

6.1 对象和类的概念及其运用

概念：

对象（object）是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位，由一组属性和施加于这组属性的一组操作构成。

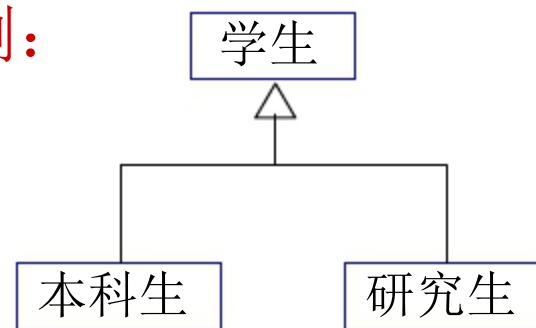
类（class）是具有相同属性和操作的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，它由一个类名、一组属性和一组操作构成。

类和对象的关系——集合与成员，对象是类的实例

在一般-特殊结构中，特殊类的对象实例在逻辑上也都是其一般类的对象实例。

不直接创建对象实例的类称为
抽象类（abstract class）

例：



主动对象（active object）——至少有一个操作不需要接收消息就能主动执行的对象
用于描述具有主动行为的事物

主动对象的类叫做**主动类（active class）**

被动对象（passive object）——每个操作都必须在消息的驱动下才能执行的对象

OO方法中的类在不同的语境下有两种不同的语义:

1. 一个类代表由它的全部对象实例所构成的群体

日常语言表达中的例子:

“公司里有管理人员、技术人员和市场人员”

“马路上汽车很多”

在**OO**模型中:

每个类都是由它的全部对象实例所构成的集合

类代表了它的全部对象实例。

2. 一个类代表属于该类的任意一个对象实例

从大量的个体中抽象出一个概念，再运用这个概念时就可以代表其中的任何一个个体，例如:

“学生有一个学号，属于一个班级，要上课”

在**OO**系统模型中定义了一个类，它就可以代表它的任何一个对象实例，例如:

汽车与发动机之间的聚合关系，表示任何一辆汽车都有一台发动机，任何一台发动机都可以装在**0—1**辆汽车上

在类的抽象层次建模

理由：

- (1) 充分性：模型中一个类描述了它的全部对象实例
- (2) 必要性：个别对象实例不能代表其他对象实例
- (3) 符合人类的思维方式：在概念层次上表达描述事物规律
- (4) 与**OOPL**保持良好的对应
- (5) 避免建模概念复杂化
- (6) 消除抽象层次的混乱

如何运用类和对象的概念

从对象出发认识问题域
将问题域中的事物抽象为对象；

将具有共同特征的对象抽象为类
用类以及它们之间的关系构成整个系统模型；

归纳

在模型中用类表示属于该类的任何对象
在类的规约中说明这个类将创建那些对象实例

在程序中用类定义它的全部对象
 编程时静态声明类的对象
 运行时动态创建类的对象

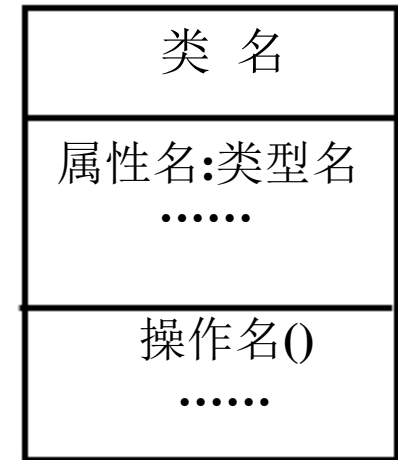
演绎

6.2 表示法

在模型中用类符号来表示一个类
它代表了属于该类的全部对象实例



压缩方式

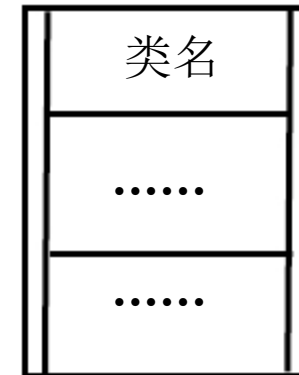
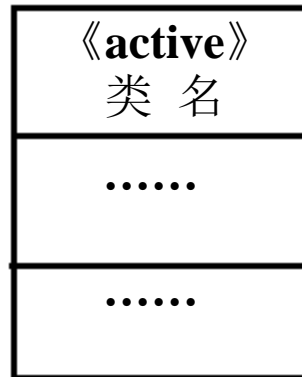


展开方式

衍型 (stereotype)
用关键字或者
用图标表示



主动类



UML2 主动类表示法

6.3 发现对象

研究问题域

亲临现场深入调查研究

直接观察并向用户及相关的业务人员进行调查和交流，考察问题域中各种各样的事物、它们的特征及相互关系

听取问题域专家的见解

领域专家——包括技术人员、管理者、老职员和富有经验的工人等

阅读相关材料

阅读各种与问题域有关的材料，学习相关行业和领域的基本知识

借鉴以往的系统

查阅以往在该问题域中开发过的同类系统的分析文档，吸取经验，发现可以复用的类

正确地运用抽象原则

对什么进行抽象——问题域
当前目标——系统责任

忽略与系统责任无关的事物

只注意与之有关的事物，抽象为系统中的对象

例如：学校的教师、学生、教务员 和 警卫

忽略与系统责任无关的事物特征

只注意与之有关的特征，抽象为对象的属性或操作

例如：教师的专业、职称 和 身高、体重

正确地提炼对象

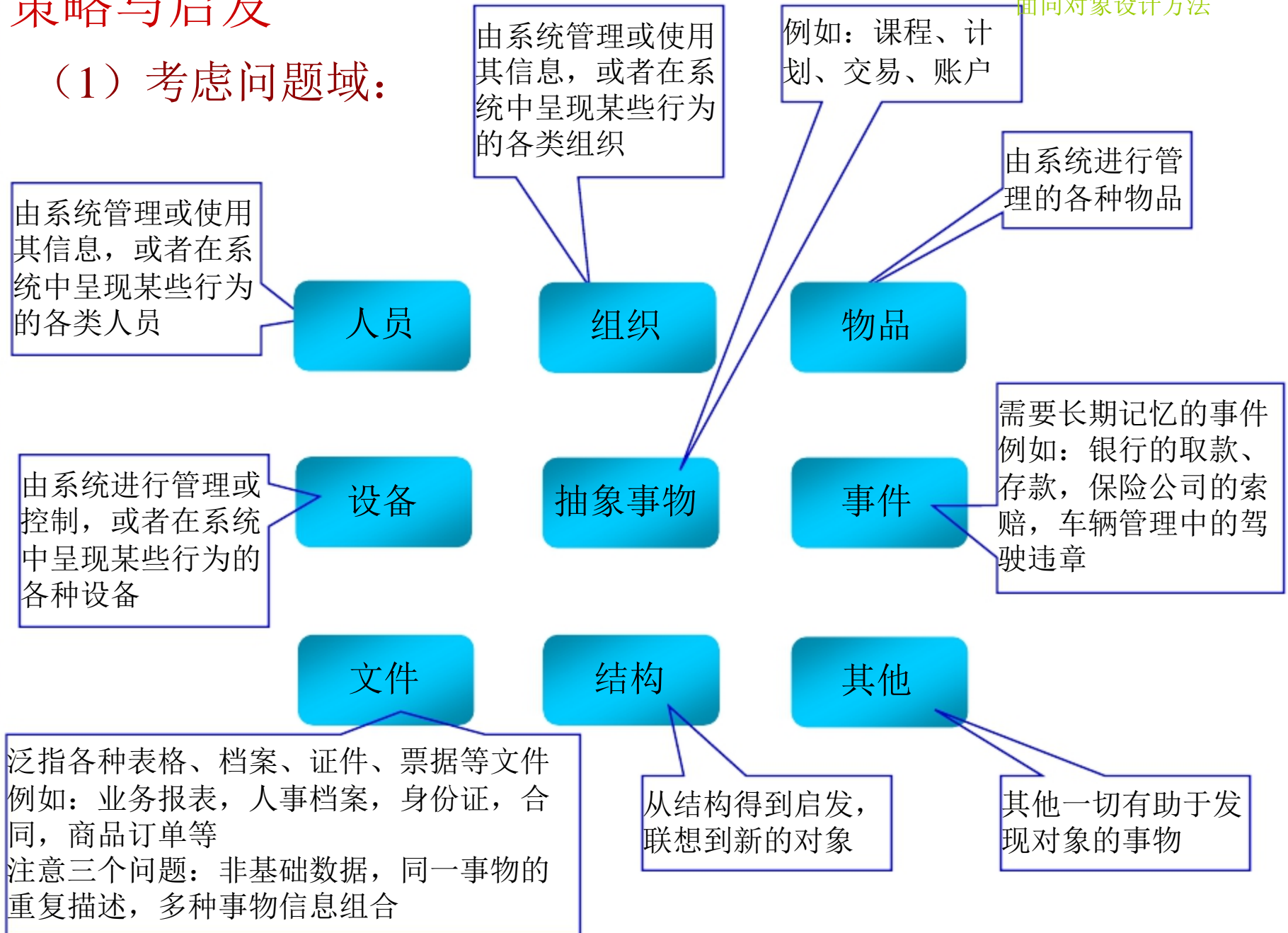
例如：对书的不同抽象

在图书馆管理系统中以一本书作为一个对象实例

在书店管理系统中以一种书作为一个对象实例

策略与启发

(1) 考虑问题域:



(2) 考虑系统边界:

考察在系统边界以外与系统交互的各类参与者
考虑通过那些对象处理这些参与者的交互

人员

设备

外系统

(3) 考虑系统责任:

检查每一项功能需求是否已有相应的对象提供,
发现遗漏的对象

审查与筛选

(1) 舍弃无用的对象

通过属性判断：

是否通过属性记录了某些有用的信息？

通过操作判断：

是否通过操作提供了某些有用的功能？

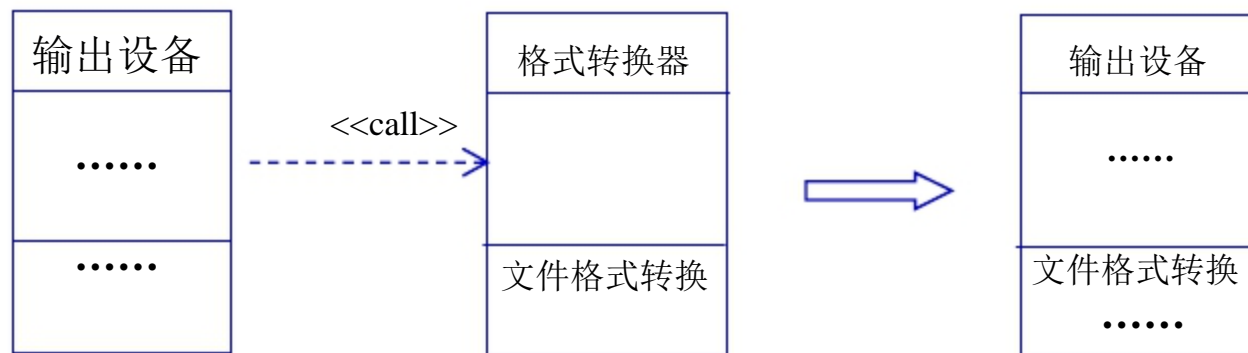
二者都不是——无用

(2) 对象的精简

只有一个属性的对象



只有一个操作的对象



(3) 与实现条件有关的对象

例如：与

图形用户界面 (**GUI**)

数据管理系统

硬件 及

操作系统 有关的对象

——推迟到**OOD**考虑

6.4 对象分类

(1) 将对象抽象为类，用类表示它的全部对象

(2) 审查和调整

类的属性或操作不适合该类的全部对象实例

例：“汽车”类的“乘客限量”属性
——进一步划分特殊类

属性及操作相同的类

经过抽象，差别很大的事物可能只保留相同的特征
——考虑能否合并为一个类

属性及操作相似的类

——考虑能否提升出一个一般类

同一事物的重复描述

例：“职员”和“工作证”
——取消其中一个

(3) 类的命名

类的名字应适合该类（及其特殊类）的全部对象实例

反映个体而不是群体

使用名词 或 带定语的名词

避免市井俚语和无意义的符号

使用问题域通用的词汇

使用便于交流的语言文字

可以用本地文字和英文双重命名