



深入学习类图

PEREC



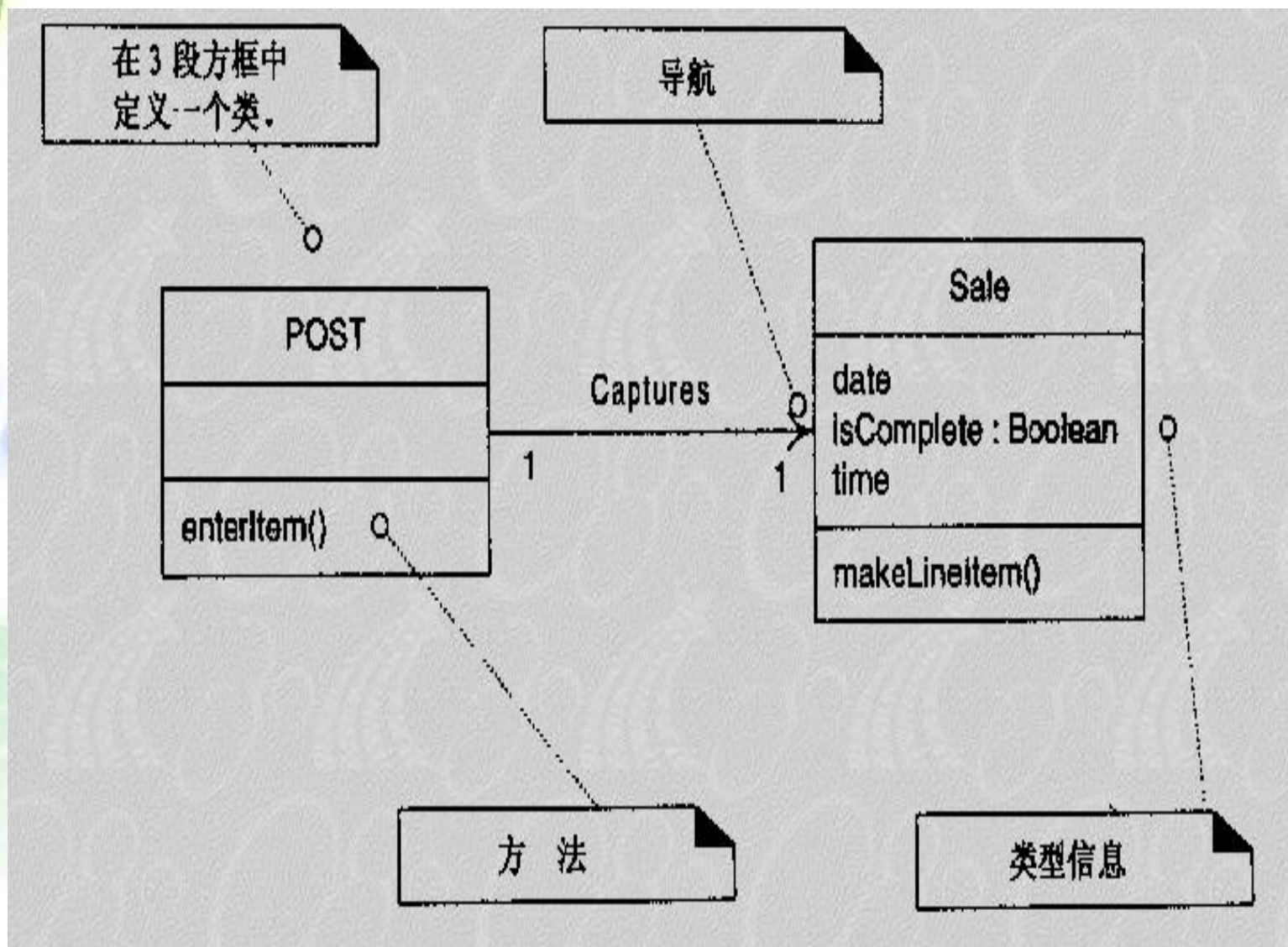
已学的类及类图知识

№ 2

⊕ 类的概念

- 类之间的关系—关联
- 多重性
- 角色
- 类的特性和操作
- 类特性的数据类型、初始值、多重性和派生的特性

PEREC





将要学习的类图知识

№ 4

- ⊕ 聚集
- ⊕ 组成
- ⊕ 约束
- ⊕ 鉴别器
- ⊕ 端点标记符

PEREC

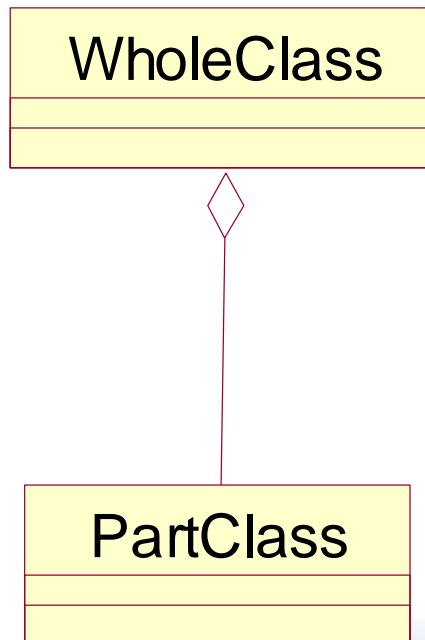


一、整体一部分关系

№ 5

1.1 聚集

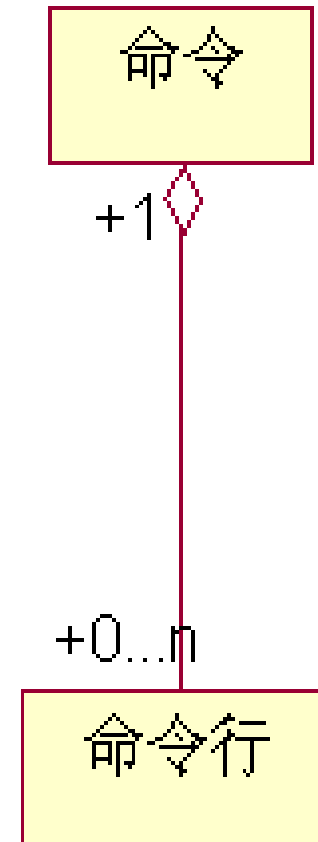
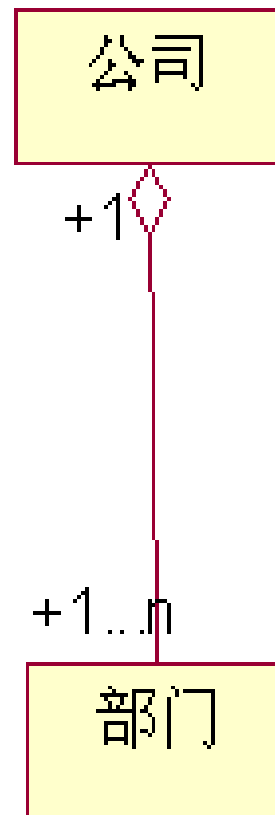
聚集描述类之间的整体一部分关系。

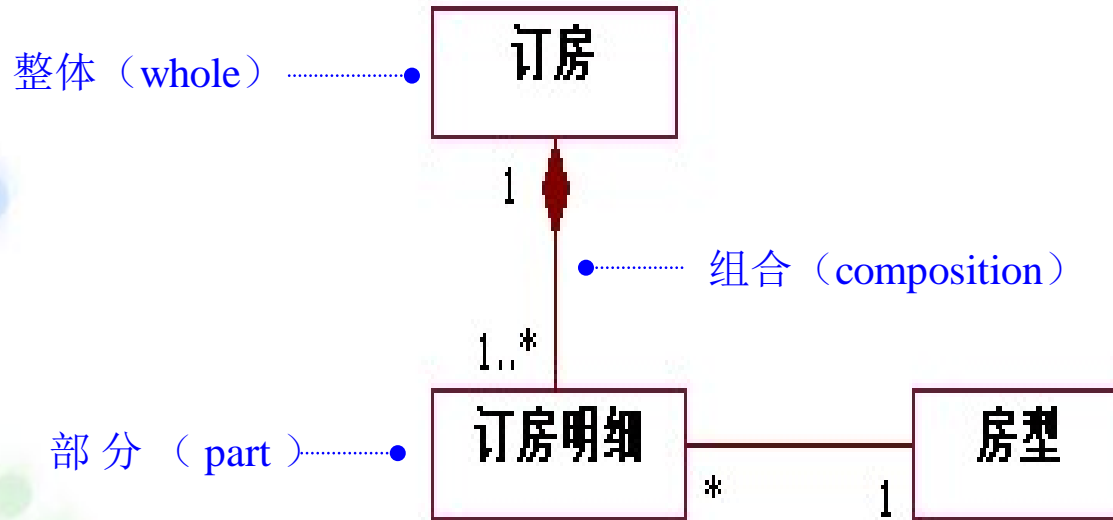




聚集关系的3个特性:

- ⊕ 构成对象不存在，聚集对象还存在。例如部门和员工
- ⊕ 每个构成对象都可以是多个聚集的构成。
人和俱乐部
- ⊕ 聚集往往是同构的，也就是说，构成对象往往是同一类的。例如森林和树木



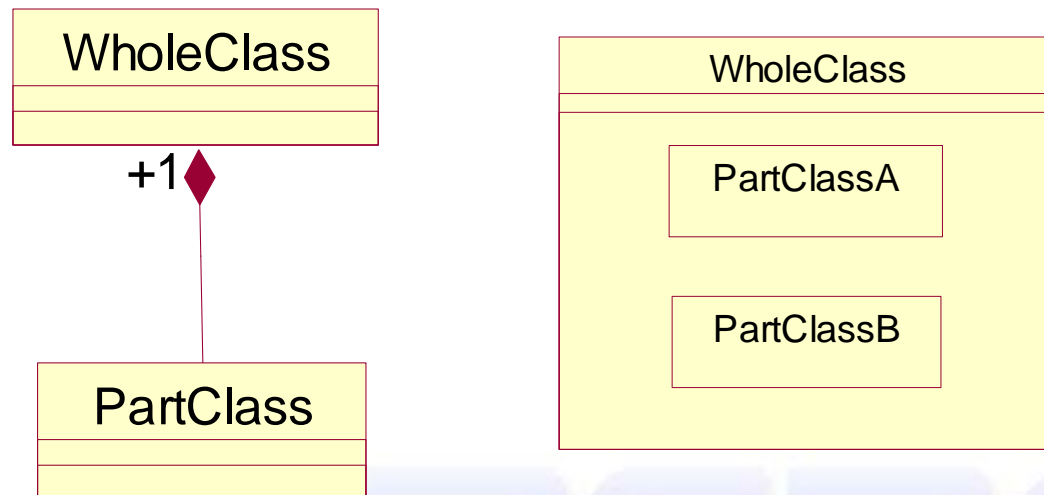


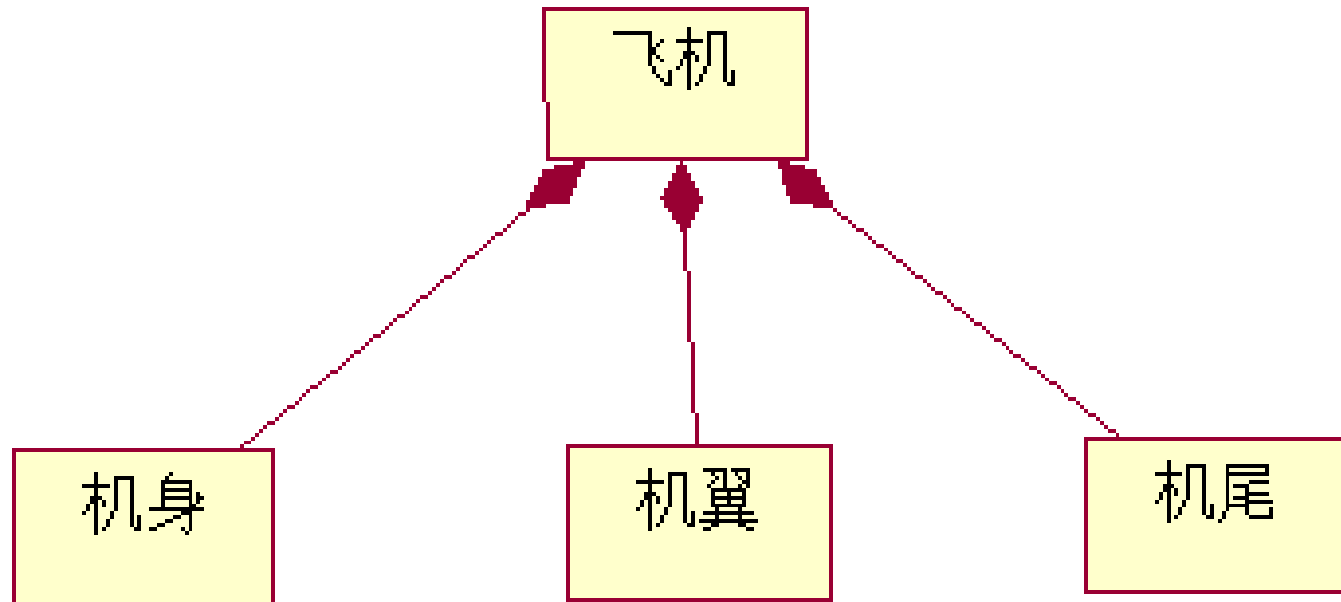


1.2 组成

组成也是一种整体—部分关系

不同于聚集关系，组成关系中的部分类不能脱离整体类而独立存在







组成关系的3个特性:

- ⊕ 如果部分不存在，整体也不存在
- ⊕ 部分类的对象是整体类对象的组成部分
- ⊕ 组成关系是典型的异构，即部分类是不同的类型



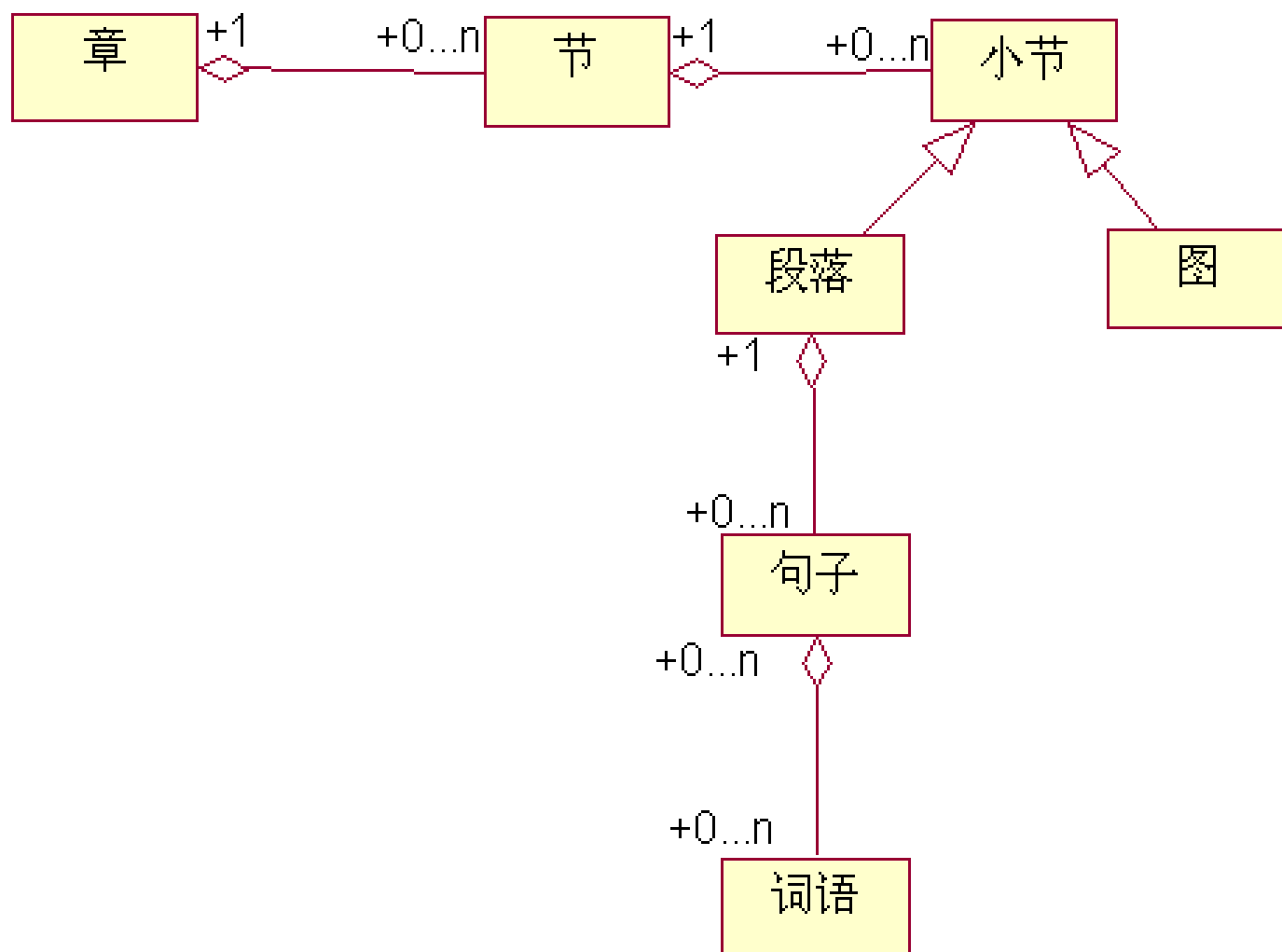
练习

№ 12

画具有下列结构的书的章节的对象聚集图：

一章由若干个小节组成，每个小节由若干个段落和图组成。段落由一些句子组成，每个句子包括一些词语。可以忽略标点符号，并且不必进一步追究图的结构

PEREC



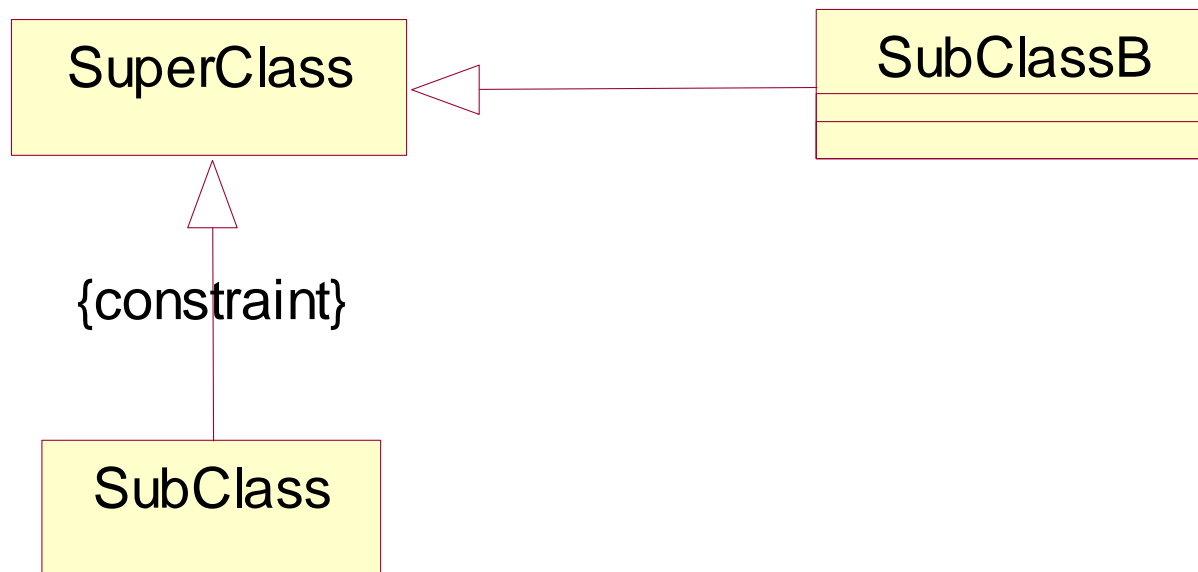


二、关联注释

№ 14

2.1 约束

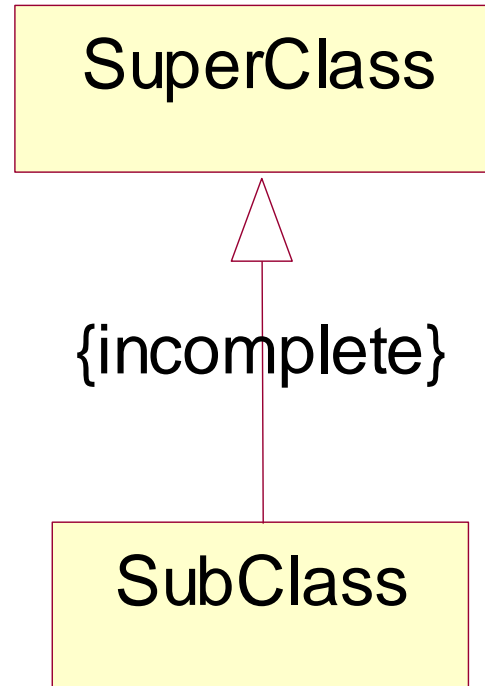
约束用来注释泛化关系



PEREC

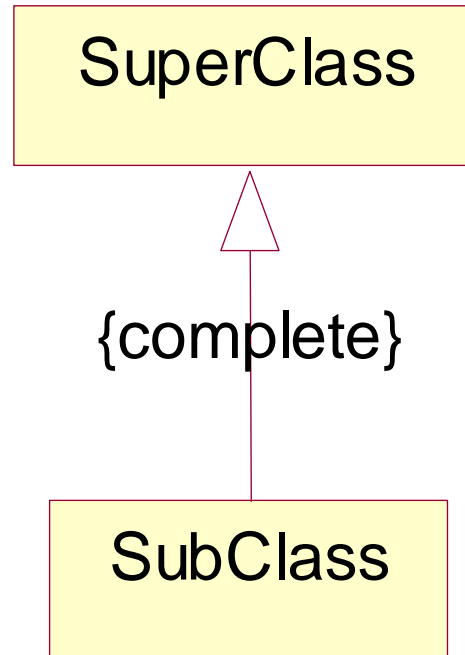


- 不完全约束



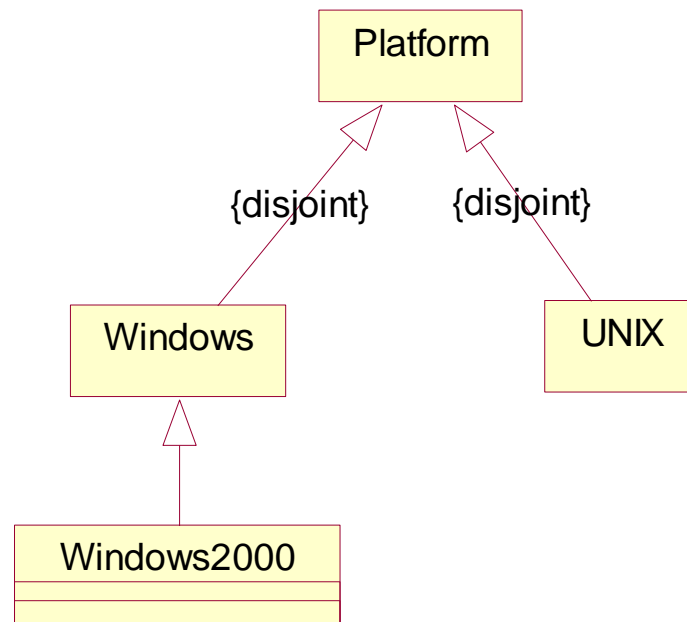


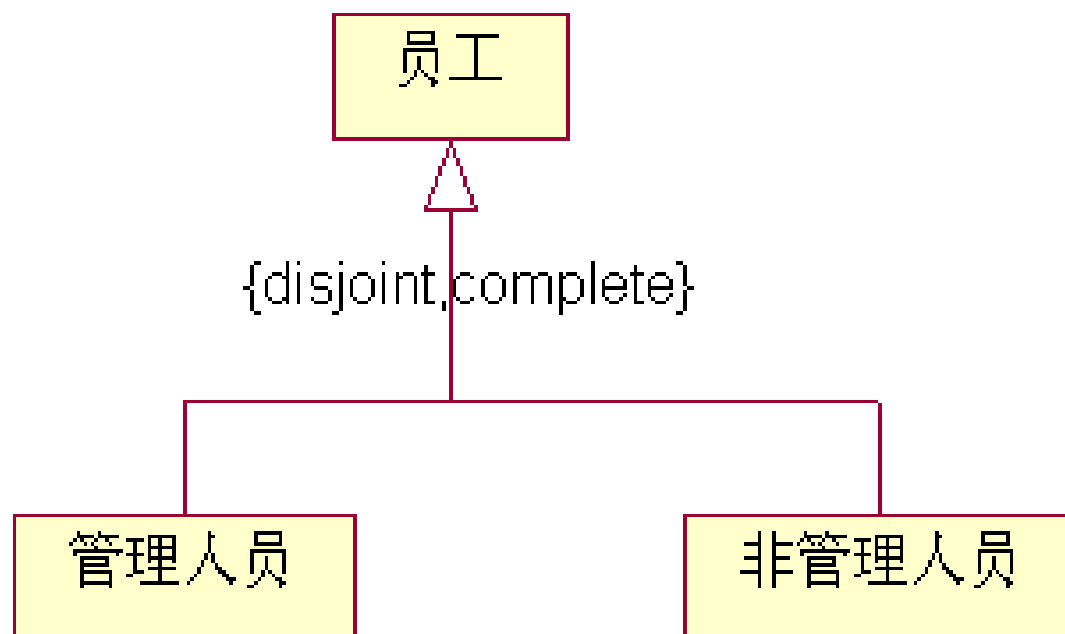
- 完全约束





- 解体约束（互斥约束）

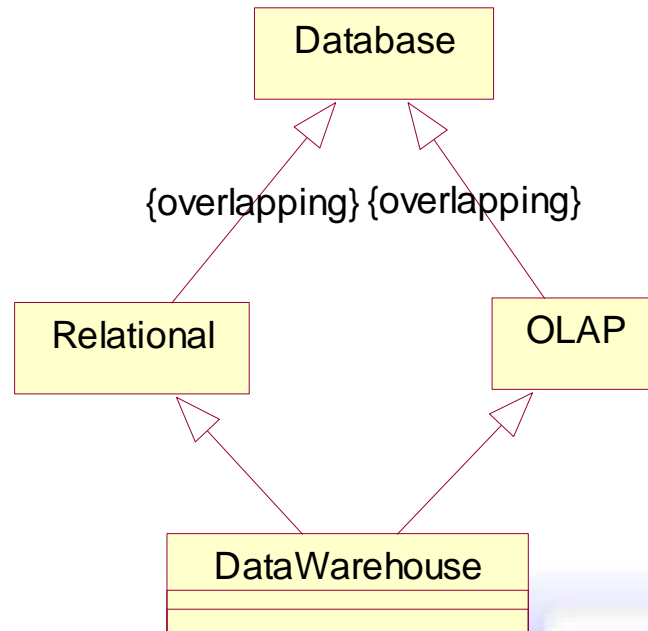






- 重叠约束

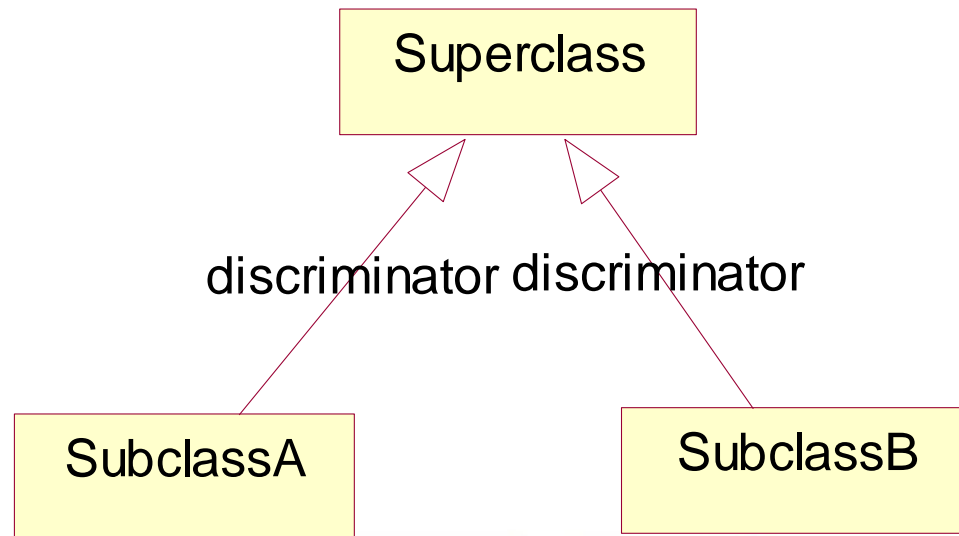
重叠约束与解体约束相反，表示子类可以共享相同的子类

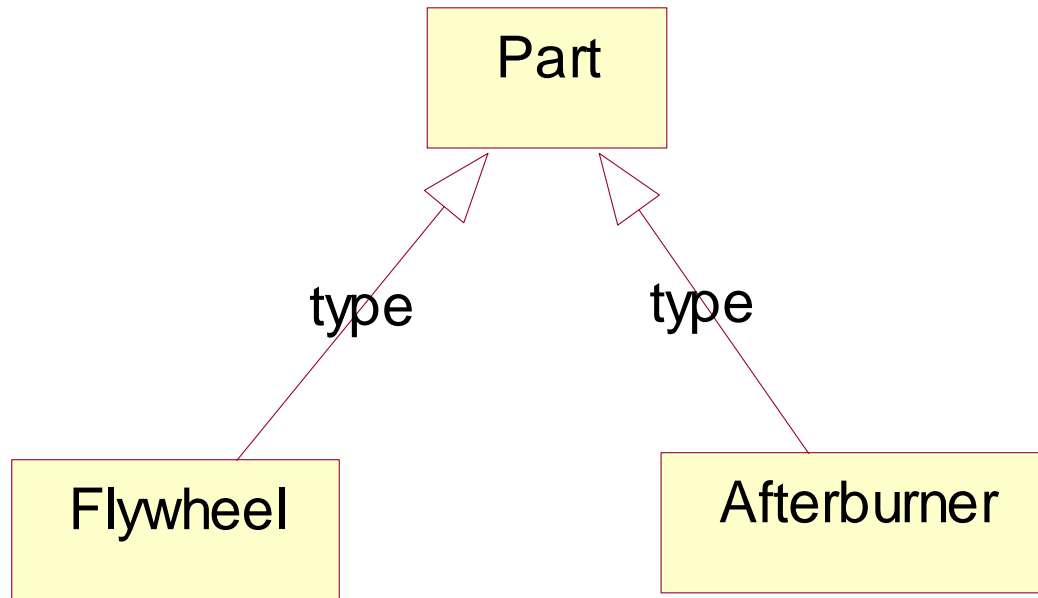




2.2 鉴别器

鉴别器用来说明泛化的作用，指示在泛化关系中，子类应用于超类时的角色







三、端点标记符

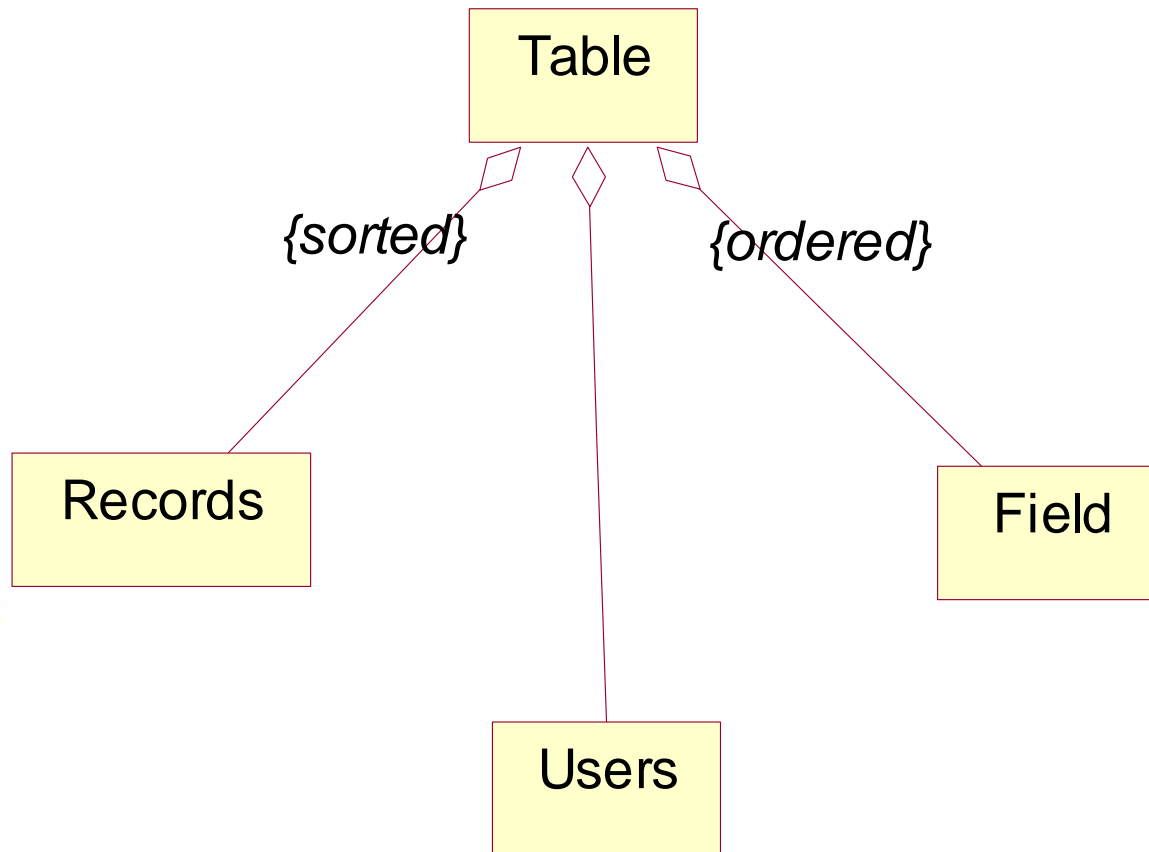
№ 22

3.1 有序和排序

有序约束用来说明一个类的对象以某个次序于另一个类的对象相关，但没有指定顺序。

排序约束说明对象之间的关联是指定顺序的，
排好序的

PEREC

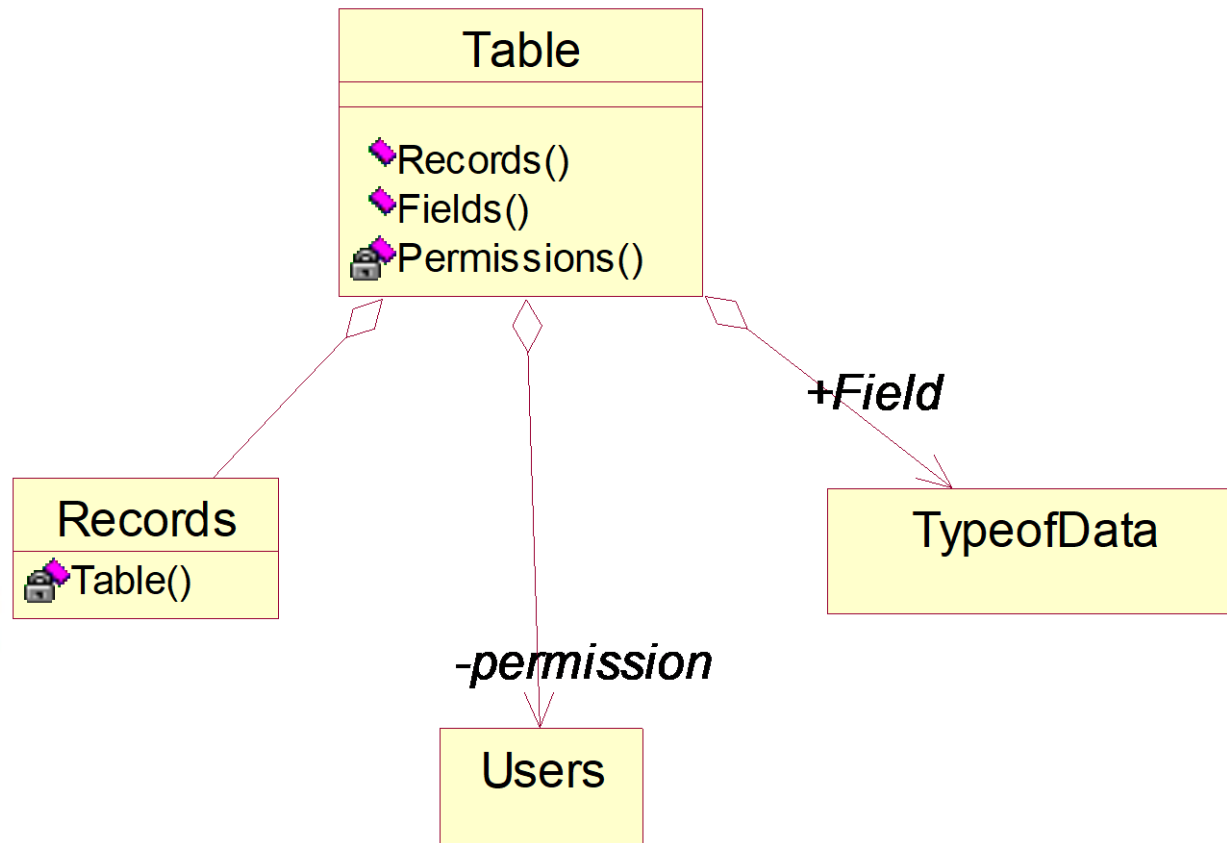




3.2 导航

导航就是在类之间通过操作进行关联访问。
导航可以是双向的也可以是单向的。

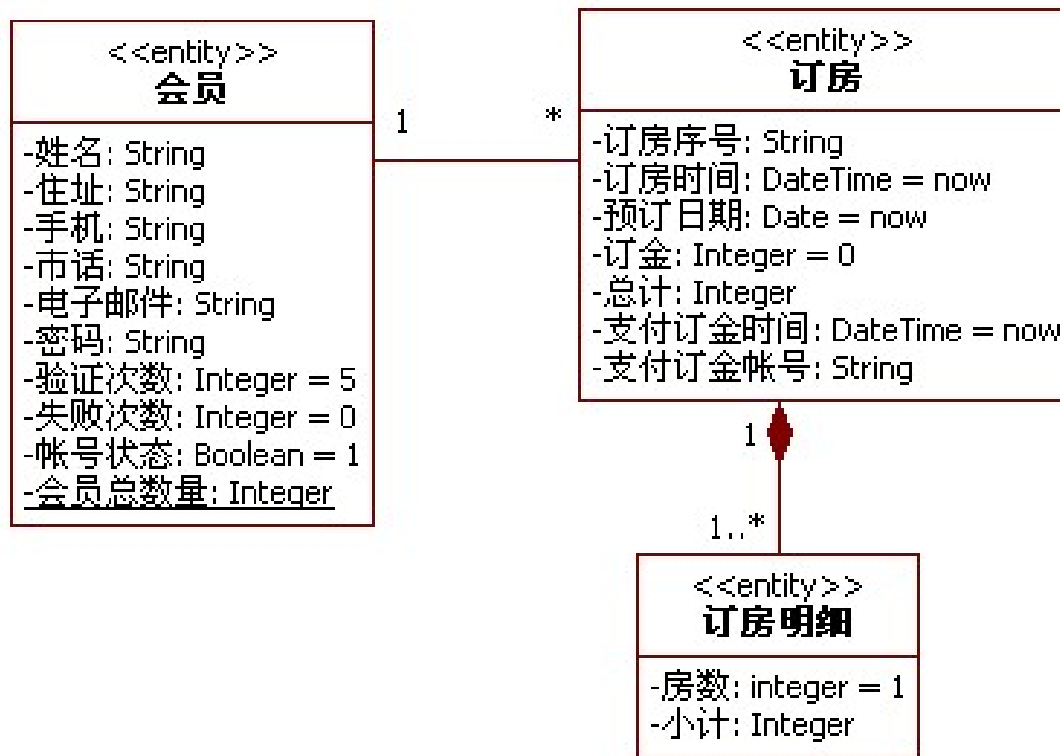
PEREC





从OO到关系型数据库

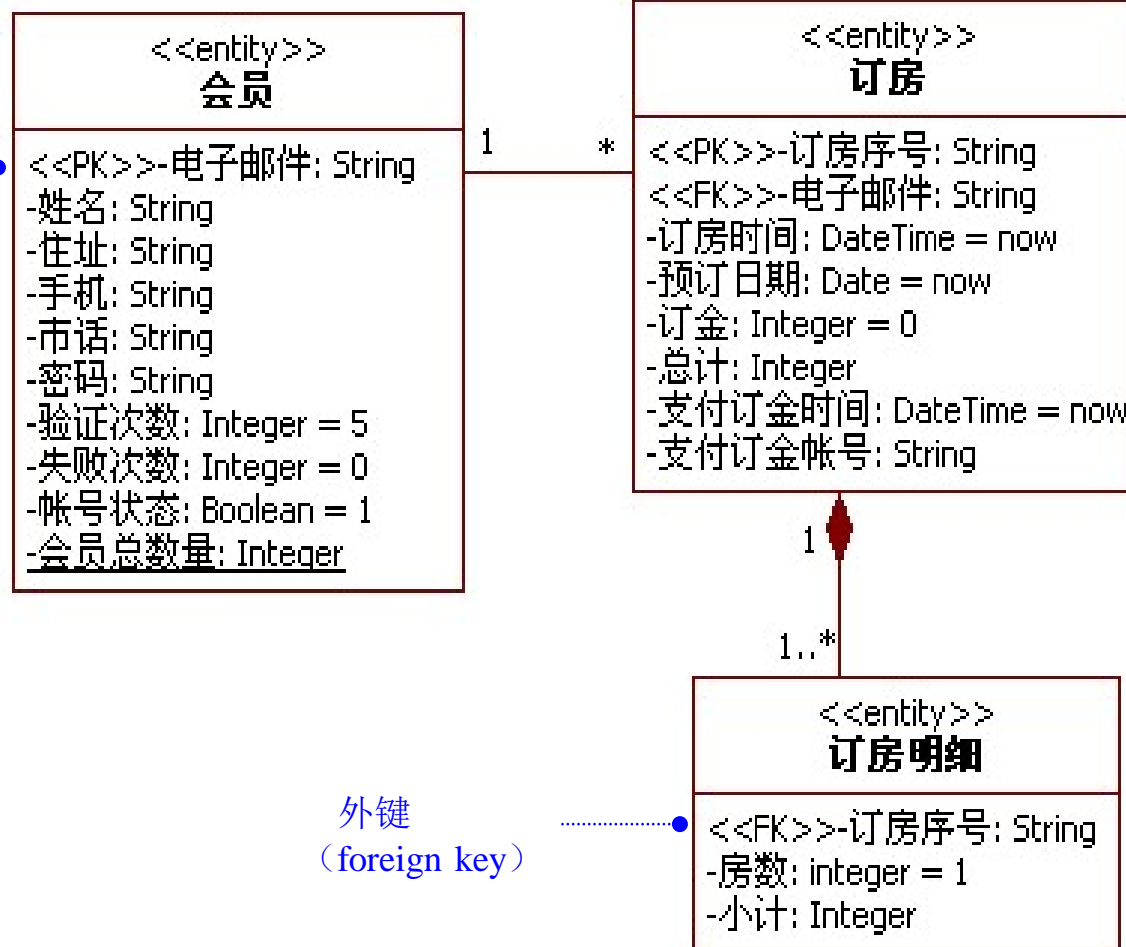
No 26



PEREC



主键
(primary key)



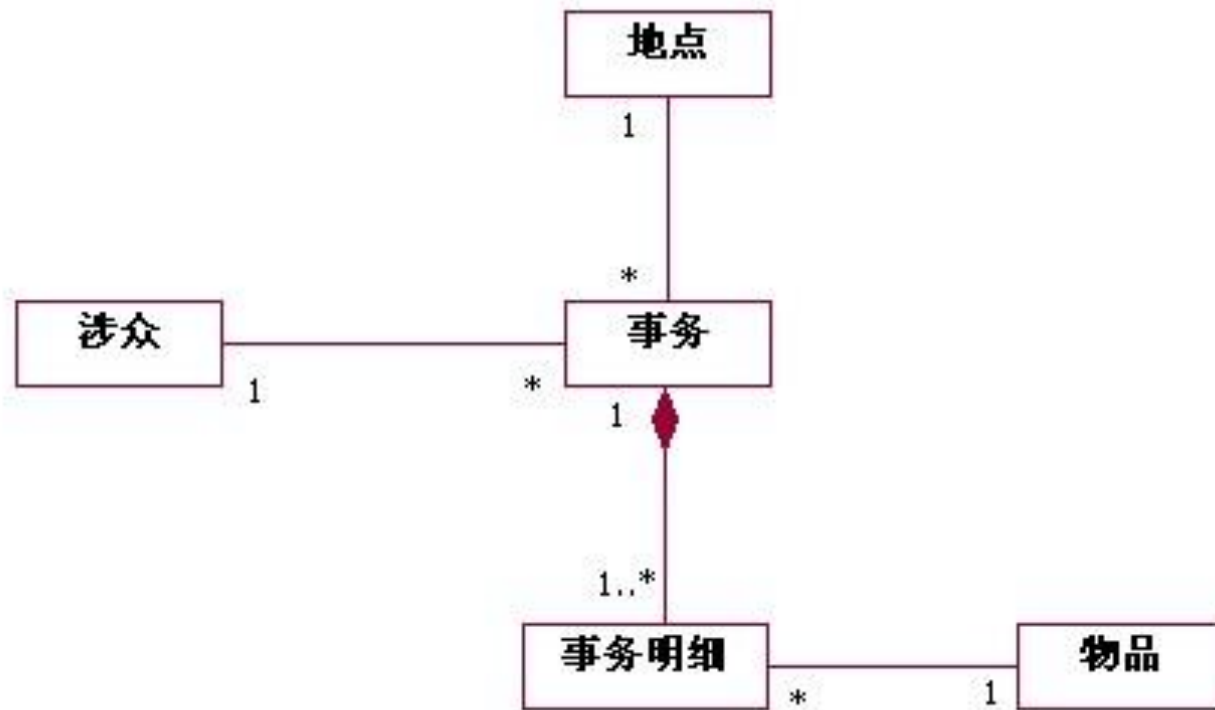
外键
(foreign key)

主键与外键



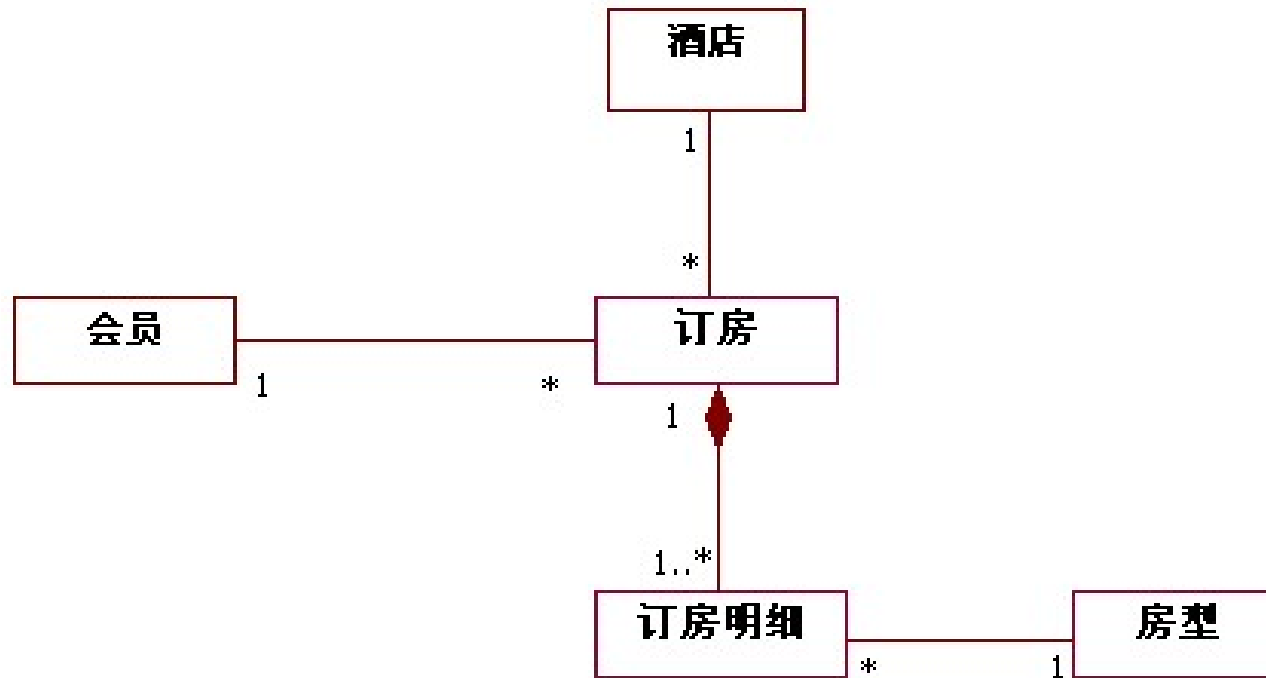
设计模式

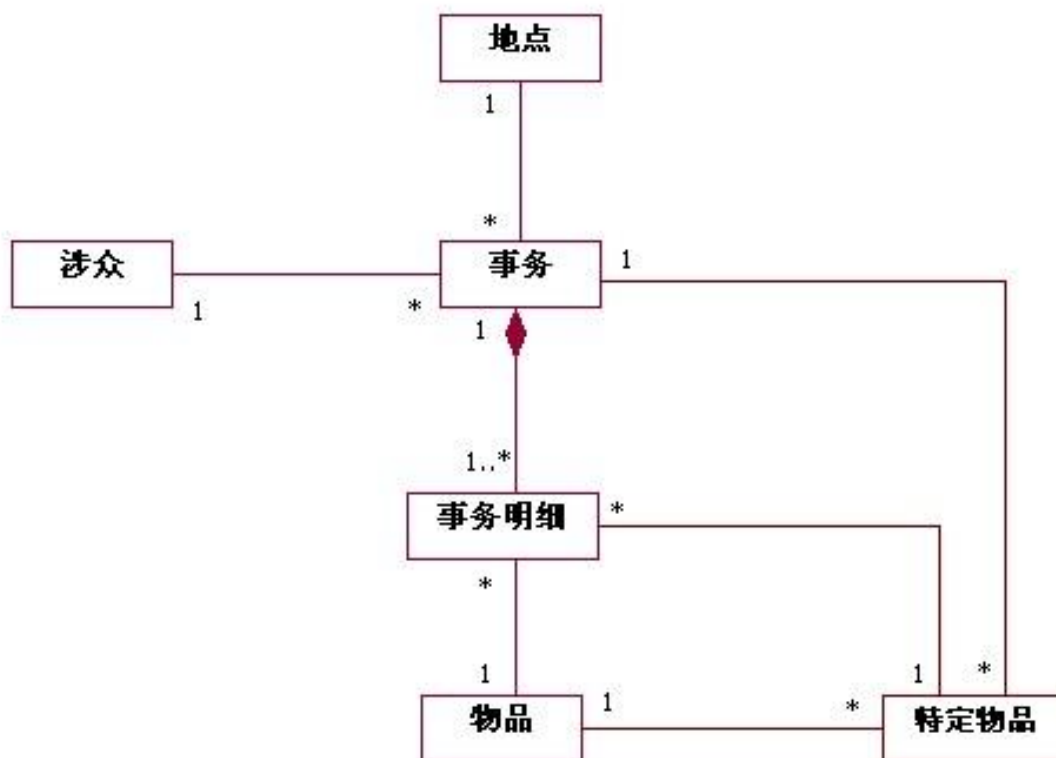
№ 28



事务模式

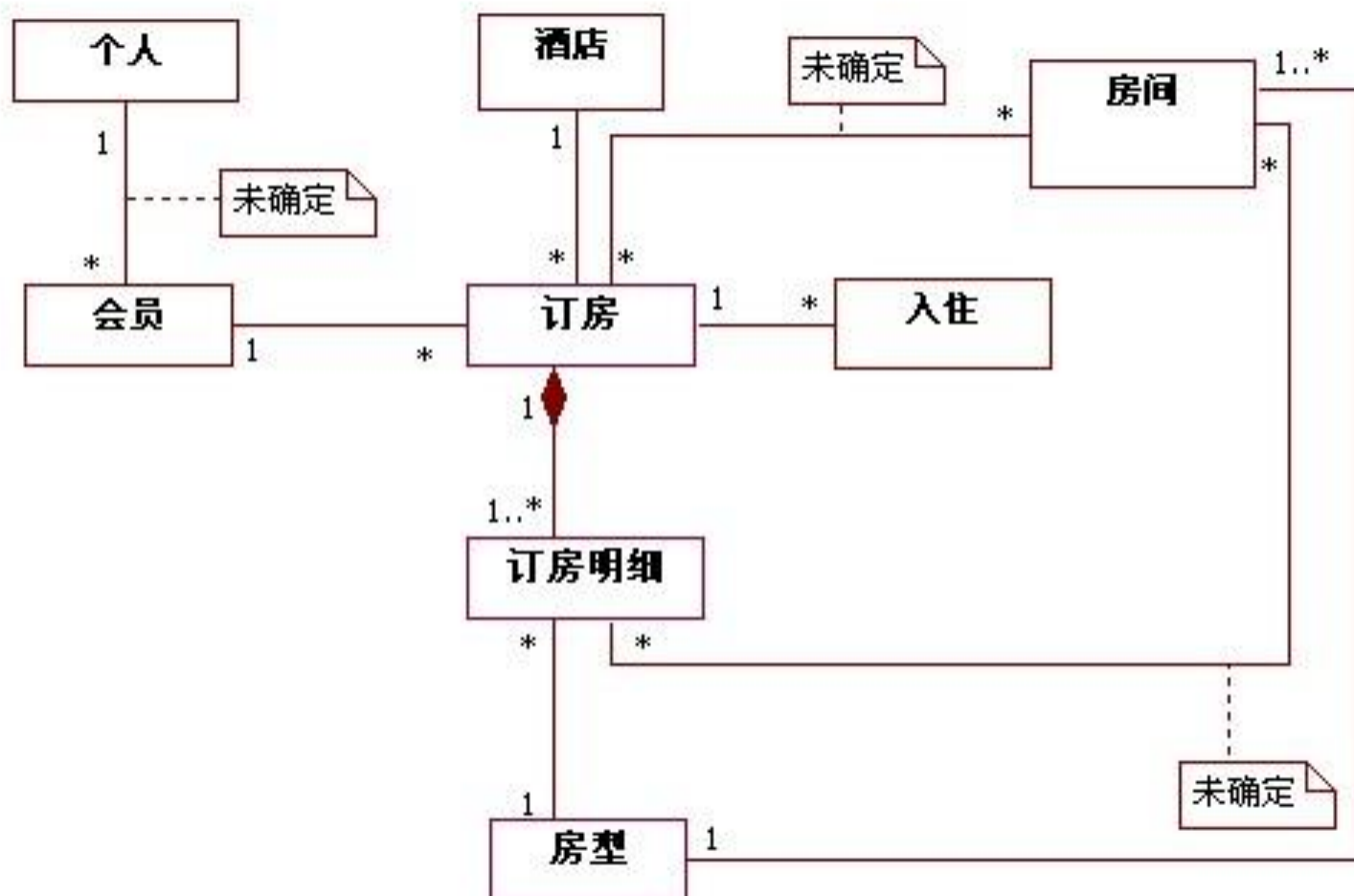
PEREC

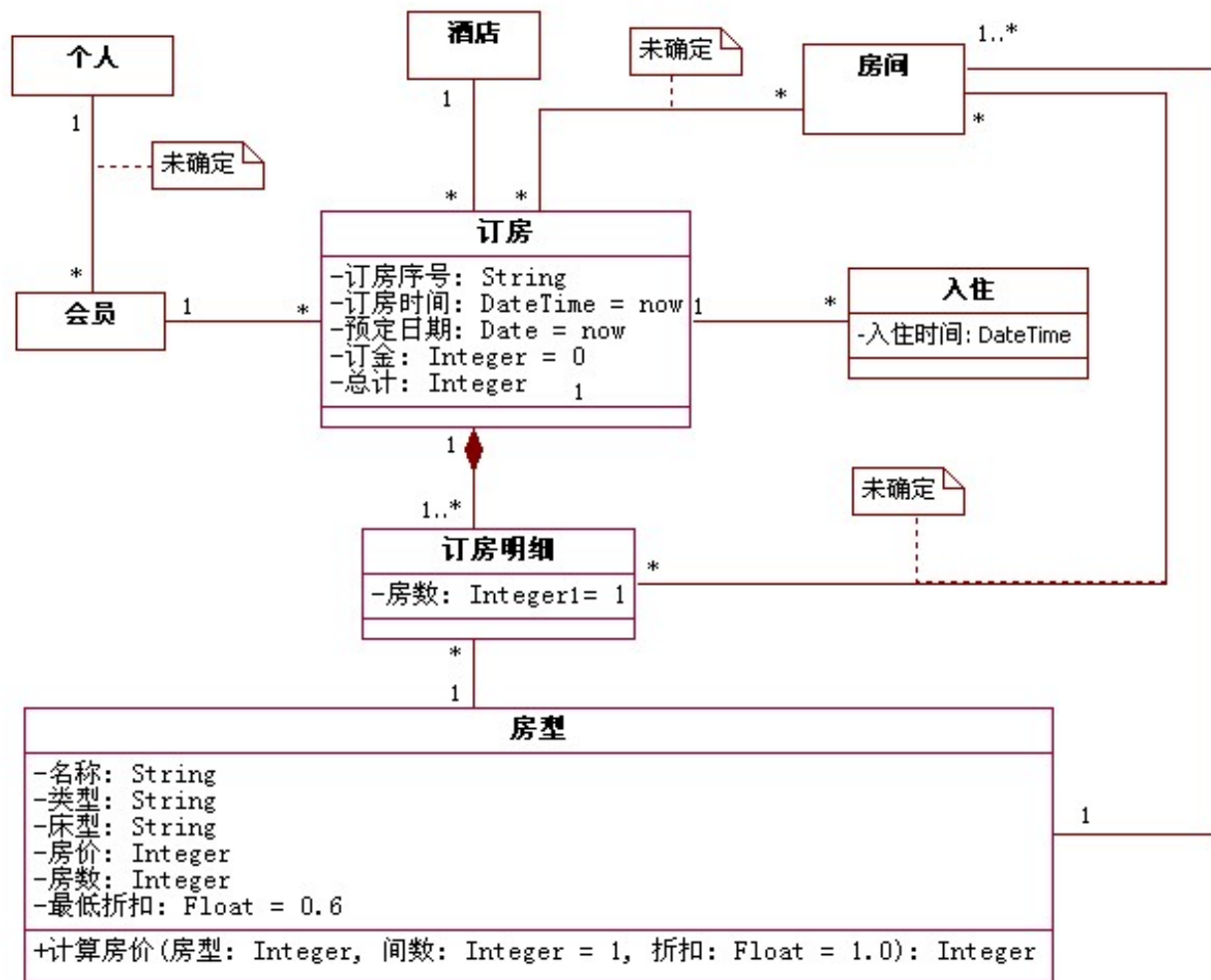




物品与特定物品

PEREC

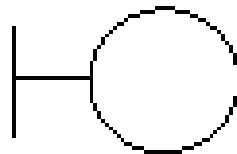




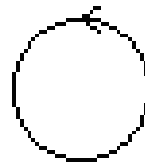


BCE模式

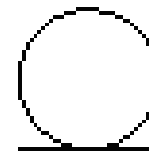
33



边界类
(boundary class)



控制类
(control class)



实体类
(entity class)

边界、控制、实体类

PEREC



BCE模式

34

- ⊕ 实体类—保存问题领域中的重要信息，封装数据结构和数据储存有关的变化。
- ⊕ 控制类—用来控制用例执行期间的复杂运算或者业务逻辑，通常针对一个用例，就会对应生成一个控制类。
- ⊕ 边界类—边界类用来隔离系统内外，通常负责接收并响应系统内外的信息。

PEREC



BCE模式

35

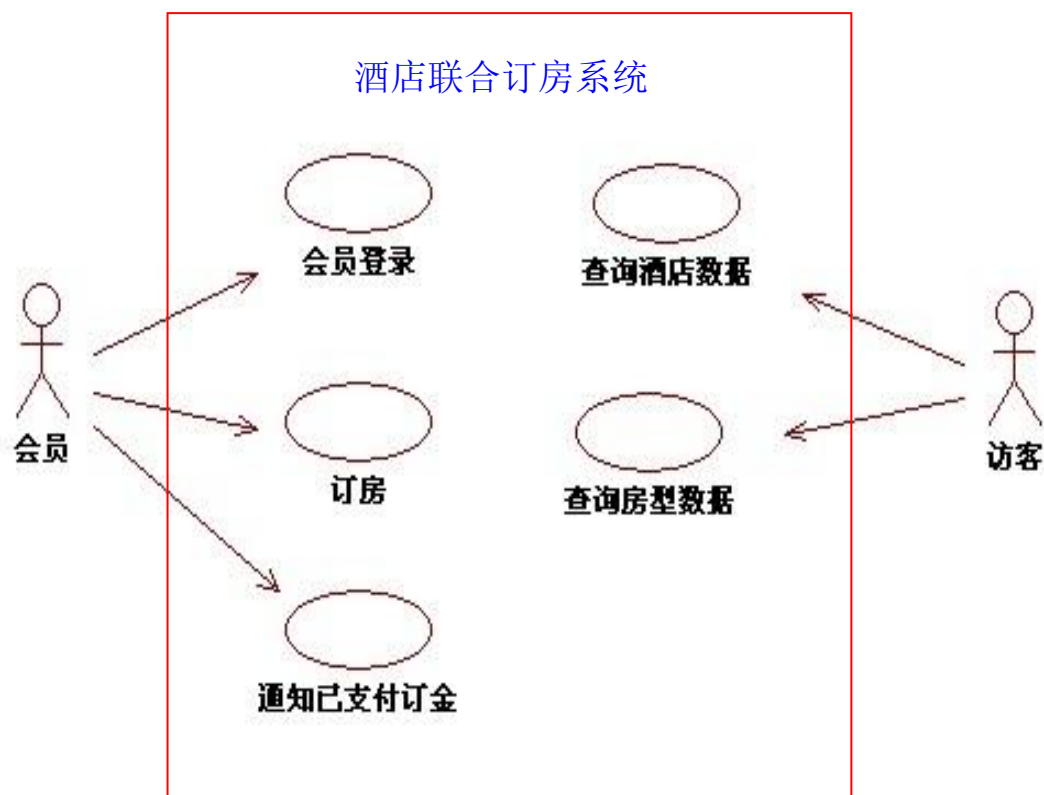
- ⊕ 一个用例，可以对应生成一个控制类。
- ⊕ 参与者对象只能跟边界对象互动。
- ⊕ 实体对象不能发送消息给边界对象和控制对象。
- ⊕ 单纯对数据表进行增删改查的话，可以不设置控制对象，让边界对象直接发送消息给实体对象。

PEREC



酒店联合订房系统

36



有分析的用例



酒店联合订房系统

37

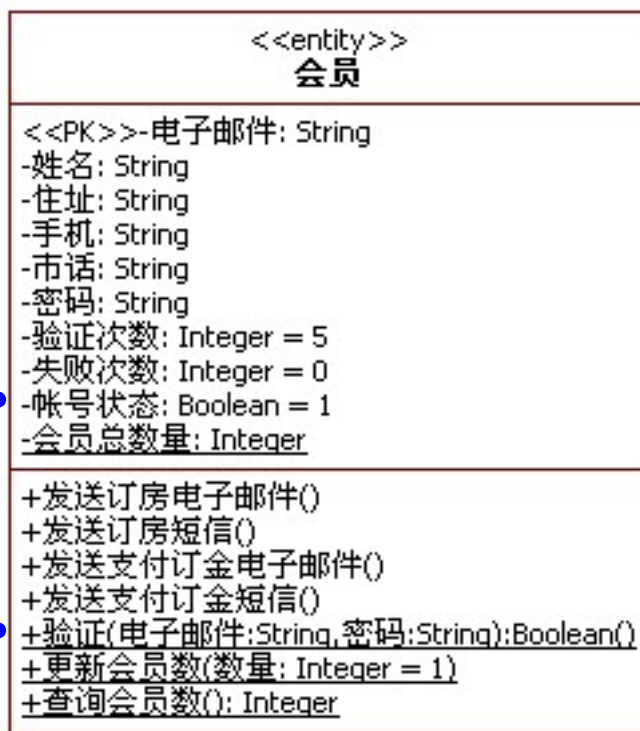
- ⊕ 会员登录
- ⊕ 查询酒店数据
- ⊕ 查询房型数据
- ⊕ 通知已预订
- ⊕ 订房



会员登录

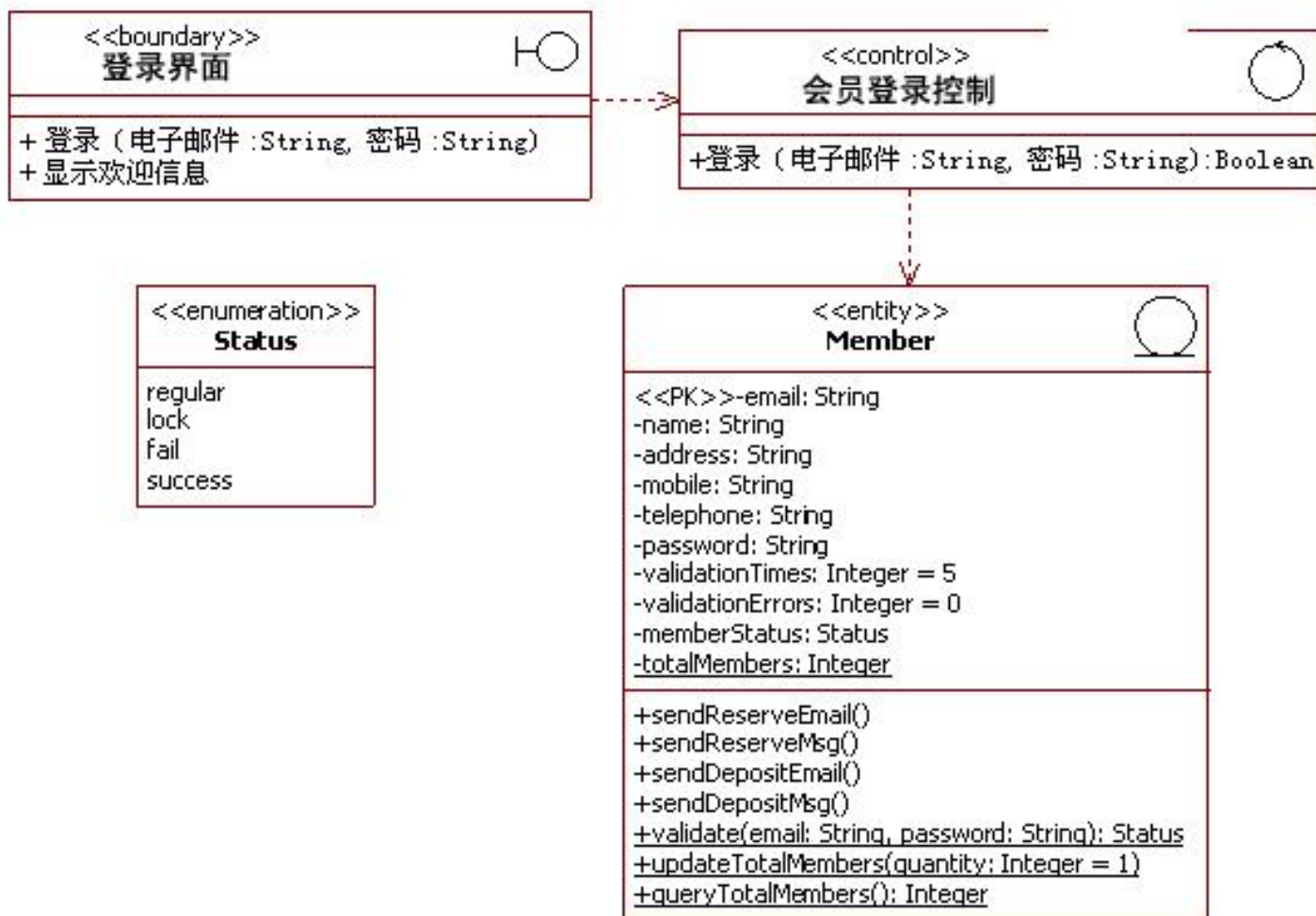
38

改用枚举类型



会员

PEREC

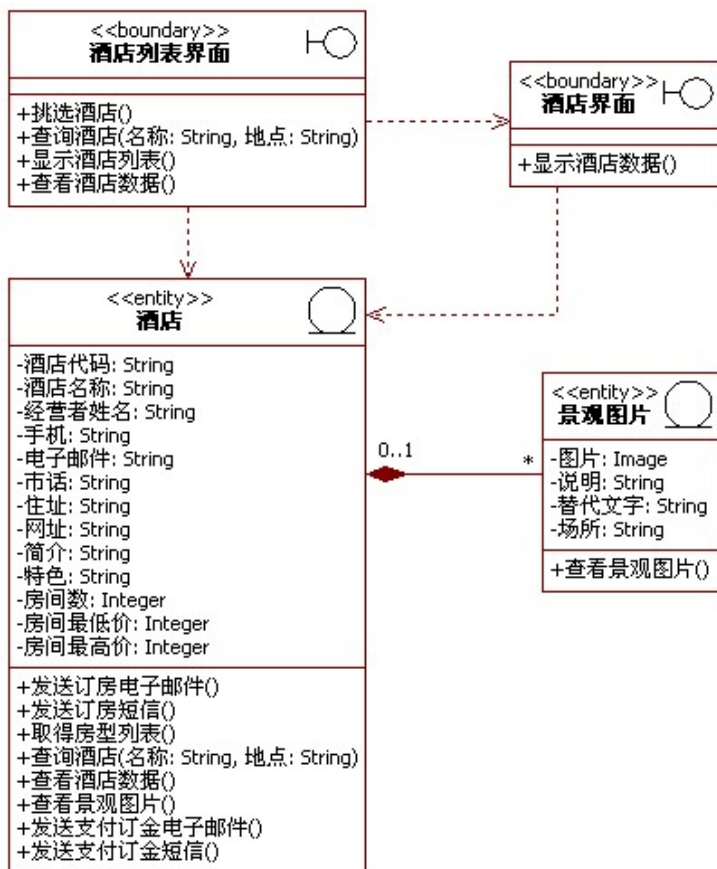


“会员登录”的BCE类图



查询酒店数据

40

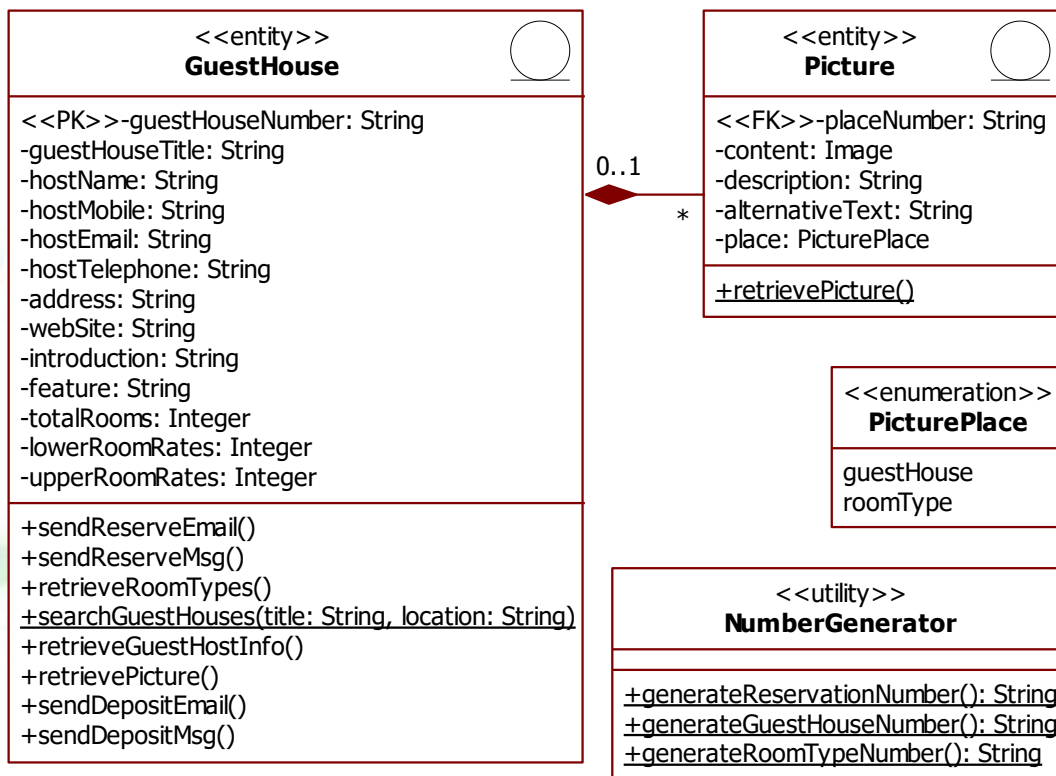


“查询酒店数据”的BCE类图



查询酒店数据

41

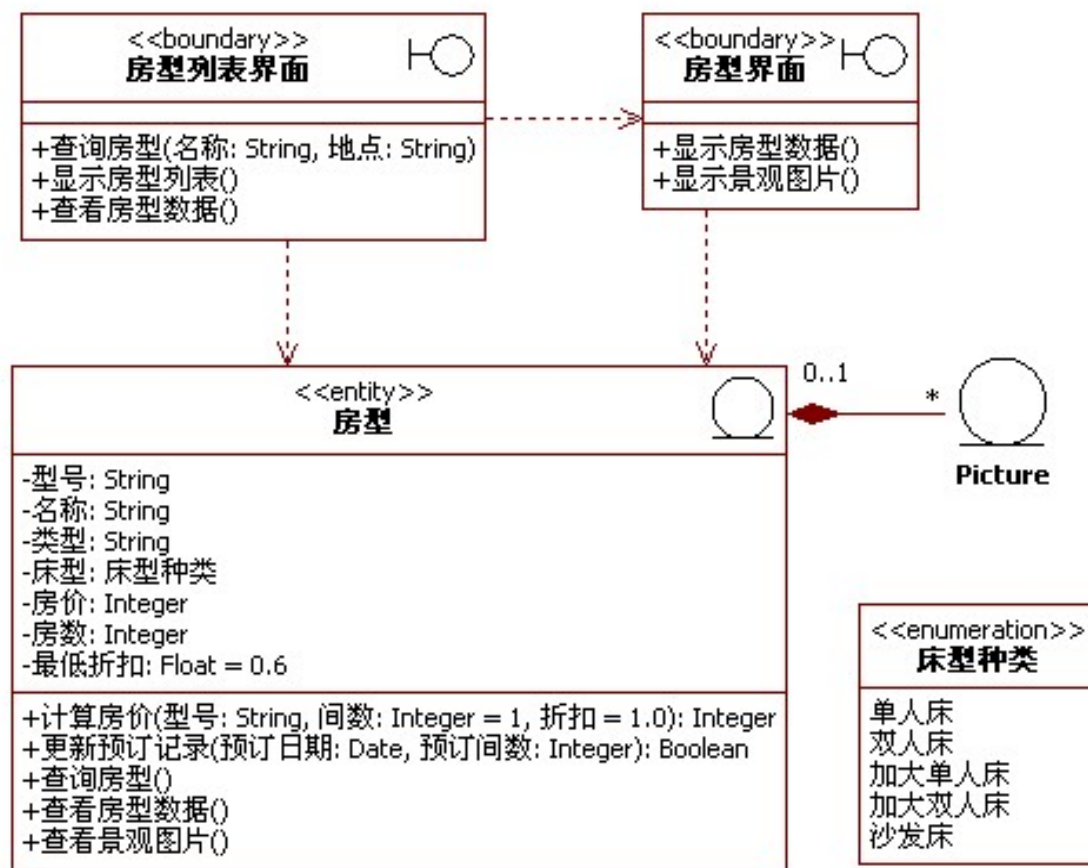


“查询酒店数据”的实体类

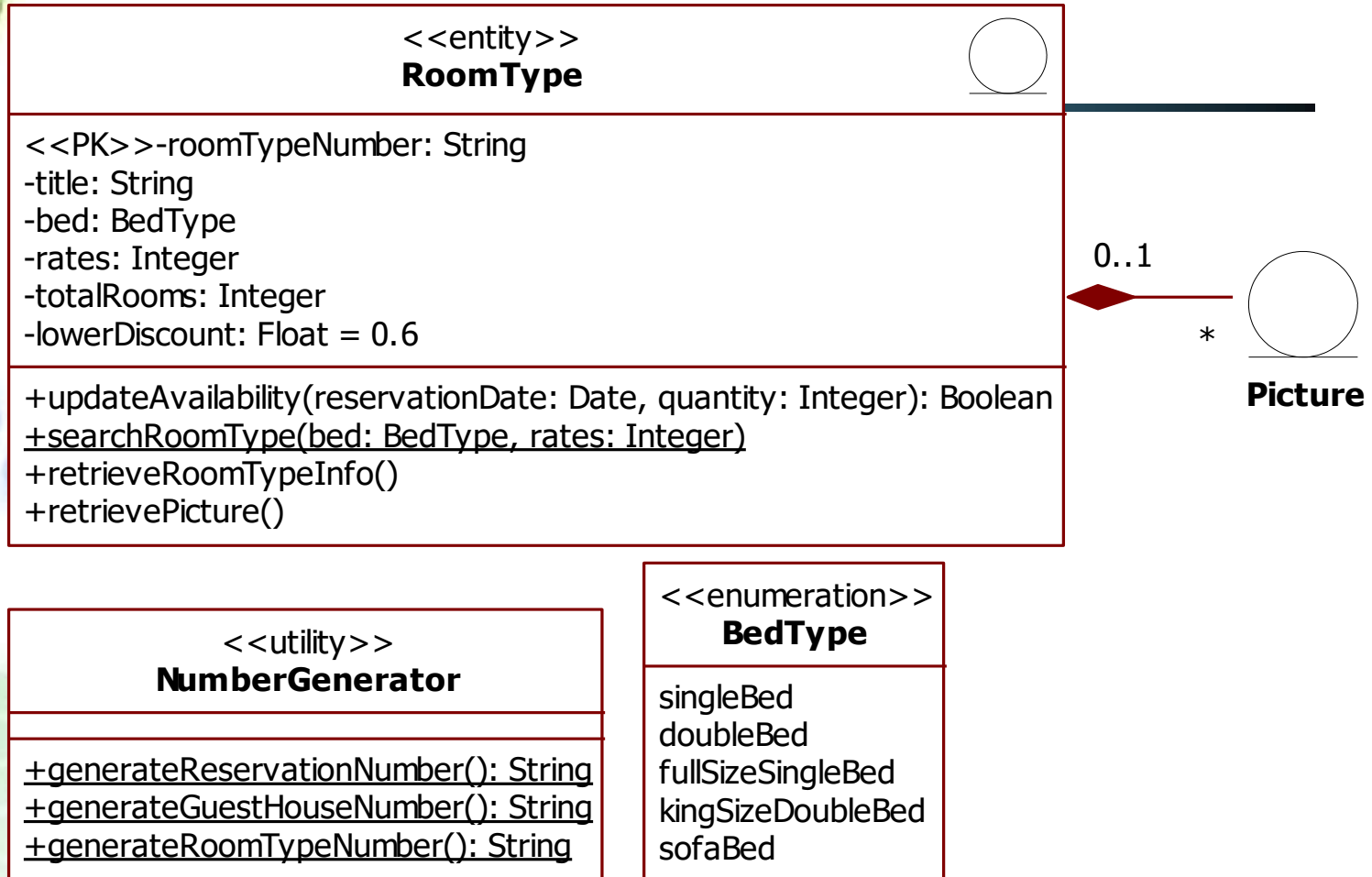


查询房型数据

42



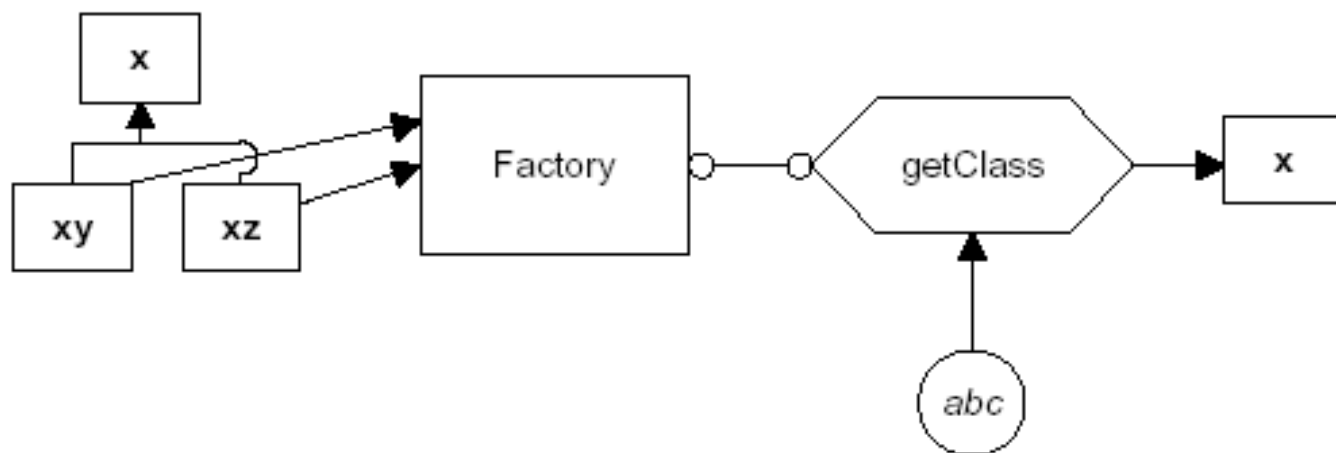
“查询房型数据”的BCE类图



“查询房型数据”的实体类



⊕ Factory pattern





例子

№ 45

⊕ 基类—Namer

```
class Namer {  
    //a simple class to take a string apart into two names  
    protected String last; //store last name here  
    protected String first; //store first name here  
    public String getFirst() {  
        return first; //return first name  
    }  
    public String getLast() {  
        return last; //return last name  
    }  
}
```

PEREC



⊕ 两个子类

```
class FirstFirst extends Namer { //split first last
public FirstFirst(String s) {
    int i = s.lastIndexOf(" "); //find sep space
    if (i > 0) {
        //left is first name
        first = s.substring(0, i).trim();
        //right is last name
        last = s.substring(i+1).trim();
    }
    else {
        First = “”; // put all in last name
        last = s; // if no space
    }
}
}
```

№ 46

PEREC



```
class LastFirst extends Namer { //split last, first
public LastFirst(String s) {
    int i = s.indexOf(","); //find comma
    if (i > 0) {
        //left is last name
        last = s.substring(0, i).trim();
        //right is first name
        first = s.substring(i + 1).trim();
    }
    else {
        last = s; // put all in last name
        first = ""; // if no comma
    }
}
}
```

No 47

PEREC



建立factory类

```
class NameFactory {  
    //returns an instance of LastFirst or FirstFirst  
    //depending on whether a comma is found  
    public Namer getNamer(String entry) {  
        int i = entry.indexOf(","); //comma determines  
        name order  
        if (i>0)  
            return new LastFirst(entry); //return one class  
        else  
            return new FirstFirst(entry); //or the other  
    }  
}
```

PEREC



如何使用

№ 49

The screenshot shows a Windows-style application window titled "Name Divider". Inside the window, there is a label "Enter name:" followed by a text input field containing "Smith, Sandy". Below this, there are two more input fields: "First name" containing "Sandy" and "Last name" containing "Smith". At the bottom of the window, there are three buttons: "Compute", "Clear", and "Close".

Name Divider

Enter name:

Smith, Sandy

First name: Sandy

Last name: Smith

Compute Clear Close

PEREC



⊕ 初始化

```
NameFactory nfactory = new NameFactory();
```

⊕ 调用方法

```
private void computeName() {  
    //send the text to the factory and get a class back  
    Namer namer = nfactory.getNamer(entryField.getText());  
    //compute the first and last names  
    //using the returned class  
    txFirstName.setText(namer.getFirst());  
    txLastName.setText(namer.getLast());  
}
```



何时使用这个模式：

- 不能确定创建那种类的对象
- 使用子类具体实现系统功能
- 你想把创建哪个类放在本地实现



其他设计模式

№ 52

创建模式:

- ⊕ factory
- ⊕ Abstract factory
- ⊕ Singleton
- ⊕ Builder
- ⊕ prototype

PEREC



结构化模式:

- ⊕ Adapter
- ⊕ Bridge
- ⊕ Composite
- ⊕ Decorator
- ⊕ Façade
- ⊕ Flyweight
- ⊕ proxy



行为模式

- ⊕ Chain of responsibility
- ⊕ Command
- ⊕ Interpreter
- ⊕ Iterator
- ⊕ mediator