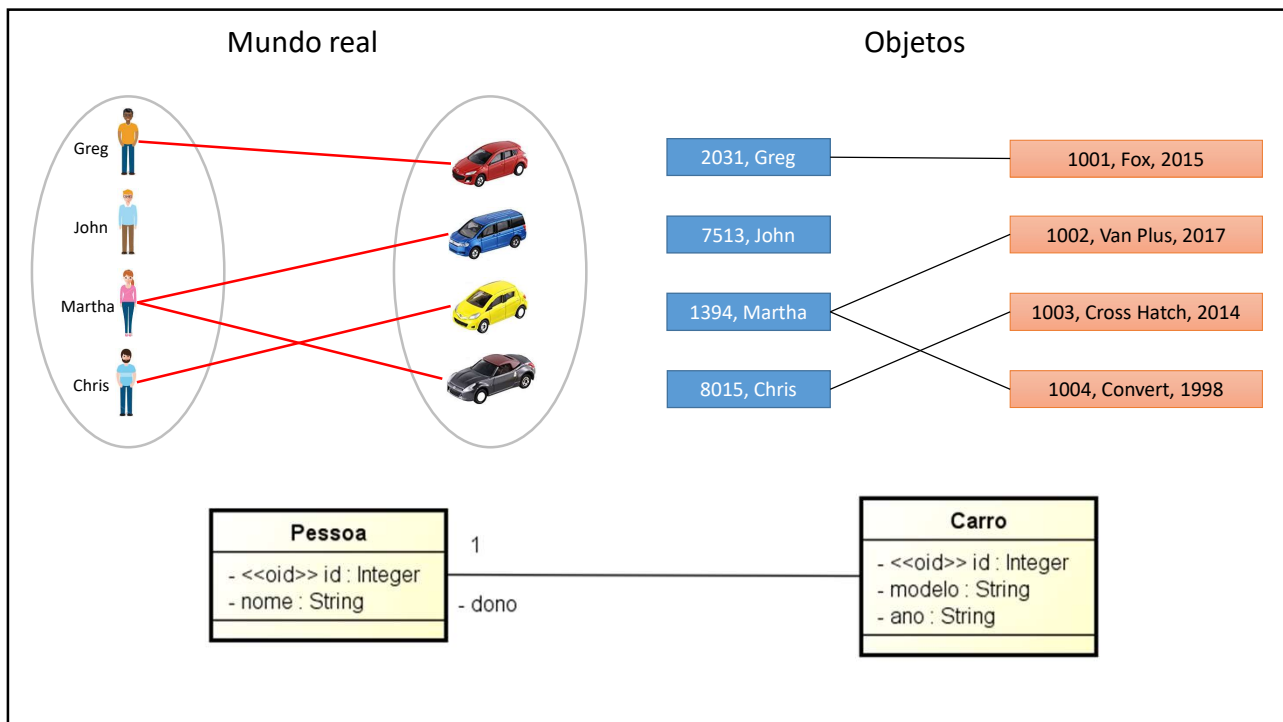


35

Recordando: quem é dono de qual carro?



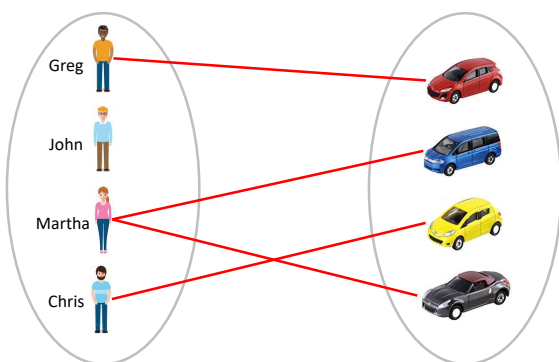
36



37

Como encontrar as
multiplicidades?

38



Pergunte para os dois lados:

1 <conceito> pode ter quantos <papel>?

1 carro pode ter quantos donos?

1 pessoa pode ter quantos carros?

Atenção:
sempre inicie a
pergunta com
"1"



Multiplicidades possíveis

"," significa "ou"

".." significa "a"

"*" significa "vários" (sem limite específico)

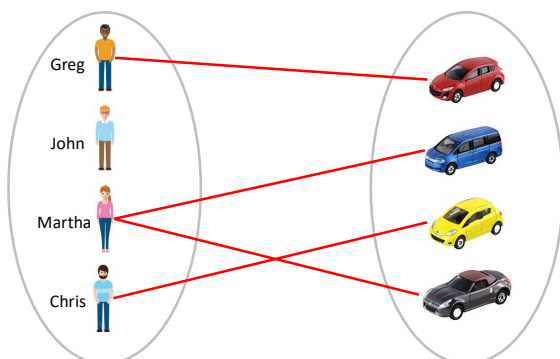
- a) 1 exatamente um
- b) 2 exatamente dois
- c) 0..1 zero a um
- d) 0..* zero ou mais
- e) * zero ou mais
- f) 1..* um ou mais
- g) 2..* dois ou mais
- h) 2..5 de dois a cinco
- i) 2,5 dois ou cinco
- j) 2,5..8 dois ou cinco a oito

Associações comuns

41

Um para muitos

Exemplo: quem é dono de cada carro?

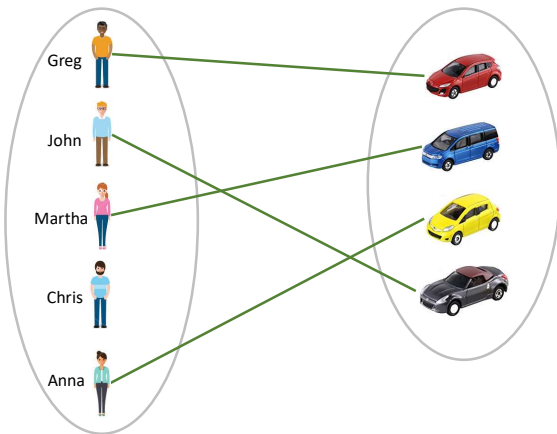


- Em um dos lados o máximo é 1
- No outro lado o máximo é "vários"

42

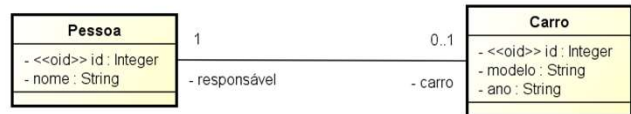
Um para um

Exemplo: quem é o responsável por cada carro?



1 carro pode ter quantos responsáveis?

1 pessoa pode ser responsável por quantos carros?

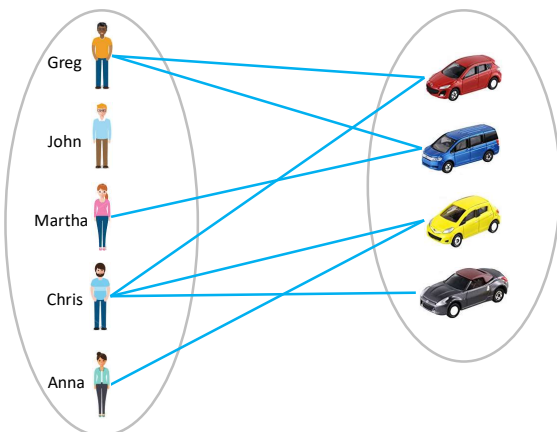


- Em ambos os lados o máximo é 1

43

Muitos para muitos

Exemplo: quem dirige cada carro?



1 carro pode ter quantos motoristas?

1 pessoa pode dirigir quantos carros?



- Em ambos os lados o máximo é "vários"

44

Resumo da aula

- Multiplicidade é a quantidade **mínima** e **máxima** de objetos que uma associação permite em cada um de seus papéis.
- Como encontrar as multiplicidades?
 - Pergunte para os dois lados: **1** <conceito> pode ter quantos <papel>?
- Multiplicidades possíveis
- Associações comuns
 - Um para muitos
 - Um para um
 - Muitos para muitos

45

Conceito dependente
Associações obrigatórias,
múltiplas e autoassociações

46

Agenda

- Associação obrigatória
- Conceito dependente
- Associações múltiplas
- Autoassociações

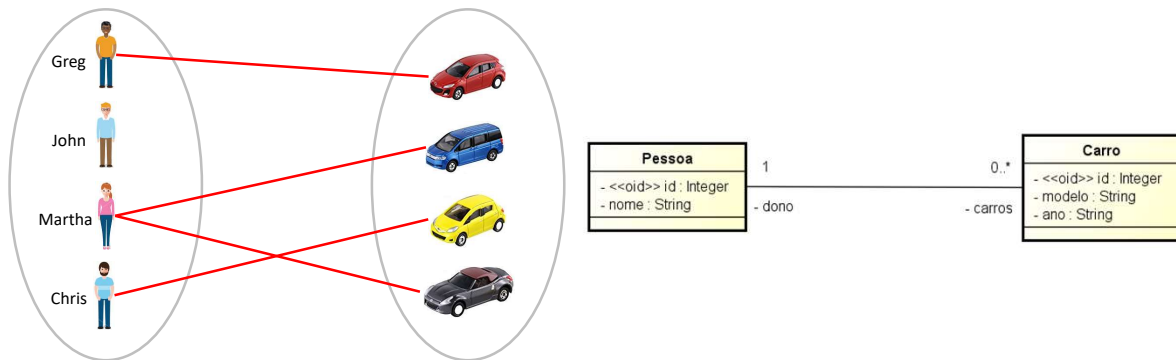
47

Associação obrigatória

48

Definição

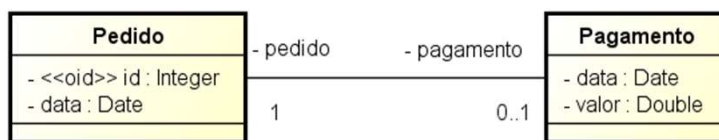
Uma associação é obrigatória se o conceito associado desempenha um papel de multiplicidade mínima maior que zero



- A associação de uma pessoa com carros não é obrigatória.
- A associação de um carro com dono é obrigatória.

49

Atenção



Embora o pagamento seja "obrigatório" em um pedido, um pedido pode existir temporariamente sem um pagamento.

Não confunda este caso com o conceito de associação obrigatória que acabamos de aprender.

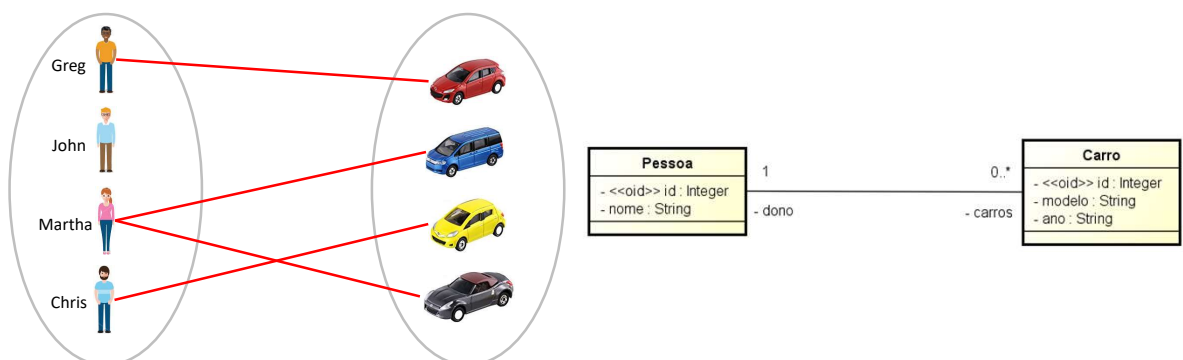
50

Conceito dependente

51

Definição

Um conceito é dependente se ele possuir pelo menos uma associação obrigatória.



- Pessoa é um conceito independente
 - Carro é um conceito dependente
- Só pode existir se existir uma pessoa dona dele
Se o dono deixar de existir, o carro também deixa de existir

52

Nota

A UML tem um símbolo que denota dependência de um modo geral, mas que não acrescenta valor prático à modelagem conceitual:



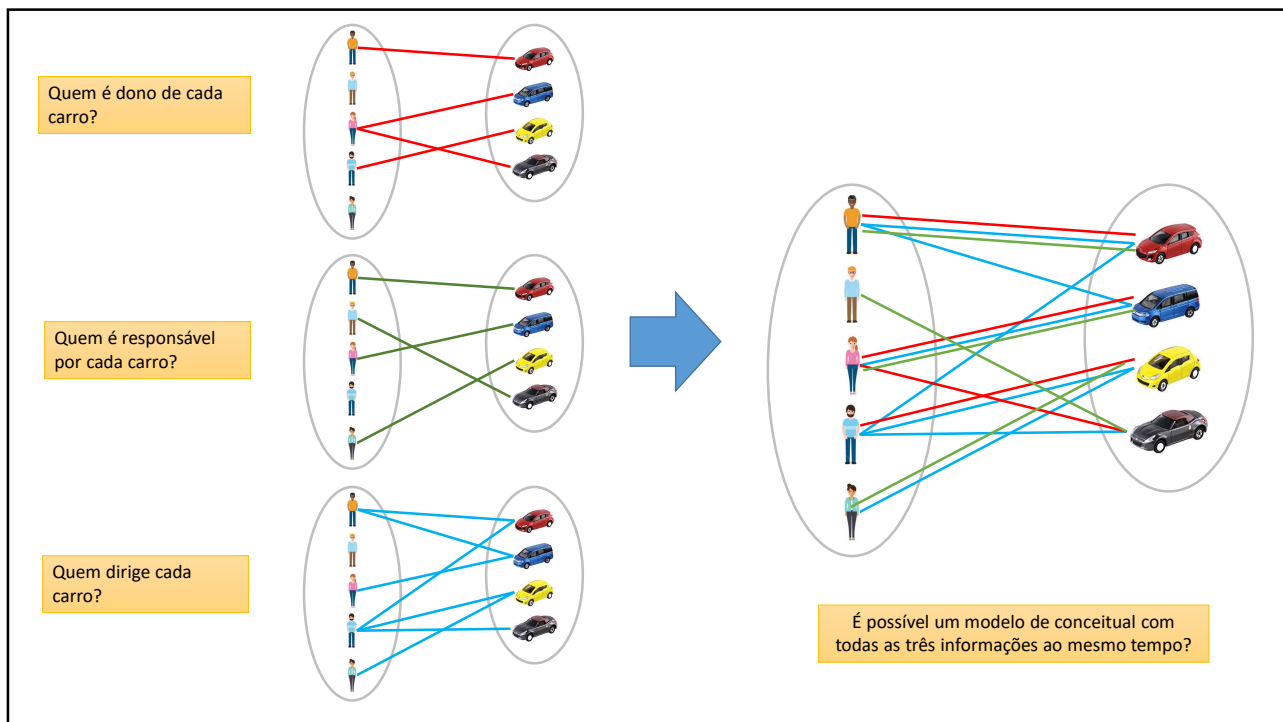
"Carro depende de pessoa"

53

Associações múltiplas

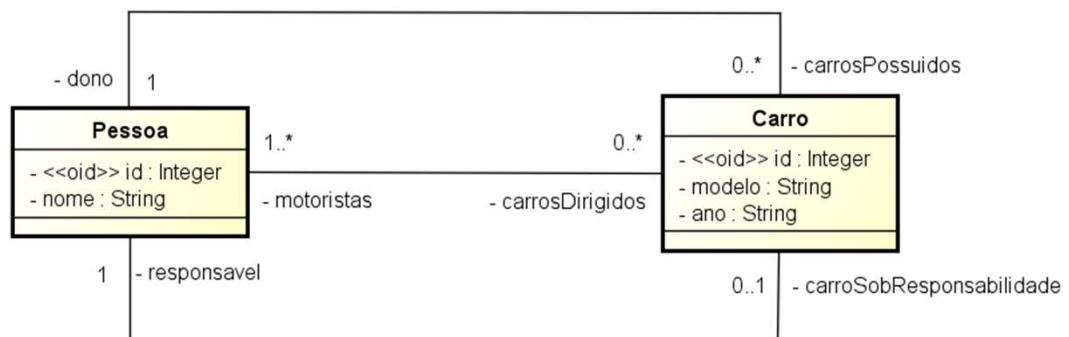
Pode haver mais de uma associação entre dois conceitos?

54



55

Sim. Mas os nomes de papel devem ser únicos.



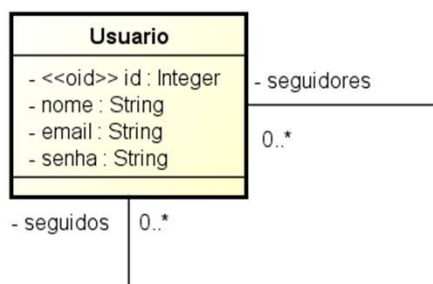
56

Autoassociações

57

Autoassociações

Quando um conceito é associado com ele próprio.



1 usuário pode ter quantos seguidores?

1 usuário pode ter quantos seguidos?

58

Resumo da aula

- Associação obrigatória
 - Conceito associado desempenha um papel de multiplicidade mínima maior que zero
- Conceito dependente
 - Possui pelo menos uma associação obrigatória
 - Só pode existir se o outro existir
 - Se o outro deixar de existir, o objeto dependente também deixa de existir
- Associações múltiplas
 - Ok. Nomes de papel únicos.
- Autoassociações

59

Associações todo-parte:
agregação e composição

60

Agenda

- Agregação
- Composição
- Ressalvas

61

Associações todo-parte

Quando um conceito é parte de outro que representa um todo, desenhamos um diamante no lado do todo.



Diamante branco: "Agregação"
O conceito parte não é exclusivo

Diamante preto: "Composição"
O conceito parte é **exclusivo**

62

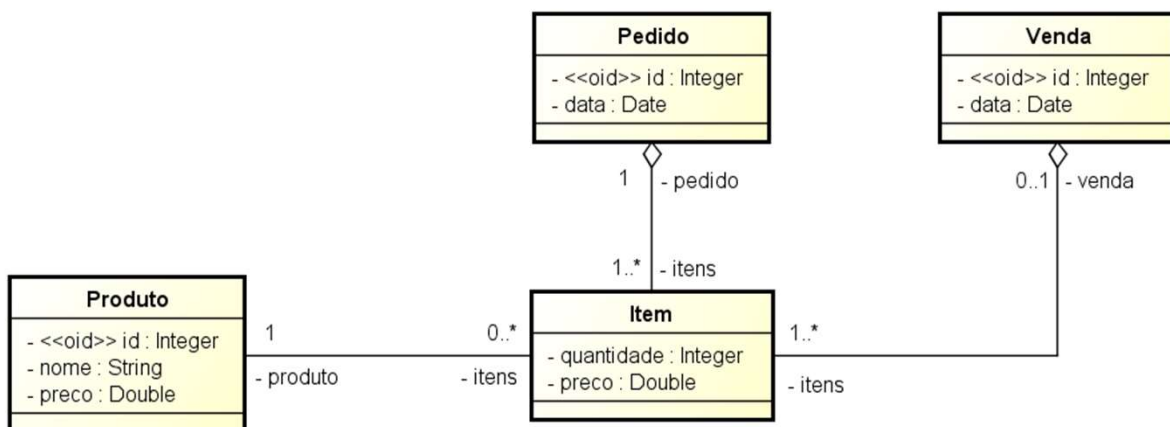
Exclusividade: 1 ou 0..1

Como a composição (diamante preto) é uma relação exclusiva, a multiplicidade no lado do diamante sempre será 1 ou 0..1



63

Agregação - exemplo 2



64

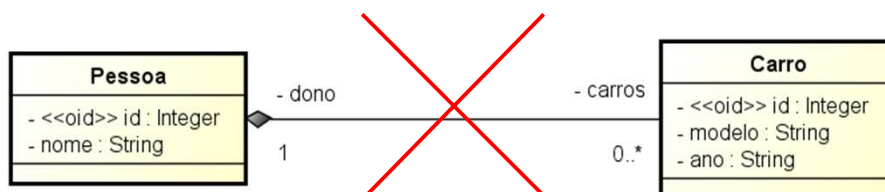
Agregação - exemplo 3



65

Ressalva 1

O diamante (seja branco, seja preto) deve ser usado **somente em casos em que realmente se trata de uma relação todo-parte**



ERRADO

66

Ressalva 2

Algumas pessoas confundem o diamante preto (composição) como se fosse uma forma de indicar a deleção em cascata dos objetos dependentes. Na verdade o que indica isso é a multiplicidade.



67

Resumo da aula

- Associações todo-parte
 - Agregação - diamante branco - mais fraca
 - Composição - diamante preto - mais forte - exclusiva (1 ou 0..1)
- Ressalvas
 - Use diamante somente quando realmente for uma associação todo-parte
 - Não é a composição que indica deleção em cascata

68

Classe de associação

69

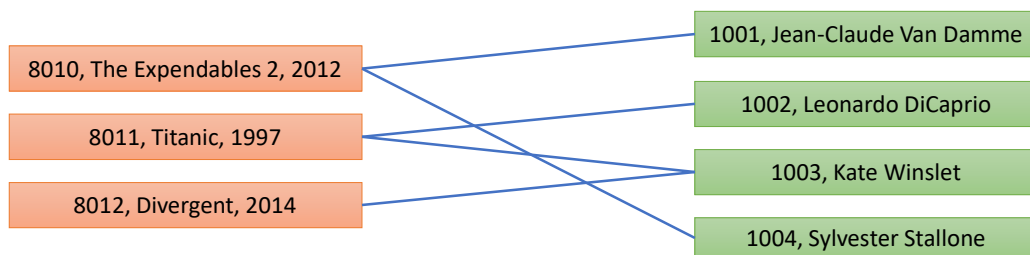
Agenda

- Exemplo motivador
- Classe de associação em associações muitos-para-muitos
- Classe de associação vs. Classe comum

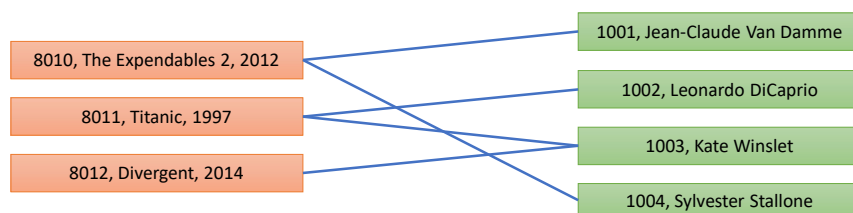
70

Exemplo motivador

Deseja-se fazer um sistema para manter um cadastro de filmes e artistas (atores/atrizes), bem como a informação de qual artista atuou em cada filme.



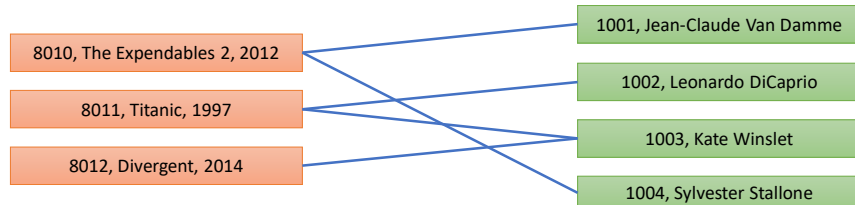
71



72

Problema

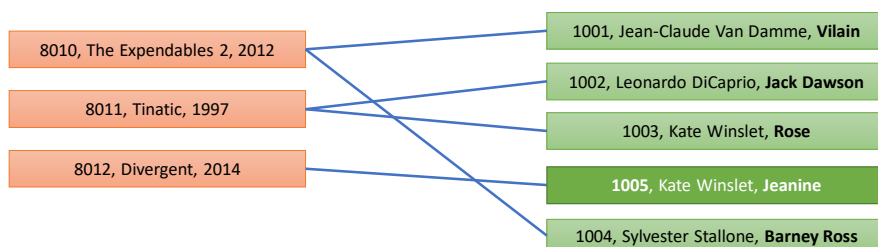
Além disso, desejo saber também o nome do personagem desempenhado por cada artista em cada filme



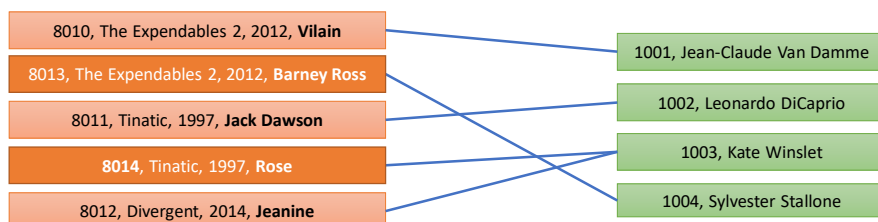
73

Problema

Além disso, desejo saber também o nome do personagem desempenhado por cada artista em cada filme



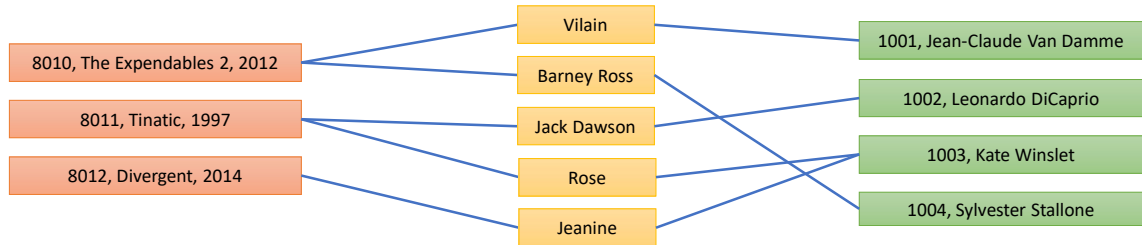
ERRADO



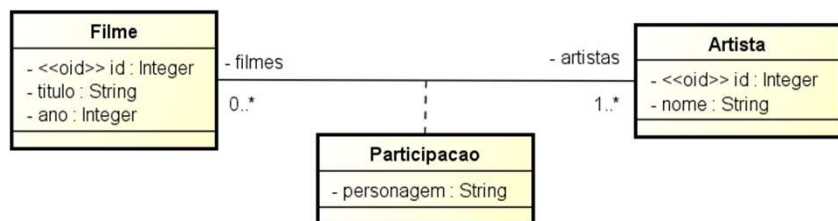
ERRADO

74

Conclusão: o nome do personagem é um dado que pertence à associação.
Neste caso, o nome do personagem deverá ser armazenado em um objeto intermediário.



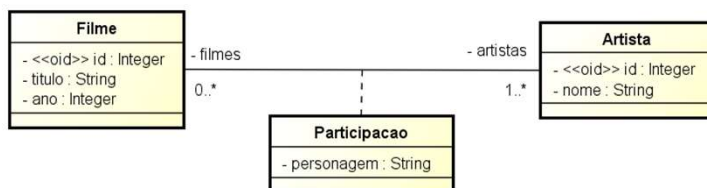
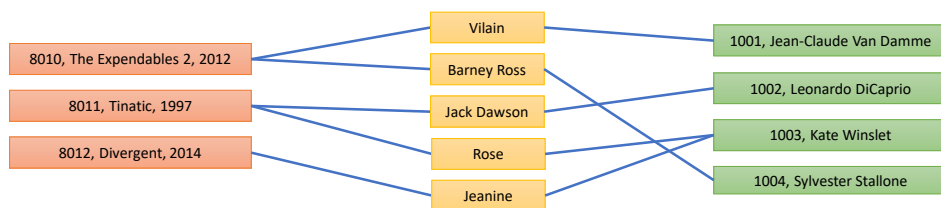
Classe de associação:



75

Atenção!

A classe de associação indica associação ÚNICA!



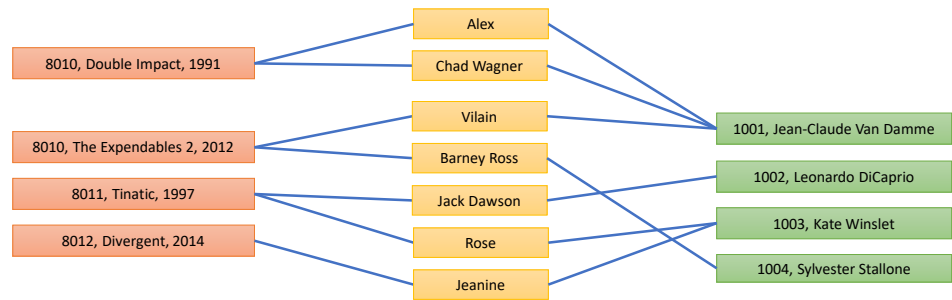
Este modelo indica que para um mesmo artista e um mesmo filme, só pode haver uma participação

Isso não pode:

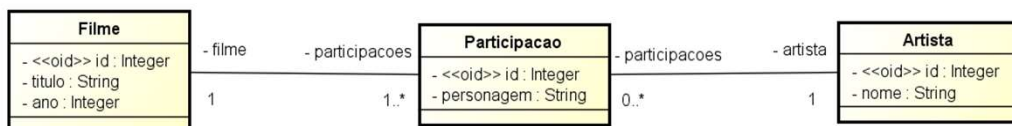
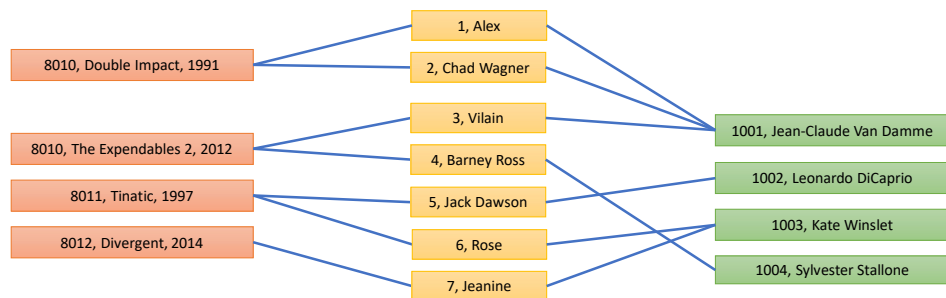


76

Então como representar um modelo no qual um mesmo artista pode representar mais de um personagem em um mesmo filme?



77



78

Resumo da aula



Associação muitos-para-muitos: ALERTA!
 Tem atributo na associação?
 SIM: então haverá um novo conceito no meio!

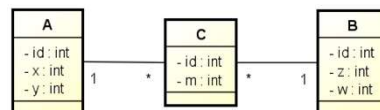
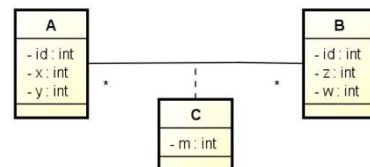
O mesmo "A" pode estar associado com o mesmo "B" mais de uma vez?



O mesmo artista pode participar do mesmo filmes mais de uma vez?

não

sim



Enumerações e Tipos Primitivos

Enumerações

Uma enumeração pode ser considerada um "meio termo" entre um conceito e um atributo.

Uma enumeração representa um conceito que possui um número finito de valores possíveis, valores estes que, para o negócio, valem a pena ser descritos.



81

Para simplificar o diagrama principal do sistema, recomenda-se incluir as enumerações em um diagrama separado, representando seu nome no diagrama do sistema diretamente como um tipo de atributo.

Diagrama de enumerações

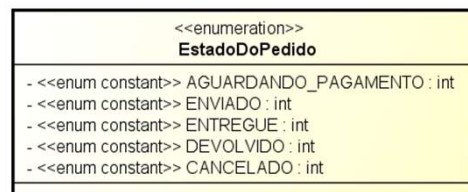
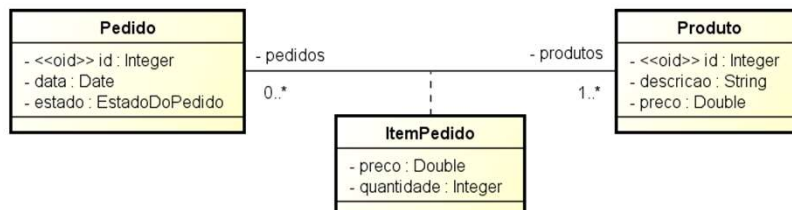


Diagrama do sistema



82

Tipos primitivos

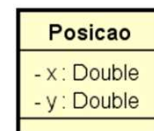
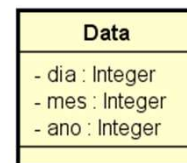
Também podem ser considerados um "meio termo" entre um conceito e um atributo.

É um conceito que, devido a sua simplicidade, "não merece" ser modelado como um conceito comum no diagrama do modelo conceitual.

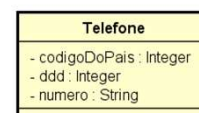
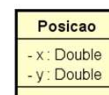
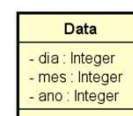
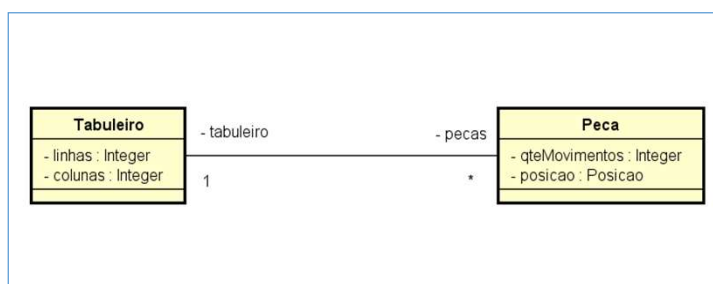
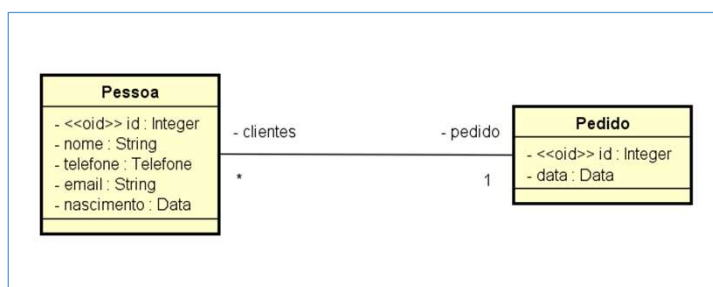
Sua regra de formação é meramente sintática e não depende de nenhum dado do sistema.

Tipicamente não possui identificador.

Outros exemplos: ISBN, CEP, Posicao3D, etc.



83



84

Ressalva

Em linguagens modernas, a data (ou data-hora) não é armazenada internamente por meio de um número inteiro para cada campo (dia, mês, ano, hora, segundo, milissegundo).

Exemplo:

Em linguagem Java, o tipo `Date`, do pacote `java.util`, armazena uma data-hora na forma de um único número inteiro longo (`long`), representando a quantidade de milissegundos que se passaram desde 0:00:00 GMT 01/01/1970

Teste no `compilejava.net`:

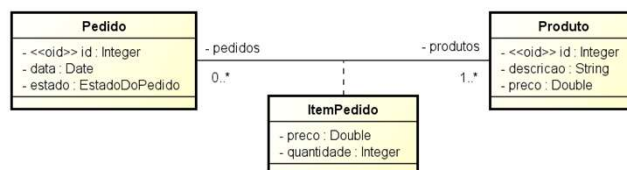
```
java.text.SimpleDateFormat sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm:ss z");
java.util.Date d1 = new java.util.Date();
java.util.Date d2 = new java.util.Date(0L);
System.out.println(d1.getTime() + ": " + sdf.format(d1));
System.out.println(d2.getTime() + ": " + sdf.format(d2));
```

85

Resumo da aula

• Enumerações

- Meio termo conceito/atributo
- Conjunto finito de valores



• Tipos primitivos

- Meio termo conceito/atributo
- Simples
- ISBN, CEP, Posicao, Posicao3D, etc.



86

Herança

87

Agenda

- Exemplo motivador
- Dois primeiros questionamentos ao se considerar herança
- Definições importantes
- Ressalvas
- Quando o uso de herança é impróprio?

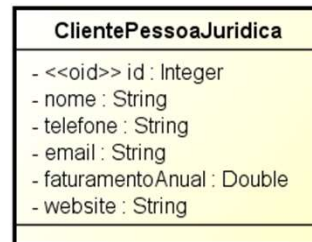
88

Exemplo motivador

Deseja-se manter um cadastro dos clientes de uma empresa, sendo que há dois tipos de clientes: pessoa física e pessoa jurídica (organizações).



8010, Greg Junior, 3928-9211, greg@gmail.com, 12/10/1985, 83911290166

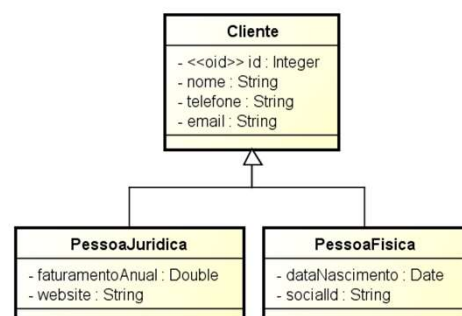
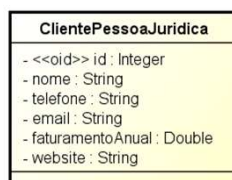


6165, Arts Inc., 30228-6160, contact@arts.com, 500000.00, www.arts.com

89

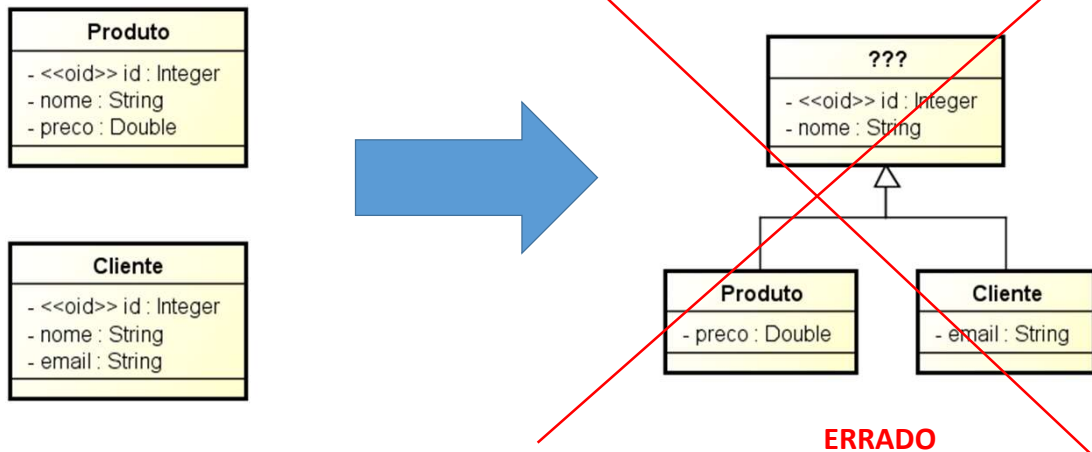
Questionamentos básicos ao se considerar herança:

- Há estrutura comum entre os conceitos?
 - id, nome, telefone, email
- Há relação É-UM entre os conceitos e um conceito mais genérico que pode representar a estrutura comum?
 - ClientePessoaFísica é um **cliente**
 - ClientePessoaJurídica é um **cliente**



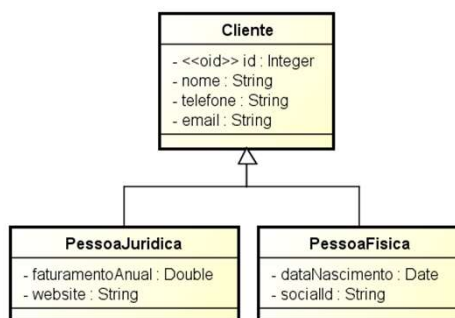
90

Exemplo no qual não há relação É-UM



91

Definições importantes



- Relação É-UM
- Relação de generalização / especialização
- Superclasse / subclasse
- Herança
 - Extensão
 - Pode adicionar elementos
 - Não pode remover elementos
- A herança é uma **associação de classes** e não de objetos (não há duas instâncias a serem ligadas)

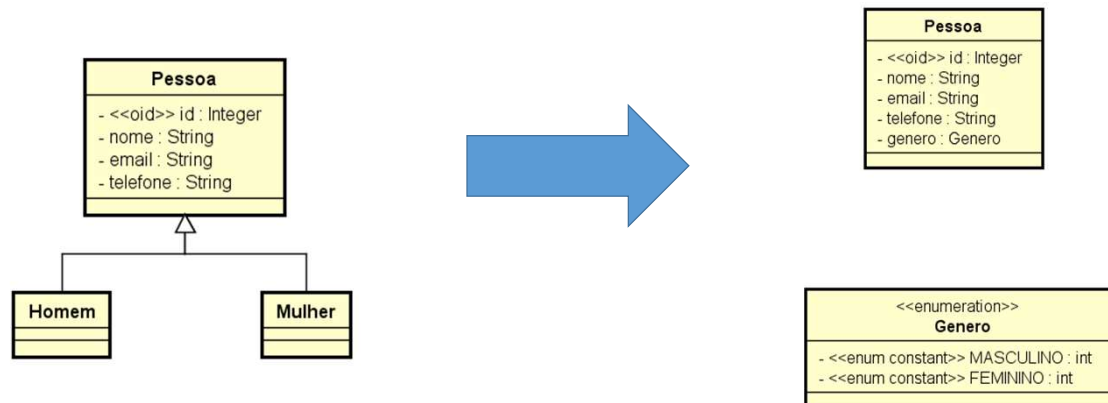
8010, Greg Junior, 3928-9211, greg@gmail.com, 12/10/1985, 83911290166

6165, Arts Inc., 30228-6160, contact@arts.com, 500000.00, www.arts.com

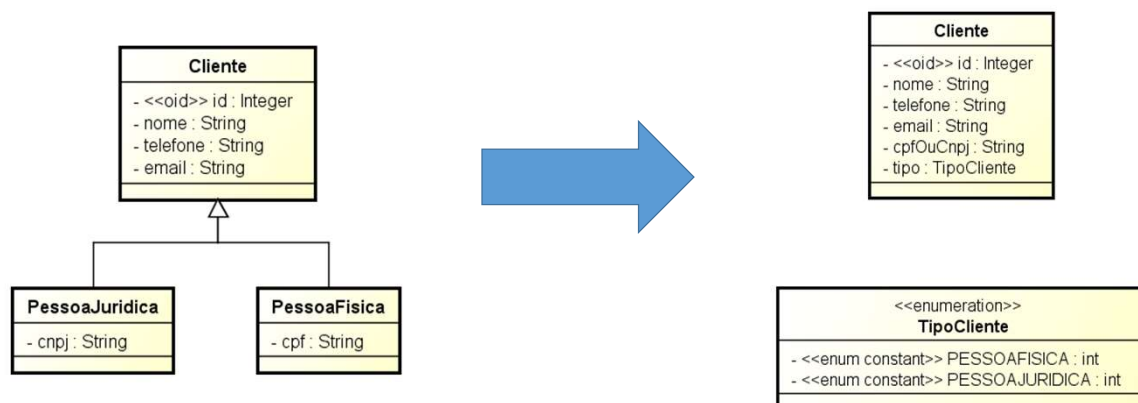
92

Ressalva 1

Não use herança se não há dados exclusivos de cada subclasse.

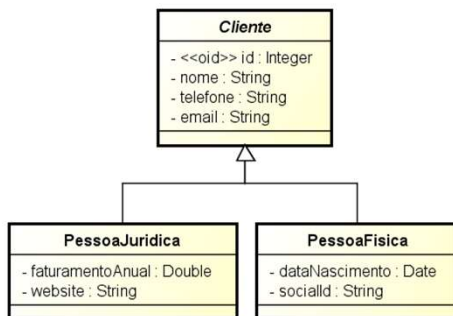


93



94

Ressalva 2



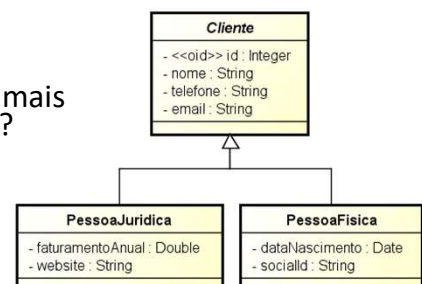
- Recomenda-se cautela com herança parcial, dando-se preferência a herança total.
- **Herança total**: somente instâncias das subclasses (PessoaJuridica e PessoaFisica) são permitidas (não são permitidas instâncias de Cliente).
- Neste caso, defina a superclasse como uma classe **abstrata** (UML = nome em itálico).

95

Resumo da aula

- **Questionamentos básicos ao considerar herança:**
 1. Há estrutura comum entre os dois conceitos?
 2. Há relação É-UM entre os conceitos e um conceito mais genérico que pode representar a estrutura comum?

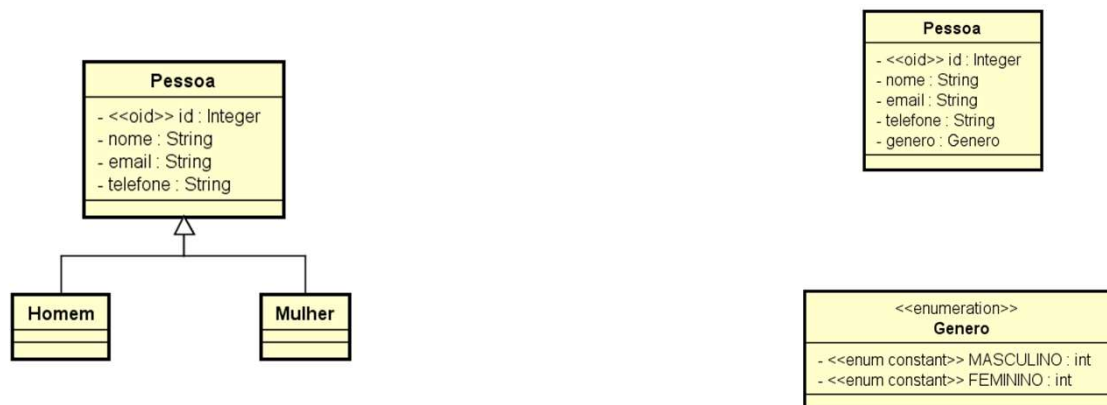
- **Definições:**
 - Relação É-UM
 - Relação generalização / especialização
 - Superclasse / subclasse
 - Herança / Extensão
 - A herança é uma associação de classes



96

Resumo da aula

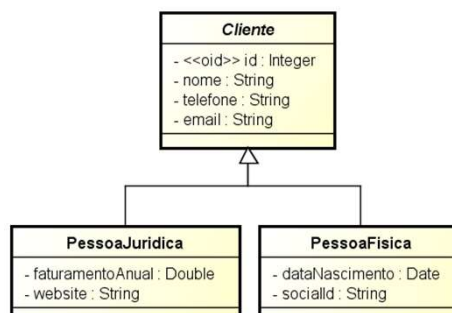
- **Ressalva 1:** Não use herança se não há dados exclusivos de cada subtipo.



97

Resumo da aula

- **Ressalva 2:** Prefira herança total. Faça a superclasse como **abstrata** (UML = nome em *itálico*)



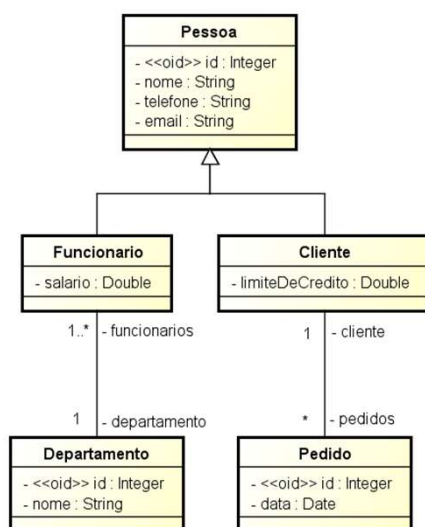
98

Agenda

- Exemplo motivador
- Dois primeiros questionamentos ao se considerar herança
- Definições importantes
- Ressalvas
- Quando o uso de herança é impróprio?

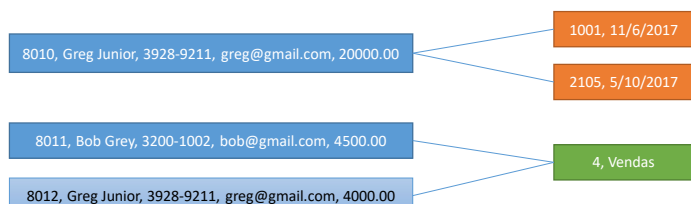
99

Uso impróprio de herança

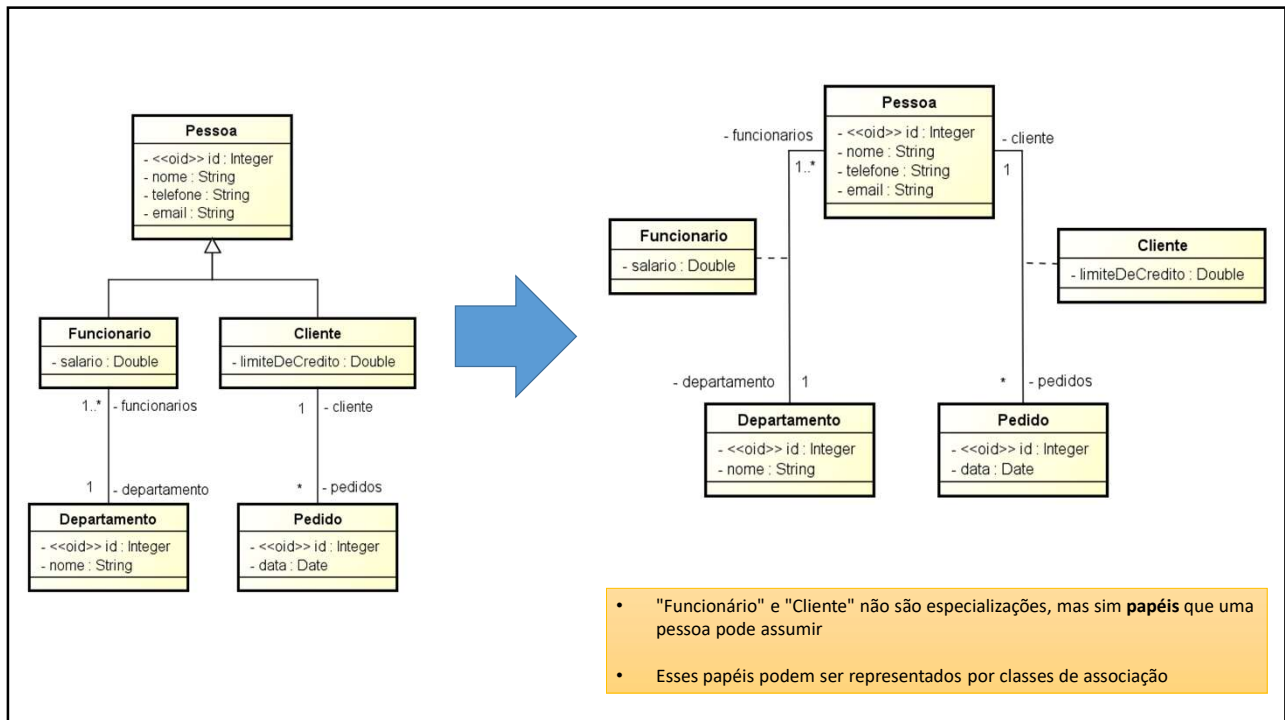


Indicativos de que o uso da herança é impróprio:

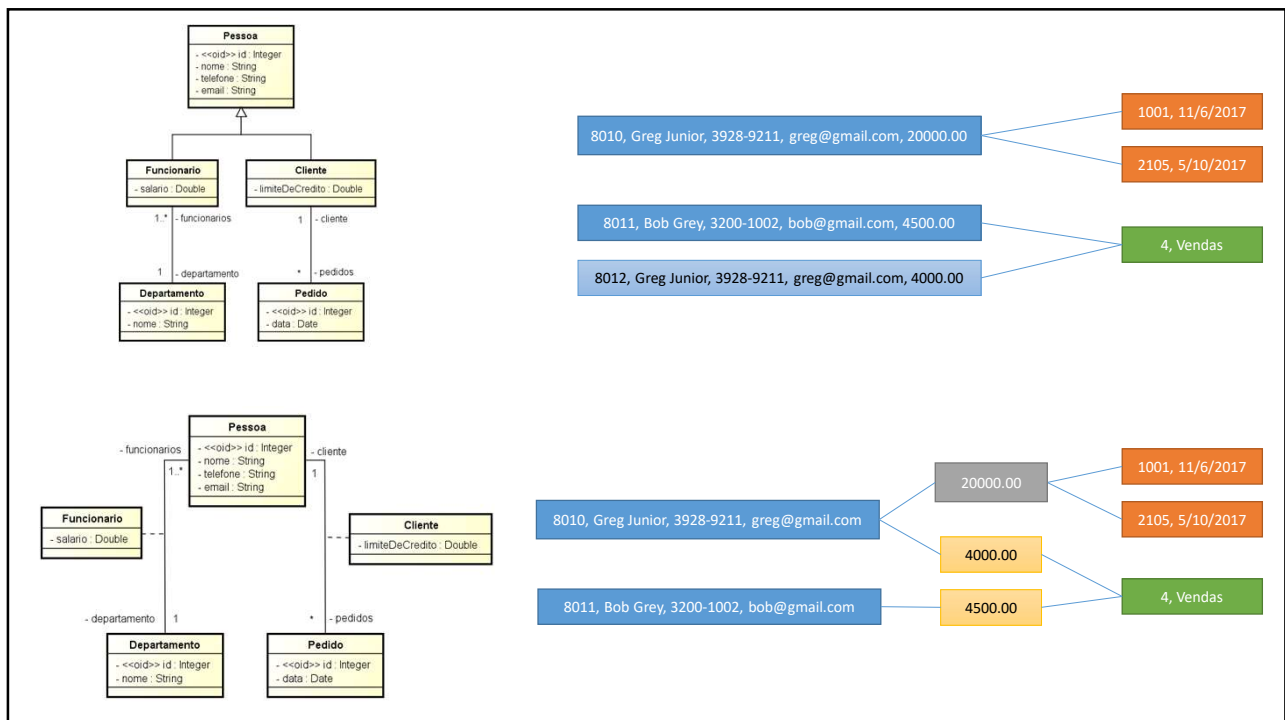
- Para o conceito da subclasse fazer sentido, ele estaria associado a outro conceito?
- Uma mesma pessoa pode ser cliente e também funcionário?



100



101



102

Resumo da aula

- Indicativos de que o uso da herança é impróprio:

1. Para o conceito da subclasse fazer sentido, ele estaria associado a outro conceito?
2. Um mesmo <Superclasse> pode ser <Conceito1> e também <Conceito2>?

Impróprio

