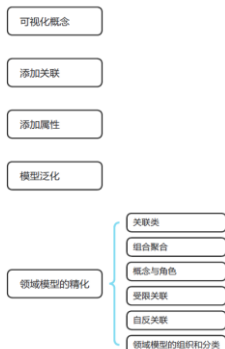
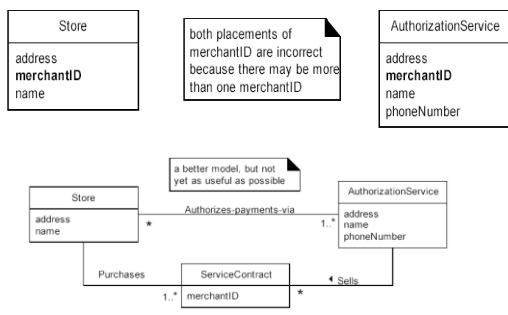


# OOA

## 标识实体



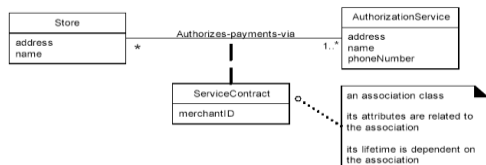
## 关联类(1/2)



## 关联类 (2/2)

需要关联类的场合:

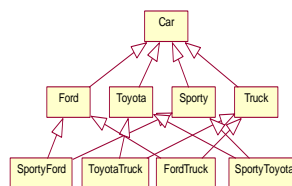
- ❖ 关联具有属性.
- ❖ 关联的实例具有持久性的生命周期.
- ❖ 概念之间具有多对多的关系, 通过关联类可以简化概念之间关系的建模.



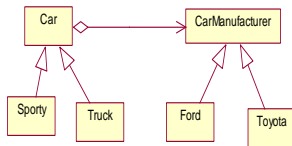
## 组合和聚合(1/3)

- 利用组合和聚合改善模型结构
- 利用组合和聚合改善概念的层次结构
- 利用组合和聚合改变不恰当的泛化
- 实例: 栈和线性表、命题逻辑公式建模

## 组合和聚合(2/3)



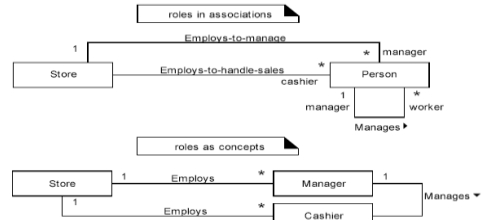
## 组合和聚合(3/3)



© 苏州大学计算机科学与技术学院

## Roles as Concepts vs. Roles in Associations

- 在领域模型里，现实世界的角色，尤其是人，既可建模为关联的角色，也可建模为离散的概念。
- 讨论



8

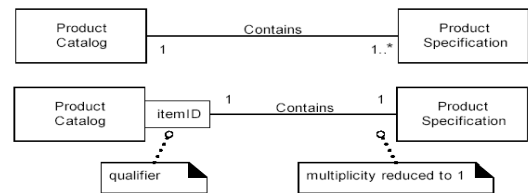
© 苏州大学计算机科学与技术学院

- 在一个班级中，一个班长可以管理班级的同学，但是班长和同学都是学生。

9

© 苏州大学计算机科学与技术学院

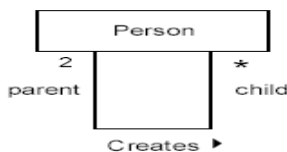
## 受限关联



10

© 苏州大学计算机科学与技术学院

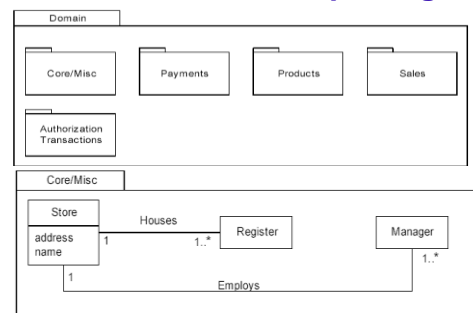
## 自反关联



11

© 苏州大学计算机科学与技术学院

## 领域模型的组织和分类--package



12

© 苏州大学计算机科学与技术学院

## UML中的组织事务

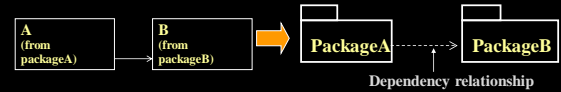
- 包是UML中用于把元素组织成组的常用的一种方式;
- 包是UML中基本的组织事务, 但也有些别的方式, 例如框架, 模型和子系统等等
  - ✓ 1) 对语义上相关的元素进行分组;
  - ✓ 2) 定义模型中的“语义边界”;
  - ✓ 3) 提供配置管理单元;
  - ✓ 4) 在设计时, 提供并行工作的单元;
  - ✓ 5) 提供封装的命名空间, 其中所有名称必须惟一
- 包的表示

13

© 苏州大学计算机科学与技术学院

## 组织事务：包

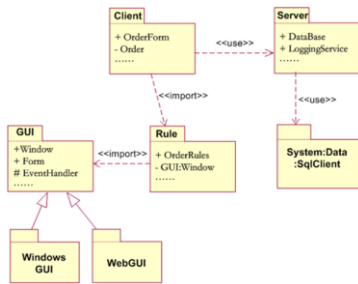
- 组织事务是UML的组织部分
- 结构事务, 行为事务和别的组织事务都可以作为包中的元素
- 不像构件, 包是概念上的, 仅存在在开发时间
- 包的表示, 可以仅包含名字, 也可含有详细的内容



Object Oriented Analysis and Design

14

## 包与包之间的关系



15

© 苏州大学计算机科学与技术学院

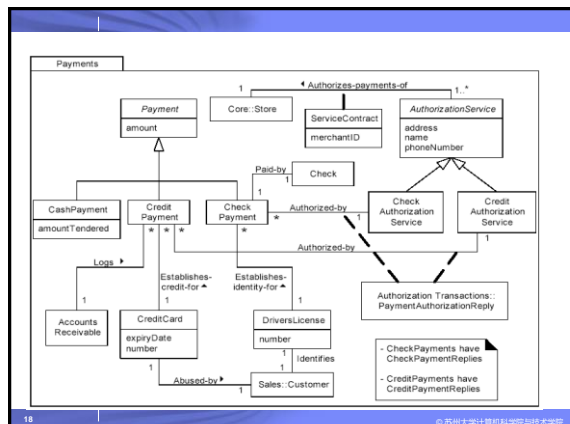
## 包与包之间的关系(续)

- <<use>>: 一种默认的依赖关系
- <<import>>: 最普遍的包依赖类型, 说明提供者包(箭头指向的包)的命名空间(包本身代表命名空间)将被添加到客户包(发出者)的命名空间中, 客户包中的元素也能够访问提供者包的所有公共元素
- <<access>>: 访问关系, 只使用提供者包中的元素, 而不想将其命名空间合并则应使用该关系
- <<trace>>: 追溯关系, 表示一个包到另一个包的历史发展, 则需要使用《trace》关系来表示
- 泛化

16

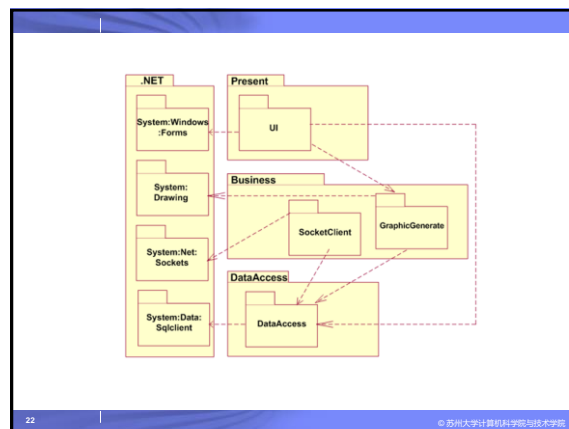
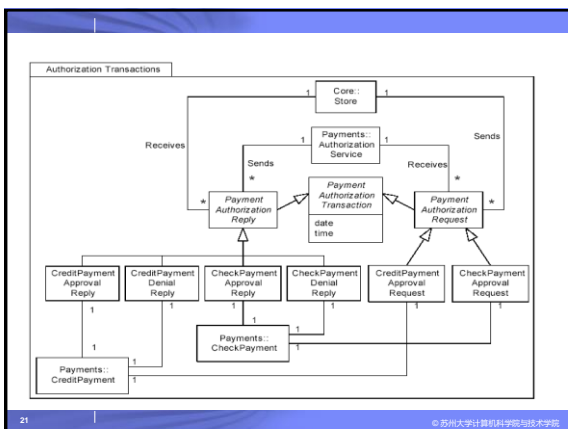
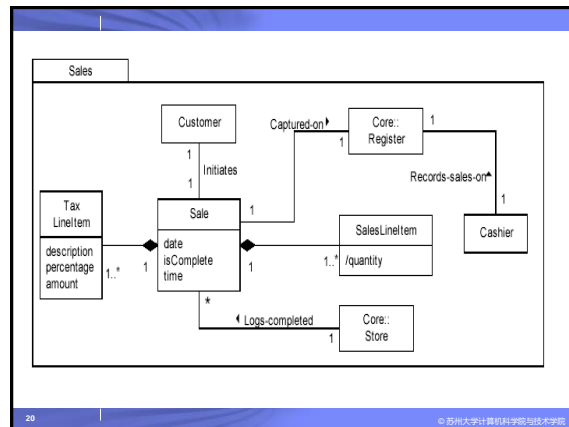
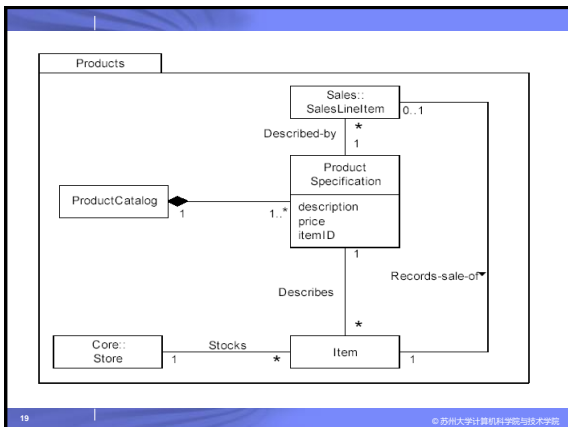
© 苏州大学计算机科学与技术学院

## 包的可见性



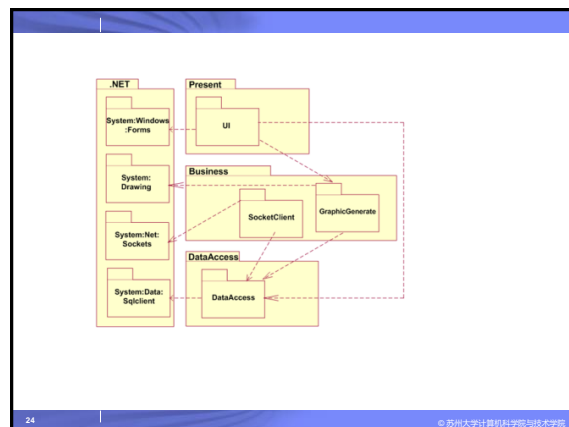
18

© 苏州大学计算机科学与技术学院



## 包图使用说明

- 每个包都应该是在概念、语义上相互接近的元素组成；
- 对每个包找出应标记为公共的元素，但应尽可能地少；
- 一般使用默认《use》构造型，在映射到编程时考虑明确《import》构造型；
- 采用泛化来对特殊包进行建模。
- 在表示这种模型时，注意只标明对每个包起核心作用的元素；另外也可以标识每个包的文档标记值，以使其更加清晰。
- 对体系结构建模
  - 对体系结构进行建模（程序分层），是包图更有意义的一个用途。体系结构是一个软件系统的核心逻辑结构
  - 常用的体系结构模式包括分层、MVC、管道、黑板、微内核等，而在应用软件中，分层和MVC



## 注意

- 包之间不要出现循环依赖的情况!
- 解决方法: 合并和分离