

# Modelo de Conceitual

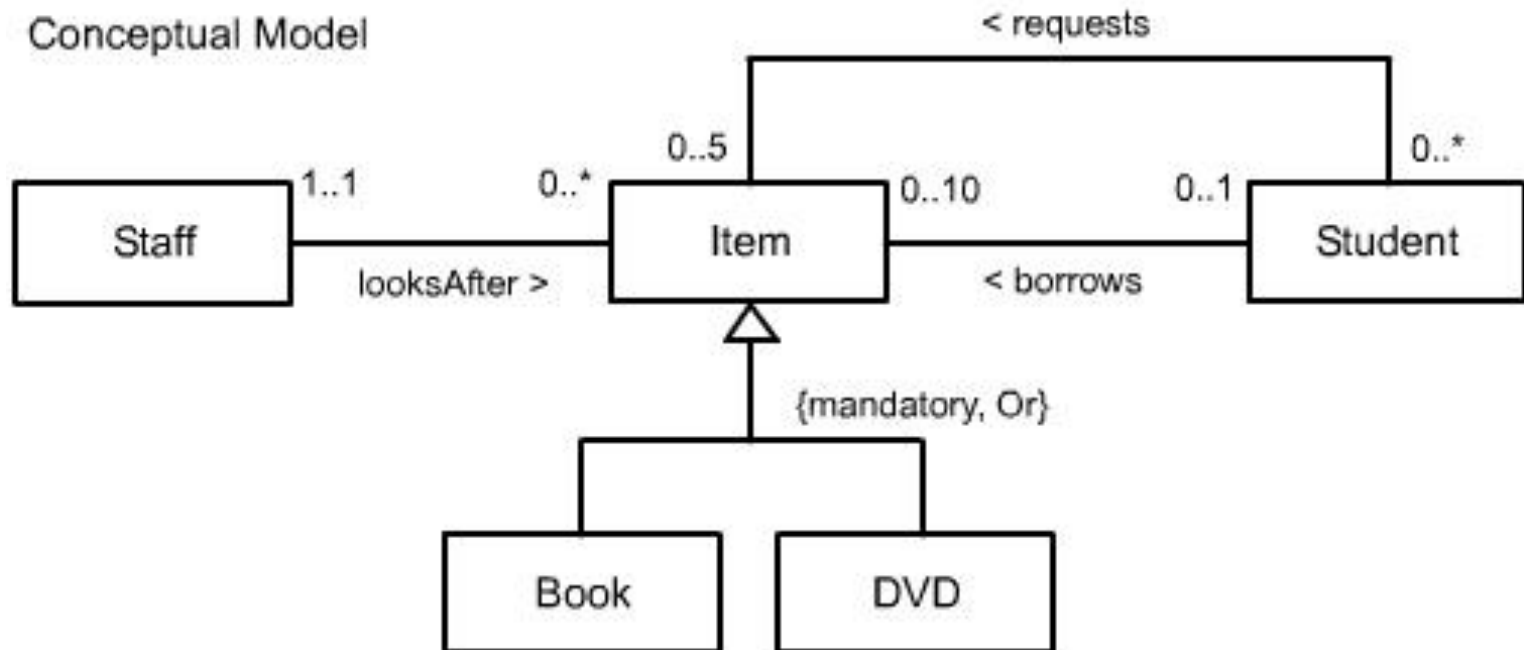
## Visualizando conceitos

### Objetivos

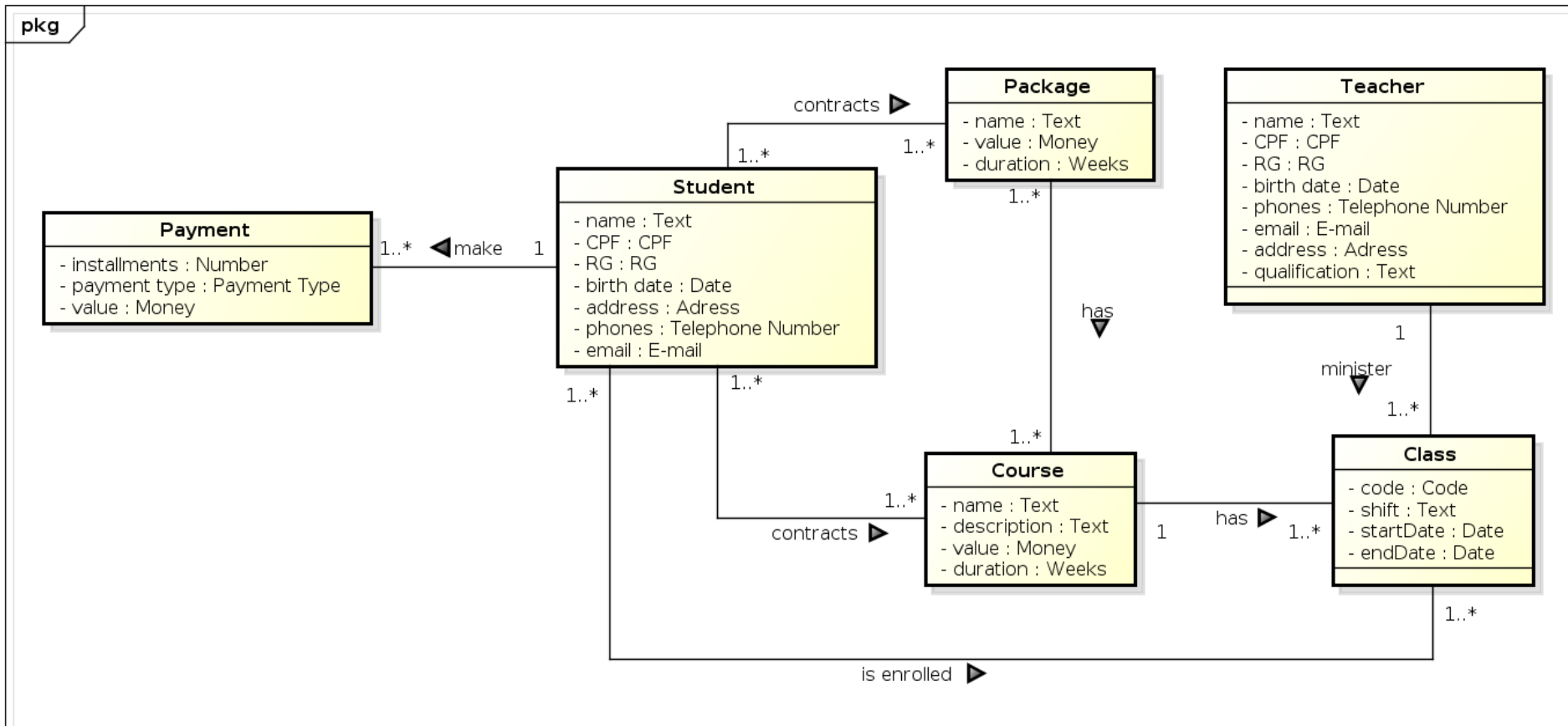
- Identificar conceitos relacionados aos requisitos atuais do ciclo de desenvolvimento
- Criar um modelo conceitual inicial (modelo de domínio)
- Distinguir entre conceitos, atributos e relacionamentos
- Entender os termos: conceito (concept), tipo (type), interface, e classe (class)

Baseado em: Larman, C. (2002) *Applying UML and Patterns – An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice-Hall Inc.

# Exemplo de Modelo de Conceitual: locadora



# Exemplo de Modelo de Conceitual: cursos pagos



# Um Modelo Conceitual ou de Domínio

---

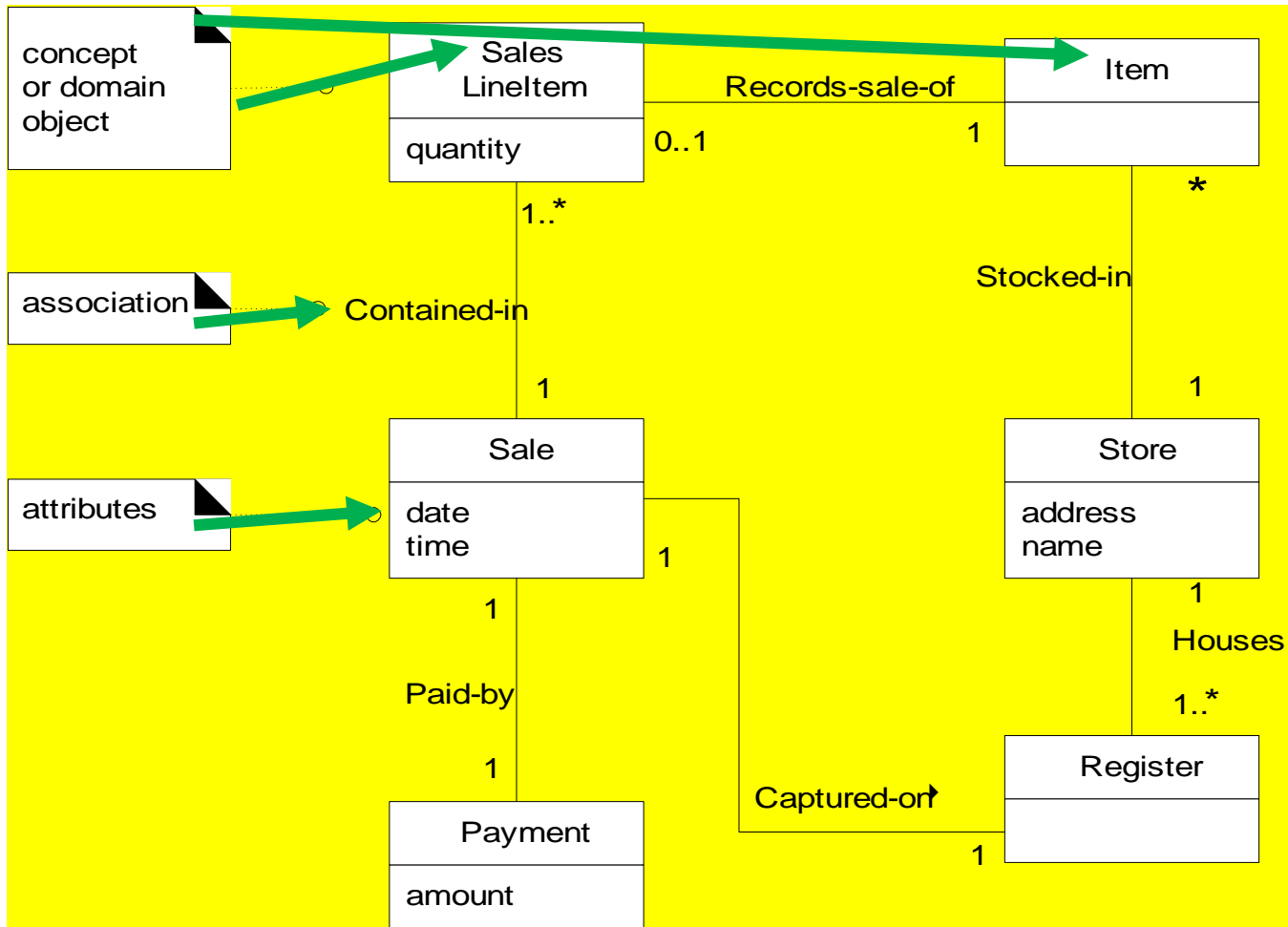
- Artefato mais importante para criar durante a análise orientada a objeto
- Ilustra classes conceituais significativas (*para o modelador*) no domínio do problema
- A UML contém notação na forma de diagramas de classe para ilustrar modelos de domínio
- É uma representação de classes conceituais do mundo real, não de componentes de software
  - Não descreve classes de software, nem objetos com responsabilidades

# Um Modelo Conceitual ou de Domínio

---

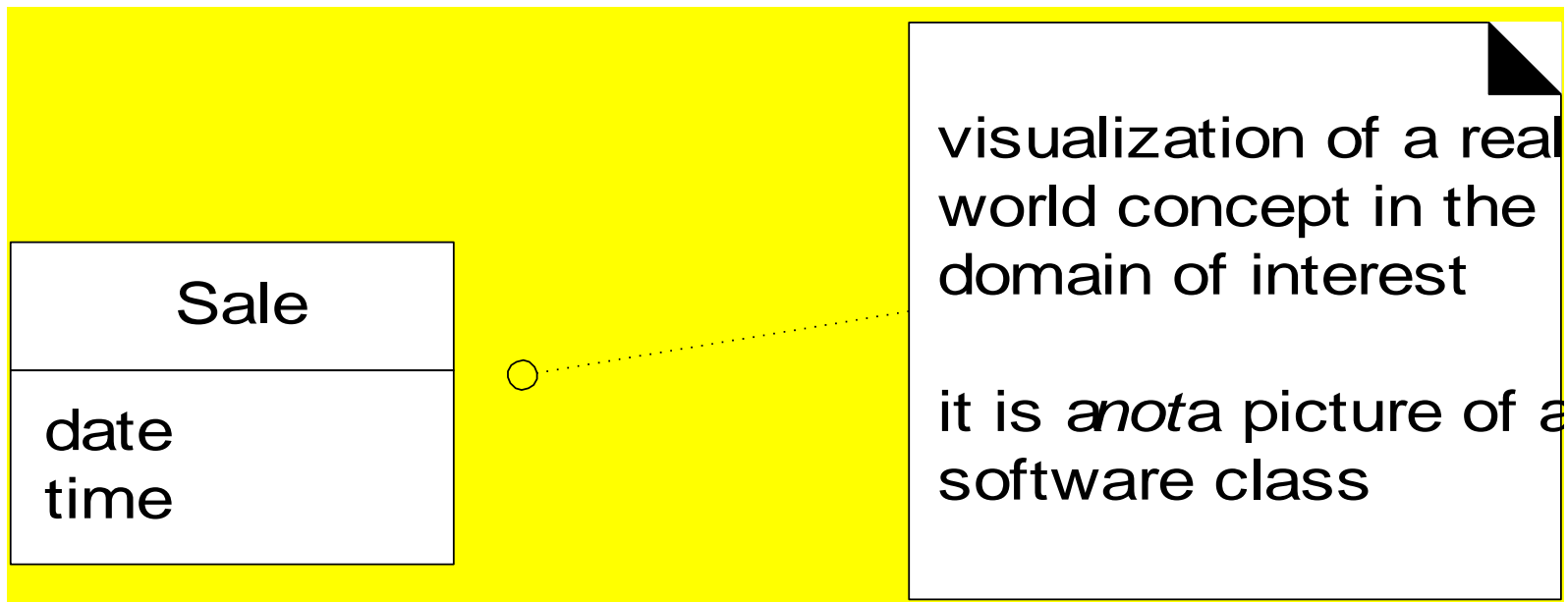
- É uma representação visual de classes conceituais ou objetos do mundo real em um domínio de interesse
- Usando a notação UML, o modelo de domínio é ilustrado com um conjunto de diagramas de classes nos quais não são definidas operações. Ele deve mostrar:
  - **Objetos** do domínio ou classes conceituais
  - **Associações** entre classes conceituais
  - **Atributos** de classes conceituais

# Modelo de Domínio Parcial – um Dicionário Visual do domínio (Ponto de Venda)

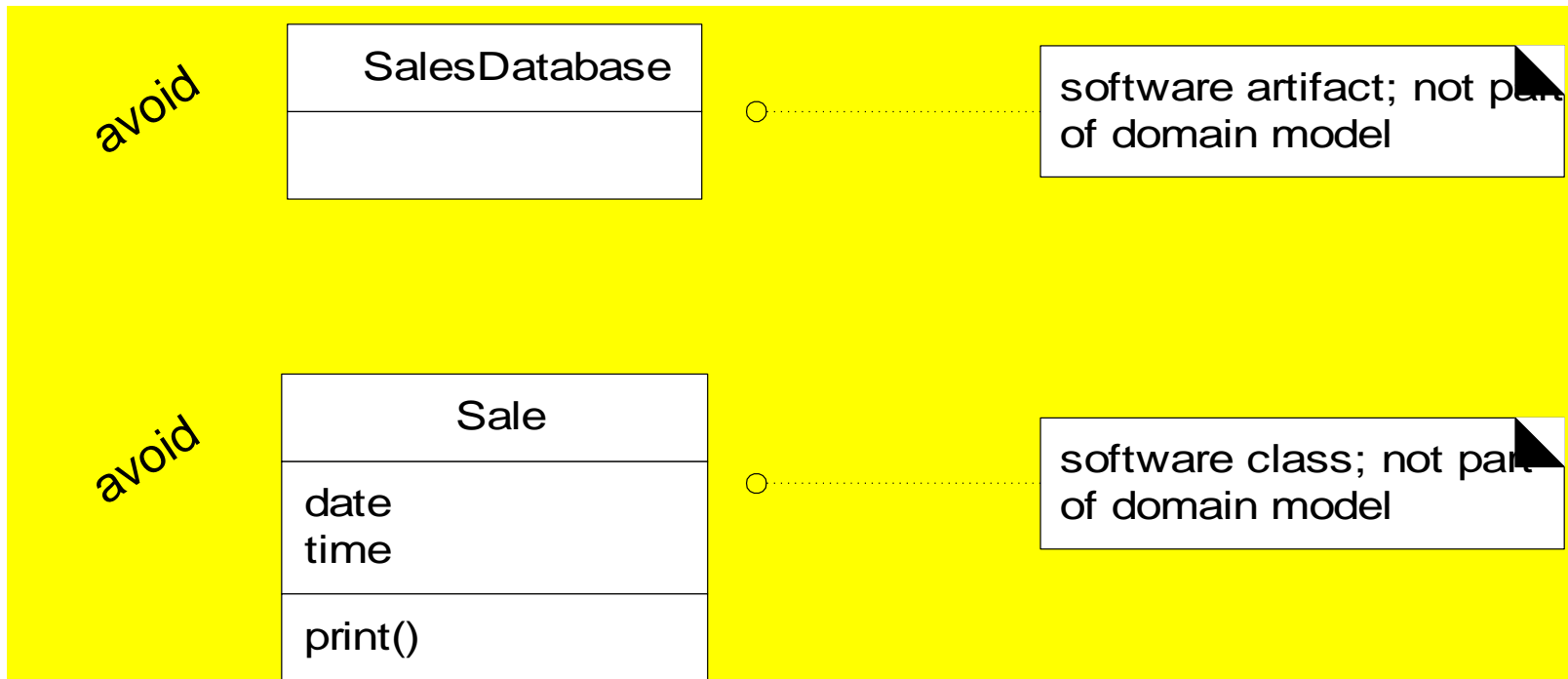


# Um Modelo de Domínio mostra classes conceituais do “mundo real”

---



# O Modelo de Domínio não mostra artefatos de software ou classes



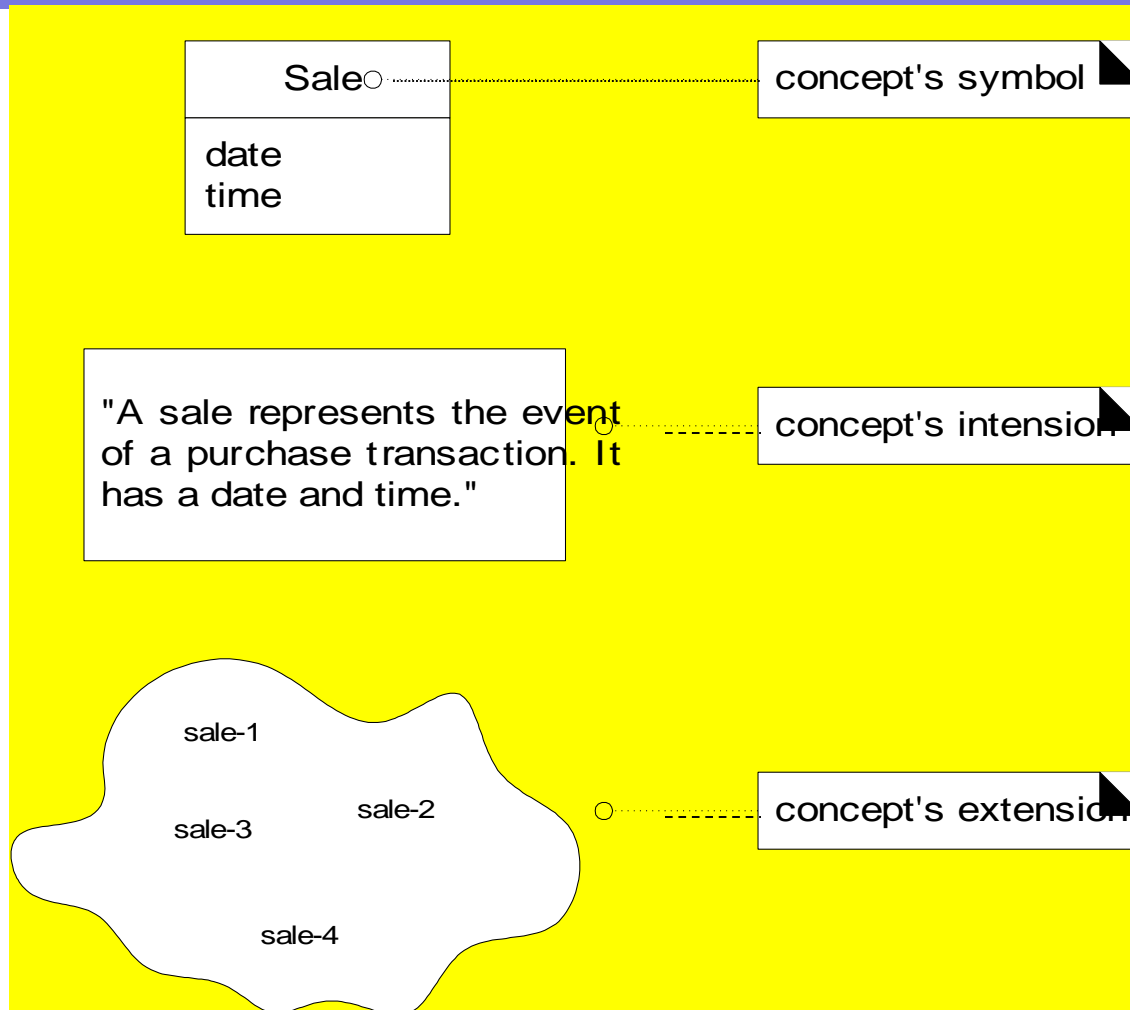


# Classes conceituais (*concept*)

---

- Informalmente são ideias, coisas ou objetos
- Mais formalmente elas podem ser consideradas em termos de seus **símbolos**, **intenção** e **extensão**
  - **símbolo**: palavras ou imagens representando uma classe conceitual
  - **intenção**: a definição da classe conceitual
  - **extensão**: o conjunto de exemplos para os quais a classe conceitual se aplica
  - Ex. A classe conceitual para uma transação de compra

# Uma classe conceitual possui: símbolo, intenção e extensão



# Associações e Atributos

- Associação:

É um relacionamento entre tipos (mais especificamente, instancias dos tipos) que indica alguma conexão significativa; Associações importantes geralmente implicam conhecimento de um relacionamento que precisa ser preservado;



# Associações e Atributos

---

Atributo:

- É o valor lógico de um objeto
- Incluir os seguintes atributos:
  - Aqueles para os quais os requisitos sugerem ou implicam uma necessidade de lembrar informação  
*ex. Um recibo normalmente inclui uma data, hora, e a gerência quer conhecê-los*
- Intuitivamente os tipos de atributos mais simples são tipos de dados primitivos [boolean, date, number, string, time]

# Modelos de Domínio e Decomposição

---

- Problemas de Software podem ser complexos
- Decomposição – “*divide and conquer*” –
  - é uma estratégia comum
- Em **análise estruturada** a dimensão de decomposição é por processos ou funções
- Em **análise orientada a objeto** a dimensão de decomposição é por coisas e entidades no domínio

# Classes Conceituais no Domínio de Vendas

---

**Importante:** 1. Uma tarefa inicial da análise é identificar diferentes conceitos no domínio do problema e documentar os resultados no modelo de domínio



Modelo parcial de domínio

\*\* Uma vez identificados os conceitos, é possível 2. mapear relações entre conceitos, e 3. propriedades desses conceitos.

# Como criar um Modelo Conceitual?

---

# Como criar um Modelo Conceitual?

Use a estratégia do cartógrafo





# Como criar um Modelo Conceitual?

Use a estratégia do cartógrafo



# Como criar um Modelo Conceitual?

---

## Use a estratégia do cartógrafo

Um modelo de domínio é um tipo de **mapa de conceitos** ou coisas em um domínio

- Use nomes existentes no território
- Exclua características (*features*) ou detalhes irrelevantes
- Não adicione coisas que não estejam lá

# Identificação de Classes Conceituais

---

- Estratégias para identificar classes conceituais:
  - 1. Use uma lista de categorias para classes conceituais
  - 2. Identifique frases nominais

# Identificação de Classes Conceituais

---

- Estratégias para identificar classes conceituais:
  - 1. Use uma lista de categorias para classes conceituais
  - **2. Identifique frases nominais**

## **Abra o Sistema DSC**

- **Veja a descrição do seu problema e cenário**
- **Veja seu Diagrama de Casos de Uso Essencial**
- **Veja a sua Lista de atributos que a solução precisa TER e SER.**

# Identificando Frases Nominais

---

- Análise Linguística:
  - Identificar os nomes e frases nominais em descrições textuais do domínio, e considerá-las candidatas a classes ou atributos
  - Os casos de uso são úteis para isso

Cenário Principal de Sucesso (exemplo de Ponto de Venda):

1. ***Customer*** arrives at a ***POS checkout*** with ***goods*** and or ***services*** to purchase.
2. ***Cashier*** starts a new ***sale***.
3. ***Cashier*** enters ***item identifier***. (...)

# Lista de Categorias para Classes Conceituais

1. Objetos físicos ou tangíveis [Máquina Registradora, Avião]
2. Especificações ou descrições de coisas [EspecificacaoProduto, DescricaoVoo]
3. Locais [Loja, Aeroporto]
4. Operações [Venda, pagamento, Reserva]
5. Itens de linha de transação [ItensVenda]
6. Funções de pessoas [Caixa, Piloto]
7. Recipientes de outras coisas [Estoque, Cesta, Avião]
8. Coisas em um recipiente [Item, Passageiro]
9. Outros sistemas (de computador, eletromecânicos) externos ao sistema [SistemaAutorizaçãoPagamentoCredito, ControleTrafegoAereo]

# Lista de Categorias para Classes Conceituais

10. Conceitos substantivos abstratos [Fome, Aerofobia]
11. Organizações [DepartamentoVendas, EmpresaAerea]
12. Eventos [Venda, Pagamento, Voo, Decolagem]
13. Processos [VenderProduto, ReservarAssento]
14. Regulamentos e Políticas [PolíticasDevolucao, PolíticasCancelamento]
15. Catálogos [CatalogoProduto, CatalogoDestinos]
16. Registos das finanças, contratos [Recibo, ContratoEmprego]
17. Instrumentos e serviços financeiros [LinhaCredito, Bolsa]
18. Manuais, documentos, livros [ListaPrecosDiarios, ManualConserto]

# Candidatos a Classes Conceituais para o Domínio de Vendas

---

- Da Lista de Categorias e análise de frases nominais, uma lista de candidatos é gerada

Ex. lista restrita ao cenário do *Process Sale*

- *Register*
- *Item*
- *Store*
- *Sale*
- *Payment*
- *ProductCatalog*
- *Productpecification*
- *SalesLineItem*
- *Cashier*
- *Customer*
- *Manager*



# Recomendações para Modelagem do Domínio

---

1. Liste as classes conceituais candidatas usando as 2 técnicas, considerando os requisitos atuais
2. Desenhe-as em um Modelo de Domínio
3. Adicione as associações necessárias para registrar relacionamentos necessários para preservar a memória (lembrar das coisas, armazenar informação)

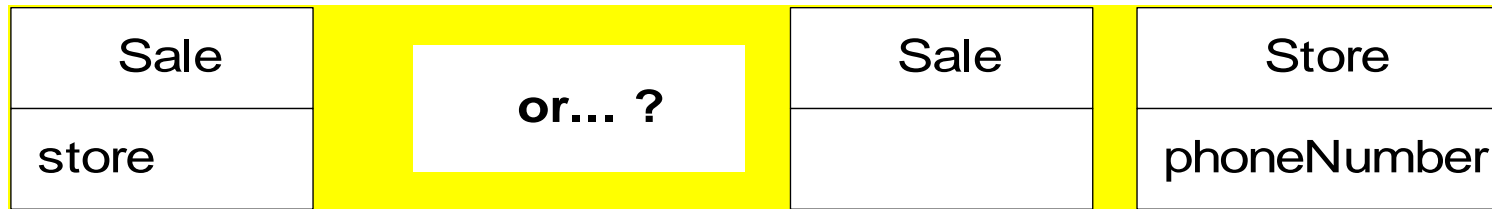
Ex: Uma Venda tem vários produtos.

4. Adicione os atributos necessários para satisfazer os requisitos (detalhes que precisam ser adicionados, armazenados, etc.)

Ex: Um produto tem código, descrição, marca, cor, etc.

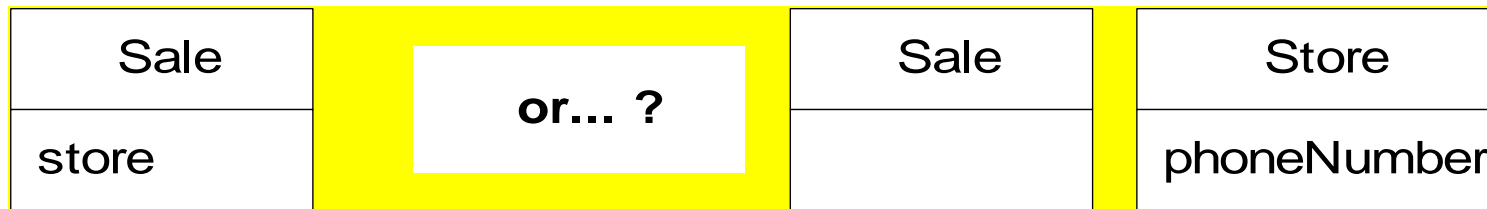
# ***Store*** deveria ser um atributo de ***Sale***, ou uma classe conceitual separada?

---



# ***Store* deveria ser um atributo de *Sale*, ou uma classe conceitual separada?**

---

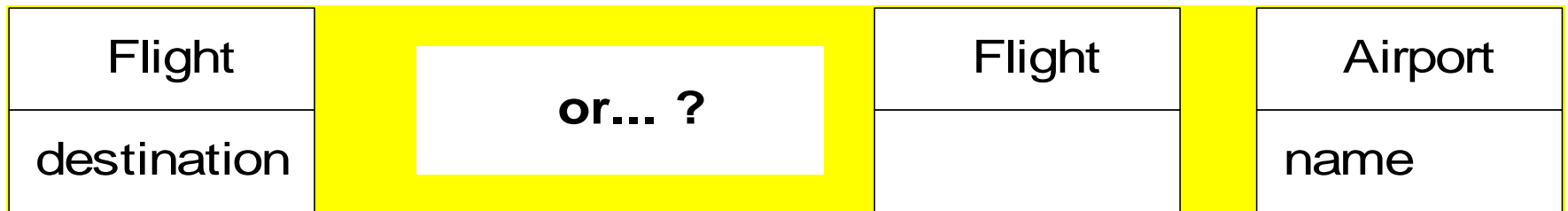


## **Importante:**

Se não pensamos em um dado elemento “X” como um número ou texto no mundo real, “X” é provavelmente uma classe conceitual, não um atributo.

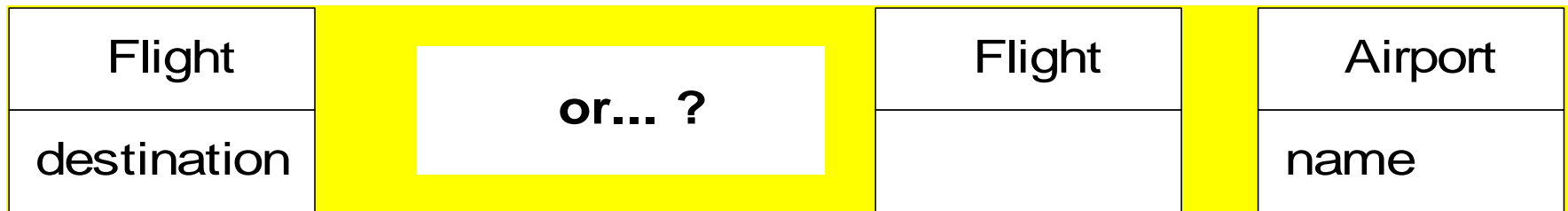
# ***Destination*** deveria ser um atributo de ***Flight*** ou uma classe conceitual separada?

---



# ***Destination*** deveria ser um atributo de ***Flight*** ou uma classe conceitual separada?

---



Sugestão:

Na dúvida, faça-o um conceito separado.

Atributos devem ser usados com parcimônia no modelo de domínio

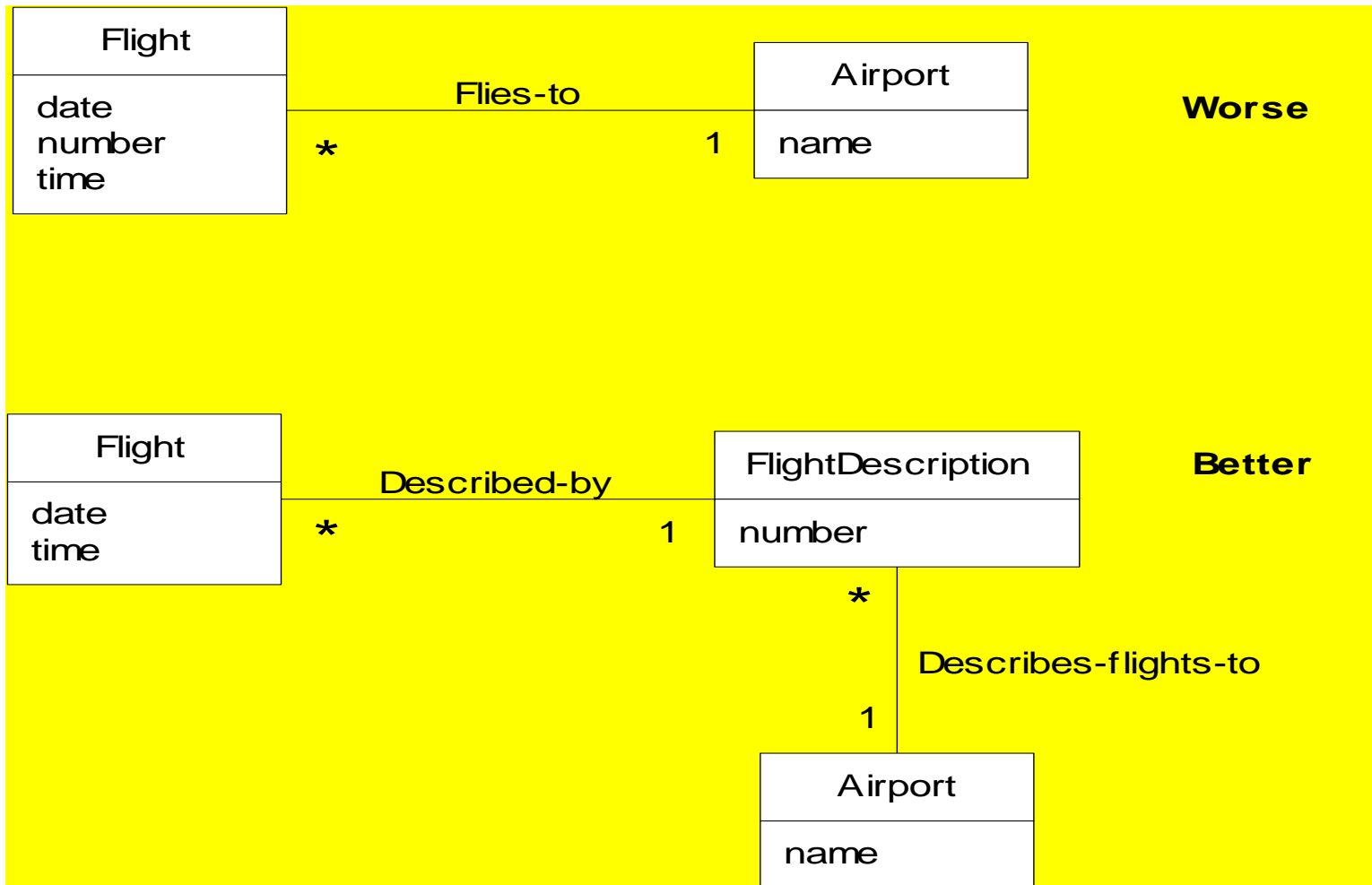
# Classes de especificação:

---

## Criar quando:

- Deve haver uma descrição sobre um item ou serviço, independente da existência atual de exemplos desses itens ou serviços;
- Deletar instâncias de coisas que eles descrevem (ex. item) resulta em perda de informação que necessita ser mantida;
- Reduzir informação redundante ou duplicada.

# Especificações sobre outras coisas



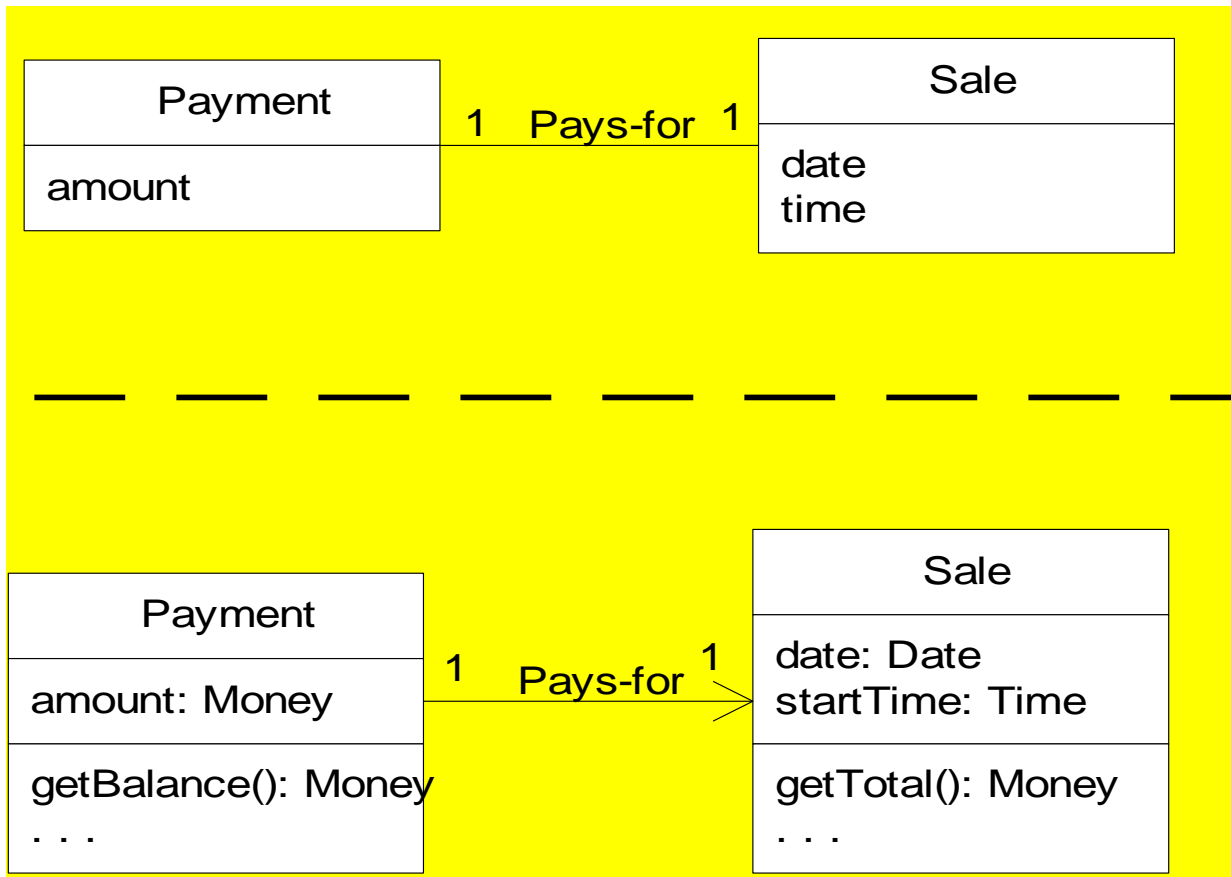
# UML, Modelos e Métodos

---

- A UML simplesmente descreve tipos de diagramas brutos
- A mesma notação é usada para 3 perspectivas e tipos de modelos:
  - **Essencial ou conceitual:** os diagramas são interpretados como descrevendo coisas no mundo real ou domínio de interesse
  - **De especificação:** os diagramas são interpretados como descrevendo abstrações de software ou componentes
  - **De implementação:** os diagramas são interpretados como descrevendo implementações de software, em uma linguagem específica (ex. Java)



# UML aplicada em diferentes perspectivas e modelos



# Referências

---

- Larman, C. (2002) *Applying UML and Patterns – An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice-Hall Inc.
- Muller, P.A. (1997) *Instant UML*, Wrox Press Ltd.

# Atividade

---

- Criar o Modelo Conceitual/de Domínio para o Projeto