

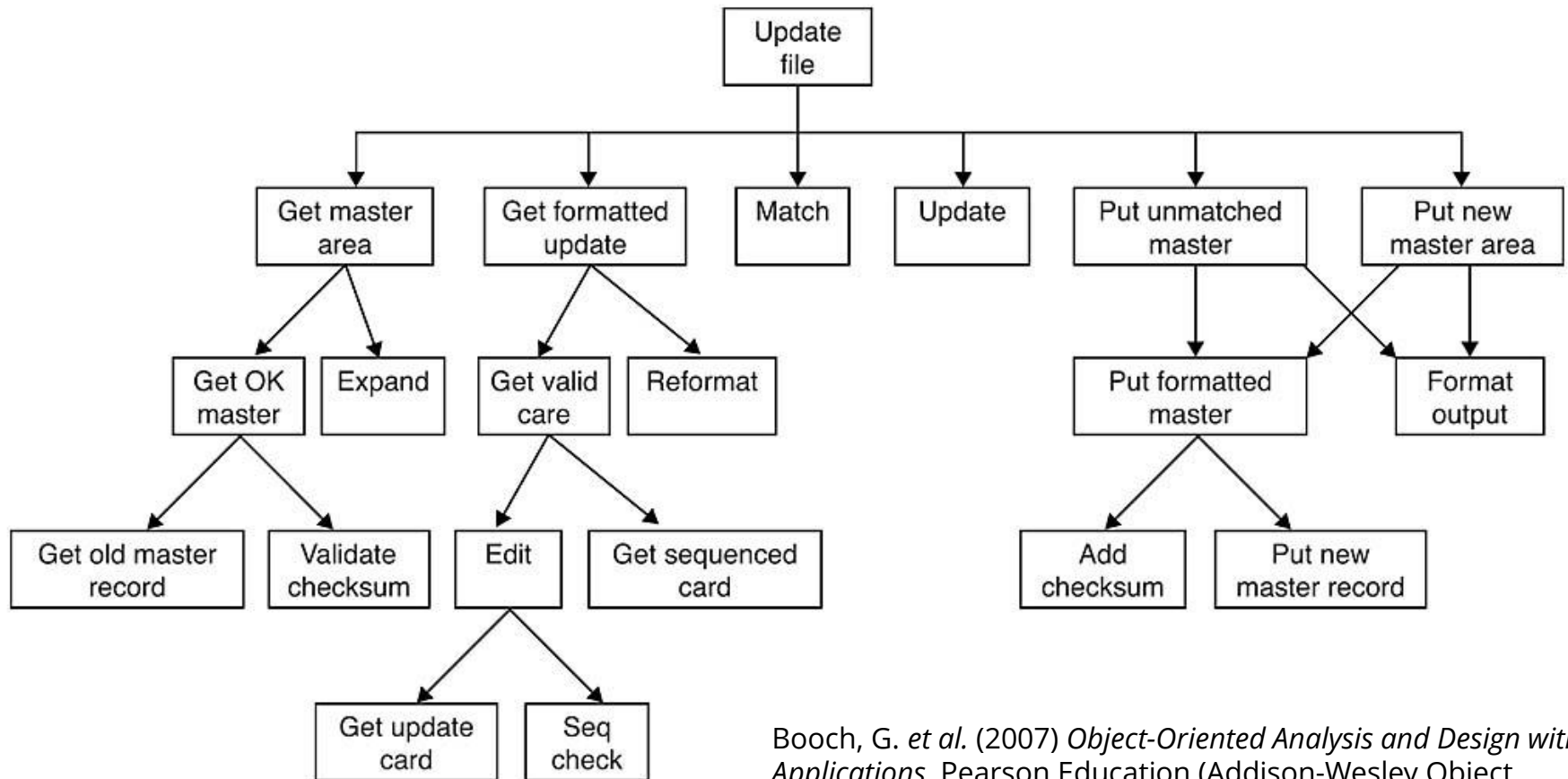


MODELO DO DOMÍNIO COM CLASSES DA UML Parte I

MODELAÇÃO E ANÁLISE DE SISTEMAS | TP

ILÍDIO OLIVEIRA ico@ua.pt
v2018-03-07

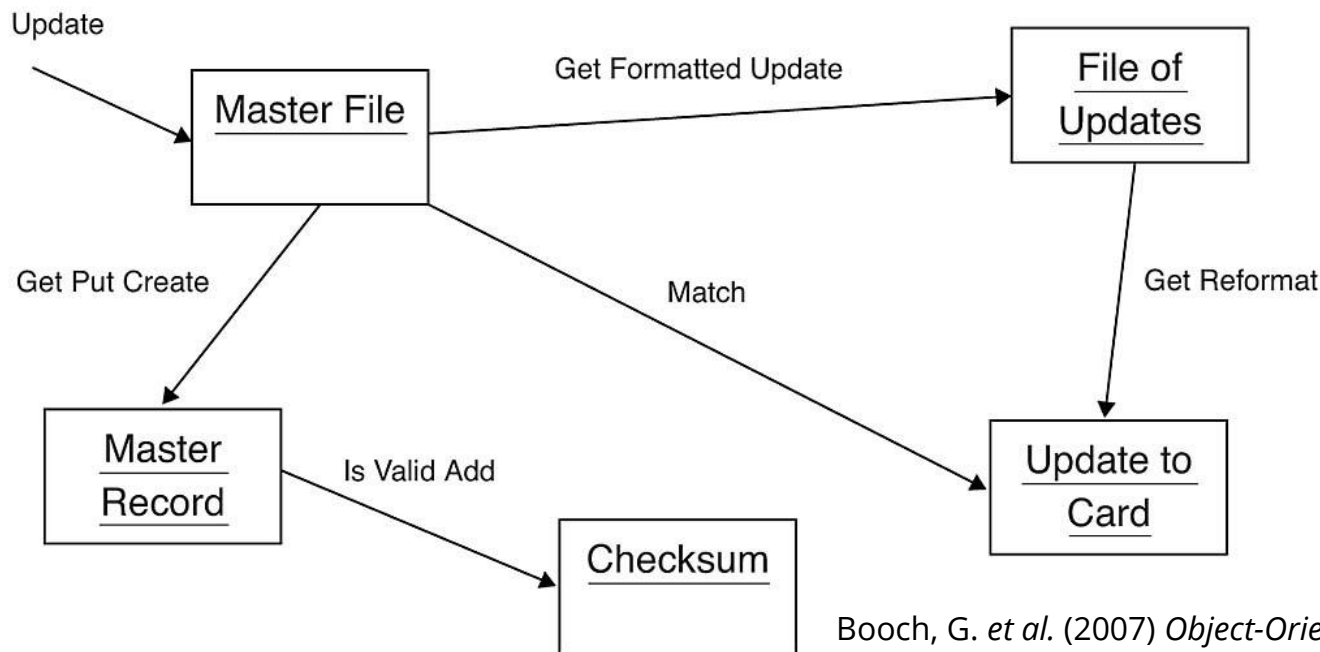
Decomposição algorítmica, *top-down*



Booch, G. et al. (2007) *Object-Oriented Analysis and Design with Applications*. Pearson Education (Addison-Wesley Object Technology Series).

Decomposição por objetos (abstrações-chave no domínio)

“Rather than decomposing the problem into steps such as *Get formatted update* and *Add check sum*, we have identified objects such as *Master File* and *Check Sum*, which derive directly from the vocabulary of the problem domain.”



Booch, G. et al. (2007) *Object-Oriented Analysis and Design with Applications*. Pearson Education (Addison-Wesley Object Technology Series).

Decomposição por objetos

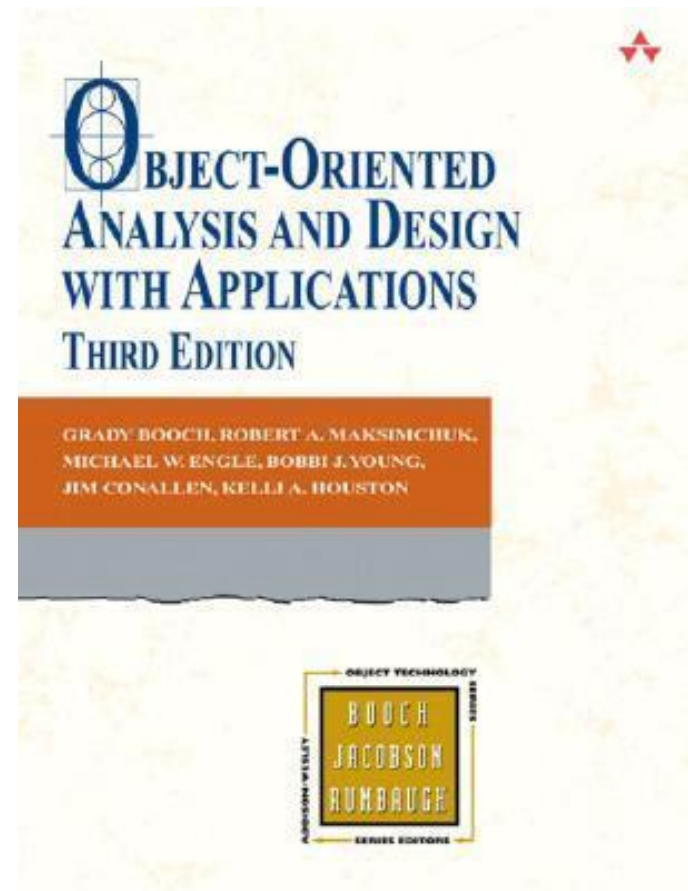
"Although both designs solve the same problem, they do so in quite different ways.

In this second decomposition, we view the world as a set of autonomous agents that collaborate to perform some higher level behavior. *Get formatted update* thus does not exist as an independent algorithm; rather, it is an **operation associated with the object** *File of Updates*.

Calling this operation creates another object, *Update to Card*. In this manner, **each object in our solution embodies its own unique behavior, and each one models some object in the real world.**

From this perspective, an object is simply a tangible entity which exhibits some well-defined behavior.

Objects do things, and we **ask them to perform what they do by sending them messages**. Because our decomposition is based upon objects and not algorithms, we call this an *object-oriented* decomposition."



Objetos e classes

Na modelação por objetos

Desenvolvimento por objetos

É uma estratégia para simplificar o espaço do problema, modularizando-o

Esquema mental comum à análise (requisitos) e programação

Classificar entidades similares

Facilita a reutilização de software

Os objetos (em OO) modelam entidades do mundo real/espço do problema

Observáveis no mundo físico

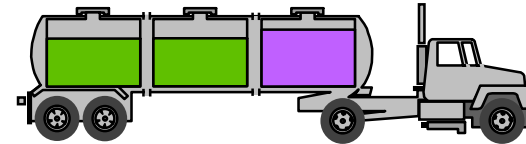
e.g.: Aluno, Avião

Conceitos

e.g.: Venda, Reserva

Abstrações próprias do software

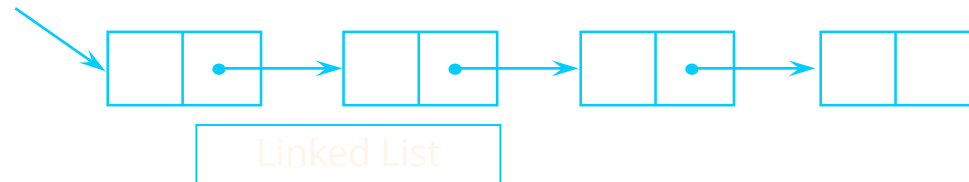
e.g.: ListaLigada, Vetor



Truck



Chemical
Process



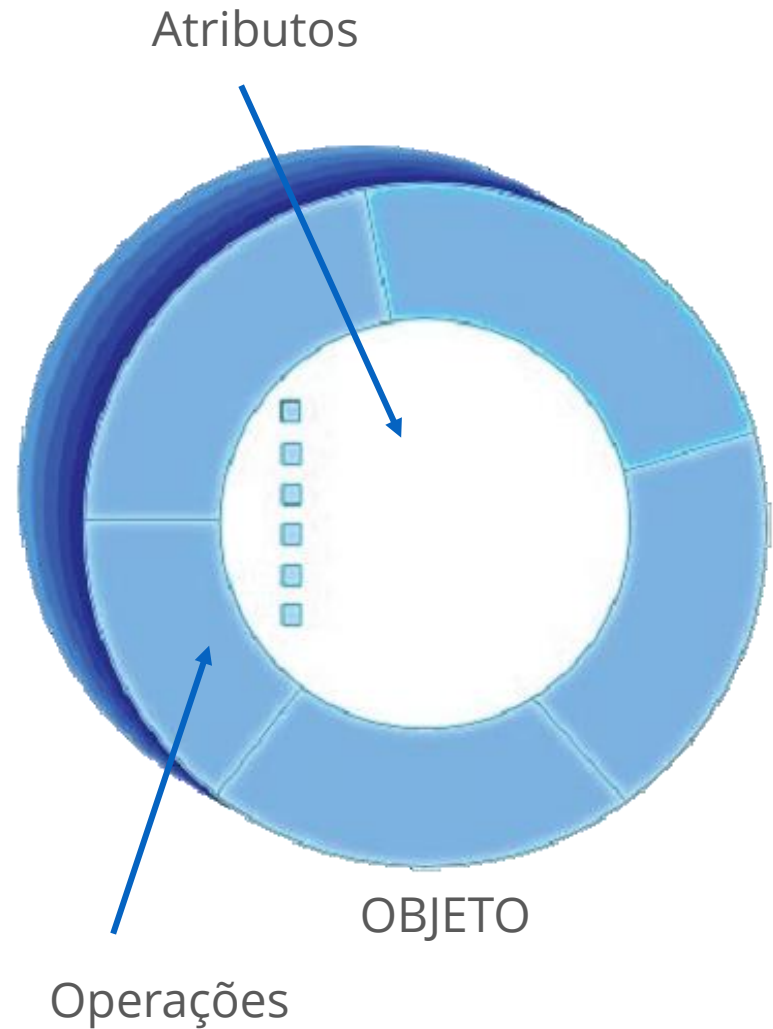
Objeto = propósito + estado + comportamento

Um objeto tem um propósito bem definido

É uma entidade com uma fronteira que encapsula **estado** e **comportamento**.

O estado é representado através de atributos e relacionamentos.

O comportamento é representado através de operações.



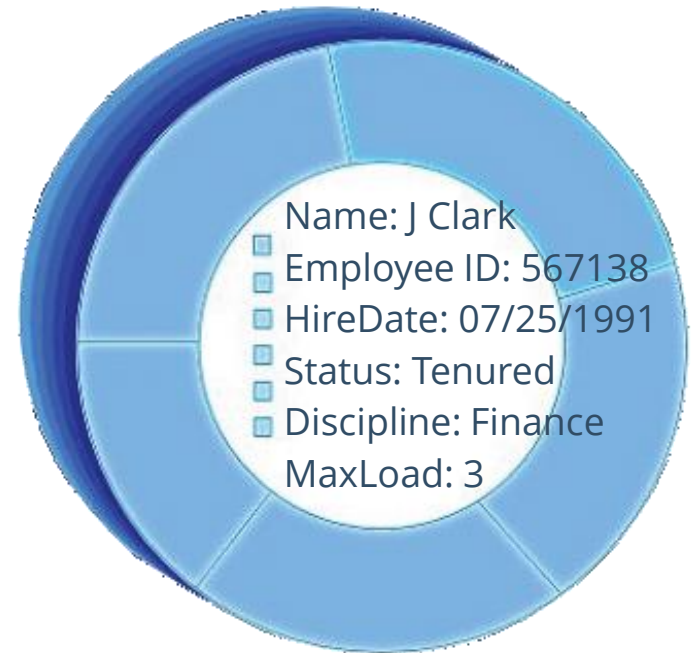
Um objeto tem um estado (interno)

O estado de um objeto corresponde a uma das condições/configurações é que é possível o objeto apresentar-se.

É normal o estado objeto mudar ao longo do tempo.



Name: J Clark
Employee ID: 567138
Date Hired: July 25, 1991
Status: Tenured
Discipline: Finance
Maximum Course Load: 3 classes

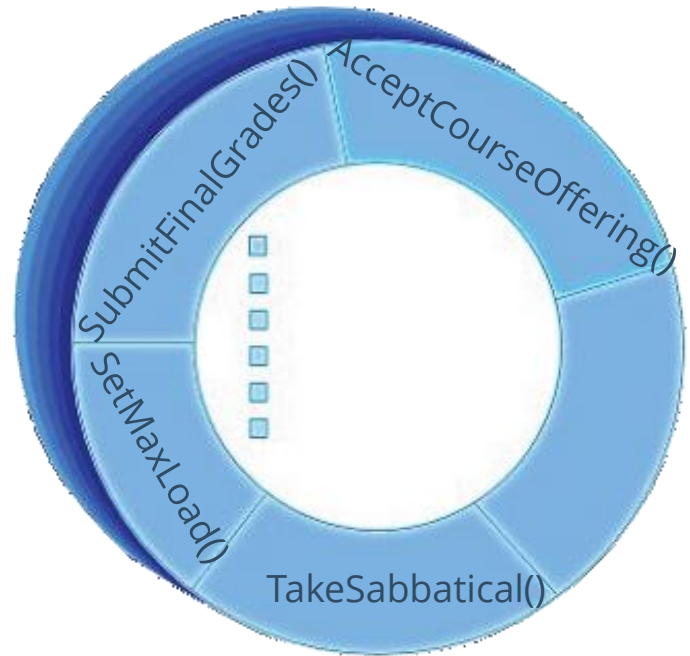


Professor
Clark

Um objeto tem comportamento (funcionalidades)

O comportamento define como é que o objeto age/reage

O comportamento visível/exposto é modelado pelo conjunto de mensagens a que responde (operações)



Professor Clark's behavior
Submit Final Grades
Accept Course Offering
Take Sabbatical
Maximum Course Load: 3 classes

Professor Clark

Objeto: estado + operações

Sabe alguma coisa

Os seus atributos

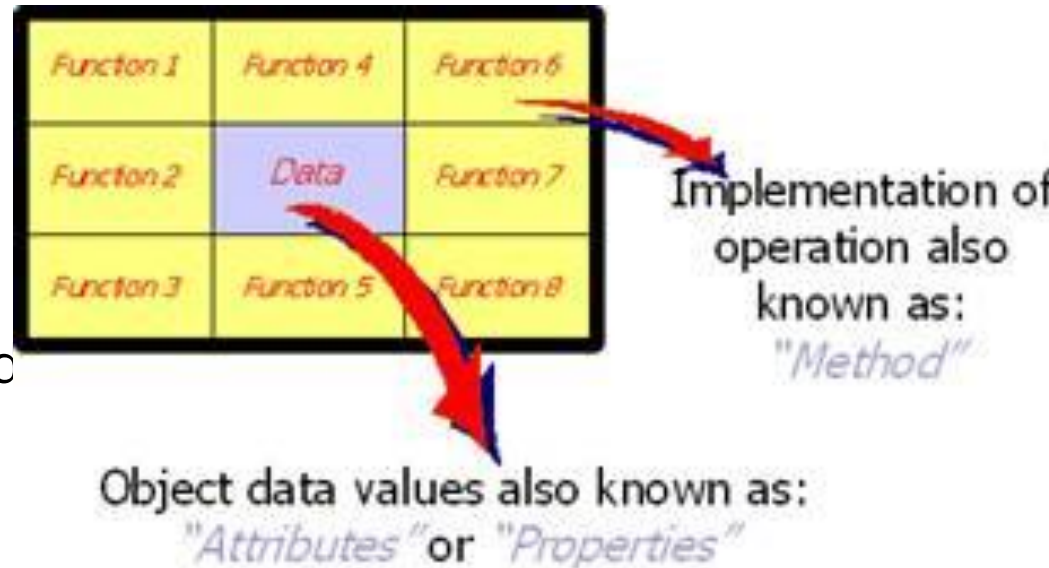
Os seus relacionamentos

Sabe fazer alguma coisa

O seu comportamento, ativado através de operações

Cápsula

Os outros (objetos) não precisam de saber o que ele sabe...



Um objeto tem identidade própria

Cada objeto tem a sua própria identidade, única, mesmo que o seu estado seja igual ao de outro objeto.



Professor “J Clark”
teaches Biology



Professor “J Clark”
teaches Biology

O que é uma classe?

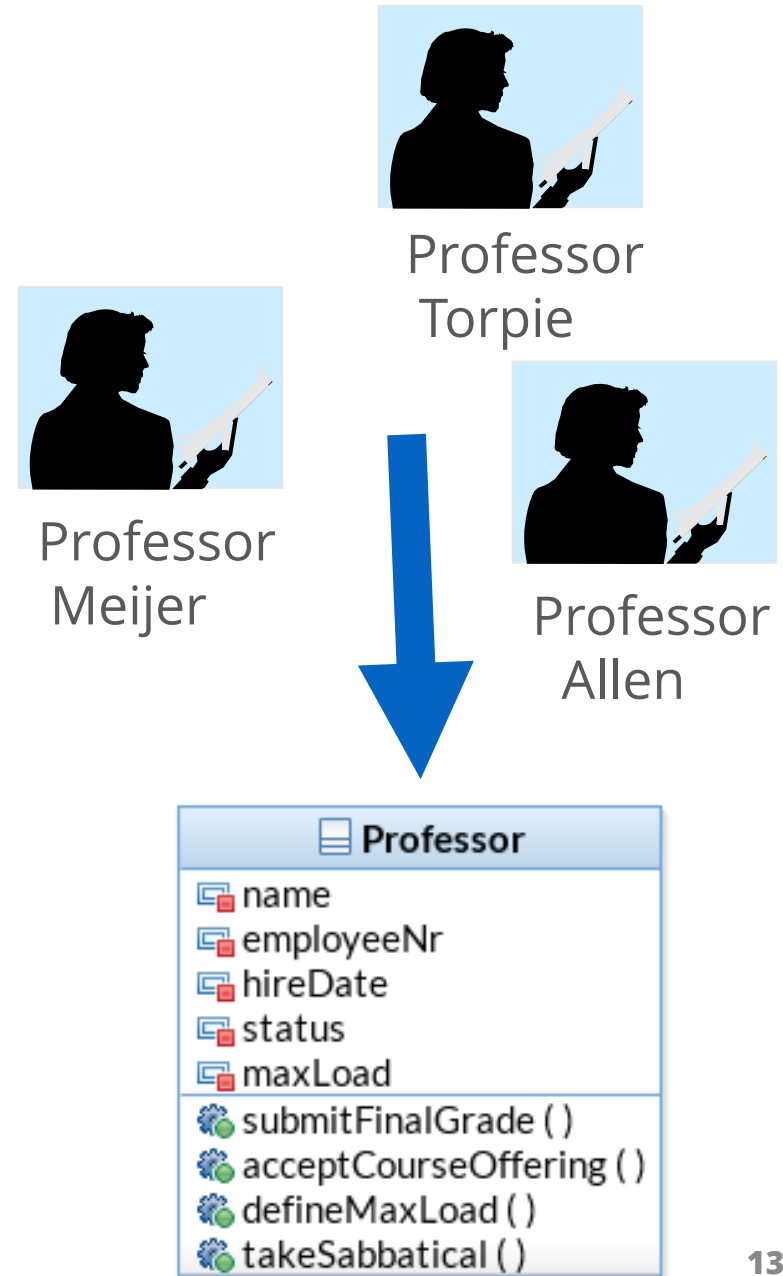
É uma categoria de objetos semelhantes, que partilham os mesmos atributos, operações, relacionamentos e semântica.

O objeto é uma instância (ocorrência) de uma classe

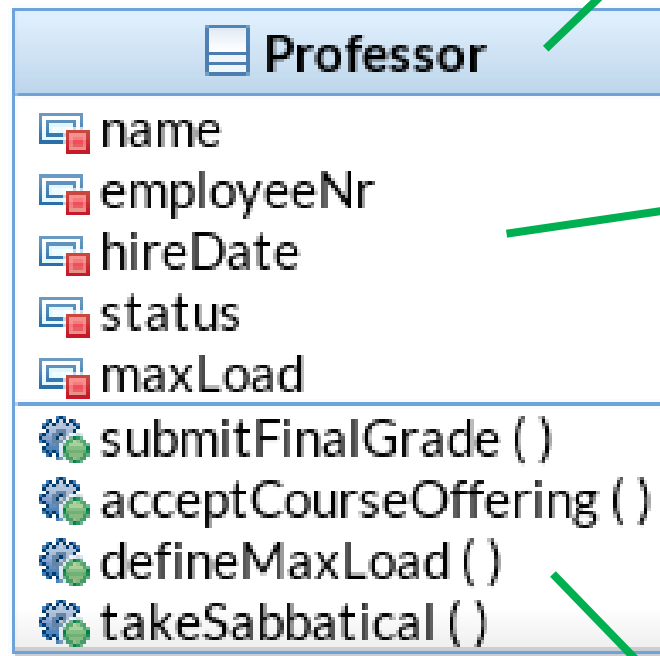
Uma classe é uma abstração

Categoriza objetos semelhantes

Enfatiza as características de interesse (e suprime as outras)



Representação de classes em UML



Nome

Atributos

Operações

Professor J Clark

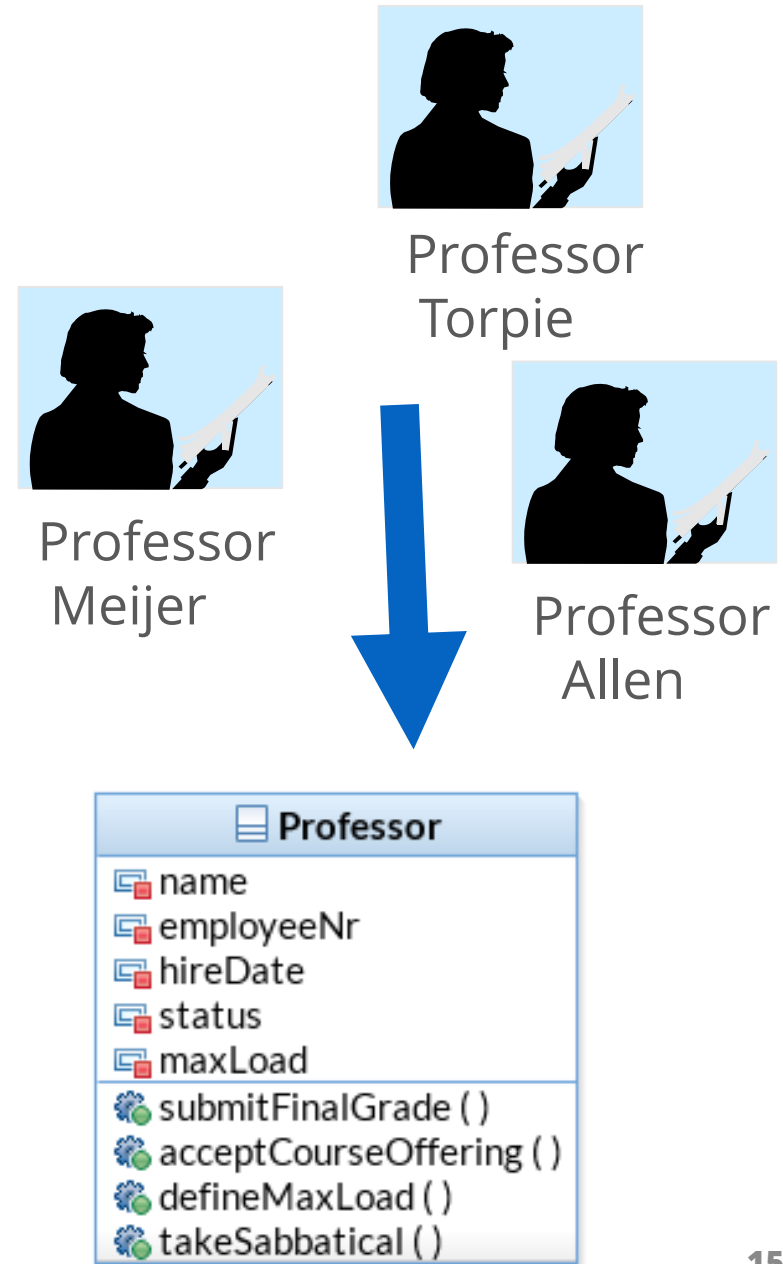
A relação entre classes e objetos

Uma classe é uma definição abstrata de um objeto

Define a estrutura e comportamento de cada objeto daquela classe/categoria.

Funciona como um molde para criar objetos.

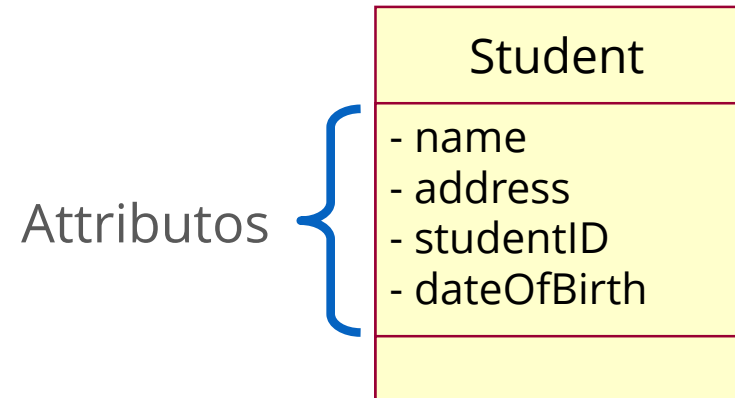
As classes não são coleções de objetos



O que é um atributo?

É uma propriedade de uma classe que descreve a gama de valores que as instâncias podem deter

Uma classe pode ter vários ou nenhum atributo



O que é uma operação?

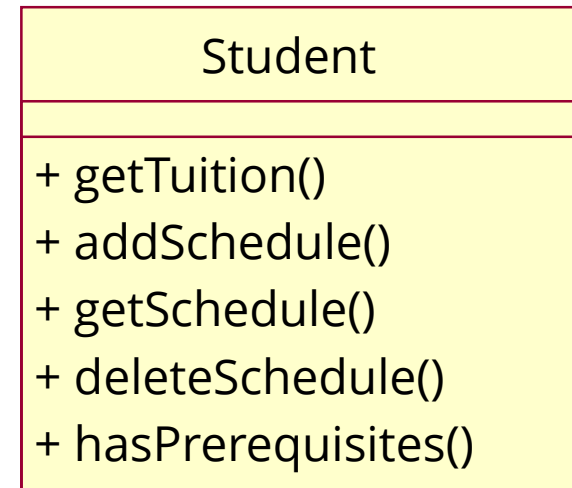
É a implementação de um serviço/ação que pode ser pedida a qualquer objeto de uma classe

É o que a classe sabe fazer

A classe pode ter várias ou nenhuma operação

Comuns: comandos e interrogações

Operações





O que é uma associação?

A relação semântica entre dois ou mais classificadores que especifica as ligações existentes entre as suas instâncias

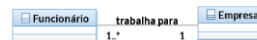
Uma relação estrutural que mostra que um tipo de objetos estão ligados a outro tipo de objetos



O que é a multiplicidade?

Nr de instâncias de uma classe que se relacionam com UMA instância da outra.

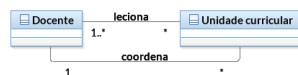
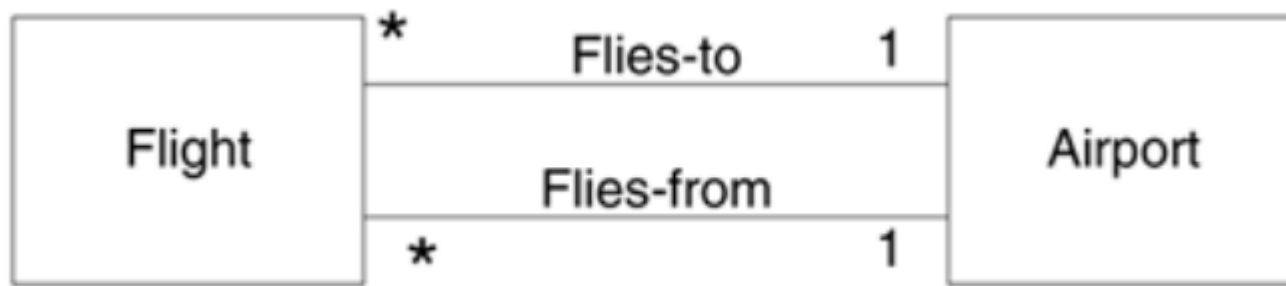
Para cada associação, avaliar cada um dos extremos



Indicação de multiplicidade

Unspecified	
Exactly One	1
Zero or More	0..*
Zero or More	*
One or More	1..*
Zero or One (optional scalar role)	0..1
Specified Range	2..4
Multiple, Disjoint Ranges	2, 4..6

Associações múltiplas entre duas classes

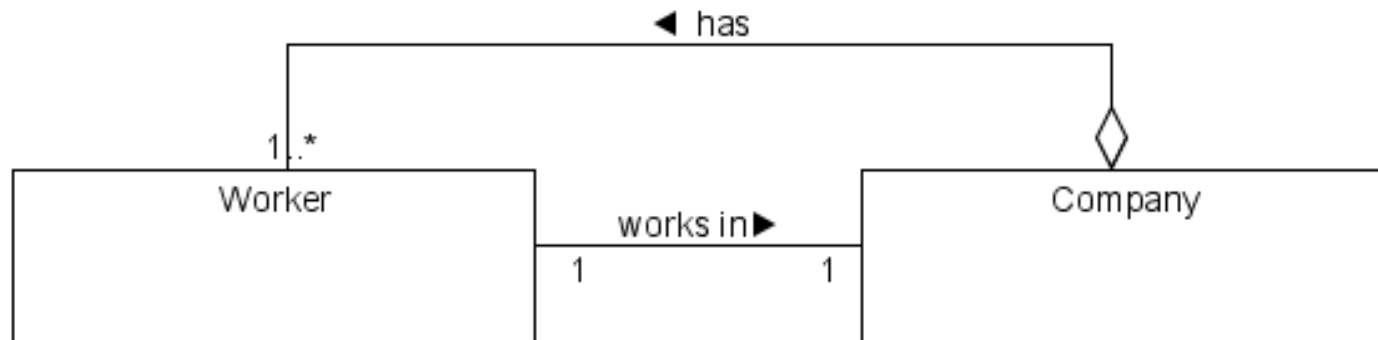


O que é uma agregação?

É uma forma especial de associação que modela uma relação de todo-parte, entre o agregador (“contentor”) e as suas partes constituintes

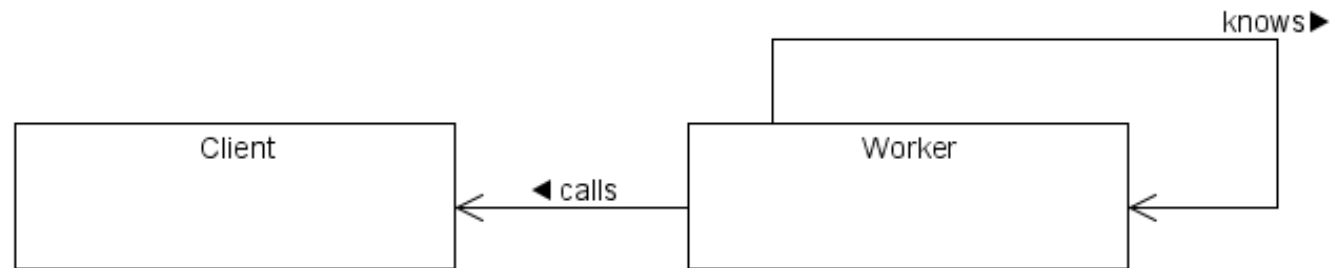
Pode ler-se “É parte de...”

A multiplicidade é usada como em outras associações



O que é a navegabilidade?

Indica a possibilidade de navegar de uma classe de partida para uma classe de chegada, usando a associação

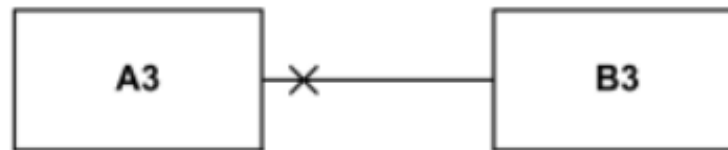




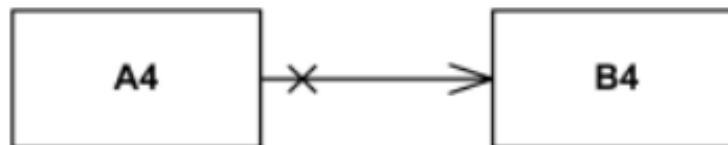
*Both ends of association have **unspecified navigability**.*



*A2 has **unspecified navigability** while B2 is **navigable from A2**.*



*A3 is **not navigable from B3** while B3 has **unspecified navigability**.*



*A4 is **not navigable from B4** while B4 is **navigable from A4**.*

O que é a generalização?

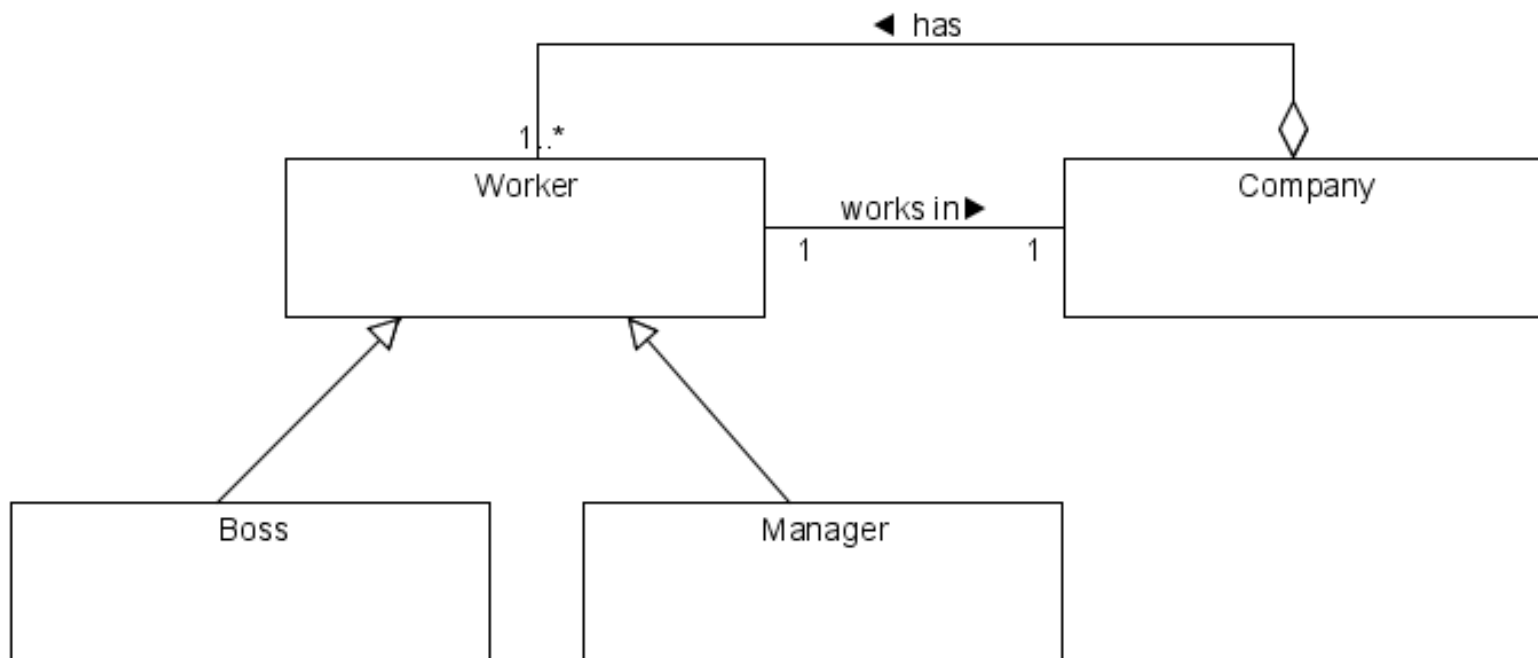
É relação entre classes em que uma especializa a estrutura e/ou comportamento de outra, partilhando todas as características

Define uma hierarquia em que a subclass herda das características da superclass

A subclasse pode sempre ser usada onde a superclasse é usada, mas não ao contrário.

Pode ler-se “é um tipo de”

Exemplo: especialização simples



O que é passado à subclasse?

A subclasse herda os atributos, operações e relacionamentos da superclasse

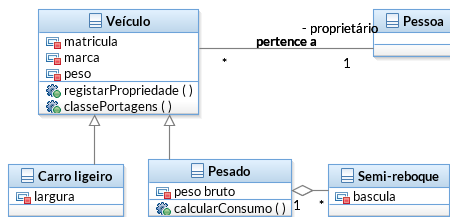
A subclasse pode:

Adicionar mais atributos, operações e relacionamentos à base herdada

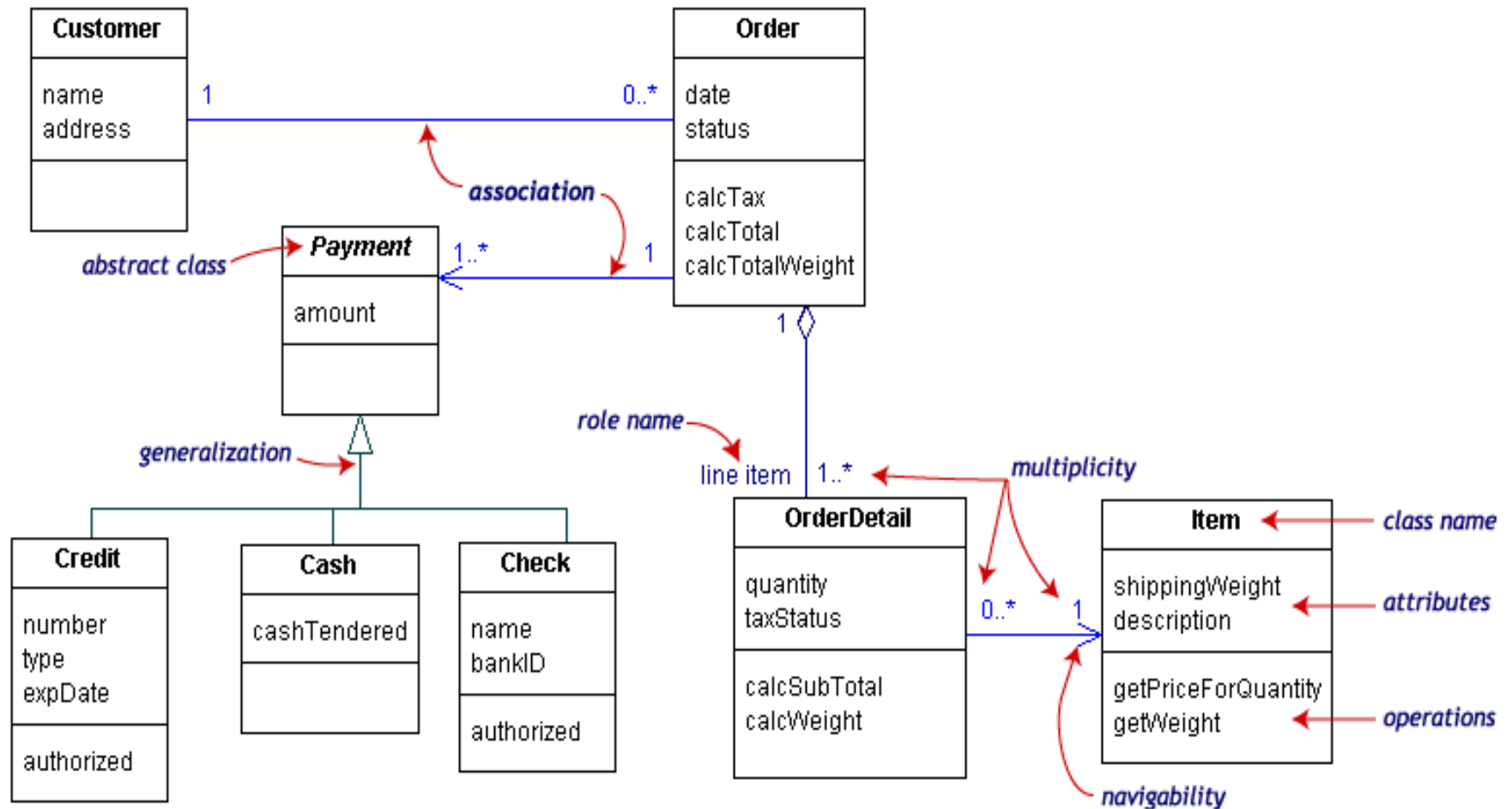
Redefinir as operações da superclasse

A herança põe em evidência as características comuns entre classes

O que é “herdado”?



Síntese da notação do diagrama de classes



Modelo do domínio

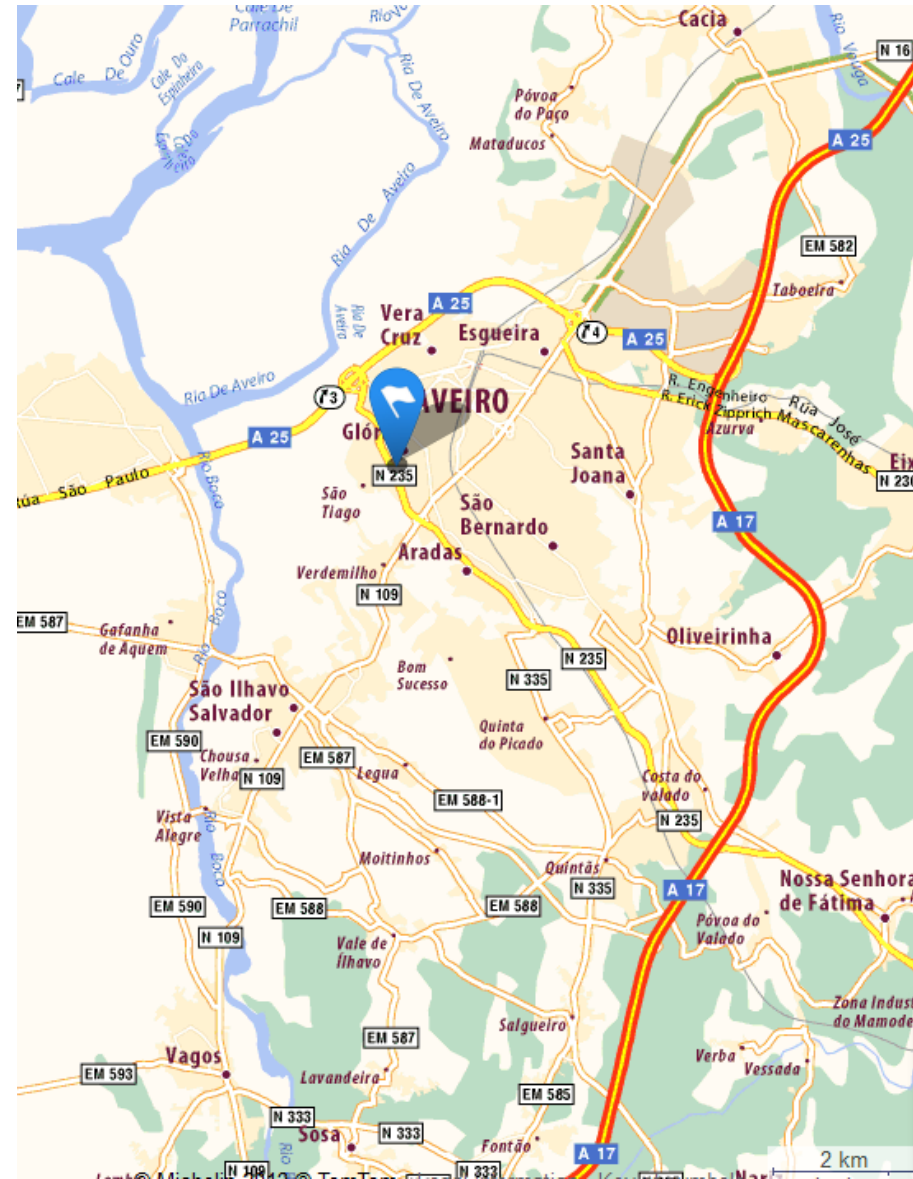
Mapa dos conceitos de um problema

O modelo do domínio é um mapa para os objetos

Mostra os conceitos de um problema
“Dicionário” visual

Perspetiva do “cartógrafo”
mostrar o que existe, com os nomes que a
“população” utiliza

Não é software!



Modelo do domínio em UML

Representado com um
diagrama de classes

Sem operações

Pode representar:

Objetos/conceitos do domínio

Associações entre esses
conceitos

Atributos (relevantes o
problema)

O modelo do domínio não tem a implementação

Classes de análise

Resultado da análise dos requisitos (**Analista**)

Neutro em relação à implementação

Não fornece diretamente o modelo de dados nem as classes de programação

Assunto de MAS

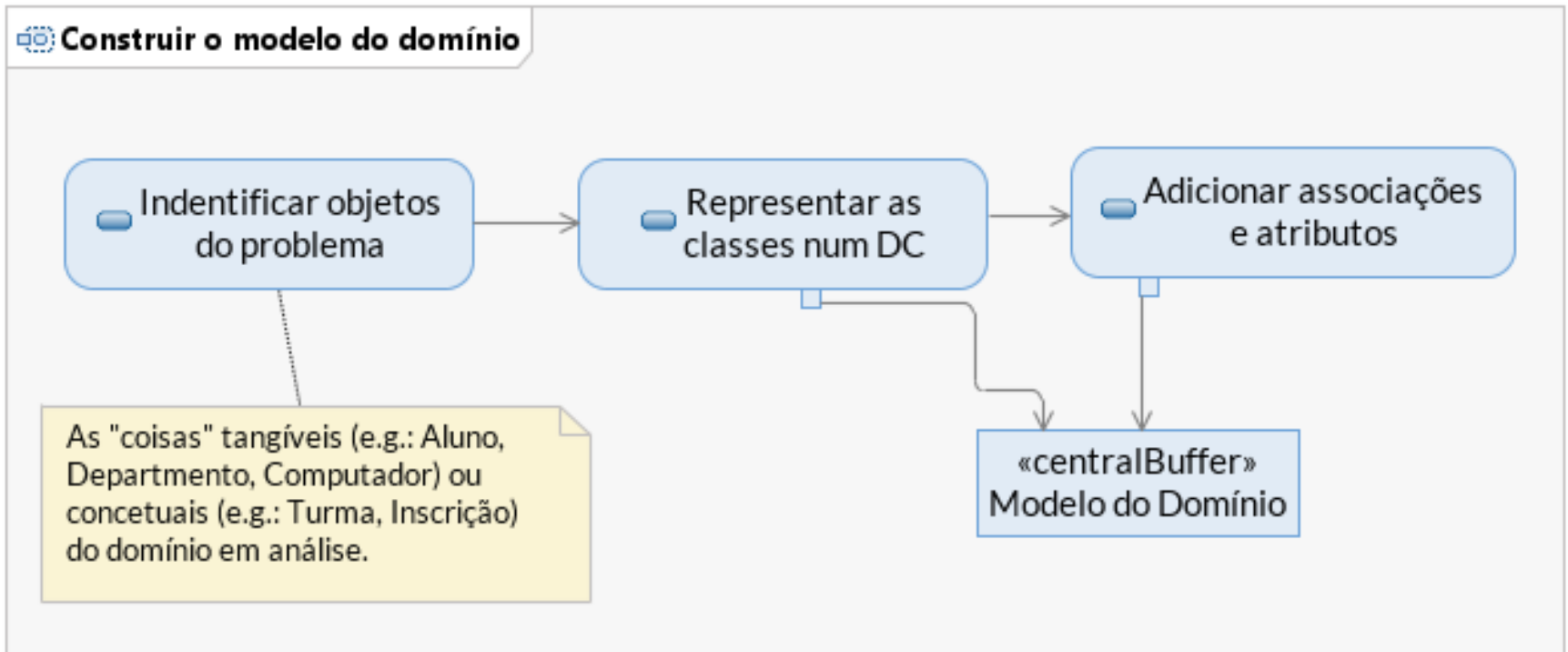
Classes em Java

Resultado do desenho/implementação (**Programador**)

Escritas numa linguagem concreta (OO)

Assunto de POO (e, em parte, de MAS)

Fluxo para criar o Modelo do Domínio



Como encontrar as classes?

Procurar numa lista de situações comuns

→ categorias de classes

Explorar documentos /relatórios existentes na área do problema

Análise de nomes

→ explorar descrições do problema à procura dos substantivos

Situações comuns para encontrar classes candidatas

Conceptual Class Category	Examples
product or service related to a transaction or transaction line item <i>Guideline:</i> Transactions are <i>for</i> something (a product or service). Consider these next.	<i>Item</i> <i>Flight, Seat, Meal</i>
where is the transaction recorded? <i>Guideline:</i> Important.	<i>Register, Ledger</i> <i>FlightManifest</i>
roles of people or organizations related to the transaction; actors in the use case <i>Guideline:</i> We usually need to know about the parties involved in a transaction.	<i>Cashier, Customer, Store</i> <i>MonopolyPlayer Passenger, Airline</i>
place of transaction; place of service	<i>Store</i> <i>Airport, Plane, Seat</i>
noteworthy events, often with a time or place we need to remember	<i>Sale, Payment MonopolyGame Flight</i>
physical objects <i>Guideline:</i> This is especially relevant when creating device-control software, or simulations.	<i>Item, Register Board, Piece, Die</i> <i>Airplane</i>

Conceito ou atributo?

Se não é natural pensar em X como um número ou texto, X provavelmente é uma classe e não um atributo

Sale
store

or... ?

Sale

Store
phoneNumber

Flight
destination

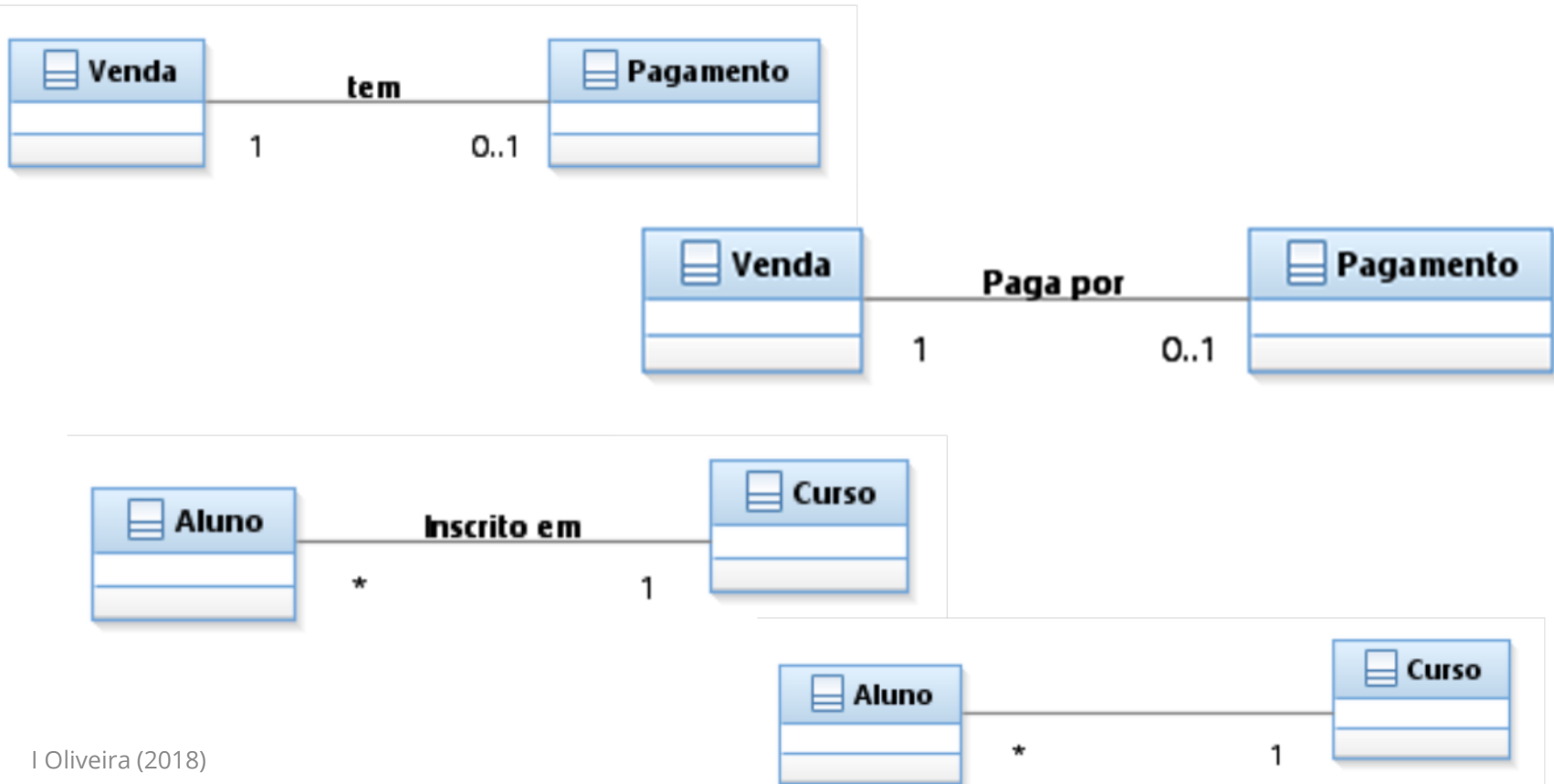
or... ?

Flight

Airport
name

Nomear as associações com Classe x Expressão verbal x Classe

A sequência deve ser legível e revelar o significado da associação



Associações comuns

Category	Examples
A is a transaction related to another transaction B	<i>CashPaymentSale</i> <i>CancellationReservation</i>
A is a line item of a transaction B	<i>SalesLineItemSale</i>
A is a product or service for a transaction (or line item) B	<i>ItemSalesLineItem (or Sale)</i> <i>FlightReservation</i>
A is a role related to a transaction B	<i>CustomerPayment</i> <i>PassengerTicket</i>
A is a physical or logical part of B	<i>DrawerRegister</i> <i>SquareBoard</i> <i>SeatAirplane</i>
A is physically or logically contained in/on B	<i>RegisterStore, ItemShelf</i> <i>SquareBoard</i> <i>PassengerAirplane</i>
A is a description for B	<i>ProductDescriptionItem</i> <i>FlightDescriptionFlight</i>

Exemplos de associações no modelo do domínio

O aluno lê um livro

Associação: A “usa”/ “interage com” B

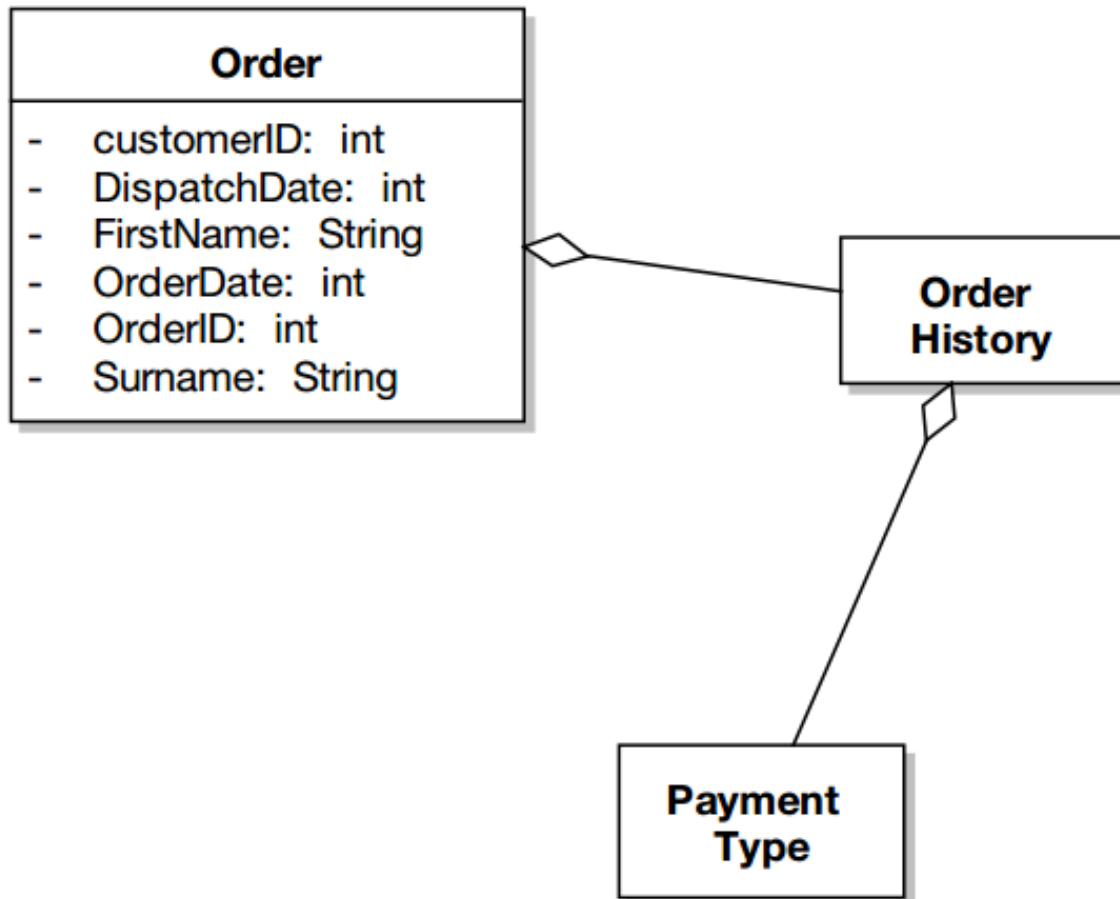
Uma biblioteca contém muitos livros

Agregação: A é um contendor/agregador de objetos B

O aluno é uma pessoa

Generalização: A generaliza as características de B/B especializa A (superclasse)

Figure 2-9 shows a domain model diagram with attributes on the Order class. What database-related problem does the diagram suggest?



- A clínica veterinária Faísca faz consultas a vários tipos de animais domésticos, e está em expansão na região de Aveiro.
- Em cada consulta só é visto um animal, por um veterinário da clínica.
- A clínica mantém a informação do dono do animal, incluindo nome, morada, telefone e NIF.
- Cada animal é tratado com elevados padrões de higiene, e tem uma ficha própria, em que se caracteriza o nome, género, espécie e data de nascimento.
- Os animais podem ser vacinados, de acordo com as vacinas indicadas para essa espécie.
- O Sr. Joaquim tem três cães, Tejo, Tamisa, Danúbio, que são pastores-alemães.

Referências

[PRE'10] Pressman, R. S. (2010). Software Engineering: a practitioners approach (seventh ed). McGraw Hill.

→ Chap. 5

[DEN'15] Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Systems analysis and design: An object-oriented approach with UML. John Wiley & Sons.

→ Chap. 3

[LAR'12] Larman, C. (2012). Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development. Pearson Education.

→ chap. 5.