ДОМЕЙН МОДЕЛ

- Атрибути и асоциации

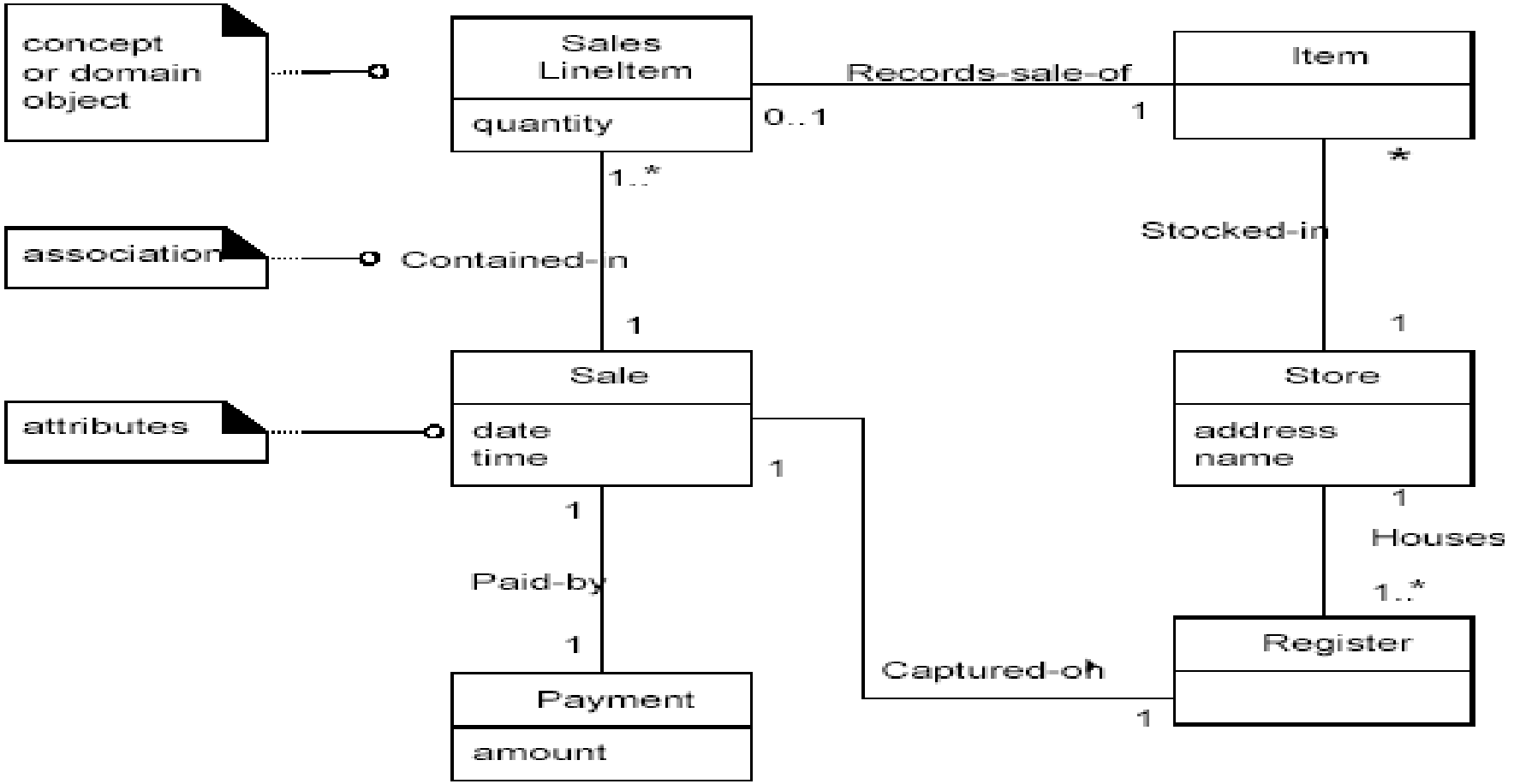
Съдържание

- Концептуални класове
- Създаване на първоначален домейн модел
- Атрибути
- Асоциации

Домейн модел

- Домейн модел - визуално представяне на концептуални класове или реални обекти в домейна
- Илюстрира:
  - Домейн обекти или концептуални класове
  - Връзки между концептуалните класове
  - Атрибути на концептуалните класове
- Домейн моделът представя реални концептуални класове, а не софтуерни компоненти.

Пример - част от домейн модел



Стратегии за определяне на концептуални класове

- По-добре е да се определи домейн модел с много подробно описани концептуални класове отколкото да не се дефинира достатъчно ясно.
- Стратегии за определяне на концептуални класове
  - Използване на концептуални класове от списък с категории
  - Идентифициране на съществителните
  - Reuse или модификация на съществуващи модели

Концептуални класове от списък с категории -1

Категория концептуален клас	**Пример**
Физически или осезателни обекти	Регистрация, самолет
Изисквания, планове или описания на предмети	Описание на полета
Места	Летище
Бизнес транзакции	Продажба, покупка, резервация

Концептуални класове от списък с категории - 2

Категория концептуален клас	**Пример**
Основни елементи от транзакция	Основен елемент на продажба
Роли на хора	Продавач, клиент, пътник
Контейнери	Склад, Кабина, Самолет
Неща от контейнер	Пътник, продукт (item)

Концептуални класове от списък с категории - 3

Категория концептуален клас	**Пример**
Друга система	Разплащателна система
Абстрактни съществителни концепции	Глад
Организации	Отдел”Продажби”
Събития	Продажба, Приземяване

Концептуални класове от списък с категории – 4

Категория концептуален клас	**Пример**
Процеси	Продаване на билет
Правила и политики	Политика за отказване на билет
Каталози	Каталог на продуктите, на полетите
Събития	Продажба, Приземяване

Концептуални класове от списък с категории – 5

Категория концептуален клас	**Пример**
Записи за финанси, работа, договори	Касова бележка
Финансови инструменти и услуги	Кредит
Наръчници, документи, препратки, книги, вестници	Дневна тарифа на билетите

Идентифициране на съществителните - Пример

Основен успешен сценарий (или основни дейности):

1. **Клиентът** пристига на **POS касата** с **покупките** и/или **услугите** за закупуване.
2. **Касиерът** стартира нова **продажба.**
3. **Касиерът** маркира **артикулите идентификатори**.
4. Системата записва **ред** от **продажби** и показва **описанието** на **атрибута, цена** и изчислява общата **сума**. Цената се пресмята от определено множество от цени.

Касиерът повтаря стъпка 2-3 докато се извърши посоченото.

5. Системата показва общата сума пресметната с **данъка**.
6. Касиерът уведомява клиента за сумата и пита за **заплащането.**
7. Клиентът се разплаща и системата управлява заплащането.
8. Системата регистрира цялата **продажба** и изпраща информация за продажбата и заплащането към външното **счетоводство** (за счетоводство и **комисионни**) и **инвентарната** система (за обновяване на инвентара).
9. Системата дава **касова бележка.**
10. Клиентът си тръгва с касовата бележка и стоките.

Кандидати за концептуални класове - Пример

- *Опис* (Register)
- *Атрибути* (Item)
- *Склад* (Store)
- *Продажба* (Sale)
- *Заплащане* (Payment)
- *Каталогизиране на продукти* (ProductCatalog)
- *Спецификация на продукт* (ProductSpecification)
- *Ред от продажби* (SalesLineItem)
- *Касиер* (Cashier)
- *Клиент* (Customer)
- *Управител (*Manager)

Създаване на Домейн модел

- Изброяване на кандидатите за концептуални класове, на базата на списъка с категории и техниката със съществителни, които са свързани с разглежданите изисквания
- Рисуване на Домейн модел
- Добавяне на необходимите връзки, които представят информация, която не трябва да се загуби
- Добавяне на необходимите атрибути

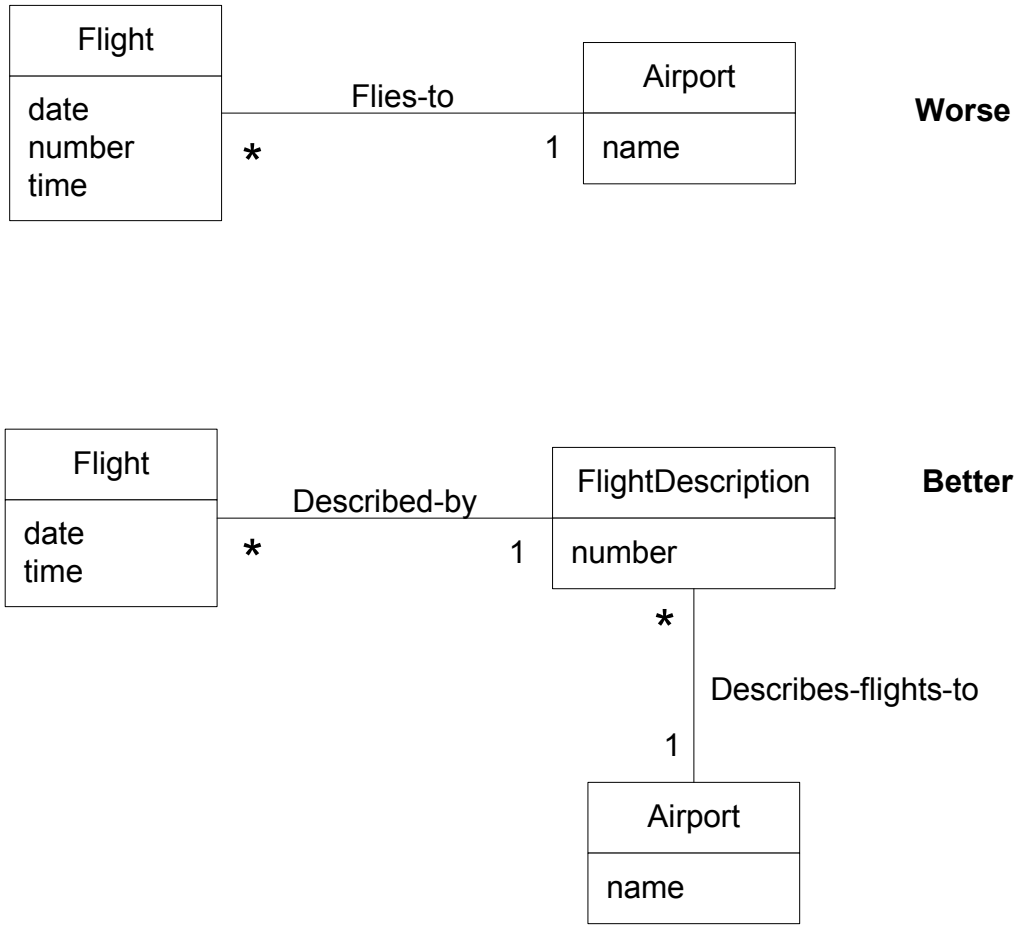
Именуване и моделиране

- Използване на съществуващи имена от областта
- Изключване на несъществената информация
- Не се добавят елементи, които не са свързани с областта
  - Стратегия на картографа

Грешки при определяне на концептуалните класове

- Представяне на нещо като атрибут, когато то може да е концепция.
  - Правило : *Ако не мислим за някой концептуален клас X като число или текст в реалния свят, то X вероятно е концептуален клас, а не атрибут.*

Описателни (description) класове



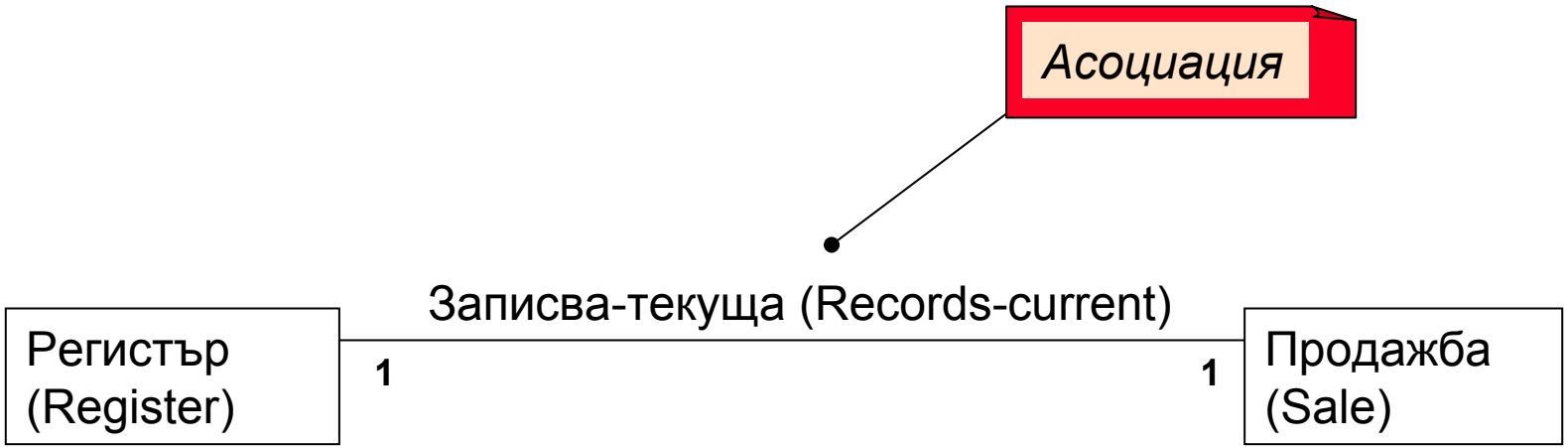
Описателните или специфициращите класове са тясно свързани с обектите, които описват.

Необходимост от описателни класове

- Има нужда да се опишат атрибути или услуги, независимо от текущото наличие на примери от тези атрибути или услуги
- С изтриването на елементите, които те описват, се губи информация, която трябва да се поддържа
  - поради неправилна асоциация на информацията с изтрития елемент
- Намалява излишествата или дублирането на информация

Асоциации - дефиниция

Асоциациите са отношения между *класове*(представители на класове), които представят смислена и интересна *връзка*.

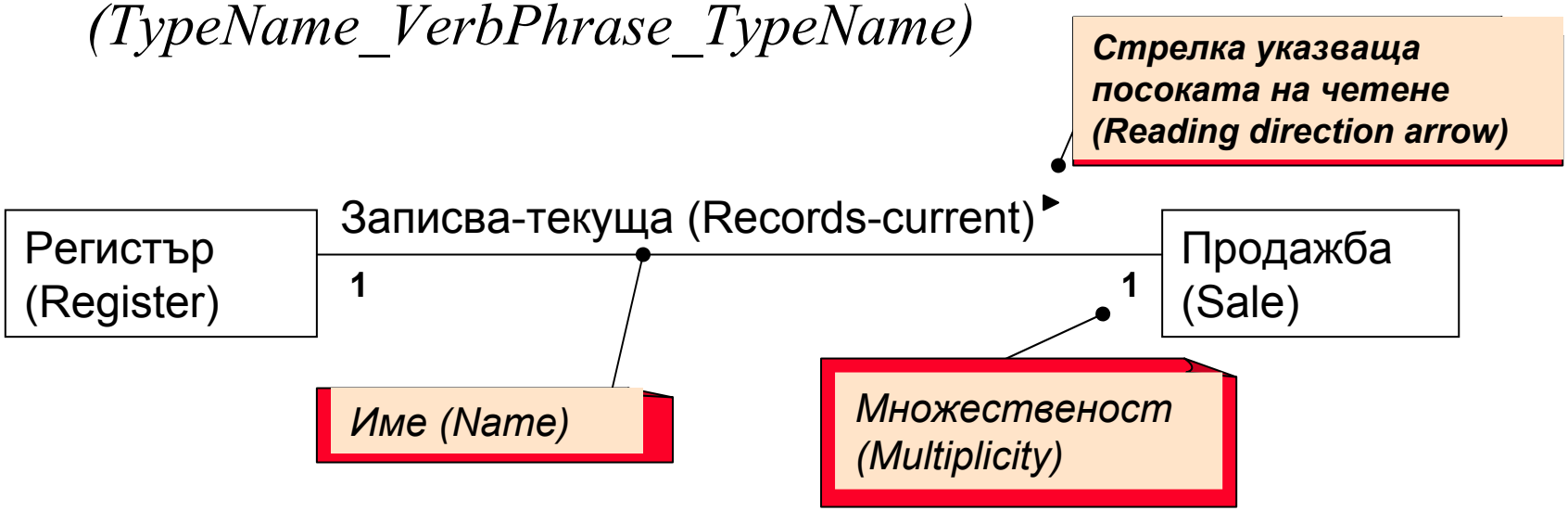


Критерии за асоциации

- *Асоциации*, за които знанията от *връзката* е необходимо да се запазят за неопределено време - асоциации "*нужно-да-знам*" (need-to-know).
- Асоциации от ***Общия списък с Асоциации***.
- Във *домейн модел*, където имаме n разични *концептуални класове* потенциално от един *концептуален клас* може да има n-1 *асоциации* до други концептуални класове
- Добавянето на много линии в диаграмата създава визуален шум и я прави трудно достъпна и неразбираема

Асоциации - UML нотация

- Представят се чрез линия
- Асоциациите са двупосочни
- Стрелка указва посоката на четене
- Име и Изрази за множественост
- *ИмеНаТип_Глагол_ИмеНаТип (TypeName_VerbPhrase_TypeName)*



Общ списък с асоциации:

Категория	***Пример***
А е физическа част от Б.	Крило – Самолет
А е логическа част от Б.	Закупена стока – Продажба
А е физически свързана с Б.	Пътник – Самолет
А е логически свързана с Б.	Полет – Разписание
А е описание на Б.	Описание на стока – Стока
А е част от транзакцията Б.	Закупена стока – Продажба
А е известен / записан / отчетен като Б.	Резервация за полет – Списък с пътниците
А е член на Б.	Пилот – Самолет
А е организационна поделиница на Б.	Отдел – Магазин Поддръжка – Самолет

Общ списък с асоциации:

Категория	***Пример***
А използва / управлява Б.	Пилот – Самолет
А комуникира с Б.	Клиент – Касиер
А е свързан с транзакцията Б.	Пътник – Билет
А е транзакция свързана с друга транзакция Б.	Резервация – Прекратяване на резервация.
А е до Б.	Закупена стока – Закупена стока
А се притежава от Б.	Самолет – Авиолиния
А е свързано чрез събите със Б.	Продажба – Клиент Продажба – Магазин

Високо приоритетни асоциации

- А е физическа част от Б.
- А е физически / логически свързано с Б.
- А се записва в Б.

Насоки

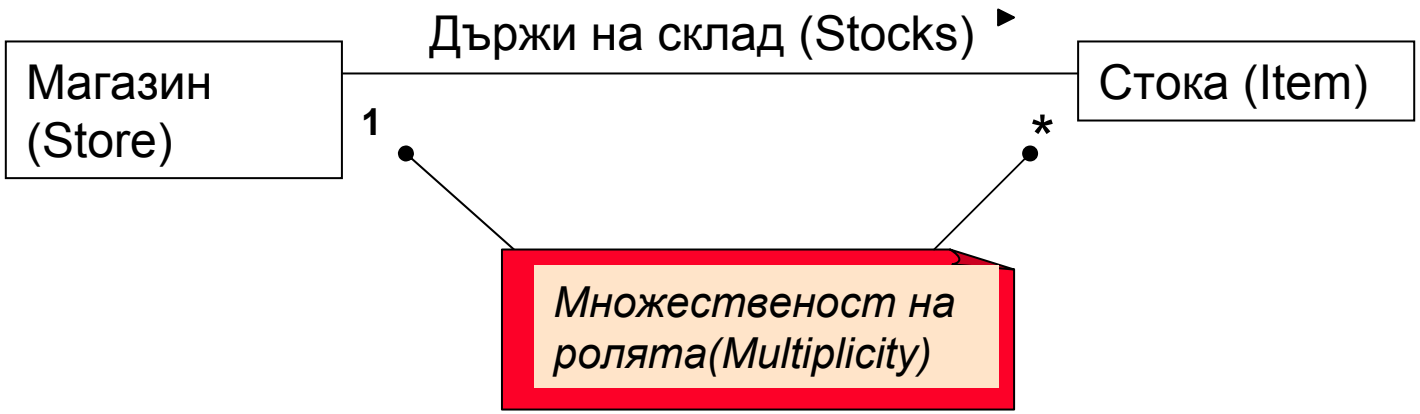
- Фокусиране върху “***нужно-да-знаеш***” асоциации.
- По-важно е да се дефинират *концептуалните класове*, отколкото да се определят *асоциациите*.
- Твърде много *асоциации*, са предпоставка за объркване на *домейн модела*, отколкото да го илюстрират.
- Избягване излишните *асоциации* и тези които могат да се извлекат от други.

Роли

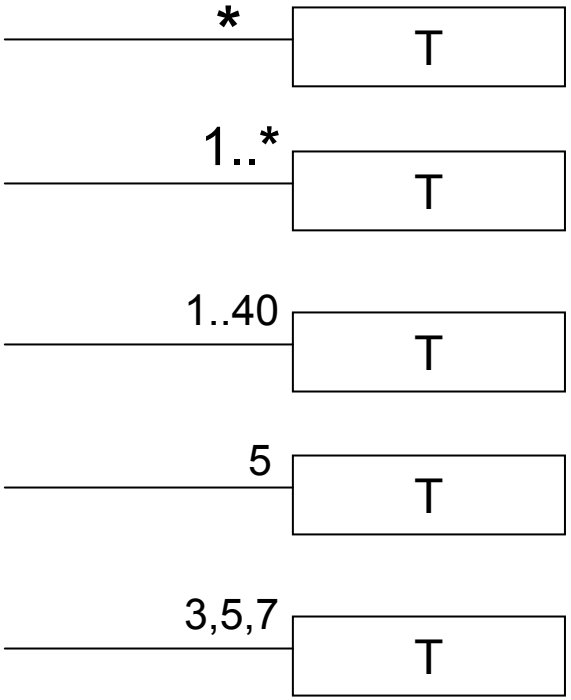
- Всеки край на асоциацията е *роля*.
- *Ролите* имат:
  - Име
  - Изрази, указващи множественост
  - Направление

Множественост

- *Множествените стойности* показват колко представителя на класа могат да се асоциират с други в един ***конкретен момент***



Множественост



→ **0 или повече**

→ **Един или повече**

→ **От 1 до 40**

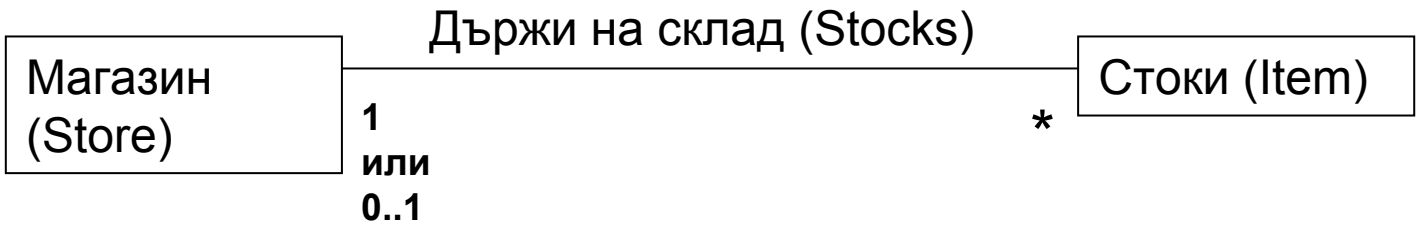
→ **Точно 5**

→ **Точно 3, 5 или 7**

Множественост - пример

- Коя множествена стойност да изберем

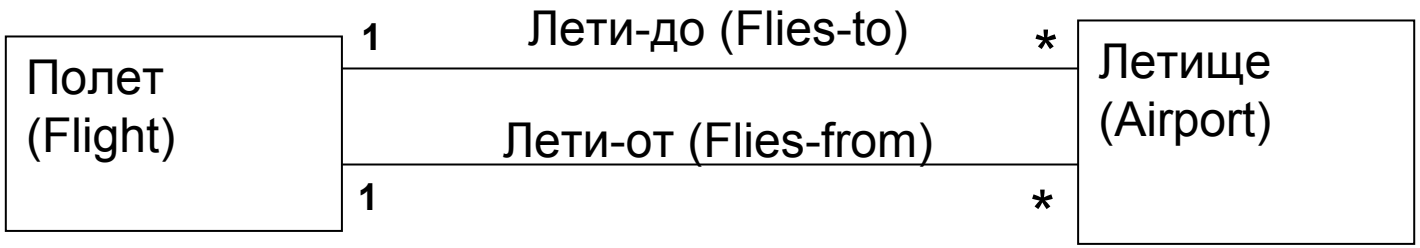
  1 или 0..1?



Множество асоциации между два типа

- Два класа могат да имат множество *асоциации* помежду си в UML диаграмата.

Пример:



Асоциации и имплементация

- Когато създаваме *домейн модела*, може да дефинираме *асоциации*, които не са необходими при *имплементацията*.

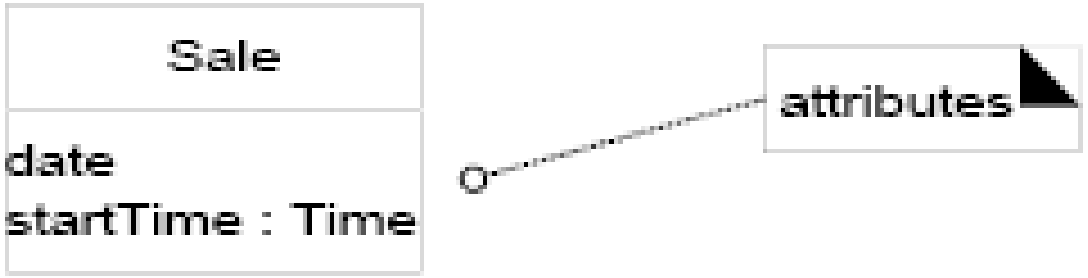
- Възможно е да открием *асоциации*, за които е необходима *имплементация*, но сме ги пропуснали в *домейн модела*.

Добавяне на атрибути

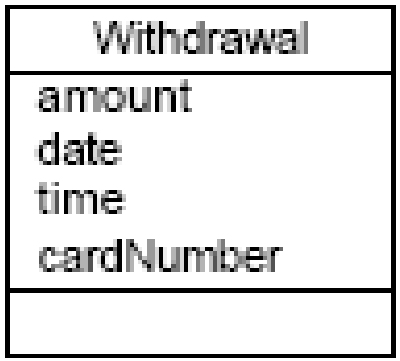
- **Атрибути** – логически данни за обекта.
- Атрибутите описват характеристиките на обектите.
- Атрибутите са нужни, за да се съхрани необходимата информация за обектите - онази информация, която трябва да бъде запомнена.

Атрибути - UML нотация

Продажба



Теглене на пари
от бакномат

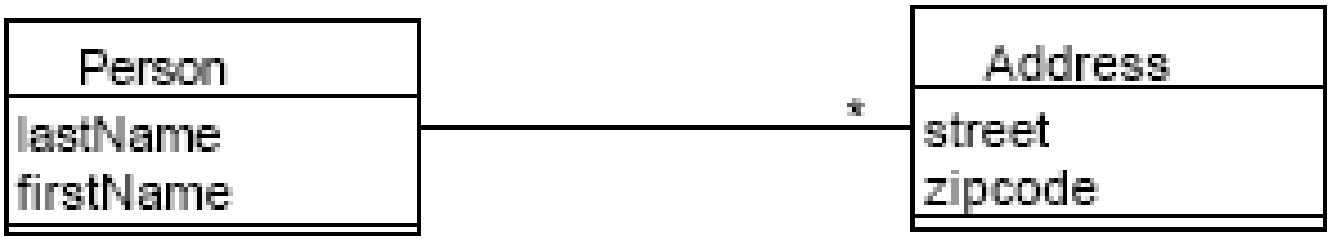


Валидни типове на атрибути

- Атрибутите в домейн модела трябва да са **примитивни** (прости) типове.
- Много атрибути включват типовете: Boolean, Date, Number, String, Time.
- Други типове: Address, Color, Geometries (Point, Rectangle), Phone Number, Postal Code…

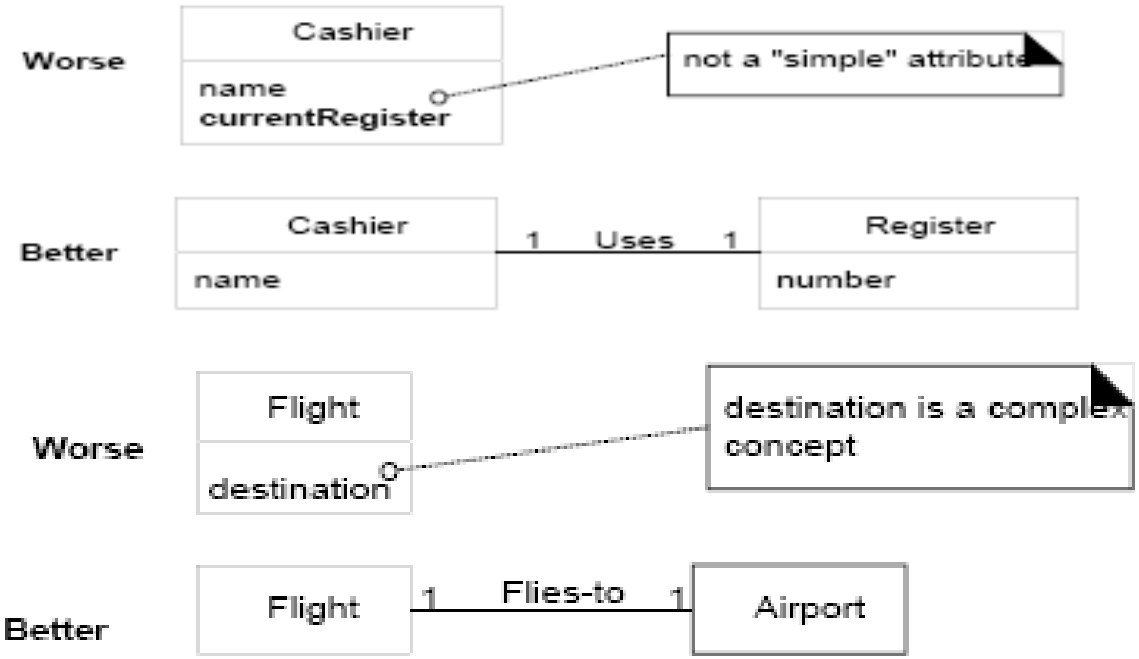
Пример за прости атрибути (1)

- Име и фамилия - прости типове (String).
- Адрес – не е примитивен тип, той също се състои от свои отделни части.



Пример за прости атрибути (2)

- Честа грешка е вместо асоциация да се прави атрибут.



Атрибути и код

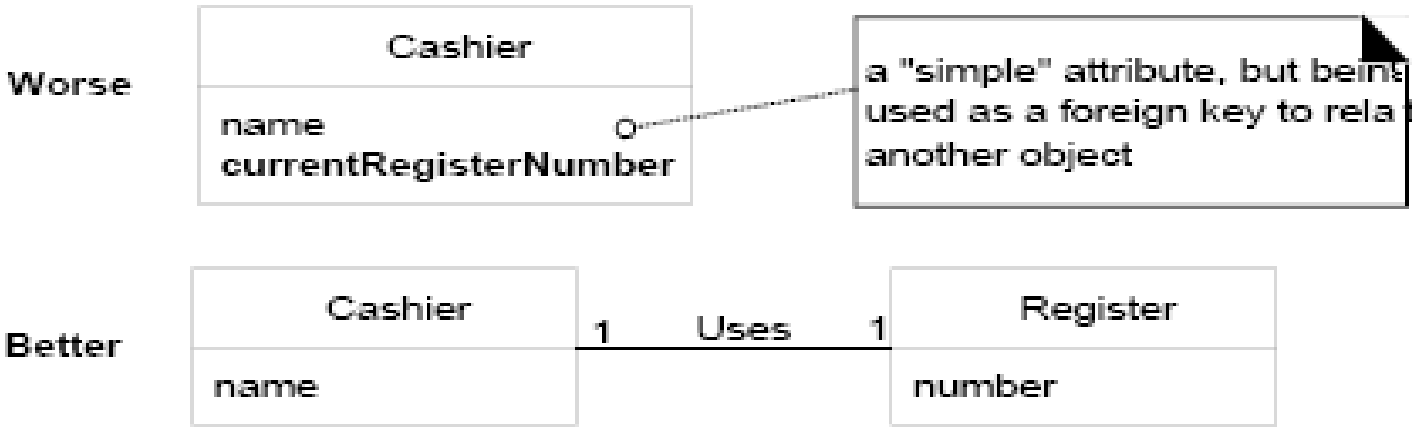
- Ограничението атрибутите в домейн модела да са прости типове НЕ означава, че типовете данни в класове в кода трябва да са прости.
- Домейн модела се фокусира върху концепциите на проблемната област. Не се вземат под внимание софтуерните компоненти.

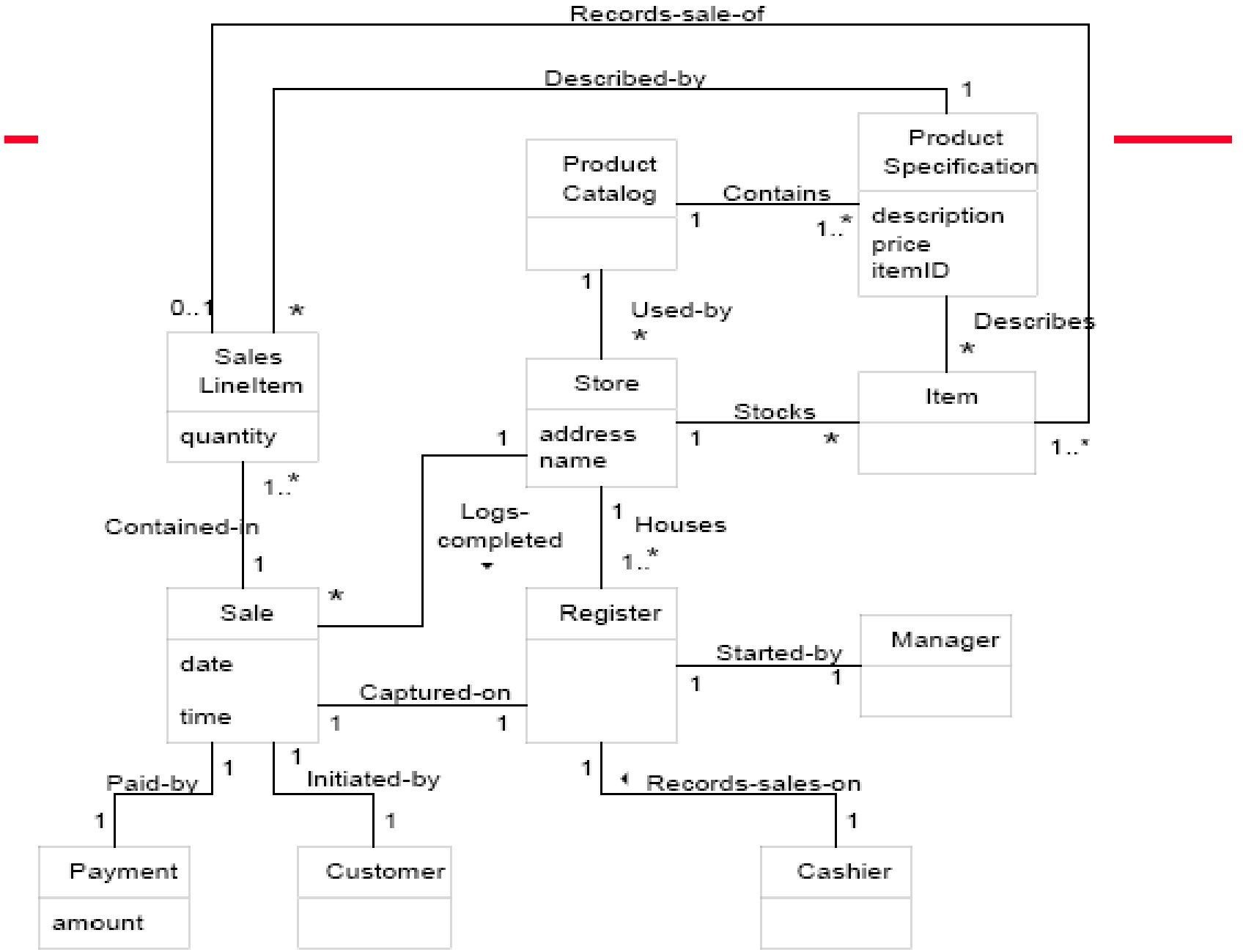
Типове данни

- Примитивни типове
  - number, string (UML **data types)**
- Непримитивни типове (или т.нар. **value objects)**
  - Phone Number
- По-добре е да се дефинира нов концептуален клас в домейн модела, отколкото атрибут

Domain and Data model

- В домейн модела използваме асоциации и атрибути.
- Не трябва да си мислим за външни ключове (foreign keys).




Records-sale-of
Described-by
1
Product Specification
description
price
itemID
Product Catalog
Contains
1
1..*
1
Used-by
*
Describes
*
0..1
*
Sales LineItem
quantity
Store
address
name
Item
Stocks
1
*
1
1..*
1..*
Contained-in
1
Logs-completed
1
Houses
1..*
Sale
*
date
time
Register
Manager
Started-by
1
1
Captured-on
1
1
1
Paid-by
1
Initiated-by
1
1
Records-sales-on
1
1
Payment
amount
Customer
Cashier

Домейн модел & UP

- Начална фаза (Inception) – не се обхваща изцяло.
- Фаза Детаилизация (Elaboration) – основно изграждане на Домейн модела
- Изискванията и обектно-ориентирания анализ са фокусирани върху изучаването на *Цели на системата, правила, ограничения*
- Дизайнът набляга на намиране на решения за удовлетворяване на поставените изисквания.