

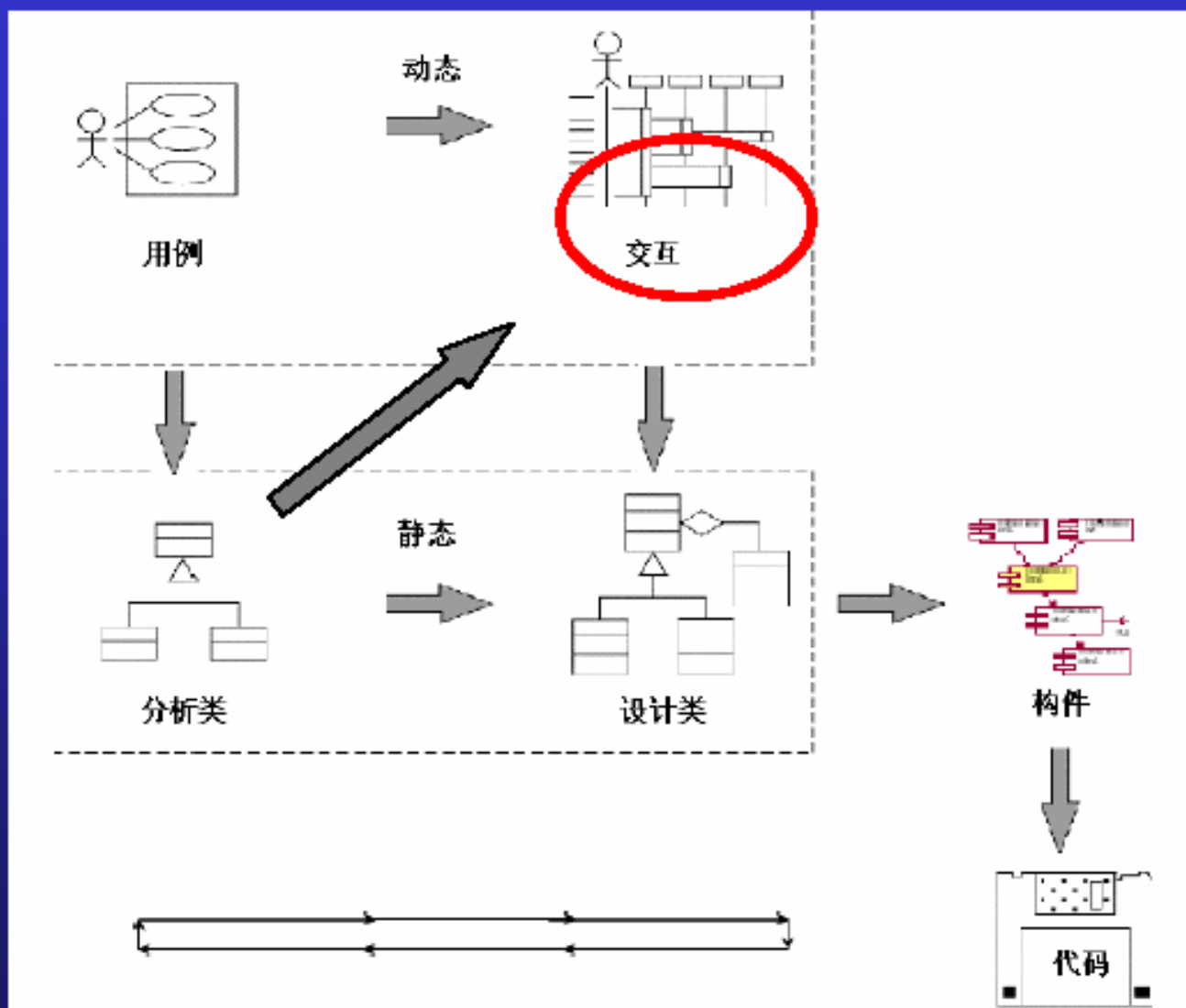


顺序图

PEREC



开发流程—动态建模





一 顺序图定义

No 3

⊕ 交互图:顺序图和协作图

⊕ 顺序图:以时间顺序来安排对象的交互

把用例的功能分配给类

建模对象交互和高层的参与者与
对象的交互

PEREC

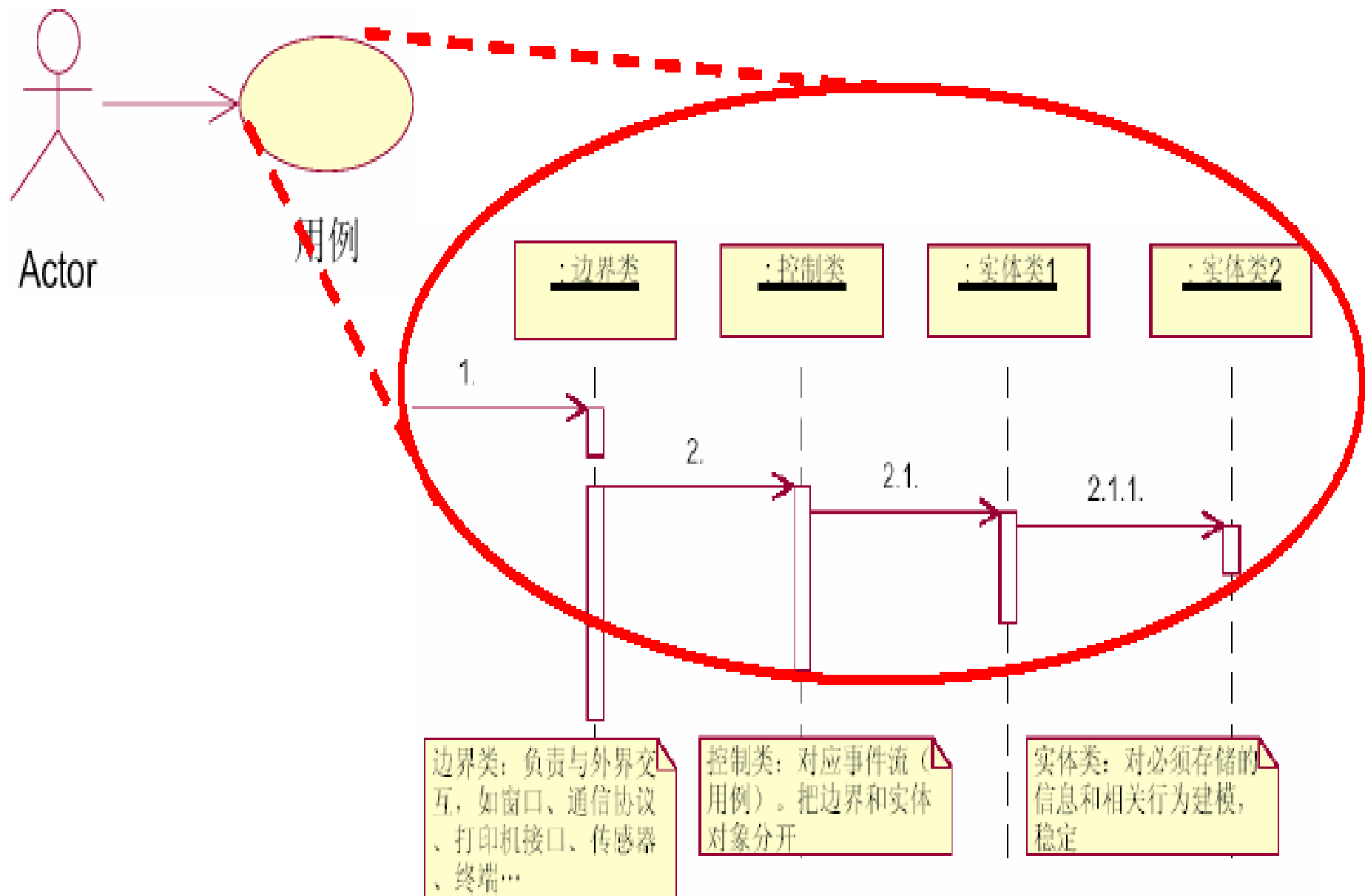


二 顺序图的标记符

№ 4

- ⊕ 活动对象: 任何在系统中扮演角色的对象,
如对象实例和参与者
- ⊕ 活动对象之间的通信: 消息
说明了对象之间的控制流,是顺序图的关键

PEREC

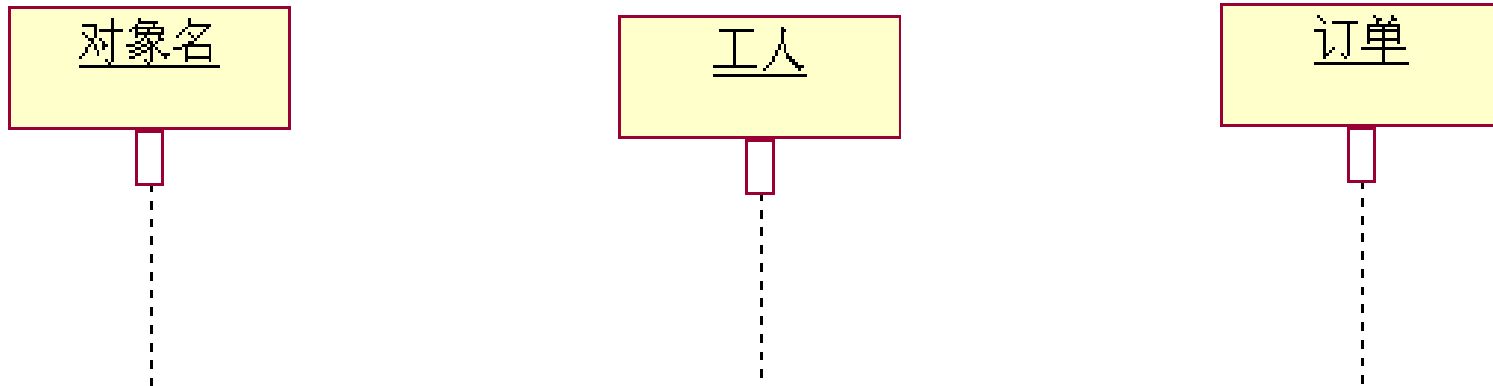




2.1 活动对象

活动对象的标记符如图所示.

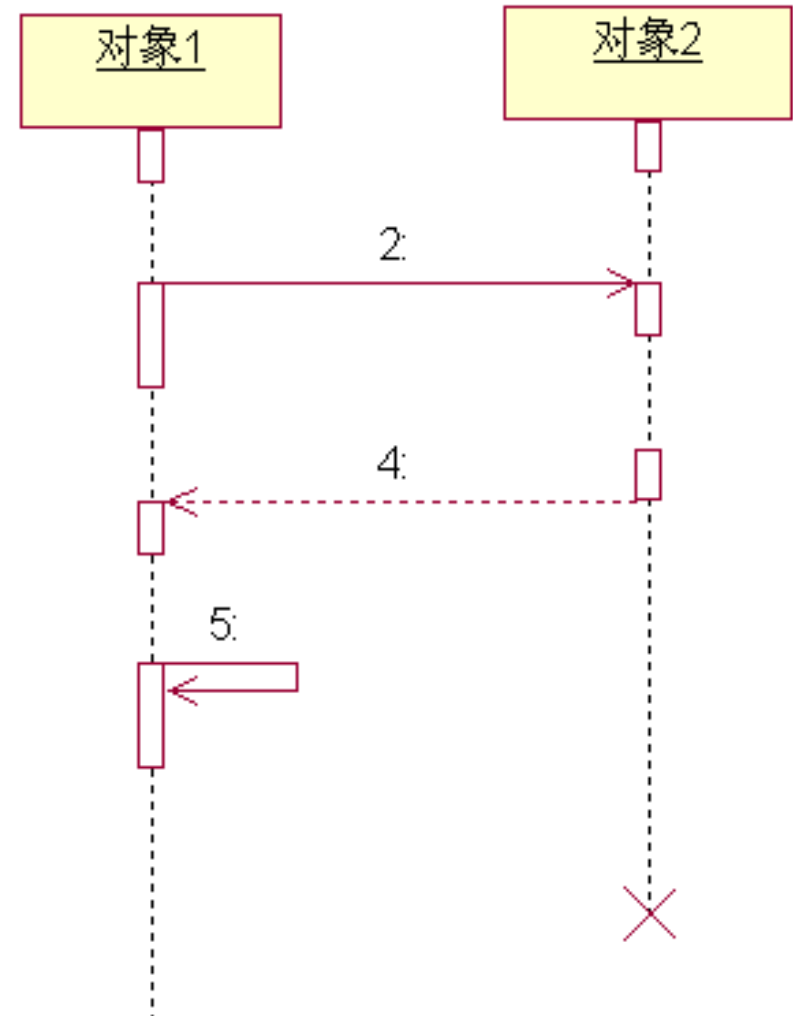
对象拖出的长虚线为生命线,说明了按照时间顺序对象所发生的事件.

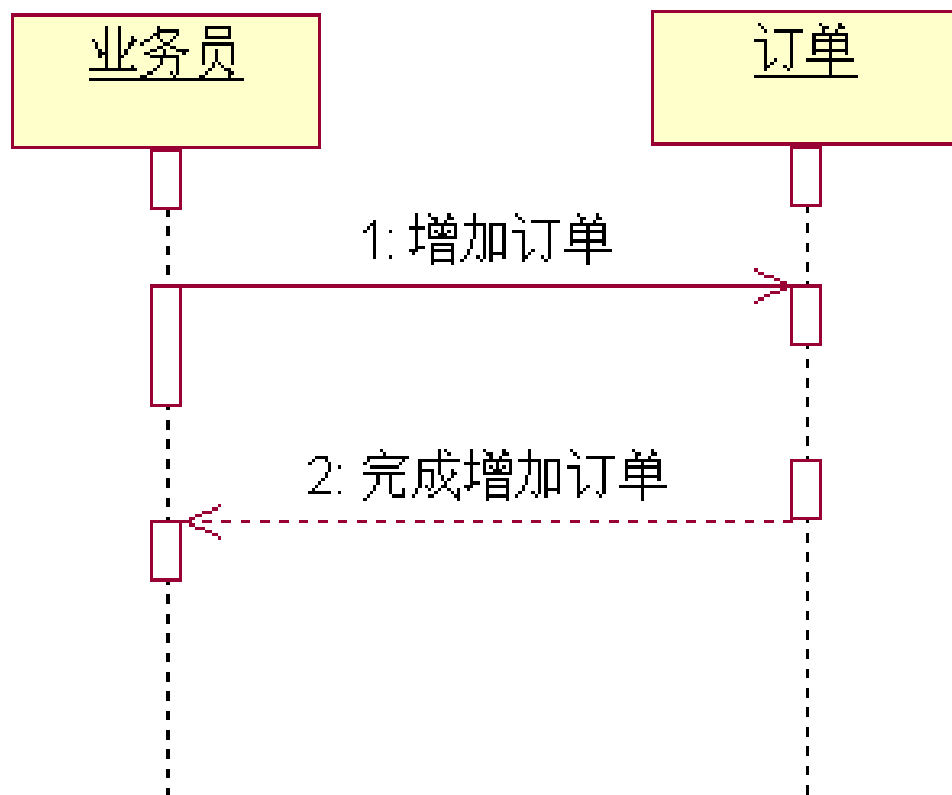




2.2 消息

消息用来说明不同对象之间的通讯。







2.3 消息类型

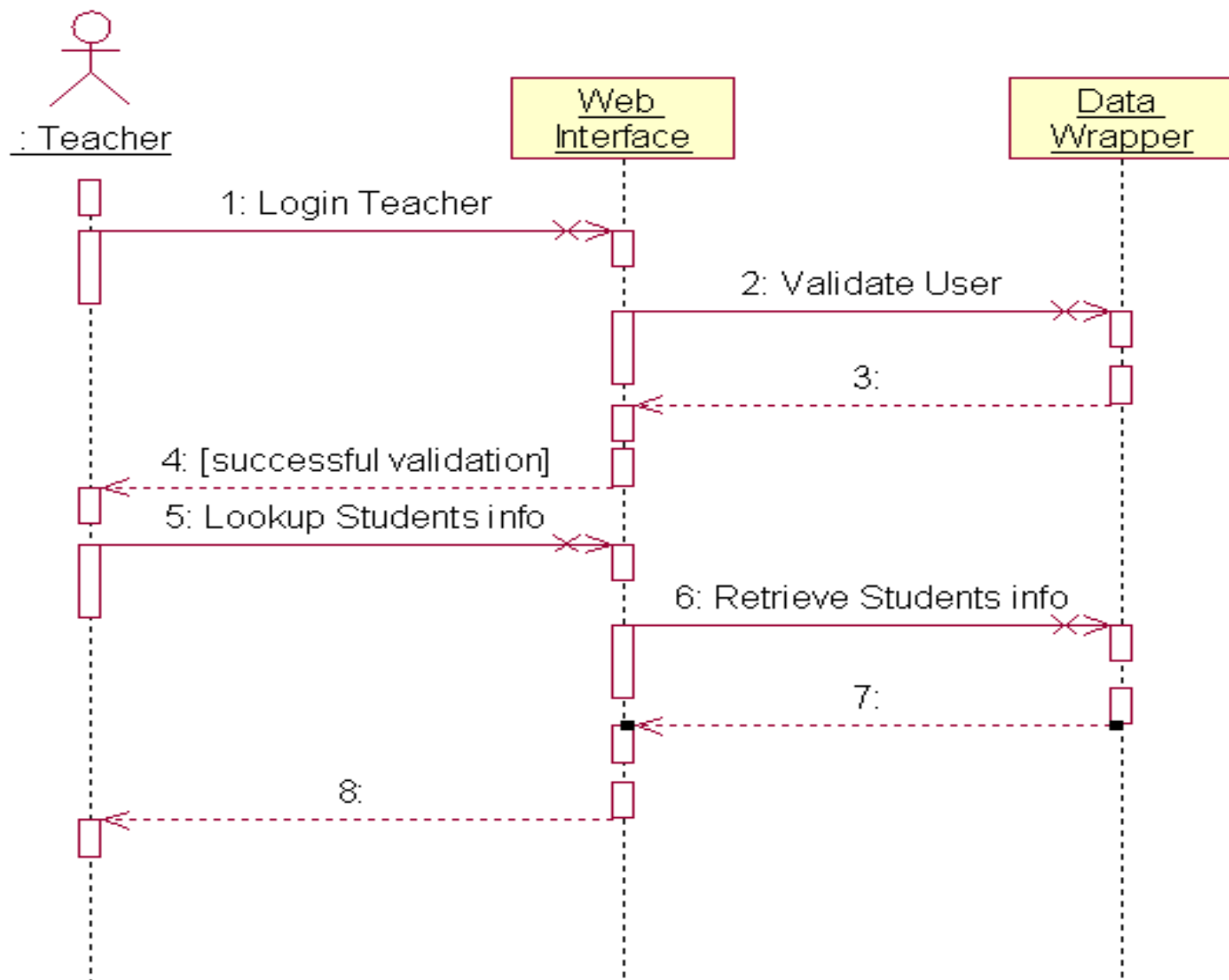
- ⊕ 同步:在消息完成之前,工作流中断
- ⊕ 返回:控制流回到活动对象,同步消息完成操作
- ⊕ 异步:活动对象不用等待消息的完成响应
- ⊕ 简单:不用详细区分同步还是异步消息



(1)同步消息

同步消息用于控制流的顺序执行操作,一个操作接着一个,前一个没有完成,后面的不能进行.

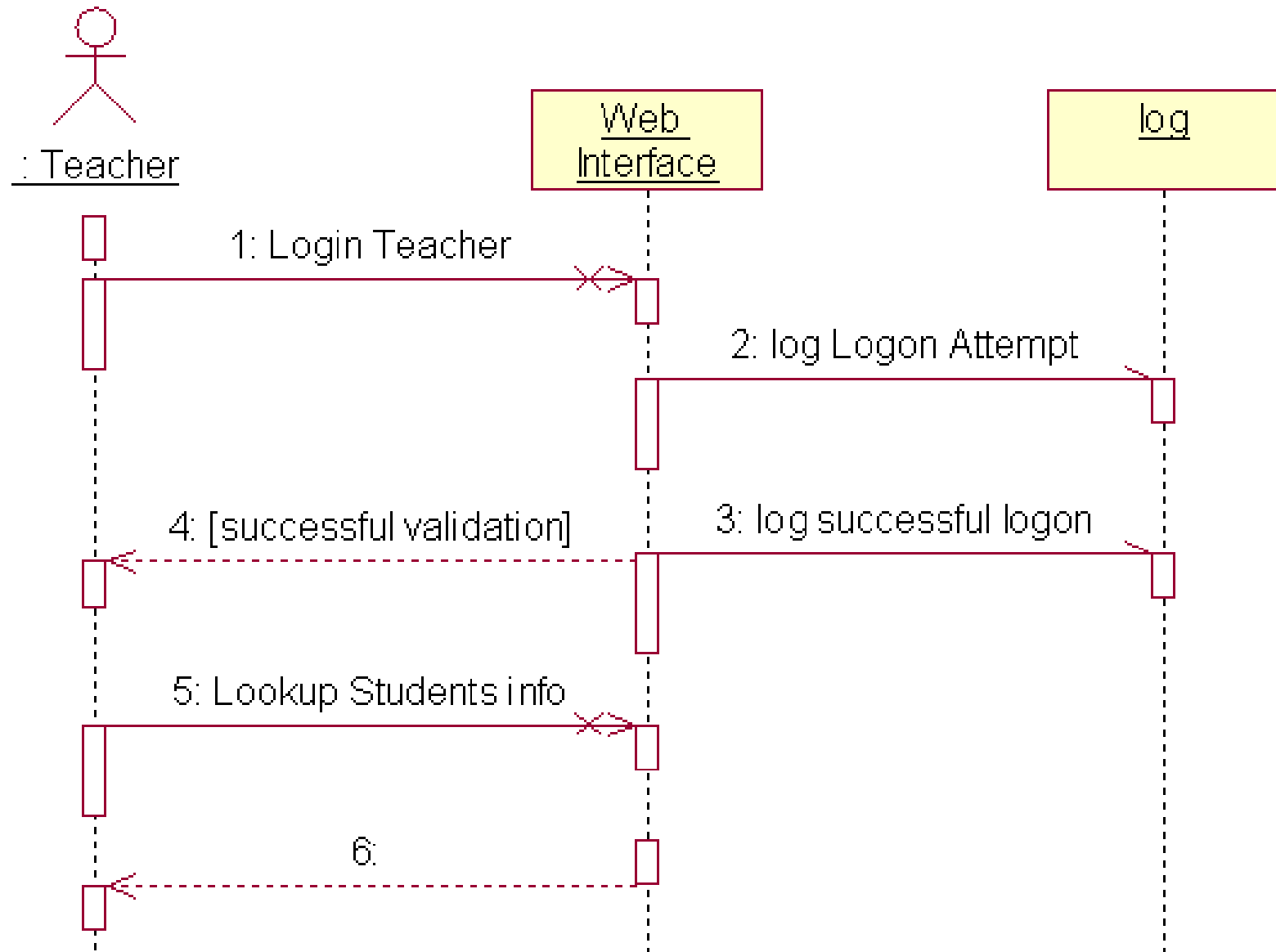






(2)异步消息

异步消息用于控制流的并行执行操作,下一个操作开始执行不需要上一个操作完成为条件.



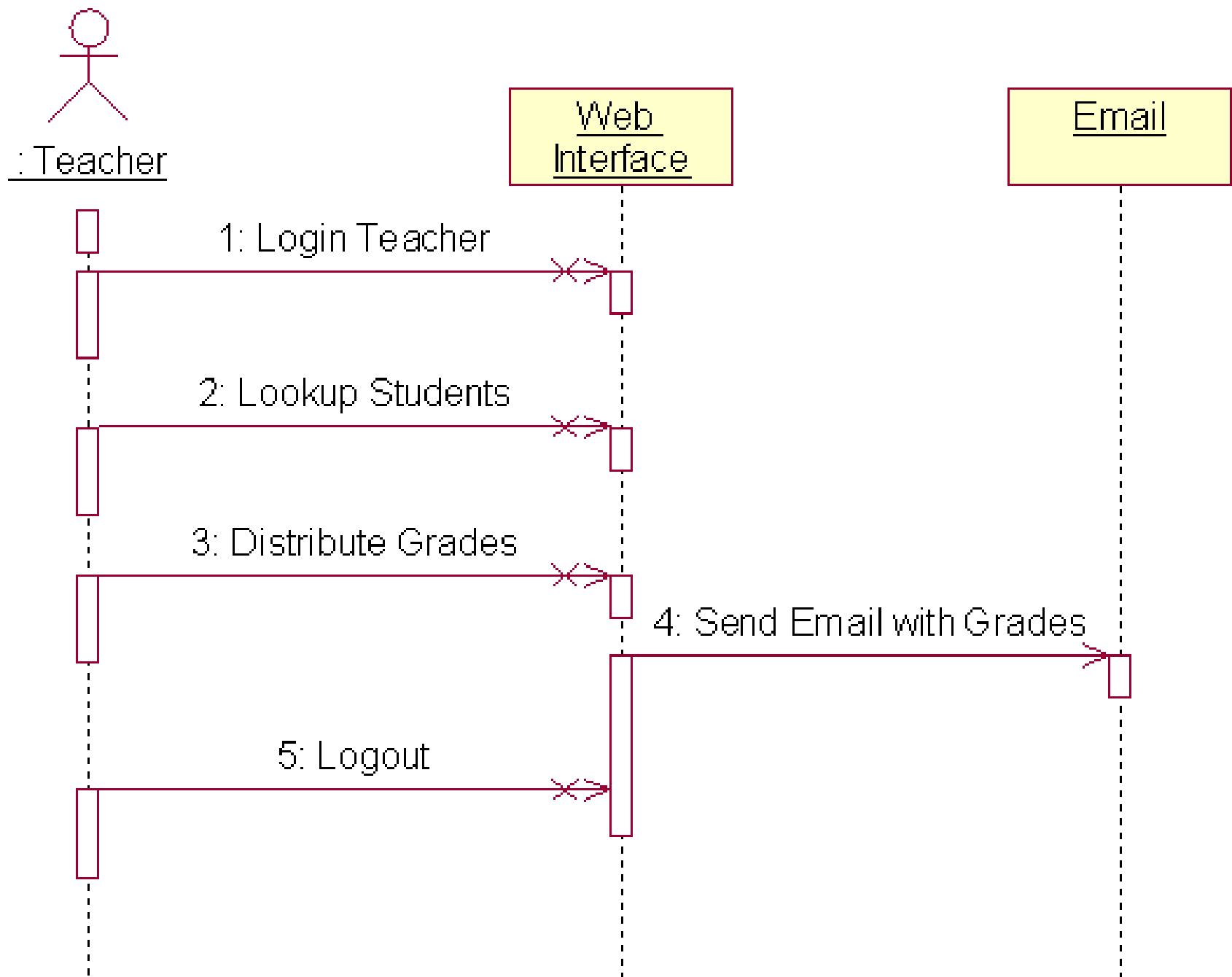


(3)简单消息

简单消息不是和同步异步消息的并列类型的消息,它只是说明不用或者不需要详细描述消息的同步和异步类型.

例如在系统的高层分析过程中,还没有设计对象的交互细节.







⊕ (4)返回消息



PEREC



三 顺序图的其它技术

№ 17

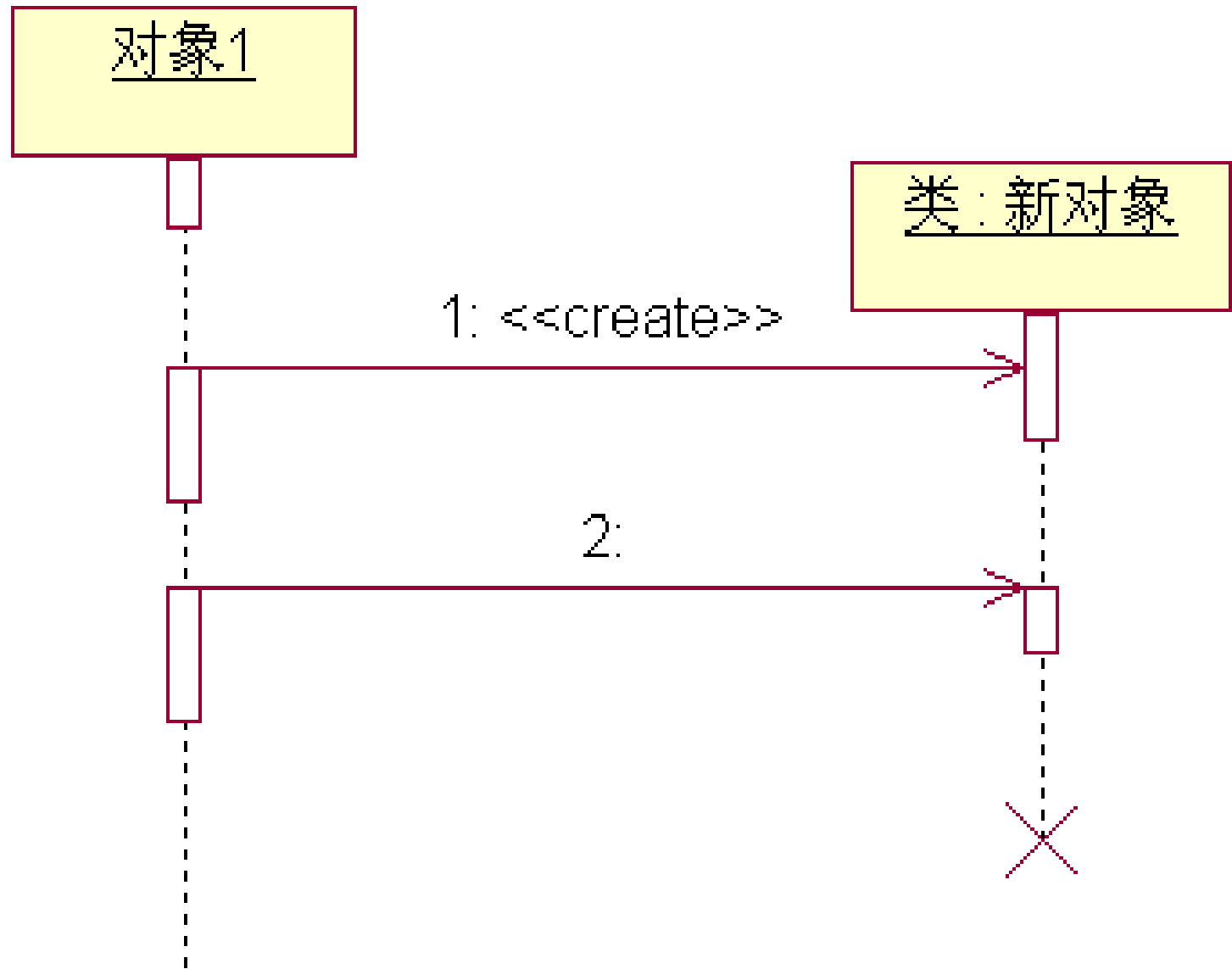
- ⊕ 创建对象
- ⊕ 使用状态
- ⊕ 分支和从属流

PEREC



3.1 创建对象

就是在控制流传递的过程中,新创建一个对象,即发送“**create**”消息来完成对象的创建,这个对象生成以后就可以和其它对象一样进行消息的传递,在该对象完成其使命以后,可以发送“**destroys**”消息来删除它.





3.2 使用状态

状态属于对象,在生命线上出现

PEREC



3.3 分支与从属流

分支和从属流是改变顺序图中控制流的两种方法,条件的不同会改变控制流的流向.

分支: 允许控制流走向不同的对象.

从属流: 允许控制流根据条件的不同而进行改变,但只允许改变为相同对象的另一条生命线分支.

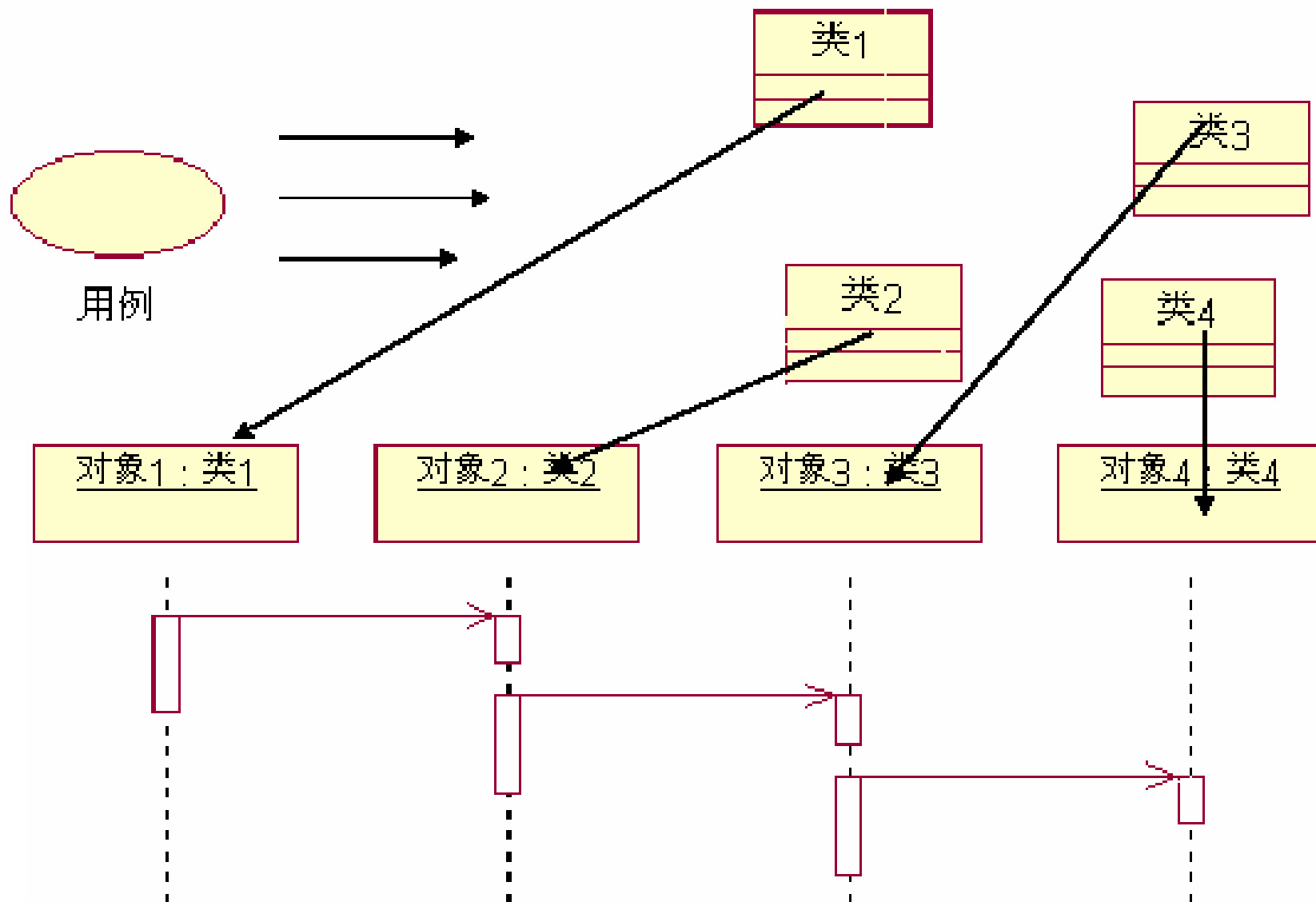


四 建模顺序图的步骤

№ 22

- ⊕ 确定 workflow
- ⊕ 寻找对象和角色
- ⊕ 从左到右布置对象
- ⊕ 添加消息和条件以便控制 workflow
- ⊕ 绘制总图

PEREC



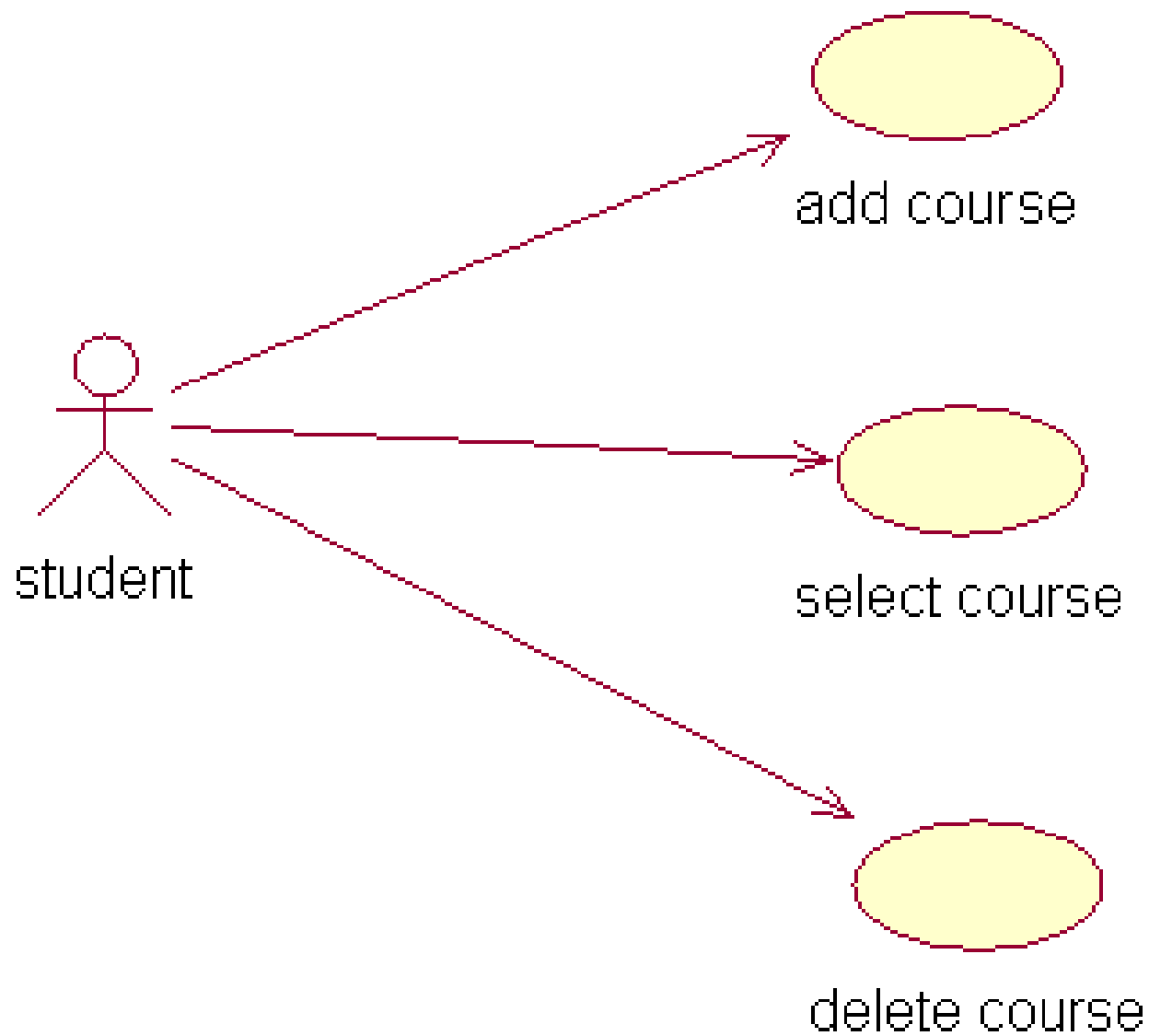


范例

№ 24

- ✦ 以选课系统中的选课用例 (**select course**), 来分析和设计序列图

PEREC





选课系统的事件流：

- (1) 学生进入选课主界面
 - (2) 学生点击选课
 - (3) 系统显示所有课程信息
 - (4) 学生选择课程
 - (5) 系统验证课程是否可选
- A1:** 课程不可选



(6) 系统提示课程选择成功，提示学生交费

(7) 用例结束

A1: 课程不可选

(1) 系统提示课程不可选及原因

(2) 学生重新选课

(3) 系统验证直到成功

(4) 转课程事件流到第6步



创建步骤:

(1) 寻找对象，从事件流中发现涉及下列对象:

- ✦ 界面
- ✦ 课程
- ✦ 对于业务层的操作，也应该有对象进行处理
- ✦ 事件流中设计的角色有：学生，数据库

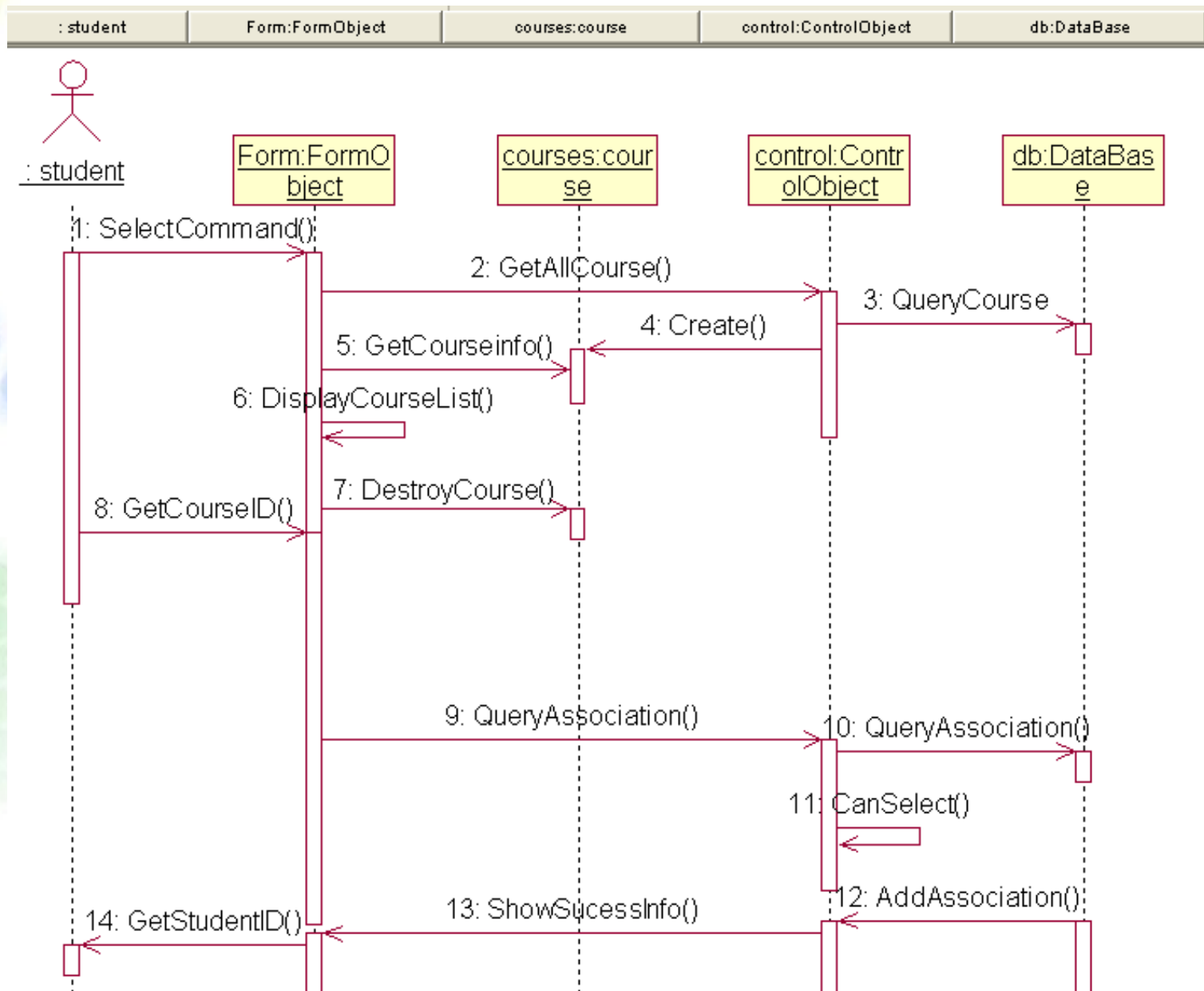


(2) 分析对象、角色之间交互的消息

- ⊕ 学生通过界面发送选课命令
- ⊕ 界面向控制对象请求课程信息
- ⊕ 控制对象向数据库发送查询数据消息
- ⊕ 控制对象暂存数据库的查询结果
- ⊕ 界面对象从控制对象中取得所有课程信息
- ⊕ 在界面上显示所有的课程信息
- ⊕ 界面对象放松命令要求控制对象删除课程信息
- ⊕ 学生选择课程



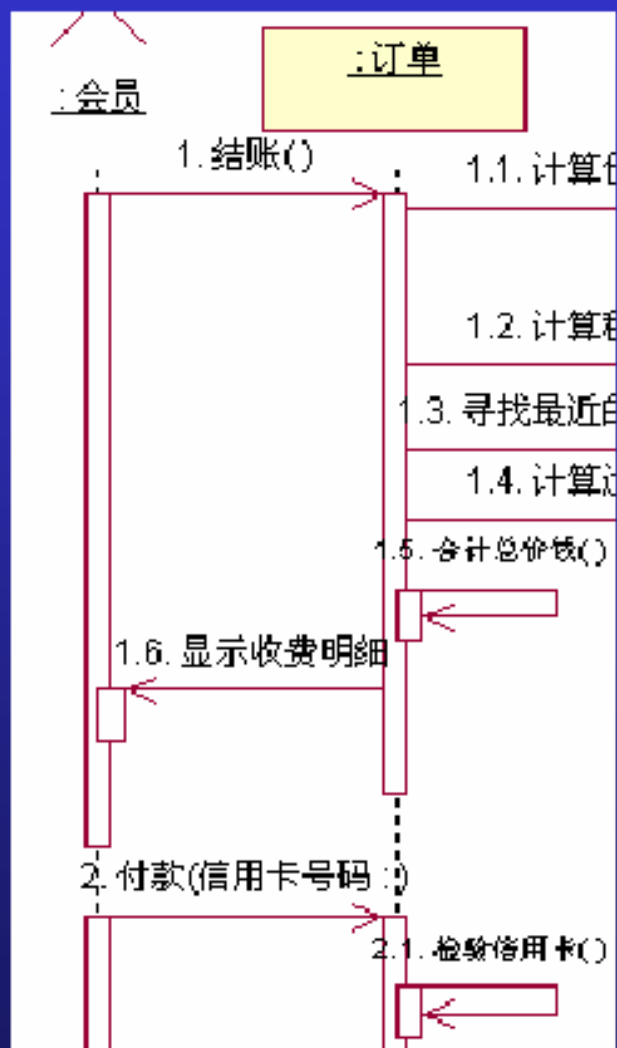
- ⊕ 界面对象要求学生输入学号
- ⊕ 界面对象向控制对象发送信息，查询该生是否可以选择该课程
- ⊕ 控制对象从数据库中查询关联信息
- ⊕ 控制对象判断是否可以选择
- ⊕ 如果可以选课，则向数据库中添加关联信息
- ⊕ 向界面对象返回信息





顺序图和类图的映射

消息的传入



说明订单有以下操作

订单. 结账()

订单. 合计总价钱()

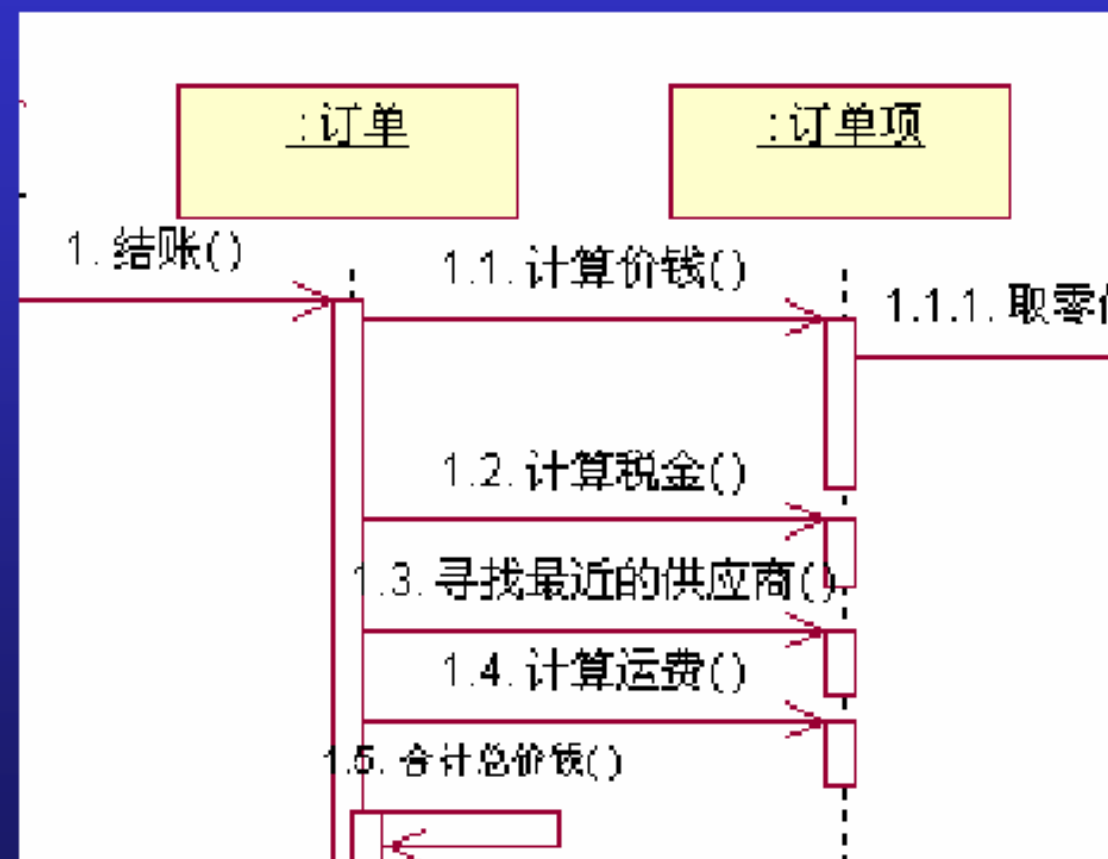
订单. 付款()

订单. 检验信用卡()



顺序图和类图的映射

消息的传出



订单. 结账()

{

...

订单项. 计算价钱()

订单项. 计算税金()

订单项. 寻找最近的供应商()

订单项. 计算运费()

...

(订单.) 合计总价钱()

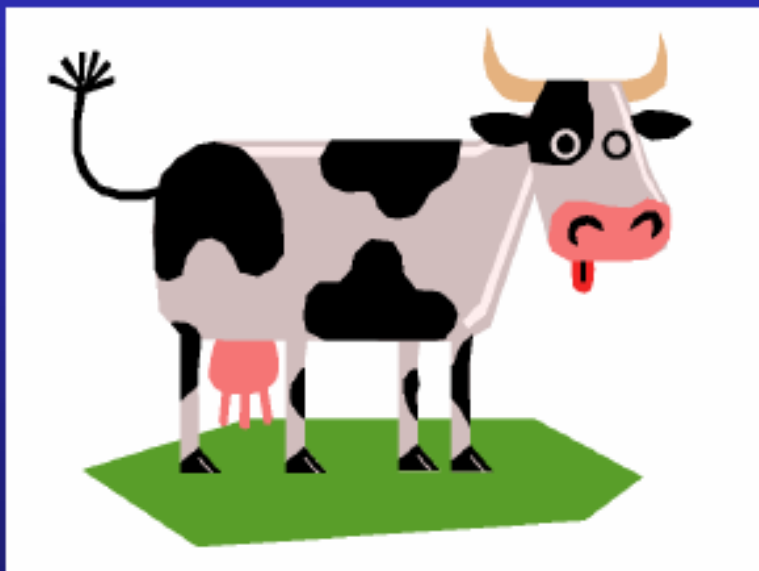
}



奶牛

No 34

❖ 每天凌晨4:30, 农夫Jones都要起床, 到牛栏去挤奶



怎样用OO方法表达这个挤奶过程?



面向过程的奶牛挤奶

- 定义函数Milk()
--float Milk(struct cow, float amount);
- 挤奶（使奶离开奶牛）的过程：

```
struct cow
{
    char    name[30];
    float   currentMilkVolume;
    const   float maxMilkVolume = 3.0;
};

struct Cow  Bessie = {"Bessie", 2.5};
// . . . other code here . . .
// now. . .milk the cow
fReturned = Milk (Bessie, 1.3);
```



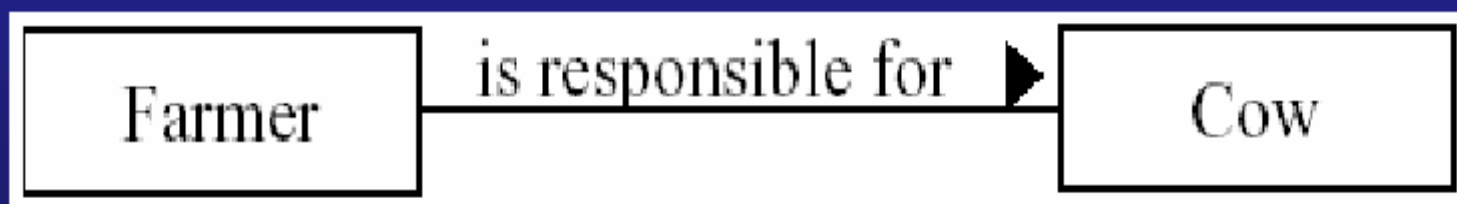
面向过程的问题

- ❖ “挤奶”操作是孤立的
 - ❖ 没有对挤奶农夫的需求
- ❖ Milk() 必须告诉哪头牛来挤奶
- ❖ Milk() 直接访问结构的内部
 - ❖ Milk() 和结构形成耦合



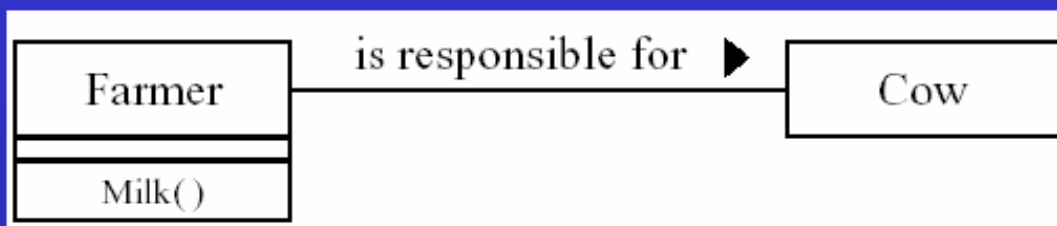
面向对象的挤奶

- ❖ 涉及的实体
 - ❖ 农夫，奶牛
- ❖ 涉及的关系
 - ❖ 农夫是奶牛的主人

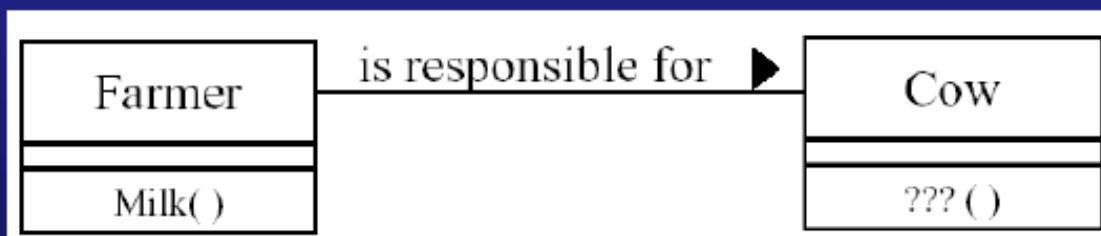




农夫挤奶牛的奶？



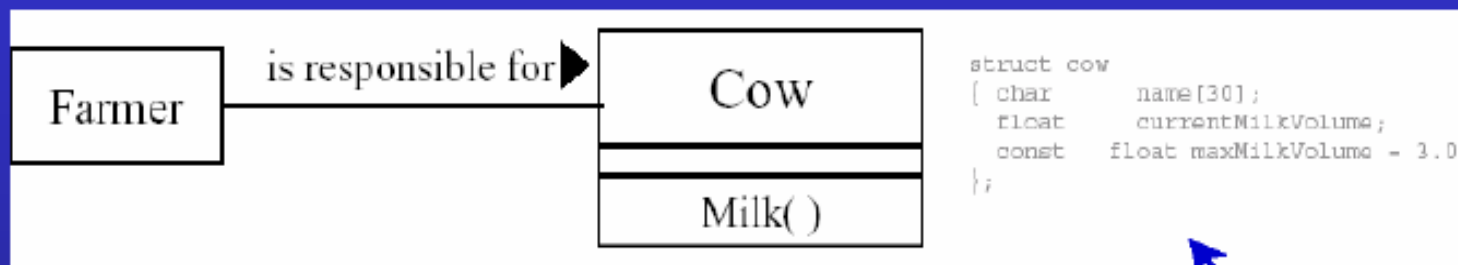
- ❖ 但这样行得通吗？
 - ❖ 牛奶在奶牛那里，不在农夫那里！
- ❖ Farmer::Milk不得不访问Cow的内部行为
- ❖ Farmer::Milk要调用Cow的哪个操作？





另一种责任分配方法

- 另一种方法一把Milk()放在Cow中



- 为了挤奶，Farmer调用Cow::Milk()操作
- 所有的“结构”信息（数据）隐藏在Cow中
- 在OO模型中，Farmer不是直接从cow中取奶，而是请求Cow自己挤奶



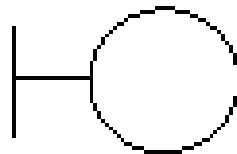
考虑...

- ❖ 谁知道奶牛有多少奶?
 - ❖ 奶牛
- ❖ 谁知道在请求挤奶时是否有奶可挤?
 - ❖ 奶牛
- ❖ 如果农夫请求挤3加仑但奶牛只有2加仑可以提供。谁来决定应该提供2, 1还是0加仑?
 - ❖ 奶牛

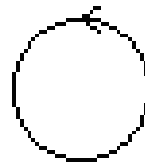


BCE模式

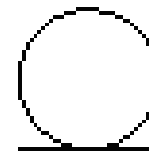
41



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

42

- ⊕ 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

43

- ⊕ 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

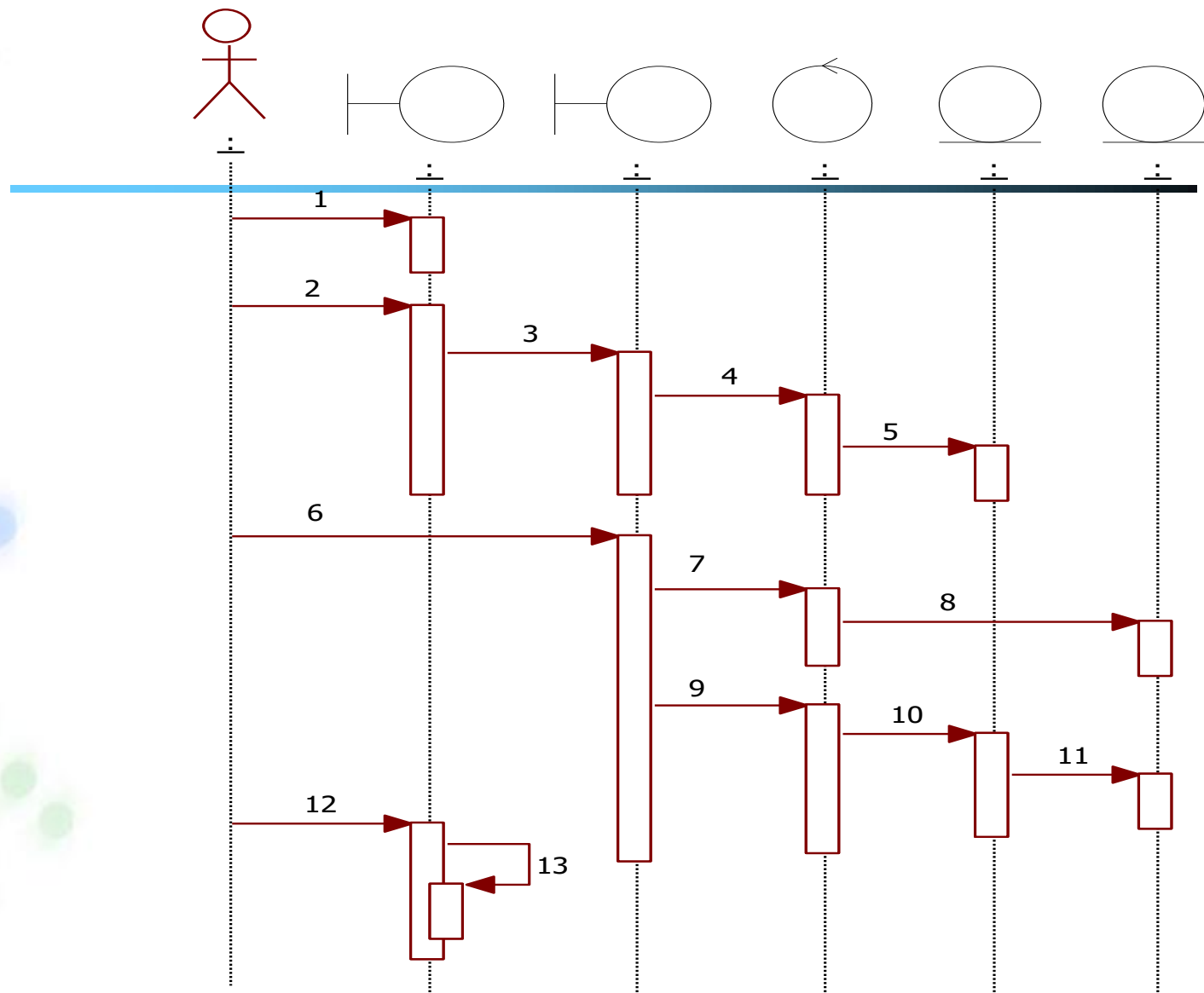


图3-6:套用BCE模式



酒店联合订房系统

45

- ⊕ 1 用例—会员登录
- ⊕ 2 用例—订房
- ⊕ 3 用例—查询酒店数据
- ⊕ 4 用例—查询房型数据
- ⊕ 5 用例—通知已预订

PEREC



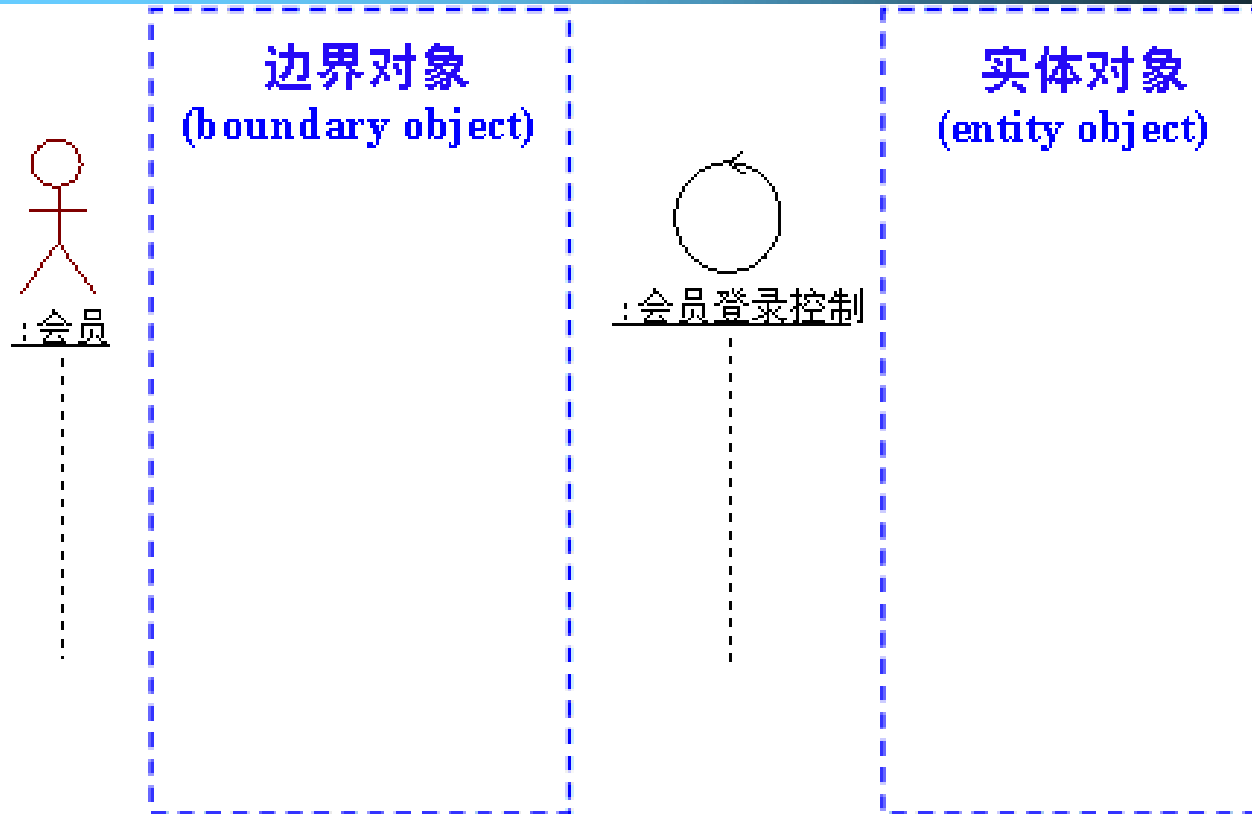
1 用例—会员登录

46

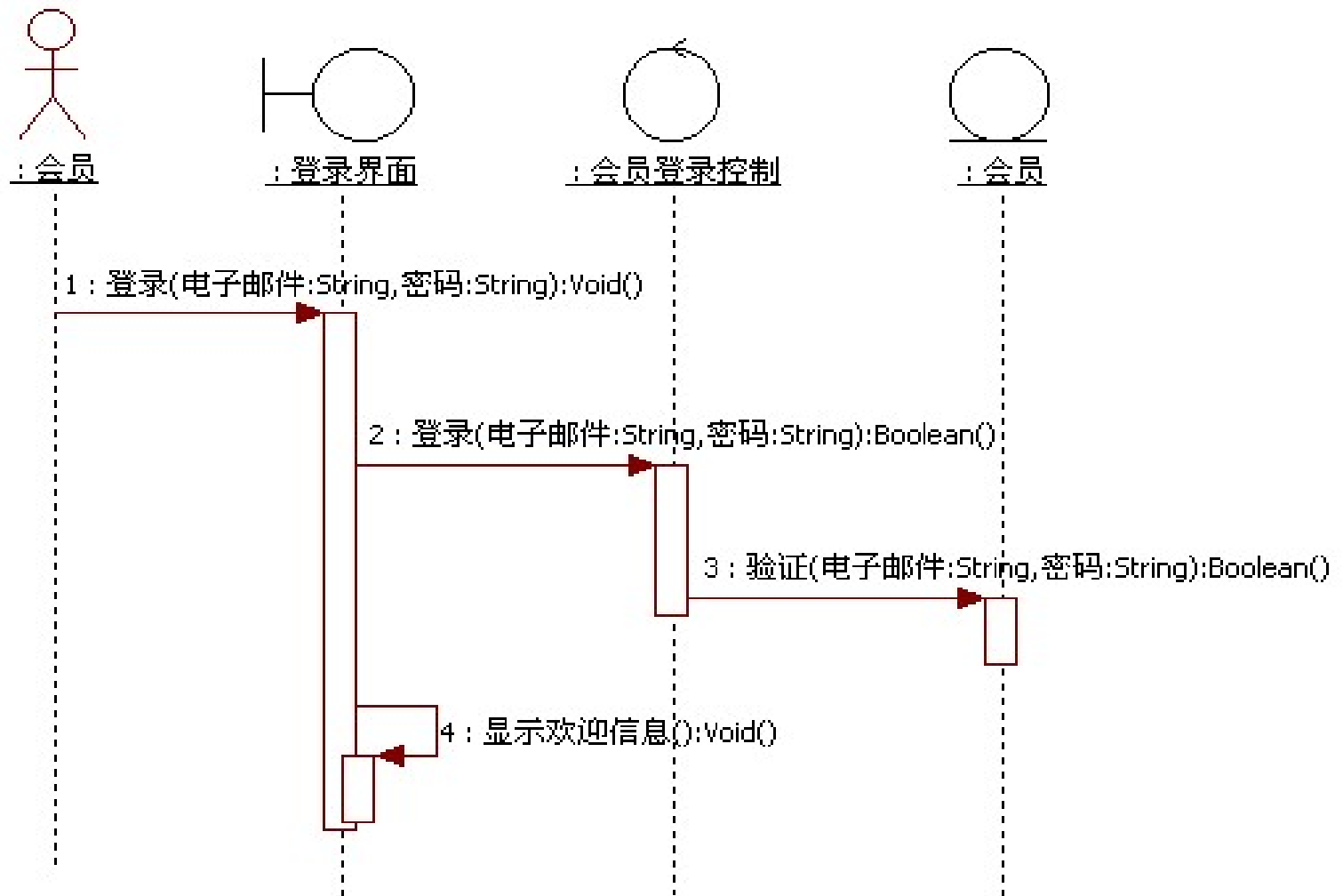
用例	会员登录		
启动者	会员	支持者	
主要流程 1.会员输入电子邮件和密码。 2.系统确认会员身分之后，出现欢迎消息。			

“会员登录”的主要流程

PEREC



“会员登录”的参与者对象和控制对象

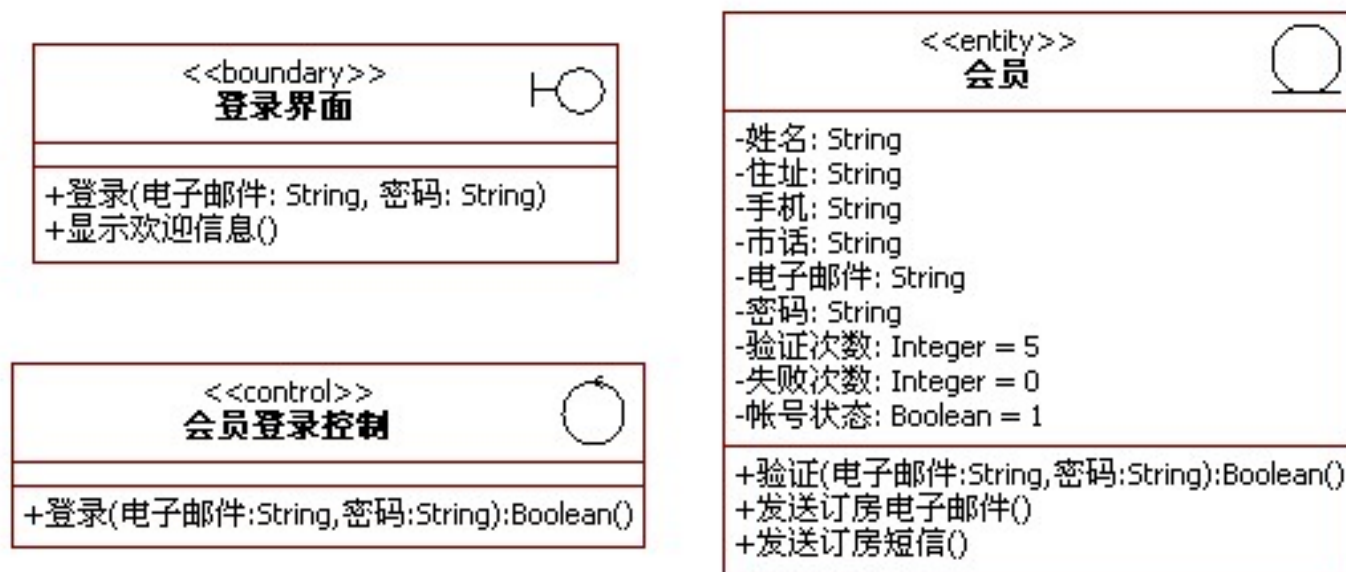


“会员登录”的序列图



1 用例—会员登录

49



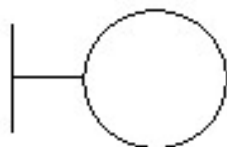
“会员登录”用例的BCE类

PEREC



1 用例—会员登录

50



登录界面

+登录(电子邮件: String, 密码: String)
+显示欢迎信息()



会员登录控制

+登录(电子邮件:String,密码:String):Boolean()



会员

-姓名: String
-住址: String
-手机: String
-市话: String
-电子邮件: String
-密码: String
-验证次数: Integer = 5
-失败次数: Integer = 0
-帐号状态: Boolean = 1

+验证(电子邮件:String,密码:String):Boolean()
+发送订房电子邮件()
+发送订房短信()

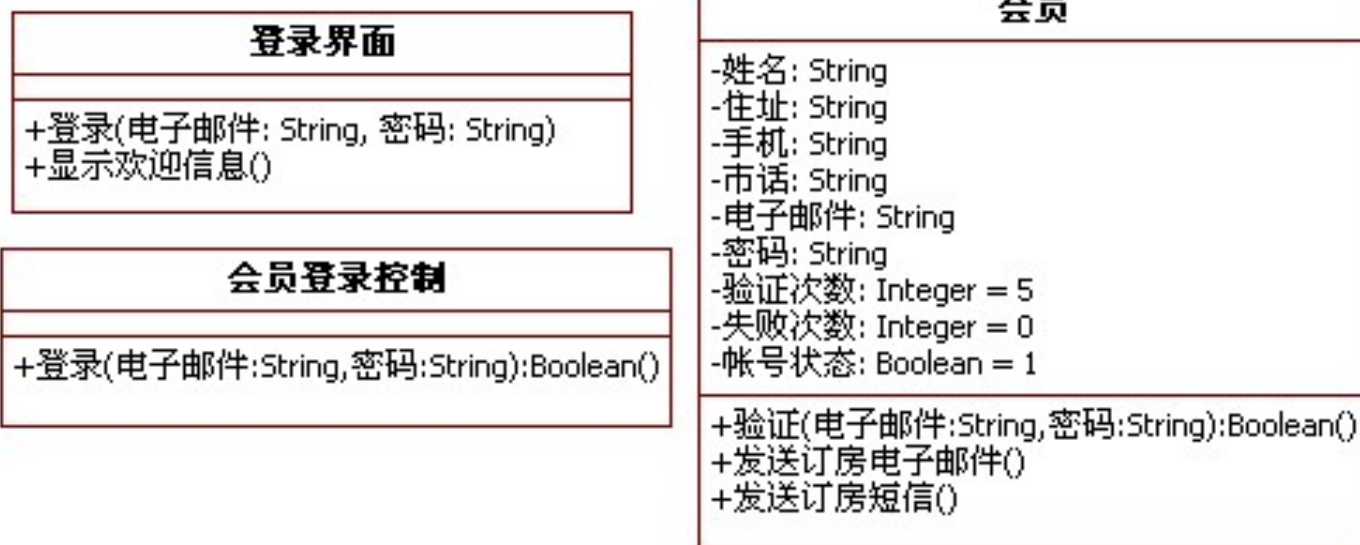
使用BCE类图标

PEREC



1 用例—会员登录

51



使用一般的类图标

PEREC



2 用例—订房

52

用例	订房		
启动者	会员	支持者	酒店经营者
主要流程 1.会员挑选一家酒店。 2.系统列出这家酒店所有的房型名称、床型、空房数和房价。 3.会员挑选预订的房型、房间数以及预订日期。 4.系统减少可预订的空房数，并且新增一条订房交易。 5.系统列出交易代号、订金与总价。 6.系统提醒会员需要48小时内支付预订金。 7.系统发送订房通知给酒店经营者和会员。			

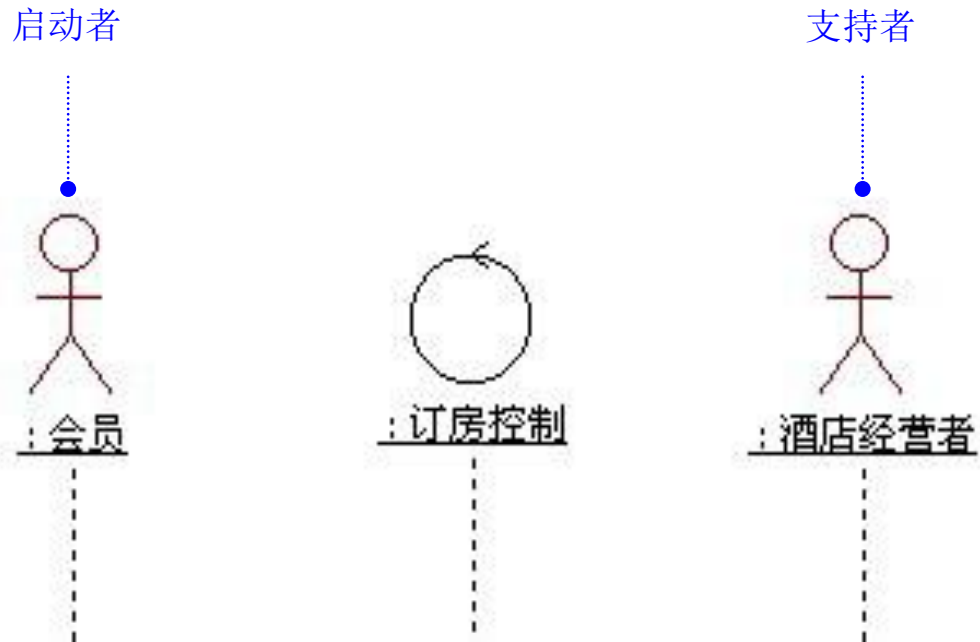
“订房”的主要流程

PEREC



2 用例—订房

53



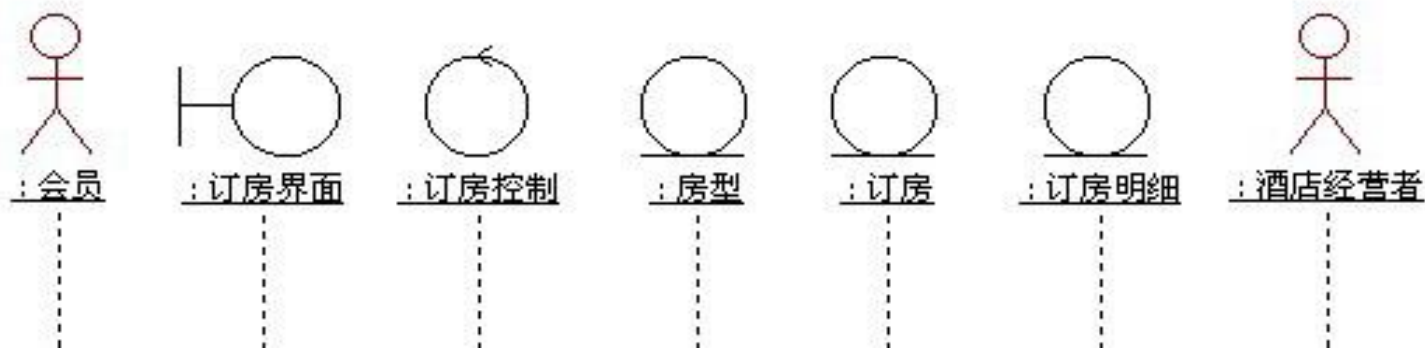
“订房”用例有两个参与者

PEREC



2 用例—订房

54



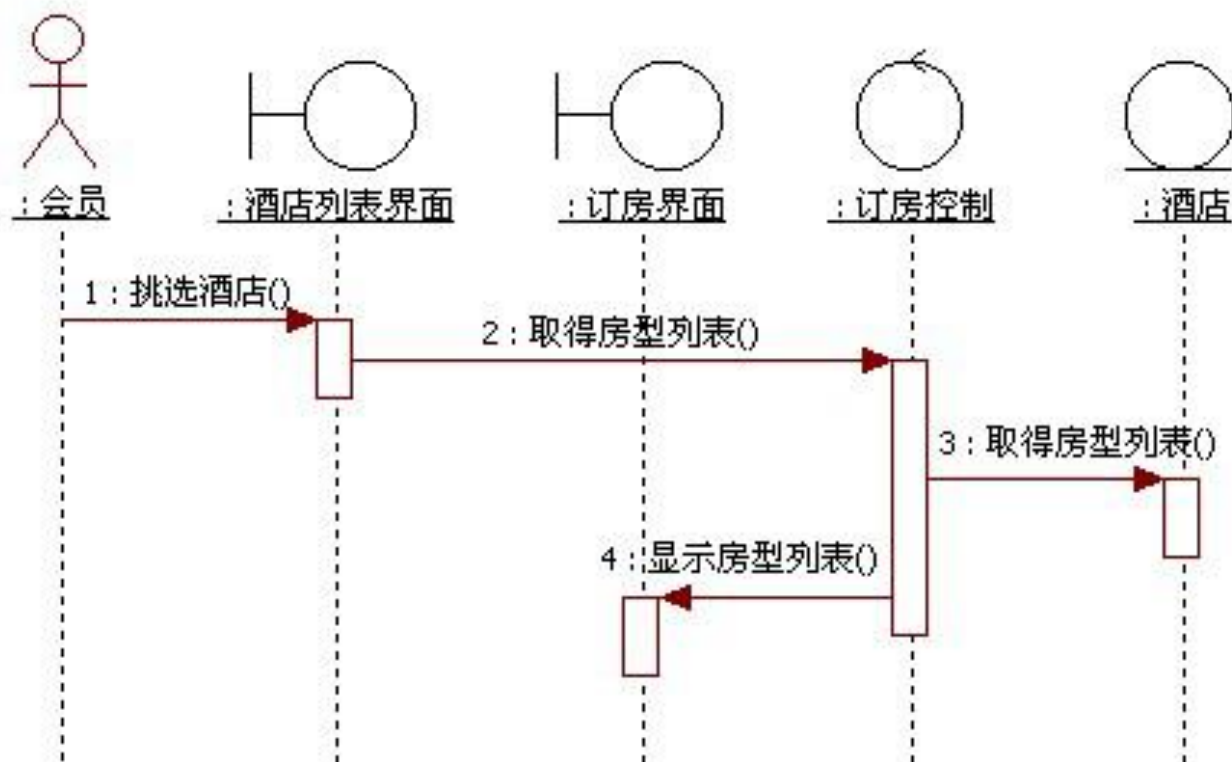
加上边界对象和实体对象

PEREC



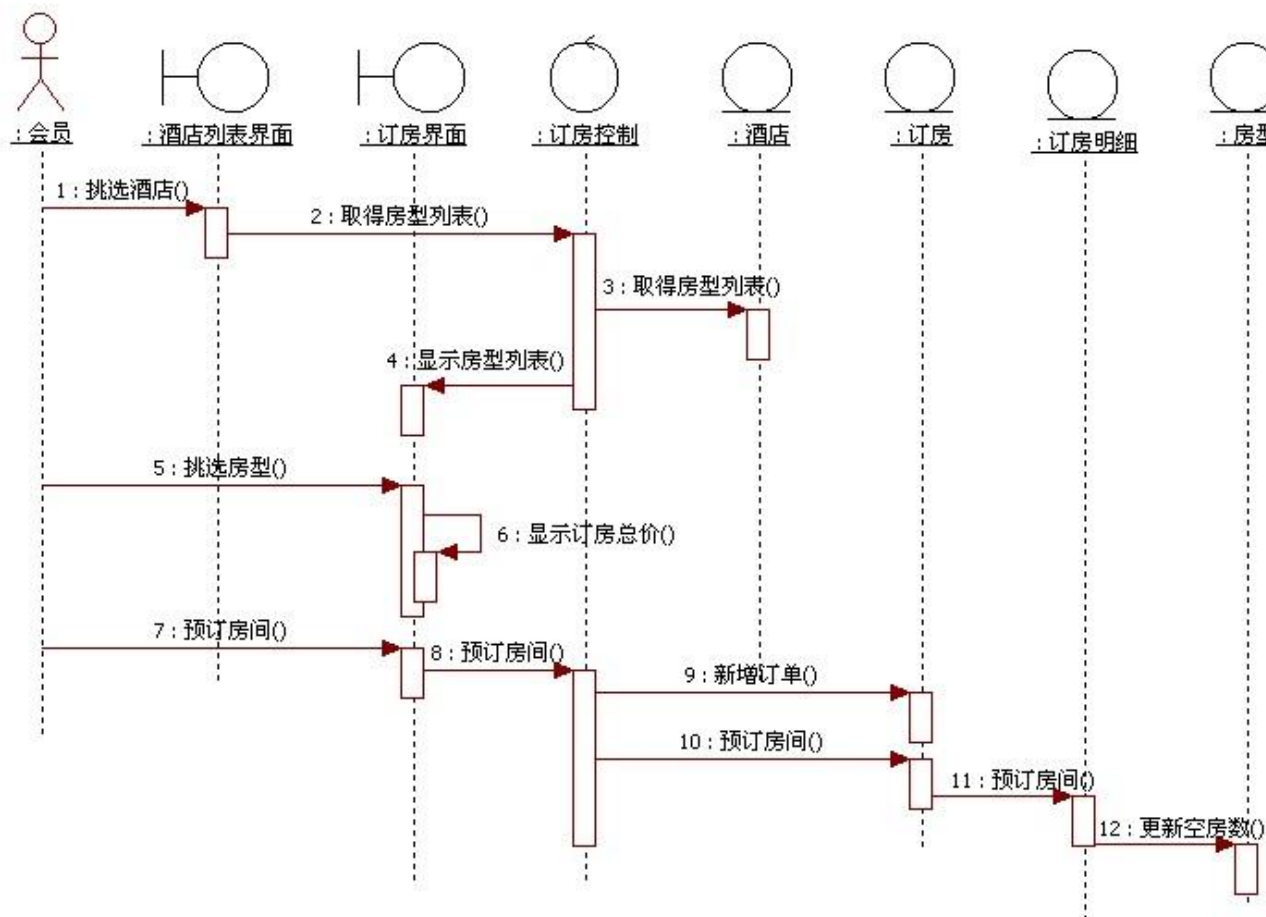
2 用例—订房

55

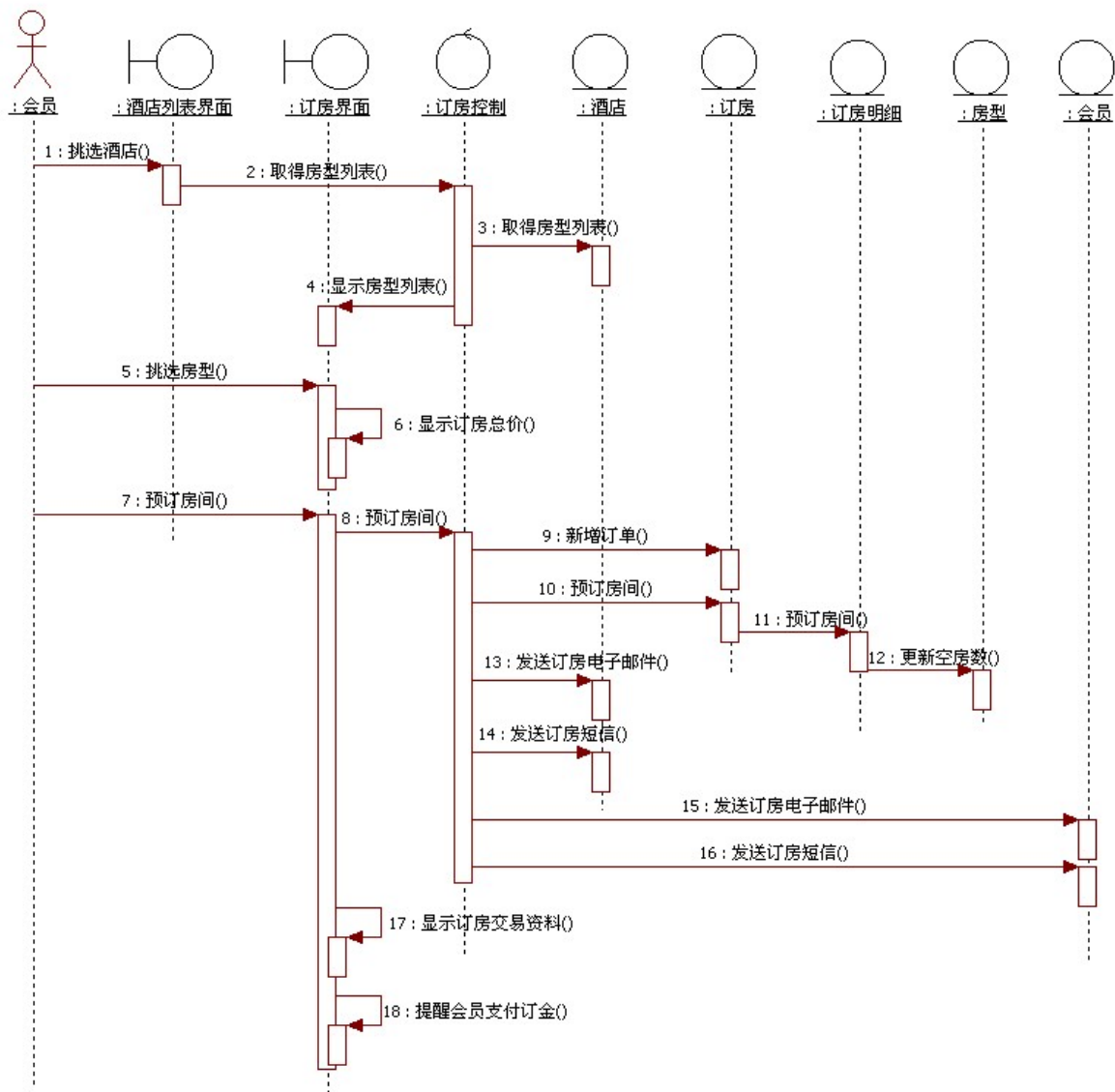


查询房型

PEREC



挑选并预订房间

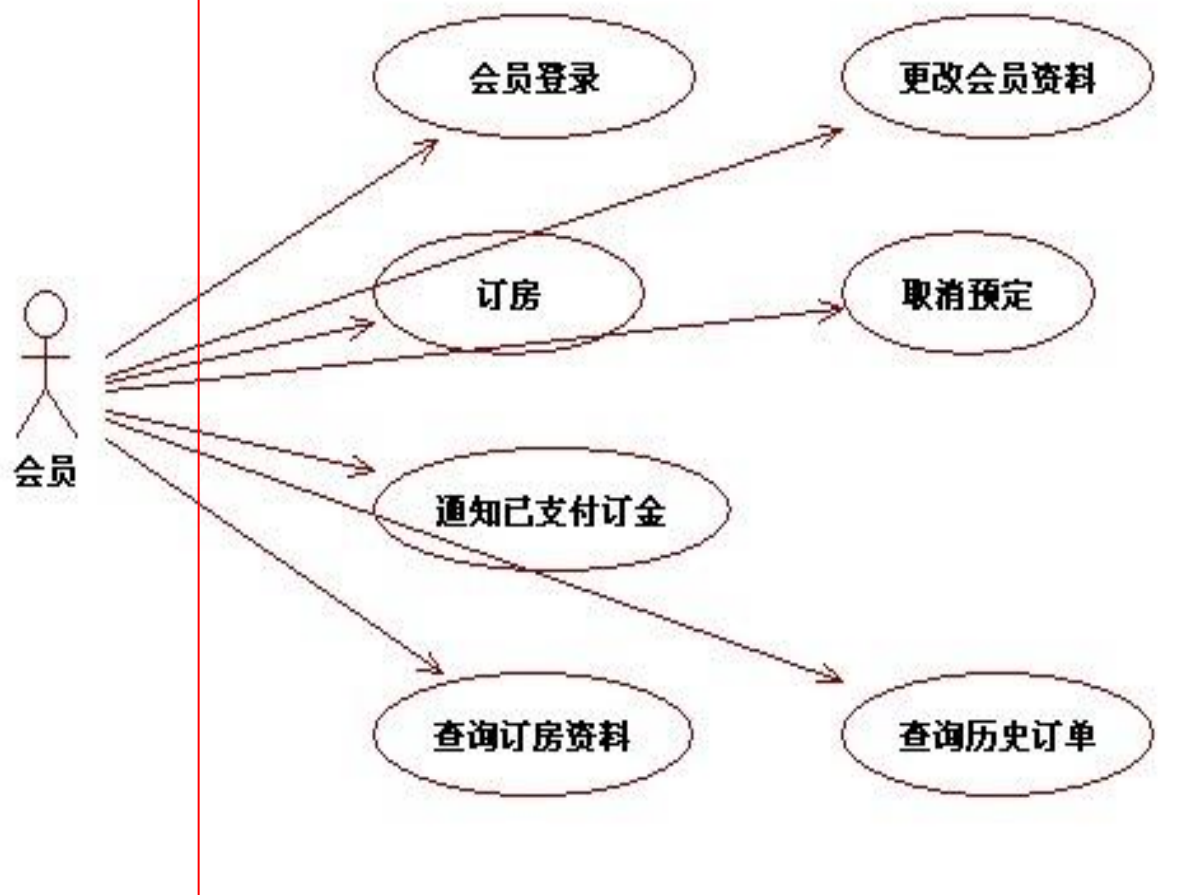


掉酒店经营者参与者



酒店联合订房系统

58



删去了酒店经营者

PEREC



用例	订房		
启动者	会员	支持者	
<p>主要流程</p> <ol style="list-style-type: none">1.会员挑选一家酒店。2.系统列出这家酒店所有的房型名称、床型、空房数和房价。3.会员挑选预订的房型、房间数以及预订日期。4.系统显示出订房总价。5.系统新增一条订房交易。6.系统减少可预订的空房数。7.系统发送订房通知给酒店经营者。8.系统发送订房通知给会员。9.系统列出交易代号、订金与总价。10.系统提醒会员需要48小时内支付预订金。			

“订房”的主要流程

2 用例—订房

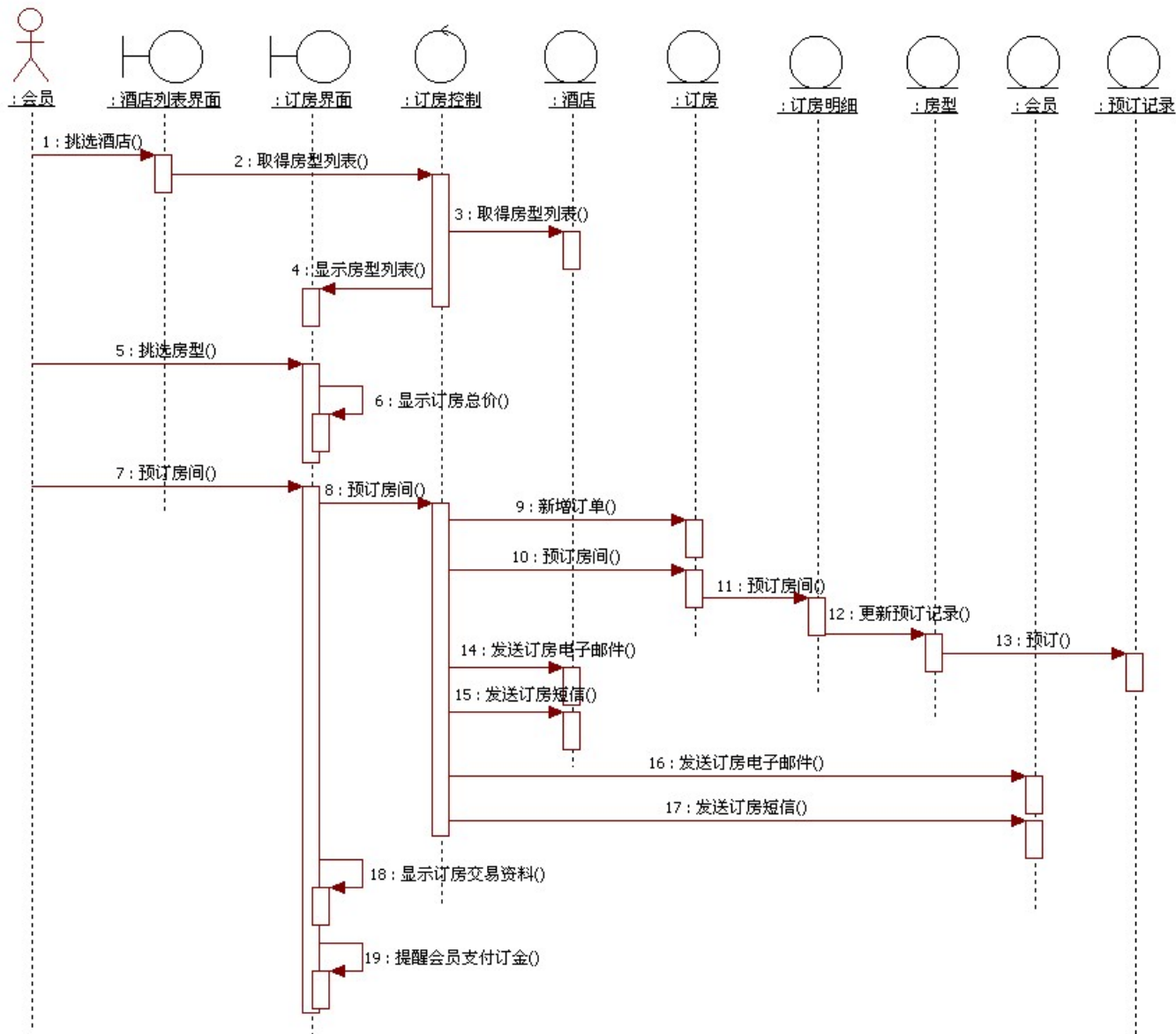
60

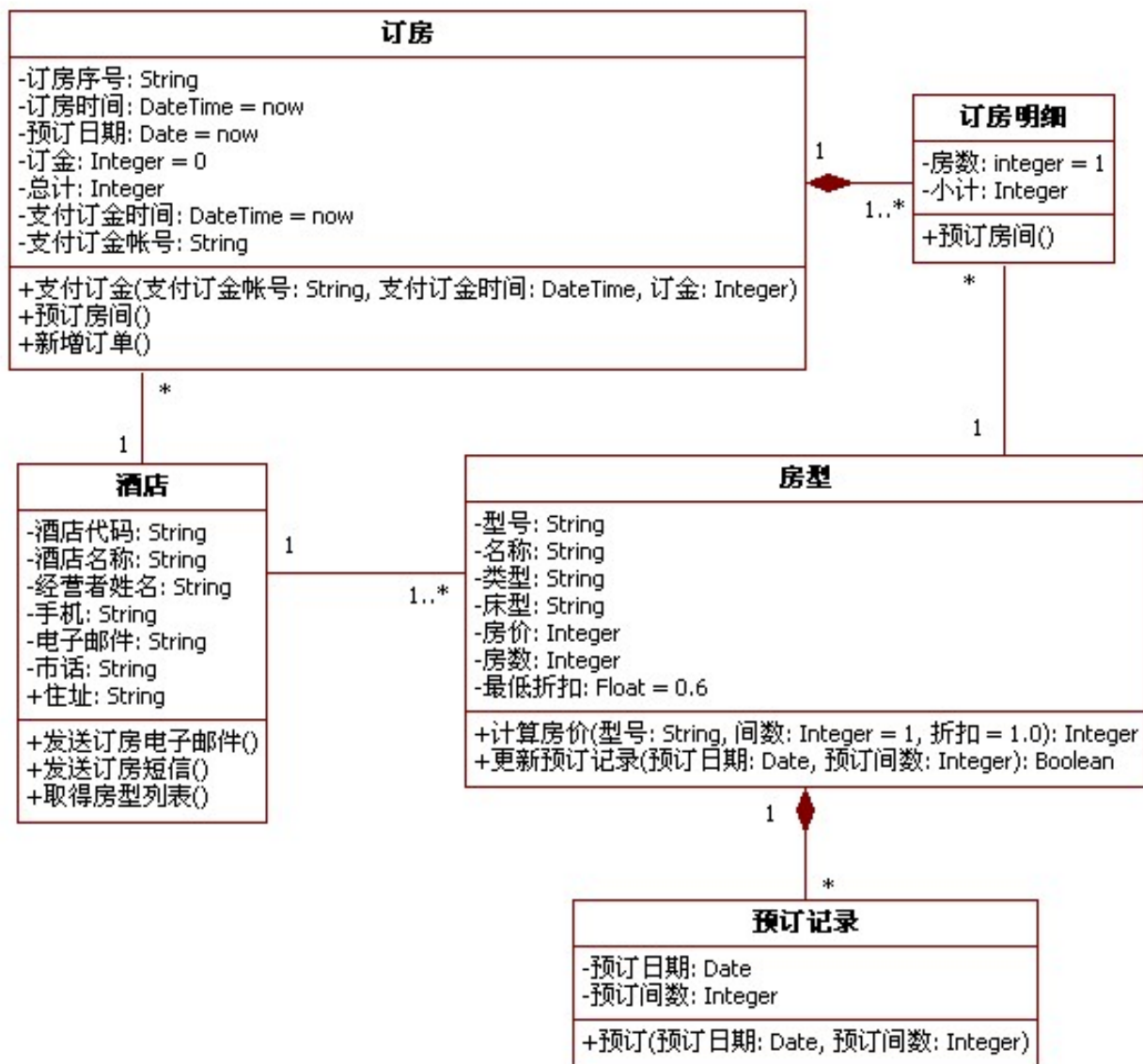


房型类

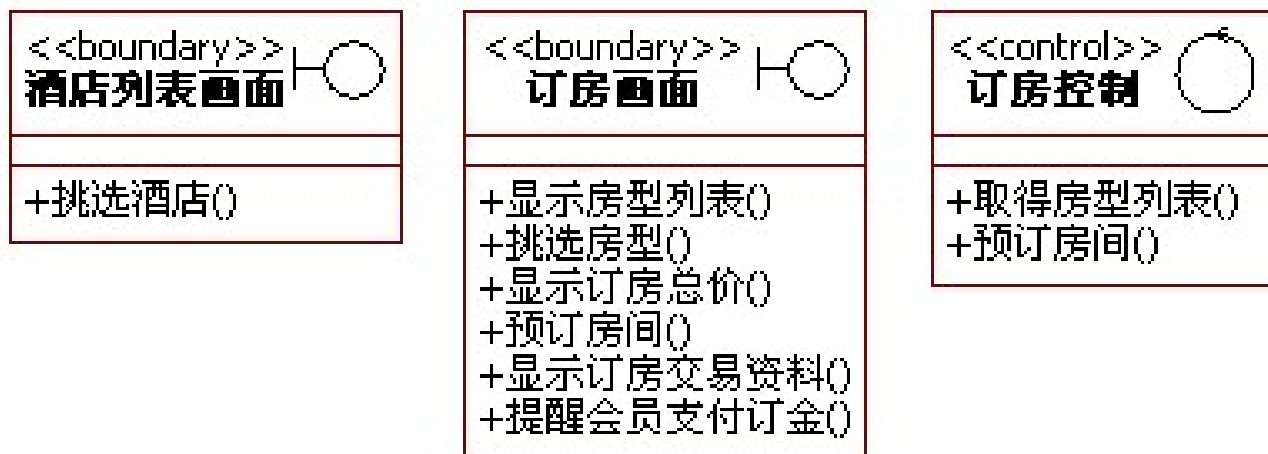


新增“预订记录”类





“订房”用例的实体类



“订房”用例的边界类和控制类