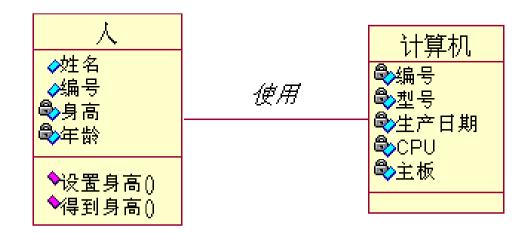
类图和对象图

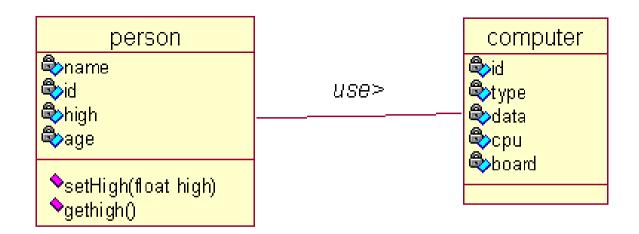
PEREC

- + 类图概念与意义
- ◆ UML中类图的标记符
- 中 建模类图的方法和步骤

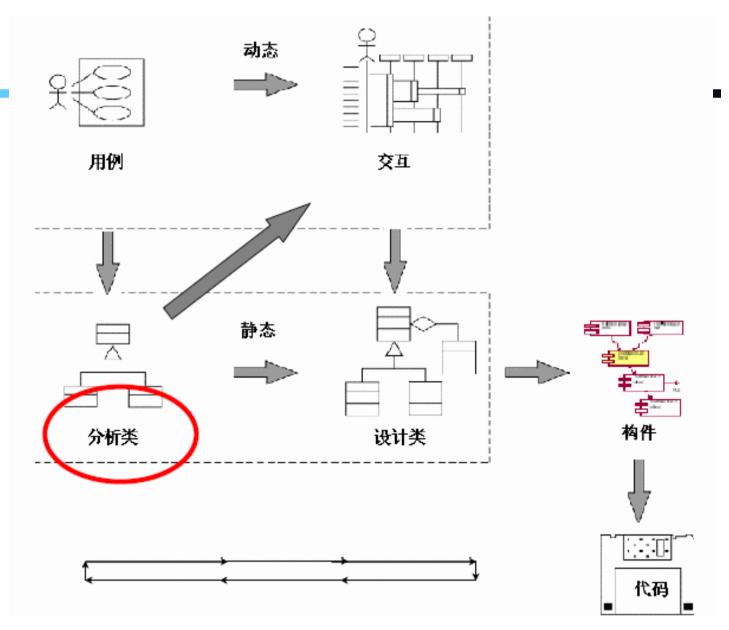
一、类图概念和意义

- ◆ 类图是由类和它们之间的关系组成的一种图示, 用来描述系统的静态模型。
- ◆ 类图是面向对象软件工程的核心





■ Nº 5



需求是分析的源头



有好的材料 才能炒出好菜

面向对象的基本原则

- ◆类(抽象和封装)
- 中 继承性
- + 多态性

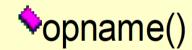
二、类图的标记符

- ⊕类:属性,方法
- → 关系:继承、聚集
- ◆ 多重性
- 中 角色

2.1 类标记







类的标记符由三部分组成: 类名、类的属性和 类的操作

类属性具有可视的 (public)、不可视的 (private)和受保护的 (protect)

发货单

- ♦总计
- ♦日期
- ◇客户

- **◇**规格说明 **◇**管理员 **◇**货单个数

发货单

发货单

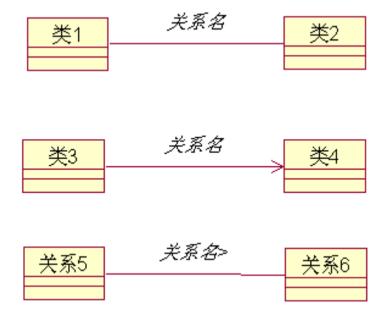
发货单

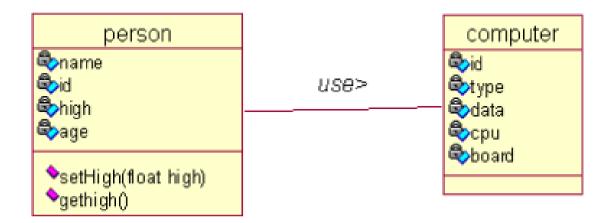
- ♦总计
- ◇日期
 ◇客户
- **◇**规格说明
- ◆管理员 ◆货单个数

```
Public class Invoice
      public double amout;
      public String customer;
     public String specification;
     public String administrator="unispecified";
     static private int number_of_invoices=0;
public Invoice() //构造函数
{number_of_invoices++;
        //类的其他操作方法
```

2.2 类之间关系的标记

类之间通过一条线来标记它们的关系





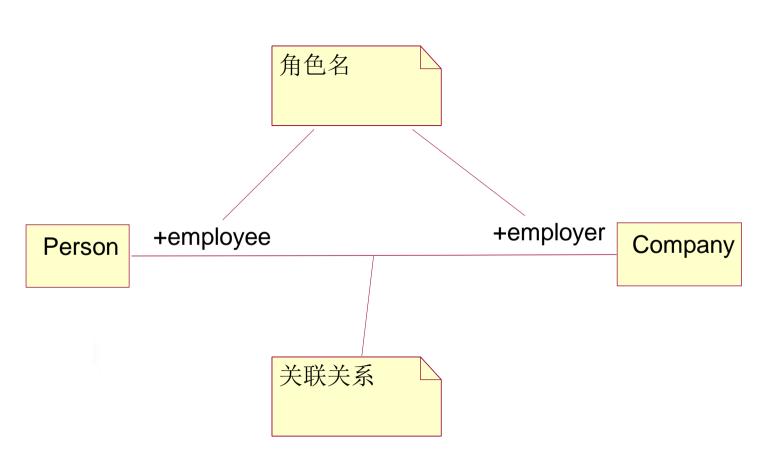
人 (person)使用计算机(computer)

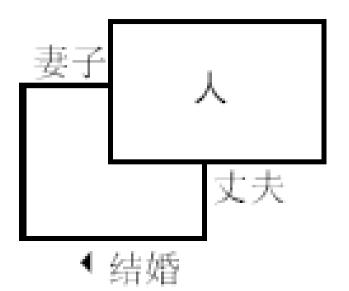
2.3 角色关联关系: 角色

- ◆ 角色的定义: 在UML里, 关联关系两端的类的 对象在对方的类里的标识, 称为角色 (role)。
- ◆ 角色在关联关系的图形化表示里由角色名 (role name)表示。
- + 当关联关系里的类被映射到程序设计语言时
 - 角色名字就成为类的一个成员变量的名字,此成员变量的类型将是另一个类的对象,或指向另一个类的指针



■ Nº 15





2.4 多重性

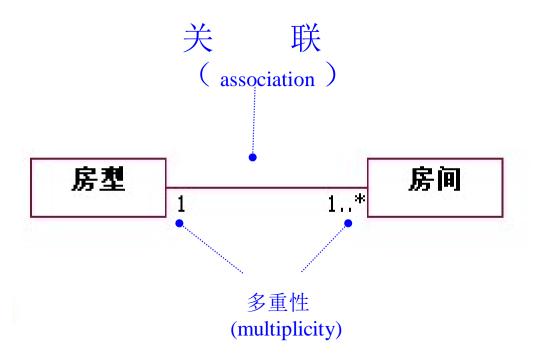
多重性用来指示一个类的多少对象与另一个类的一个对象相关。



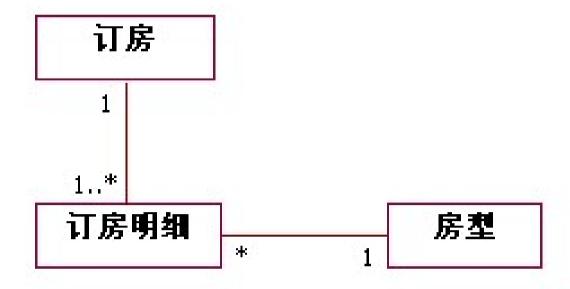








订房 * 1..* 房型



关联关系:角色

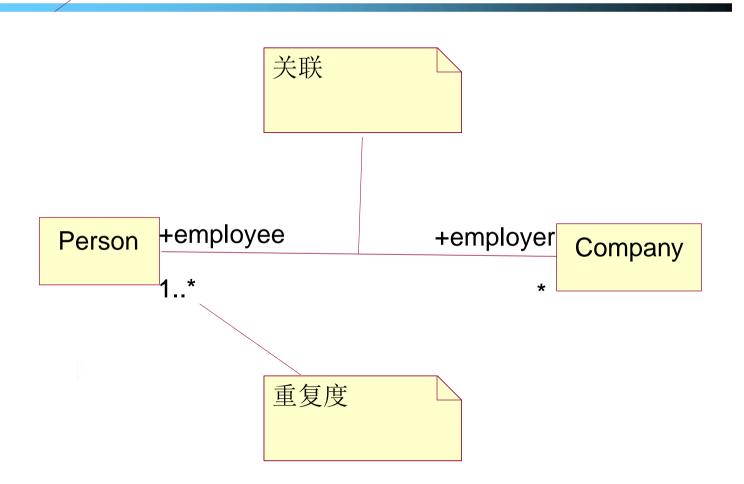
→ 角色的重复度

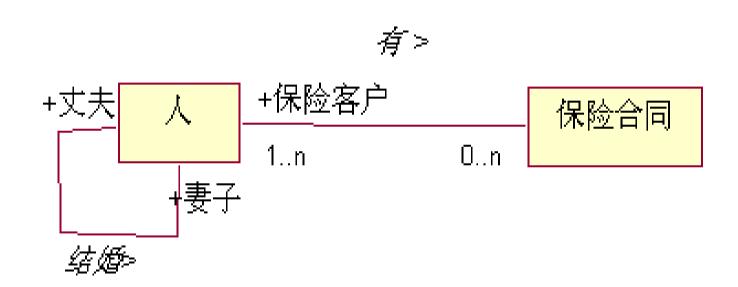
- 当一个类的对象以关联关系的角色的形式出现在另一个类里时,此角色可以在此类中出现一次,也可以出现多次。角色的重复度指的就是角色的这种出现次数。
- ◆ 角色重复度的定义: 在UML里, 角色重复度被定义为关联关系的实例的两端所连接的对象的数目。

关联关系:角色

- 母 角色重复度的图形化表示:
- ◆ 角色重复度被图示为一个表达式,它被放置在 关联关系中靠近相应角色的一侧,此表达式确 定了一个常数值或值的范围,例如:
 - 重复度1: 代表对象作为角色必须出现且只出现1次。
 - 重复度0..1:表示对象作为角色可以出现0次或1次。
 - 星号(*):代表任意多次,例如:
 - ◆重复度0..*表示对象作为角色可以出现0次任意多次;
 - ◆ 重复度1..*表示对象作为角色必须至少出现1次且可以出现 多至任意多次。

关联关系: 角色





三 对象:类的实例

- 中 对象是类的实例,类是相似对象的抽象
- → 对象的标记符与类相似,是一个带有文本"对象名:类名"的方框

对象名:类名





确定按钮:按钮

◆名称 = 确定
◆颜色 = 红

作家

姓名: string

年龄: integer

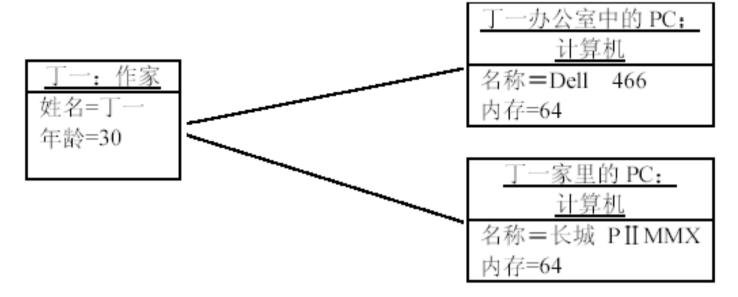
0..* 使用▶ 1..*

计算机

名称: string

内存: integer

(a) 类图

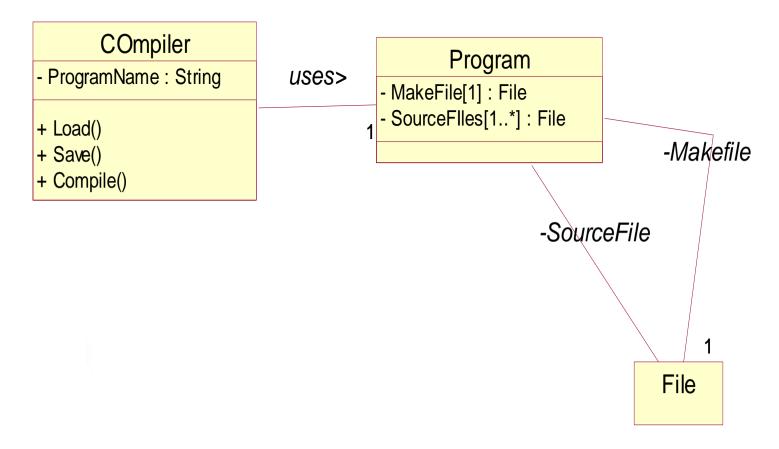


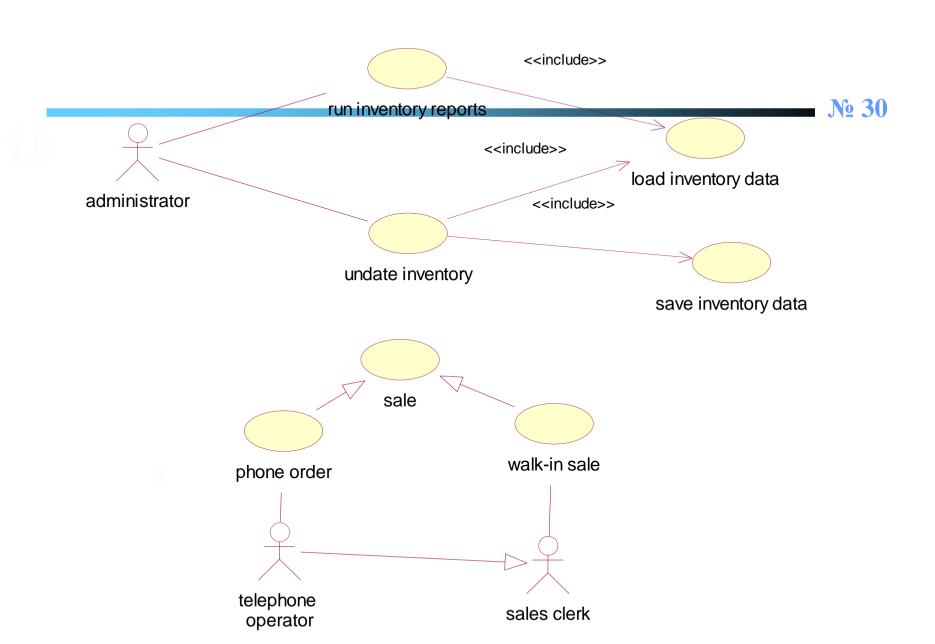
(b) 对象图

四.包

◆ 包是组织类的一种方式,或者叫做类的容器

练习:图6.26





五.建模类图

- + 确定类及其关联
- + 确定类的特性和操作
- 中 确定关联的多重性和角色

分析类及其识别

分析类

• 分析类的概念

- 在分析模型中,分析类是概念层次上的内容,用于描述 系统中较高层次的对象。
- 分析类直接与应用逻辑相关,而不关注于技术实现的问题。

• 分析类的类型

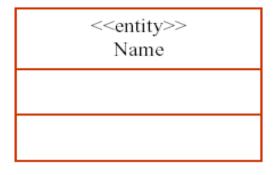
-实体类:表示系统存储和管理的永久信息

- 边界类:表示参与者与系统之间的交互

- 控制类:表示系统在运行过程中的业务控制逻辑

实体类

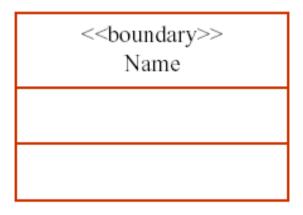
- 实体类
 - 描述必须存贮的信息及其相关行为
 - 通常对应现实世界中的"事物"
- 实体类的 UML 表示





边界类

- 边界类
 - 描述外部的参与者与系统之间的交互
 - 类型: 用户界面、系统接口、设备接口
- 边界类的 UML 表示

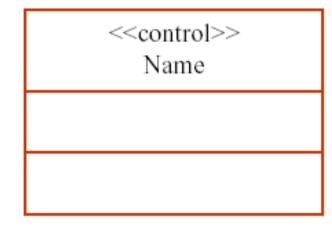




控制类

• 控制类

- 描述一个用例所具有的事件流控制行为
- 实现对用例行为的封装,将用例的执行逻辑与边界和 实体进行隔离
- 控制类的 UML 表示

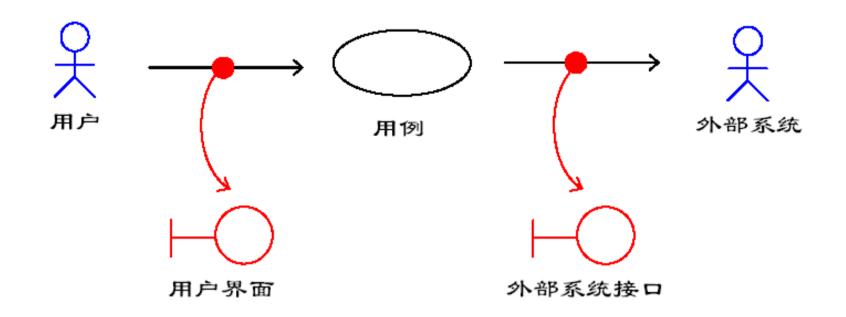




识别分析类

• 识别边界类

- 通常,一个参与者与一个用例之间的交 互或通信关联对应一个边界类。



识别分析类

- 识别边界类应当注意的问题
 - 边界类应关注于参与者与用例之间交互的<mark>信息或者响应的事件,不要描述窗口组件等界面的组成元素;</mark>

- 在分析阶段,力求使用用户的术语描述界面;

- 边界类实例的生命周期并不仅限于用例的事件流,如果两个用例同时与一个参与者交互,那么它们有可能会共用一个边界类,以便增加边界类的复用性。

• 思考: 如何识别 MiniLibrary 的边界类?

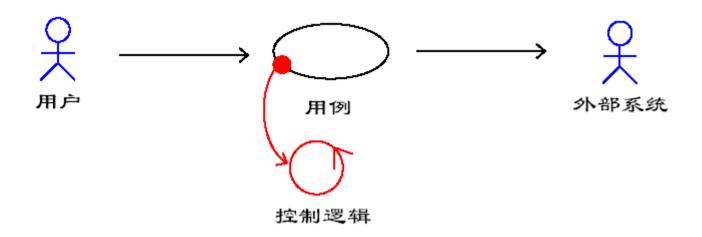
MiniLibrary: 识别边界类

边界类	说明
LoginForm	注册用户进行登录的操作界面
BrowseForm	注册用户进行查询浏览的操作界面
MakeReservationForm	普通读者预订图书的操作界面
Remove Reservation Form	普通读者取消预订的操作界面
ManageBorrowersForm	图书管理员管理读者的操作界面
ManageTitlesForm	图书管理员管理图书资料的操作界面
Manage Items Form	图书管理员管理书目的操作界面
LendItemForm	图书管理员登记借书的操作界面
ReturnItemForm	图书管理员登记还书的操作界面
MailSystem	与邮件系统的接口

识别分析类

• 识别控制类

- 控制类负责协调边界类和实体类,通常在现实世界中没有对应的事物。
- -一般来说,一个用例对应一个控制类。



• 识别控制类应当注意的问题

- 当用例比较复杂时,特别是产生分支事件流的情况下,一个用例可以有多个控制类。
- 在有些情况下,用例事件流的逻辑结构十分简单, 这时没有必要使用控制类,边界类可以实现用例的 行为。

- 如果不同用例包含的任务之间存在着比较密切的联系,则这些用例可以共用一个控制类, 其目的是复用相似部分以便降低复杂性。

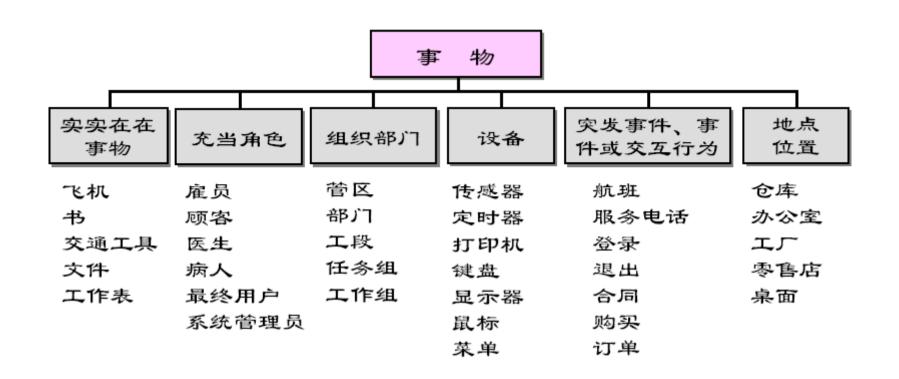
MiniLibrary: 识别控制类

控制类	说明
BrowseControl	负责执行注册用户的查询浏览
${\it Make Reservation Control}$	负责执行普通读者的预订图书
Remove Reservation Control	负责执行普通读者的取消预订
ManageBorrowers Control	负责执行图书管理员对读者的管理
Manage Titles Control	负责执行图书管理员对图书资料的管理
ManageItemsControl	负责执行图书管理员对书目的管理
LendItemControl	负责执行图书管理员登记借书
ReturnItemControl	负责执行图书管理员登记还书

识别分析类

• 识别实体类

-实体类通常是用例中的参与对象,对应着现实世界中的"事物"



• 识别实体类应当注意的问题

- 实体类的识别质量在很大程度上取决于分析人员书写 文档的风格和质量;
- 自然语言是不精确的,因此在分析自然语言描述时应 该规范化描述文档中的一些措辞,尽量弥补这种不足;
- 在自然语言描述中,<mark>名词</mark>可以对应类、属性或同义词等多种类型,开发人员需要花费大量的时间进行筛选。
- 思考: 如何识别 MiniLibrary 的实体类?

MiniLibrary: 识别实体类

MagazineItem

实体类	说明				
BorrowerInfo	普通读者的基本信息				
Loan	普通读者的借书记录				
Reservation	普通读者的预订信息				
Title	图书资料的基本信息				
Item	书目				
(由于图书资料中包括书籍和杂志等类型, 因此可以进一步划分子类)					
BookItem	书籍的基本信息				

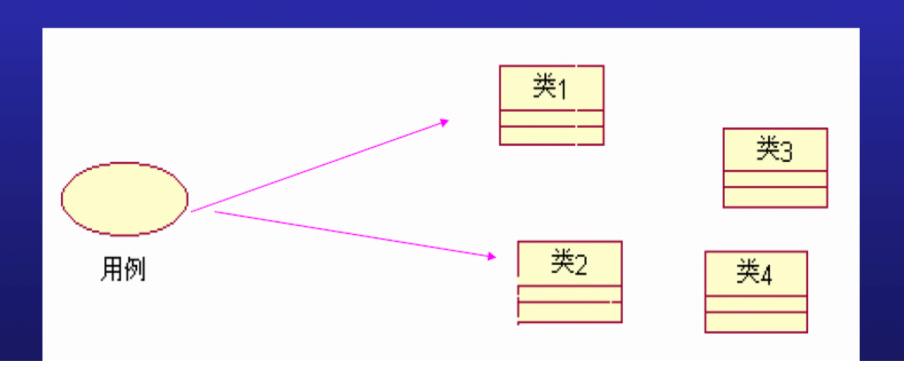
杂志的基本信息

□ 识别类及其属性

- 审查属性
- 识别类之间的泛化
- 审查泛化
- 识别类之间的聚合/组合
- 审查聚合/组合
- 识别类之间的连接

——识别类和属性

- 回顾需求文档,抽取对应于业务实体或事件的名词
- 出现遗漏时,返回需求文档进行修改
- 将名词进行分类、抽取出合适的类



——从用例文档抽取名词

零件	潜在会员	带格式的价格文件	
会员	查询条件	带格式的库存文件	
零件的形状	零件的大小	零件的几何特征	
零件的类别	零件的编号	货管员	
零件的详细信息	零件的价格	经理	
会员的用户名	零件的库存量	货管员的用户名	
会员的公司名	会员的密码	货管员的密码	
联系人	会员的联系信息	经理的用户名	
注册界面	会员信息	经理的密码	
供应商	订单		
订单的下单日期	订单项的购买数量		
订单项	账户		
订单的送货地址	库存		
账户的状态	订单的税金		
订单项的价钱	运费		
供应商	收费明细		
订单总价			

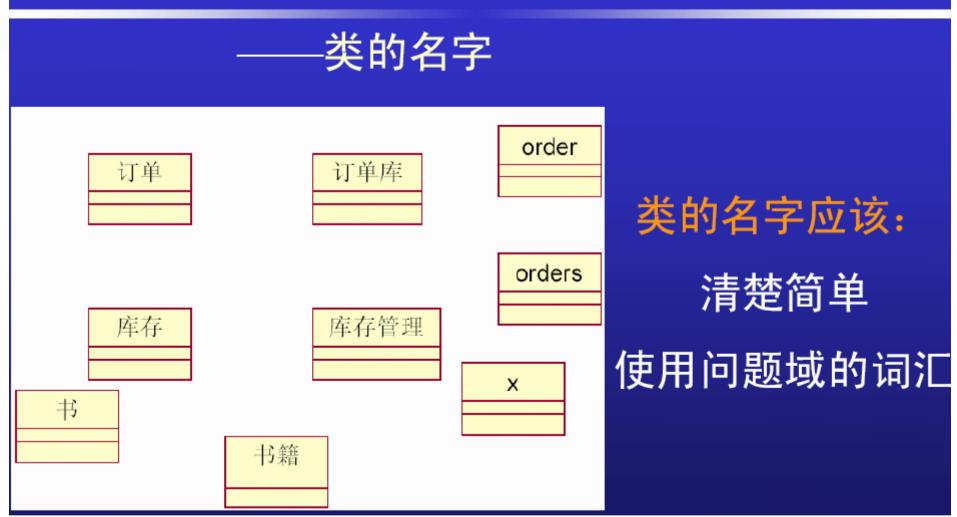
——分析

❖注意以下几点:

- ◆冗余的同义词(同一概念,不同表达)
- ◆在系统责任之外的名词
- ◆需要重新落实的模糊名词
- ◆实质上是操作或属性的名词(问一问: 有没有结构,有没有行为?)

——分析结果

零件	类	潜在会员	系统责任之外	带格式的价格文件	待定
会员	类	查询条件	会员的属性?	带格式的库存文件	待定
零件的形状	零件的属性	零件的大小	零件的属性	零件的几何特征	零件的属性
零件的类别	零件的属性	零件的编号	零件的属性	货管员	类
零件的详细信息	零件的属性	零件的价格	零件的属性	经理	类
会员的用户名	会员的属性	零件的库存量	模糊的名词	货管员的用户名	货管员的属性
会员的公司名	会员的属性	会员的密码	会员的属性	货管员的密码	货管员的属性
会员的联系人	会员的属性	会员的联系信息	会员的属性	经理的用户名	经理的属性
注册界面	界面类	会员信息	冗余	经理的密码	经理的属性
供应商	类	订单	类		
订单的下单日期	订单的属性	订单项的购买数量	订单项的属性		
订单项	类	账户	会员的属性		
订单的送货地址	订单的属性	库存	类		
账户的状态	会员的属性	订单的税金	订单的属性		
订单项的价钱	订单项的属性	运费	订单的属性		
供应商	类	收费明细	模糊的名词		
订单总价	订单的属性				



类图(初步)



- 识别类及其属性
- 审查属性



- 识别类之间的泛化
- 审查泛化
- 识别类之间的聚合/组合
- 审查聚合/组合
- 识别类之间的连接

◆ 是否在系统责任之内

◆是否描述类对象的特征

◆ 是否存在冗余

❖ 是否有复杂结构的属性

——属性是否在系统责任之内





——属性是否描述类对象的特征



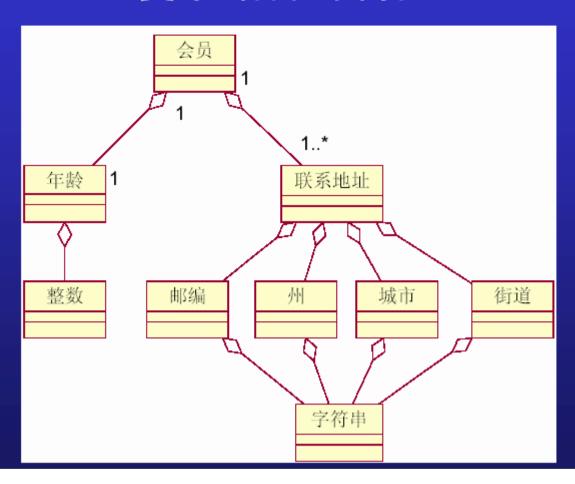
要能说得通: 什么的什么

——属性是否存在冗余



常见冗余如: 出生年月一一年龄

——复杂结构的属性



——复杂结构属性的处理

联系地址的内容包括: 小州、城市、街道、邮编



- ❖1:1—可以在原类内展开
- ◆1:N—独立出去形成关联

是否有复杂结构的属性(2)

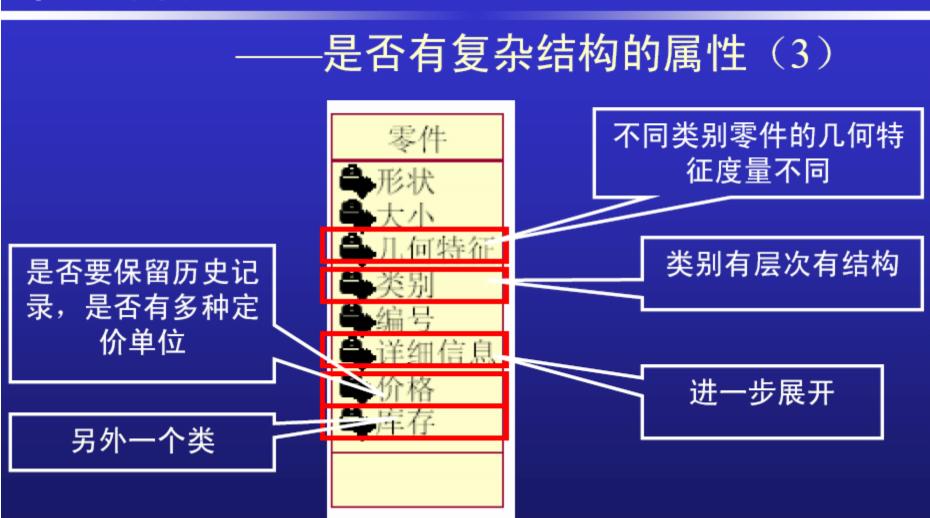
会员 会用密公联电传 Email Email

抽取出去,独立成类

₩户 账户状态 账户余额

可以抽取出去,独立成 类,也可以保留在会员类

中



根据对需求的理解进行细化

◆库存数量

