

面向对象分析与设计 Object-Oriented Analysis and Design

北京理工大学软件学院
马 锐
Email: mary@bit.edu.cn

第5章 静态图

5.1 概述

5.2 类图

5.3 包图

5.1 概述

➤ 描述系统的静态特性

- ❖ 类图
 - ❖ 描述系统中类的静态结构
- ❖ 对象图
 - ❖ 类图的实例
- ❖ 包图
 - ❖ 表示包与包、包与类之间的关系

5.2 类图

5.2.1 类图

5.2.2 对象图

5.2.3 类之间的关系

5.2.1 类图(1)

➤ 表示

- ✓ 由水平线划分的三个分栏矩形
- ✓ 最上面的那个名称栏包含类名；
- ✓ 中间的分栏包含属性列表；
- ✓ 最下面的分栏包含操作列表
- ✓ 每个属性和操作各占一行，

手机
-生产厂商
-型号
-颜色
+Telephone()
+SMS()
+Camera()
+Blog()

➤ 类名

- ❖ 包含该类所有对象
- ❖ 反映每个对象个体，而不是整个群体
- ❖ 采用名词，或带有定语的名词
- ❖ 适当的语言

5.2.1 类图(2)

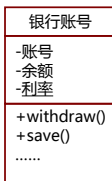
➤ 属性

- ❖ [可见性]属性名[:'类型']['='初始值]
- ❖ 可见性分为
 - ❖ + (公有的)
 - ❖ # (受保护的)
 - ❖ - (私有的)
 - ❖ ~ (包范围，只有在同一包中声明的类能够使用该属性)

5.2.1 类图(3)

- 实例属性与类属性
- 类属性是描述类的所有对象共同特征的一个数据项，对于任何对象实例，类属性值都相同

通过在类范围属性名和类型表达式画下划线的方式，表示类范围的属性，否则属性是实例范围的。



7

5.2.1 类图(4)

操作

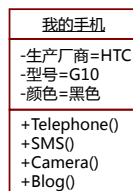
- 类对象被要求执行的服务
- [可见性]操作名[(参数列表)][:]返回类型
- 实例操作与类操作



8

5.2.2 对象图(1)

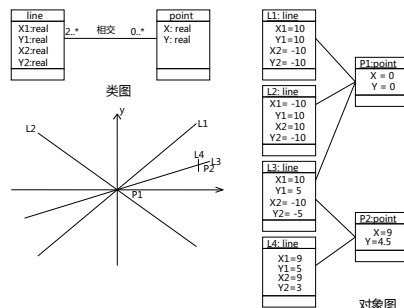
对象图



类图与对象图

9

5.2.2 对象图(2)



10

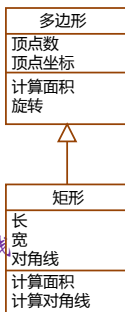
5.2.3 类之间的关系(1)

类之间的关系

- 泛化(继承)
- 聚合
- 关联
- 依赖

继承的表示

- 从特殊类到一般类的一条实线
- 在一般类一端有一空心三角



11

5.2.3 类之间的关系(2)

抽象类

- 没有实例的类，它把一些类组织起来，提供一些公共的行为，这些行为作用于其子类的实例

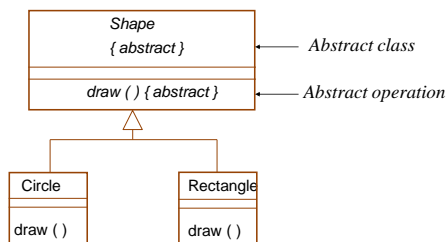
表示

- 通过把类名写成斜体的，或是在类名后标以{abstract}表示抽象类

- 抽象操作带有标记“{abstract}”，或把操作的特征标记写成斜体来表示它是抽象的

12

5.2.3 类之间的关系(3)



13

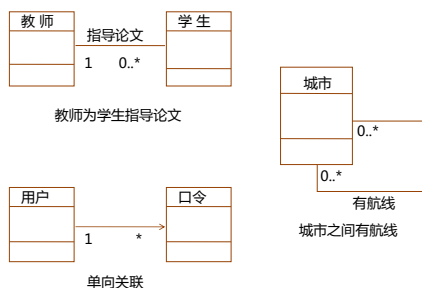
5.2.3 类之间的关系(4)

➤ 关联

- 如果类的对象之间通过属性有连接关系，那么这些类之间的语义关系就是关联
- 两个类之间可以有多个关联
- 二元关联表示成连接两个类符号的实线，可以在实线上标识出关联名
- 两个端点可以连接到相同的类，但是端点不同
- 双向关联与单向关联

14

5.2.3 类之间的关系(5)



15

5.2.3 类之间的关系(6)

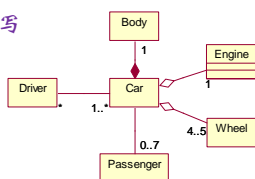
✦ 关联的多重性

- 在关联的两端写有表示数量约束的数字或符号，称为关联的多重性
- 含义：本端的一个对象可能需要的另一端对象的个数
- 表示：由逗号分隔的一系列正整数区间
 - 下限..上限
 - 星号(*)用于上限，表示无限制

16

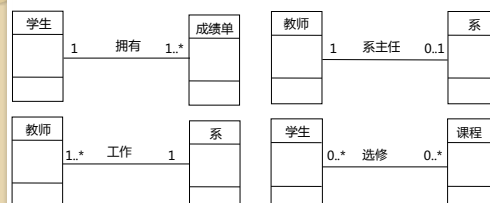
5.2.3 类之间的关系(7)

- n : 表示 n
- $m..n$: m 到 n 范围内的任意数值 (包含 m 和 n)
- $p..*$: 从 p 到无穷大的任意数
- $*$: $0..*$ 的缩写
- $0..1$: 可选



17

5.2.3 类之间的关系(8)



18

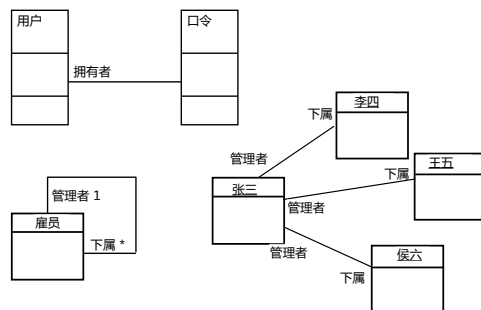
5.2.3 类之间的关系(9)

◆ 关联角色

- 在关联的每一个端点上有一个角色。每一个角色具有一个名字，用来描述其类被其他的类看作是什么，该名字称为**角色名**
- 当需要强调一个类在一个关联中的确切含义时，使用关联角色名
- 两个类之间有多个关联时，使用关联角色有助于理解关联
- 在关联中使用角色名，可以省略关联名

19

5.2.3 类之间的关系(10)



20

5.2.3 类之间的关系(11)

◆ 链

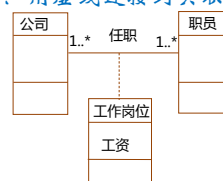
- 对象间的语义连接，是关联的一个实例
- 把二元链表示为两个实例之间的路径，可以把一个实例与它自身之间的链表示为一个具有单一实例的环
- 如果有关联名应在其下画线表示链名
- 在链的各端可以表示角色名
- 一旦在两个对象之间建立了链，每一个对象都可以请求另一个对象的操作

21

5.2.3 类之间的关系(12)

◆ 关联类

- 如果在具有关联关系的类中，存在着一个属性放在哪个类中都不合适的情况，就考虑使用关联类
- 表示：用虚线连接到关联路径的类符号



22

5.2.3 类之间的关系(13)

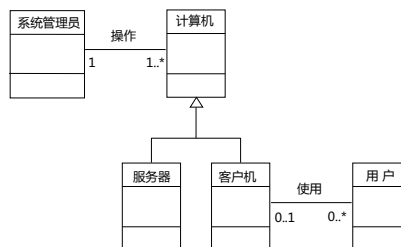
◆ 关联类与二元关联之间的转换



23

5.2.3 类之间的关系(14)

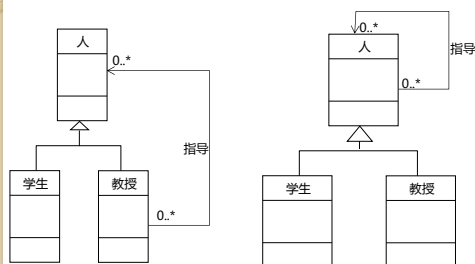
◆ 特殊类继承一般类的关联



24

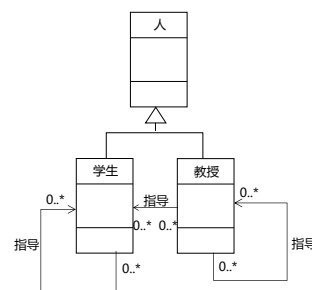
5.2.3 类之间的关系(15)

一般类与特殊类之间的关联



25

5.2.3 类之间的关系(16)

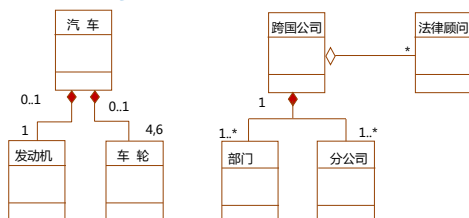


26

5.2.3 类之间的关系(17)

聚合的表示

- 带有空心菱形（组合则为实心）的线段，菱形指向聚集端



27

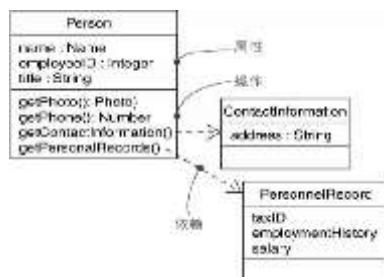
5.2.3 类之间的关系(18)

依赖的表示

- 一个依赖规约了两个或多个模型元素之间的一种语义关系，对目标元素的改变可能需要改变该依赖中的源元素
- 表示
 - 两个模型元素之间的虚线箭头
 - 在箭头尾部的模型元素（客户）依赖箭头头部的模型元素（提供者）
- 依赖多以方法参数的形式出现

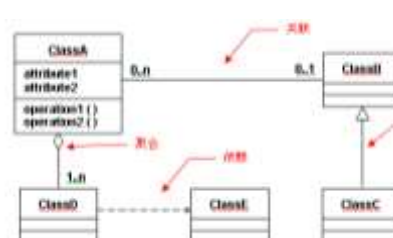
28

5.2.3 类之间的关系(19)



29

5.2.3 类之间的关系(20)



30

5.3 包图(1)

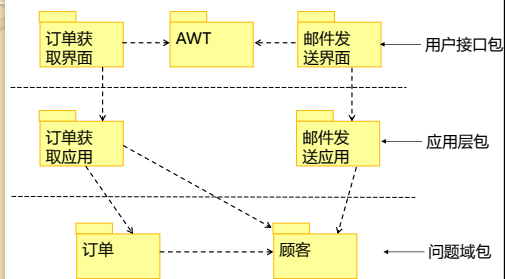
➤ 包图

- 包是对模型元素进行分组的机制
- 包图描述如何对模型元素分组以及分组之间的依赖关系
- 包是一种模型元素，可包含其他包，可用于各种不同的图
- 一个模型元素只能被一个包拥有
- 表示
 - 带有标签的文件夹
 - 包的名字



31

5.3 包图(2)



32

5.3 包图(3)

➤ 包的层次性

- 包可以嵌套

➤ 包中元素的命名

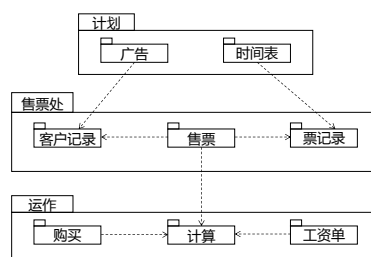
- 一个包形成了一个名字空间
- 包内名字必须唯一

➤ 包中元素的可见性

- + (公共的)、- (私有的)、# (受保护的)、~ (范围的)

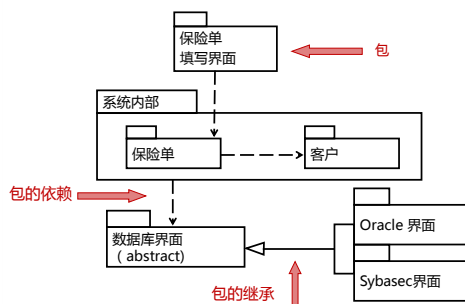
33

5.3 包图(4)



34

5.3 包图(5)



35

5.3 包图(6)

➤ 包间的关系

- 拥有关系 (包内有包)
- 泛化关系
- 引入依赖
 - 两个包之间的一种许可依赖关系
 - 提供者包的内容附加到客户包的公共命名空间
- 表示
 - 带有箭头的虚线，其上标有串 `<import>`，箭头指向提供者

36

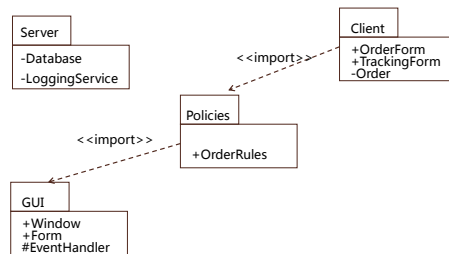
5.3 包图(7)

访问依赖

- 两个包之间的一种许可依赖关系
- 提供者包的内容附加到客户包的**私有命名空间**
- 表示
 - 带有箭头的虚线，其上标有串 `<<access>>`，箭头指向提供者

37

5.3 包图(8)



38