交互图

主讲：华俊昌

目录

CONTENT

01 交互图概述

02 序列图

03 协作图

04 创建序列图实例

05 课堂练习

01

Part one

交互图概述

交互图的定义

- 交互图是用来描述对象之间以及对象与参与者之间的动态协作关系以及协作过程中行为次序的图形文档。
- 交互图作用
  - 交互图是用来描述对象之间以及对象与参与者之间的动态协作关系以及协作过程中行为次序的图形文档
  - 帮助分析人员对照检查每个用例中所描述的用户需求是否已经在类中实现，需要补充哪些类，或者补充类的接口。
- 交互图分类
  - 序列图（Sequence Diagram）
  - 协作图（Collaboration Diagram）

序列图和协作图的区别

- 序列图强调消息发送的时间顺序
- 协作图则强调接收和发送消息的对象的组织结构
- 序列图和协作图语义等价
  - 序列图和协作图在语义上是等价的，共享相同的基本模型
  - 两个图都可以表示另一个图所不能表示的某些东西
  - 序列图和协作图可以实现两者之间的等价转换，而不丢失任何信息
  - 它们源自不同的方法

02

Part two

序列图

什么是序列图

序列图定义

序列图（sequence diagram）也称时序图或者顺序图。是显示对象之间交互的图，这些对象是按时间顺序排列的。它是强调消息时间顺序的交互图。

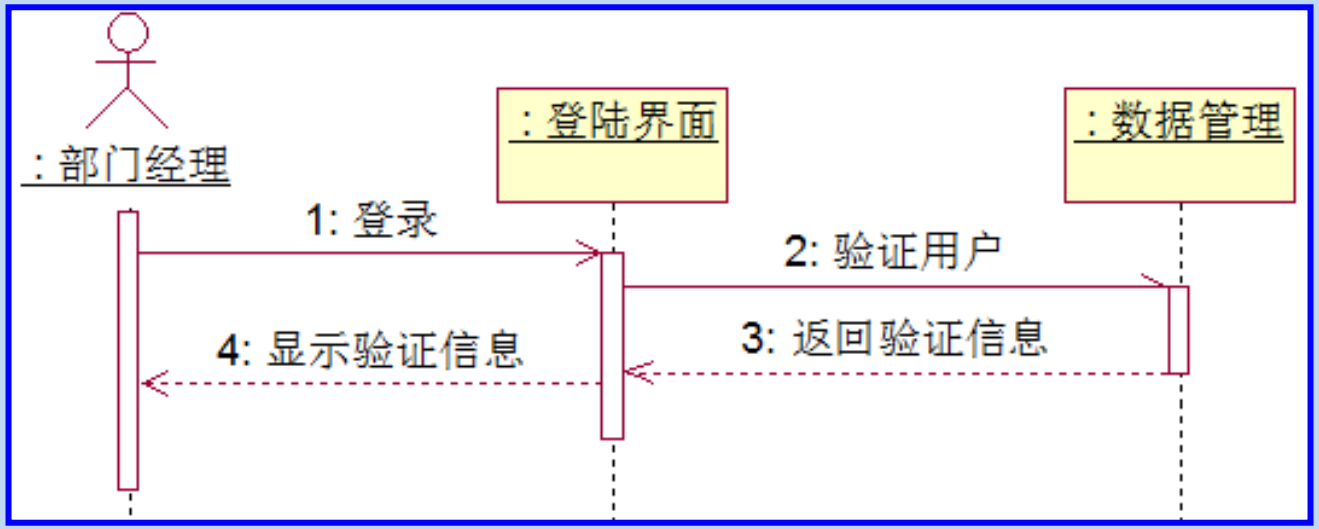
序列图描述了对象之间传送消息的时间顺序，用来表示用例中的行为顺序。

序列图将交互关系表示为一个二维图。其中，纵轴是时间轴，时间沿竖线向下延伸。横轴代表了在协作中各独立的对象。

序列图定义

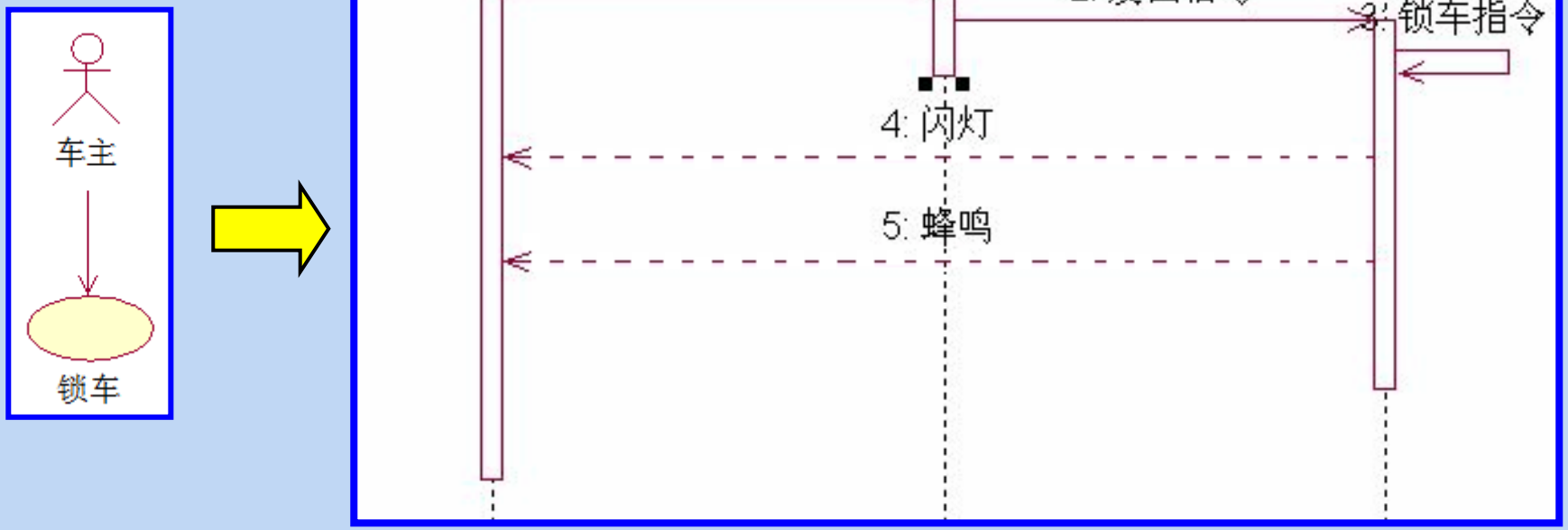
序列图作用：

- 通常用来描述用例的实现：表明了由哪些对象，通过消息相互协作来实现用例的功能
- 可以根据对象之间的交互关系来定义类的职责



序列图示列

自动车锁系统中的锁车用例



用例图

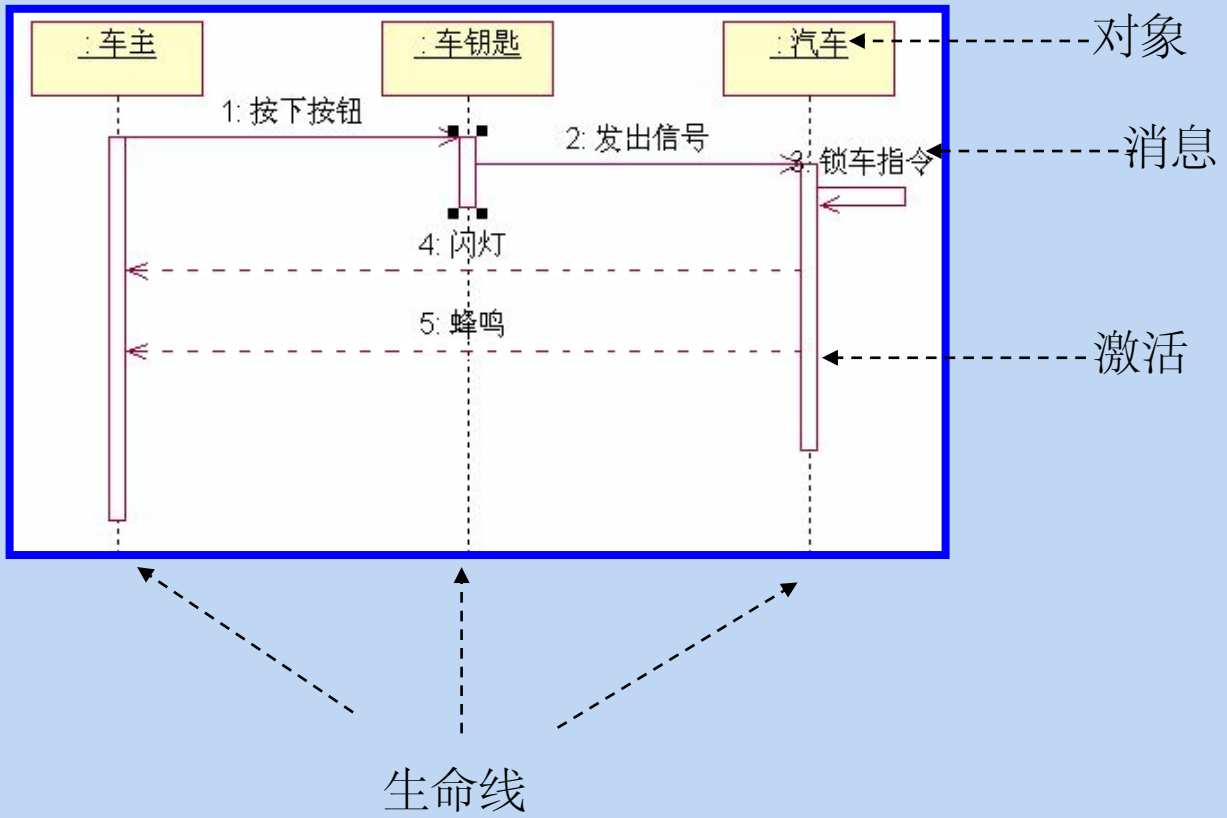
序列图

序列图的组成

序列图的组成

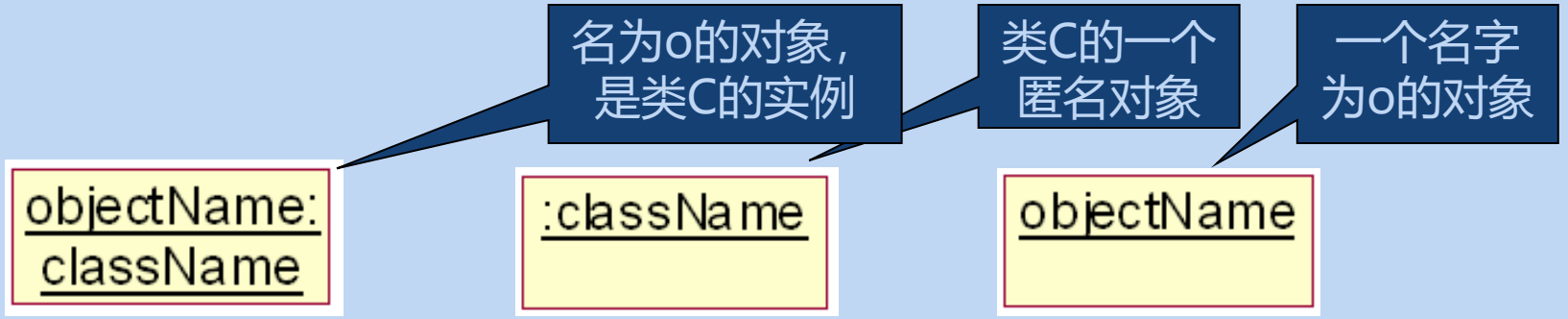
组成元素

- **对象（Object）**
- **生命线（Lifeline）**：表示对象的生存期
- **激活**(控制焦点)**（Activation）**：表示对象正在执行某个操作
- **消息（Message）**：表示对象之间的通讯

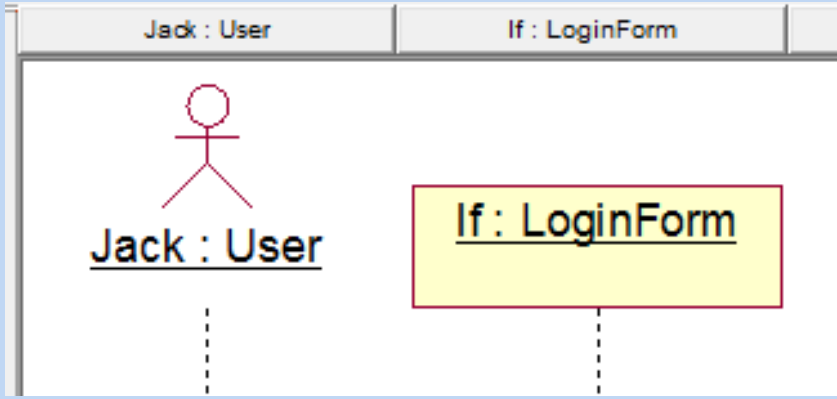


序列图的组成-对象

- 对象的命名方式：对象名:类名



- 序列图中的对象
  - 系统的参与者
  - 任何有效的系统对象
  - 用矩形框代表一个对象
- 对象排列次序没有要求。主要参与者，次要参与者
- 对象置于顺序图的顶部，对象的位置不在顶部。
- 支持同一个类的多个对象在一起工作



序列图的组成-对象

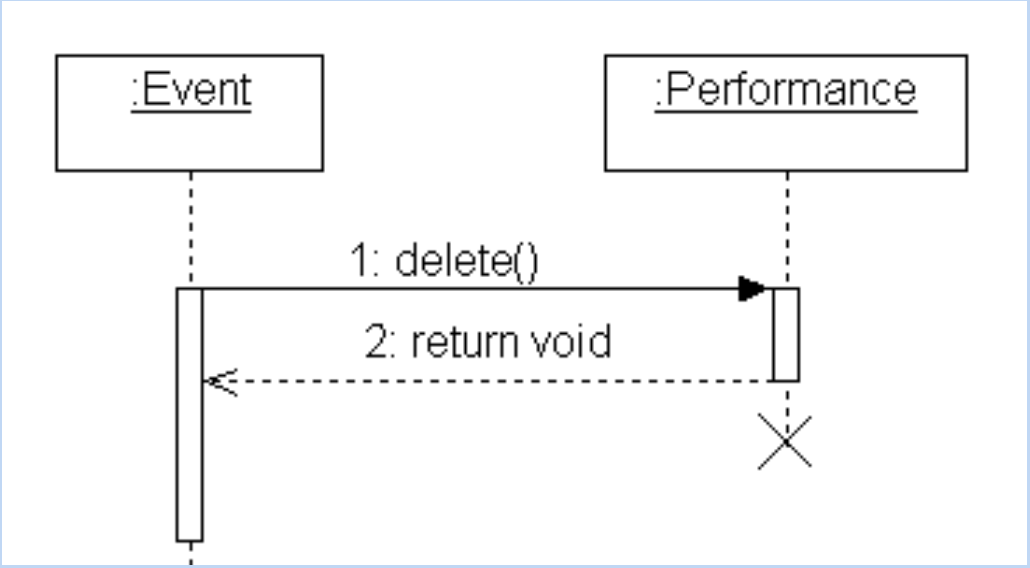
- 对象的类型
  - Persistent（持续）：对象能够保存在数据库或其它持续存储器中
  - Static（静态）：保存在内存中，直至程序终止才会销毁
  - Transient（临时）：临时保存在内存中

序列图的组成-对象

- 对象的位置
  - 对象置于顺序图的顶部：意味着在交互开始的时候对象就已经存在
  - 对象的位置不在顶部：表示对象是在交互的过程中被创建的
  - 主角：一个交互的发起对象，通常在最左边
  - 反应系统角色：与主角交互的对象，通常在图的右边

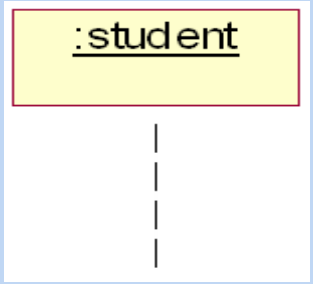
序列图的组成-对象

- 对象的撤销
  - 撤销一个对象：在其生命线终止点放置一个“X”符号，该点通常是对删除或取消消息的回应。



序列图的组成-生命线

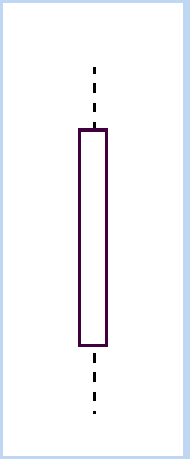
- 生命线是一条垂直的虚线，表示顺序图中的对象在一段时间内的存在。每个对象的底部中心的位置都带有生命线。

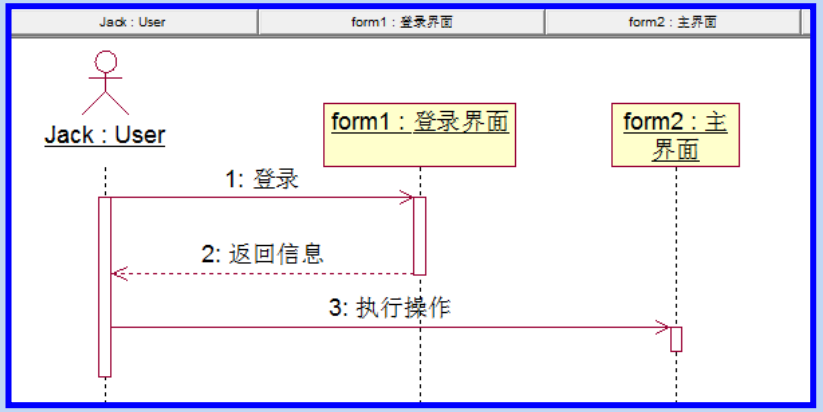


- 生命线是一个时间线，从顺序图的顶部一直延伸到底部，所用的时间取决于交互持续的时间。
- 对象与生命线结合在一起称为对象的生命线，对象的生命线包含矩形的对象图标以及图标下面的生命线。

序列图的组成-激活

- 在对象的生命线上，包含一个矩形，表示对象处于激活状态
  - 处于激活状态的对象正在执行某个任务
  - 对象在完成自己的工作后，去除激活，处于空闲状态
  - 顶端与激活时间对齐，底端与完成时间对齐
- 激活表示该对象被占用以完成某个任务，去激活指的则是对象处于空闲状态、在等待消息。
- 为了表示对象是激活的，可以将该对象的生命线拓宽成为矩形。其中的矩形称为激活条或控制期，对象就是在激活条的顶部被激活的，对象在完成自己的工作后被去激活。
- 有些对象会一直控制顺序图所描述的过程，不仅仅回应消息，还监视整个交互过程，则该对象的激活条是整个对象生命线的长度





序列图的组成-消息

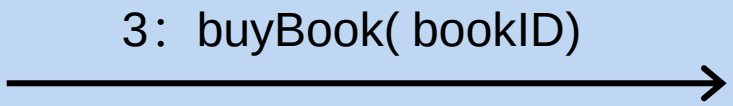
- 消息的定义：
  - 一个对象(发送者)向另一个或几个其他对象(接收者)发送信息号
  - 一个对象(调用者)调用另一个对象(被调用者)的操作
- 消息的组成
  - 包括消息号、消息名、消息参数。
  - 例如：1.login（uName, uPwd）

序列图的组成-消息

- 消息的种类
  - 简单消息
  - 调用消息（同步消息）
  - 异步消息
  - 返回消息
  - 反身消息
  - 阻止消息
  - 超时消息

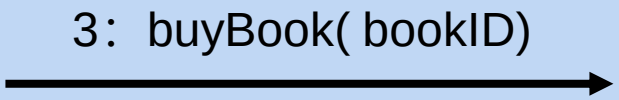
消息-简单消息

- 表示普通的控制流。只表示控制如何从一个对象传递给另一个对象，而没有描述通信的任何细节。
- 主要用于通信细节未知或者无需考虑通信细节的场合。即主要用于不知道消息是同步还是异步的场合，但通常表示异步消息。
- 图形表示



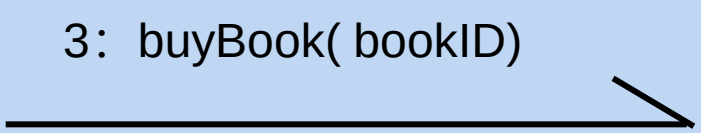
消息-调用消息（同步消息）

- 调用消息的发送者把控制传递给消息的接收者，然后停止活动，等待消息接收者放弃或返回控制。
- 通常表示同步消息。箭头是实心。
- 应该有一个配对的返回消息，但为了图的简洁，省略返回消息
- 图形表示



消息-异步消息

- 发送者将该消息发送给接收者后，无需等待接收者消息处理的完成而继续执行另一个活动。
- 异步消息就是发送发只负责将消息发送给接收方，至于接收方如何响应，发送方则不需要知道。对接收方来说，在接收到消息后它既可以对消息进行处理，也可以什么都不做。
- 图形表示

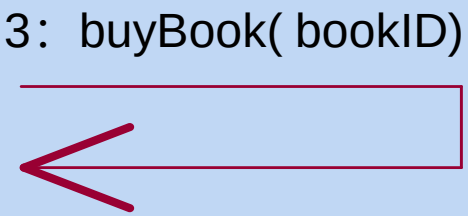


消息-返回消息

- 表示控制流显式地从接收消息的对象返回。
- 图形表示

消息-反身消息

- 消息发送方和消息接受方是同一个对象
- 如果一条消息只能作为反身消息，则应该作为类的一个私有方法，而不是一个消息
- 图形表示



消息-阻止消息

- 消息发送者发出消息给消息接收者，如果接收者无法立即接受消息，则发送者放弃这个消息
- 图形表示

消息-超时消息

- 消息发送者发出消息给消息接收者并按指定时间等待，如果接收者无法在指定时间内接受消息，则发送者放弃这个消息
- 图形表示

03

Part three

协作图

什么是协作图

协作图定义

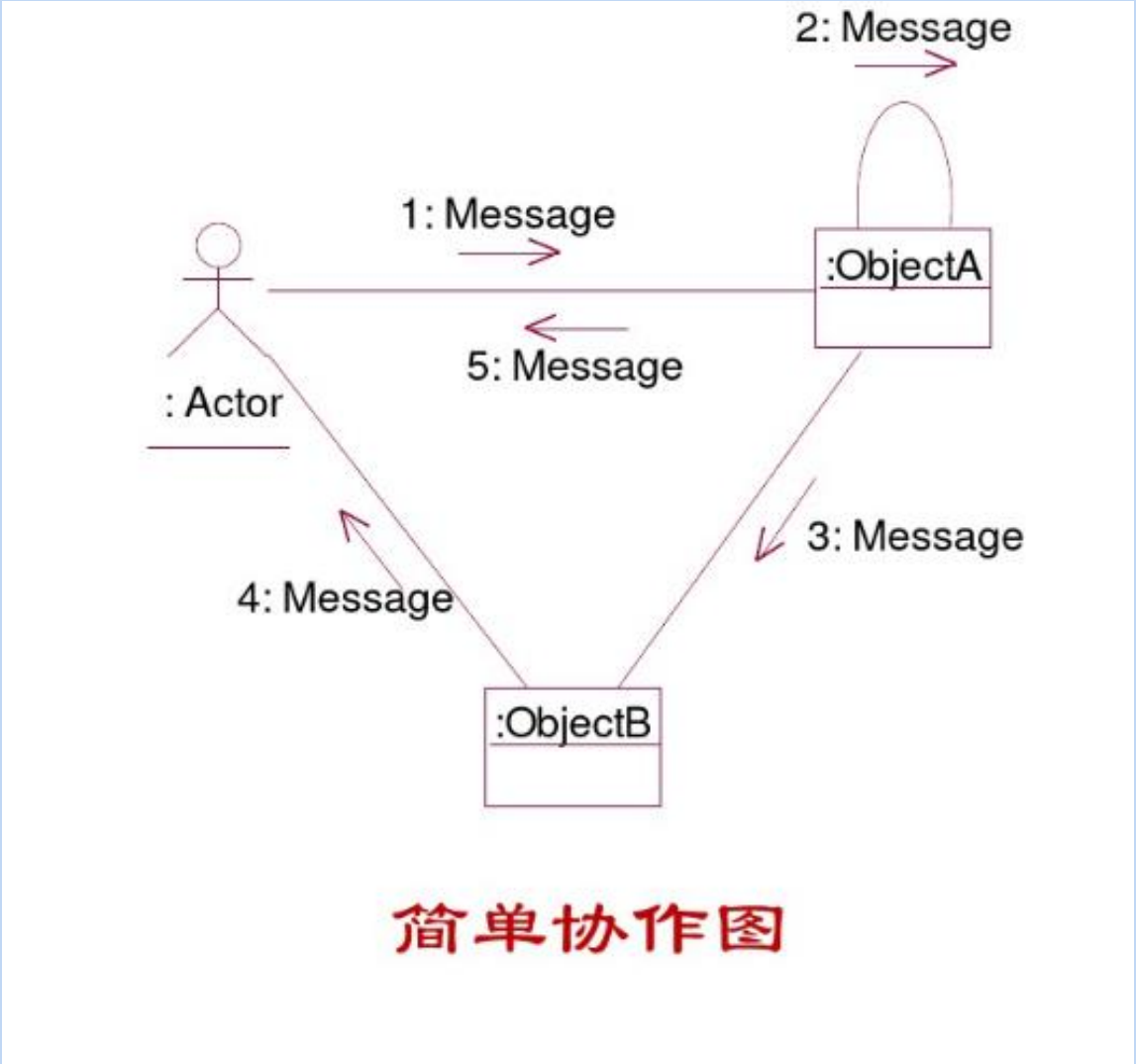
协作图（Collaboration Diagram）是UML 1.X版本的概念，在当时的认识中，认为它就是描述一个协作的，所以称为协作图。

UML2.0重新定义的协作的概念,并认为所谓协作图与协作概念实际上没有必然联系，为了避免误会，所以将它改名为通信图（Communication Diagram）,因此很多书籍中使用通信图代替协作图

协作图定义

序列图强调的是交互的时间顺序，而协作图强调的是交互情况和参与的交互对象的整体组织。

从另一个角度来看，序列图按照时间顺序布局；而协作图按照空间组织布局。



协作图组成

协作图组成

- 组成元素
  - 对象（Object）
  - 链接（Link）
  - 消息（Message）

协作图组成-对象

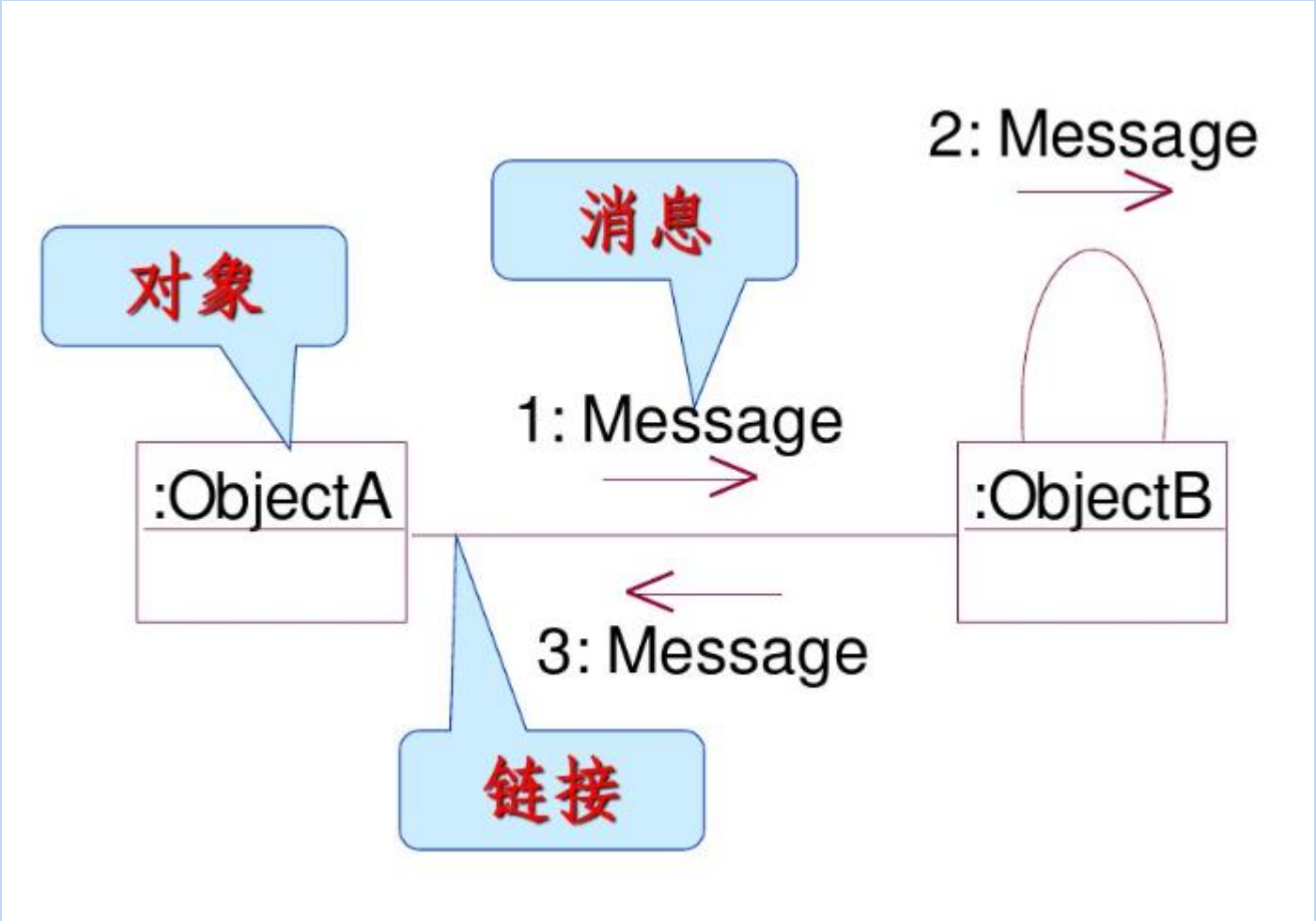
- 协作图的对象与序列图的对象类似，用一个矩形框代表一个对象，框内写对象名：所属类名
- 协作图中无法表示对象的创建与撤销，所以对象在协作图中的位置没有限制。

协作图组成-链接

- 链接是两个对象间的连接。
- 它的图形符号是一个连接再两个对象间的实线表示，在连接线上可以标注对象名。

协作图组成-消息

- 消息代表协作图中的对象间通过链接发送的消息。
- 消息由一个对象发出，由消息所指的对象接收。形成消息流。
- 消息流上标有消息的序号和对象间发送的消息，消息的序号表明了消息发送的先后顺序。
- 一条消息会触发接收对象中的一项操作。



04

Part four

创建序列图实例

序列图实例

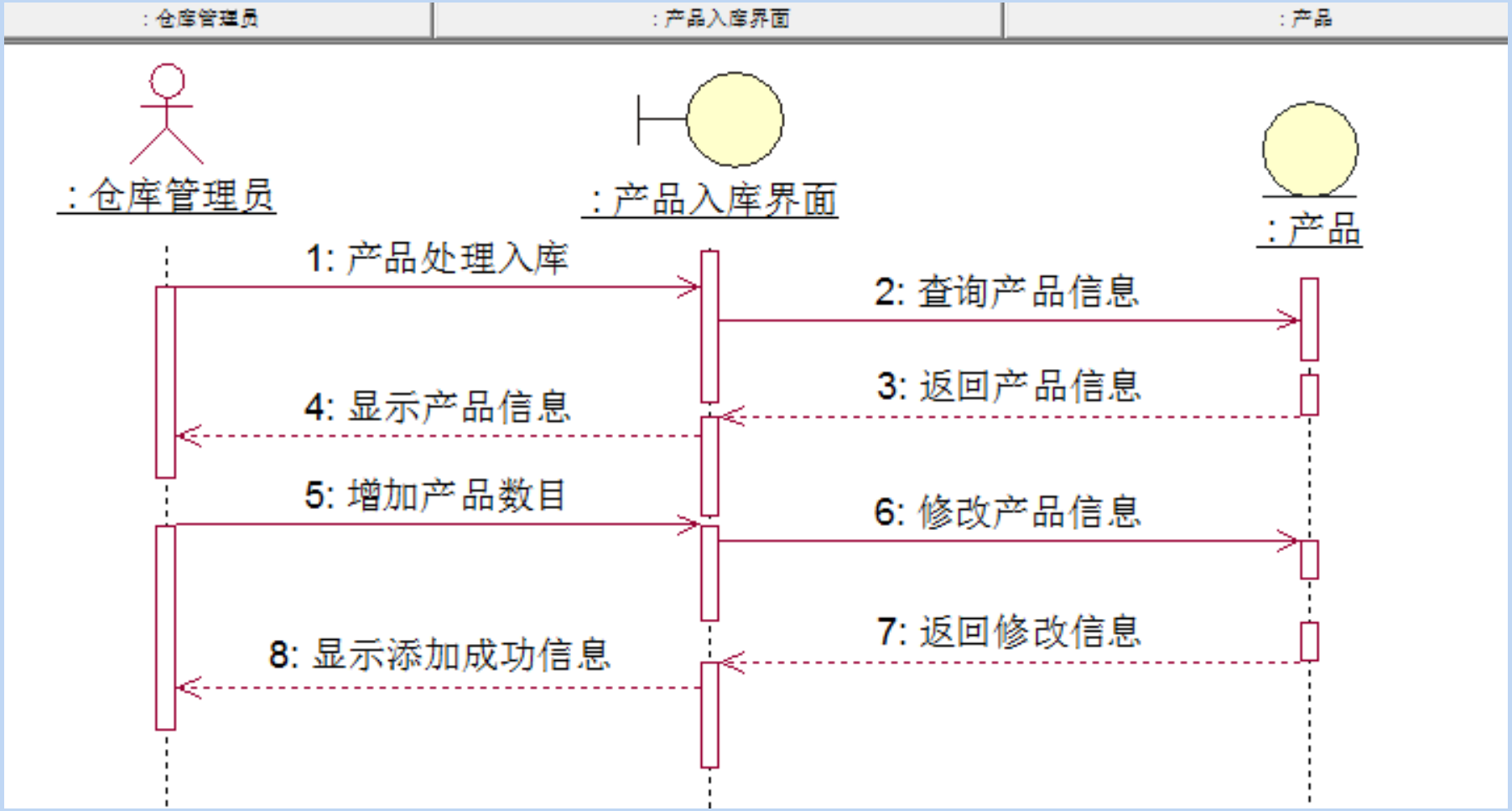
序列图的绘制顺序

- 确定交互的范围；
- 识别参与交互的对象和活动者；
- 设置对象生命线的开始和结束；
- 设置消息；
- 细化消息；

创建序列图实例-练习1

- 仓库管理员处理产品入库工作流程，基本工作流程如下：
  - 1.仓库管理员将通过库存管理子系统将生产的产品处理入库。
  - 2.仓库管理员通过库存管理子系统中的产品入库界面获取产品信息。
  - 3.产品入库界面根据产品的编号将产品类实例化并请求该类产品信息。
  - 4.产品类实例化对象根据产品的编号加载产品信息，并提供给产品入库界面。
  - 5.仓库管理员通过产品入库界面增加产品数目。
  - 6.产品入库界面通过产品类实例化对象修改产品信息。
  - 7.产品类实例化对象向产品入库界面返回修改信息。
  - 8.产品入库界面向仓库管理员显示添加成功信息。

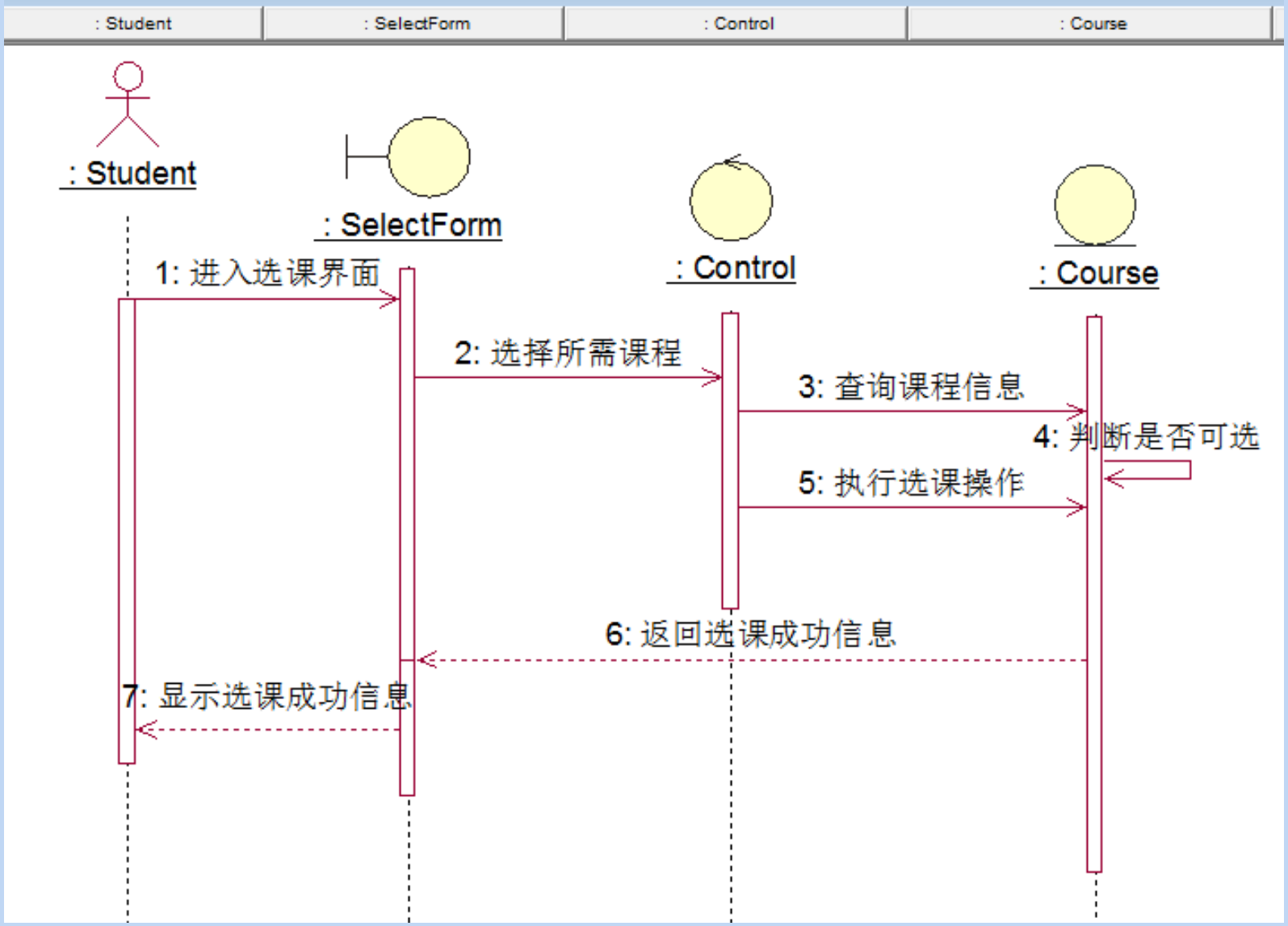
创建序列图实例1


：仓库管理员
：产品入库界面
：产品
1: 产品处理入库
2: 查询产品信息
3: 返回产品信息
4: 显示产品信息
5: 增加产品数目
6: 修改产品信息
7: 返回修改信息
8: 显示添加成功信息

创建序列图实例-练习2

- 为选课系统中的选课用例绘制序列图。假设学生已经成功登陆系统，选择的事件可以简化如下：
  - 选课事件：
    - 1.进入选课主界面。
    - 2.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。
    - 6.系统提示课程选择成功
    - 7.用例结束。
  - 错误事件：
    - 5A: 课程不可选。
      - 5A1：系统提示课程不可选原因。、
      - 5A2：重新选课。

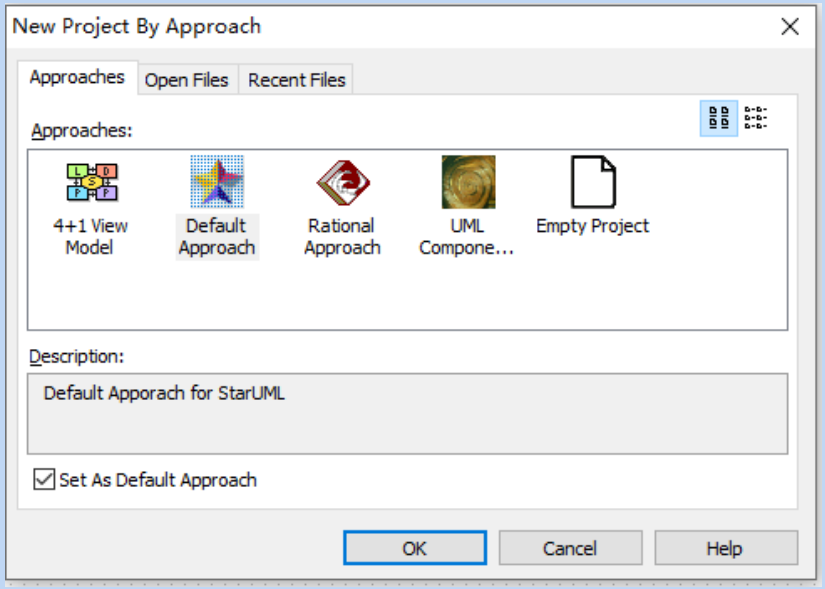
创建序列图实例2

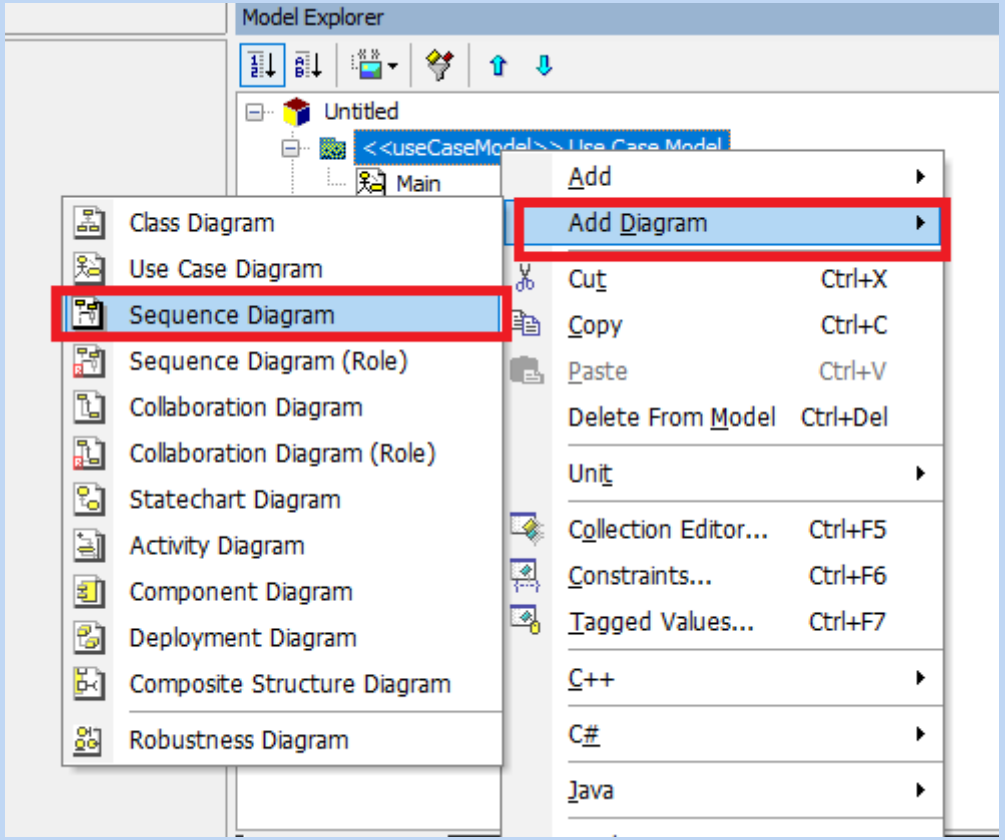

: Student
: SelectForm
: Control
: Course
1: 进入选课界面
2: 选择所需课程
3: 查询课程信息
4: 判断是否可选
5: 执行选课操作
6: 返回选课成功信息
7: 显示选课成功信息

StarUML绘制序列图

创建序列图

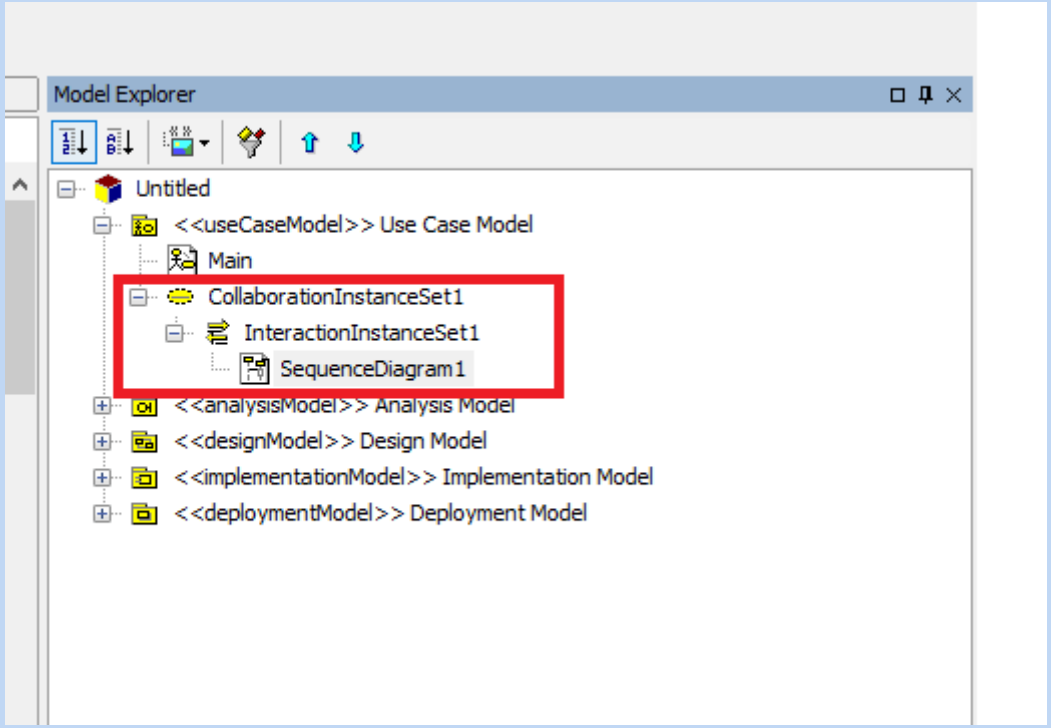
- 打开StarUML后，创建一个DefaultApproach，在Use Case Model树节点右击，在弹出的快捷菜单中选择Add Diagram|Sequence Diagram命令建立新的序列图。





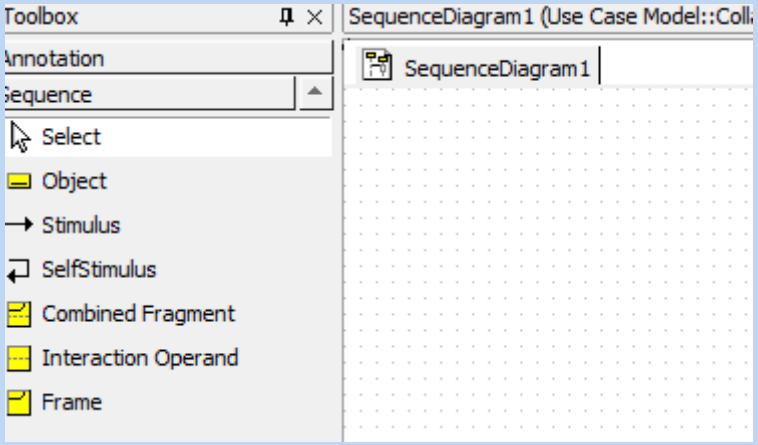
创建序列图

- 创建新的用例图，在Use Case Model树型下就多了一个名为SequenceDiagram1的图标，这个图标就是新建的序列图图标。选中此图标，用快捷键F2为新创建的用例图重命名，或者下方Properties属性窗口中修改Name。



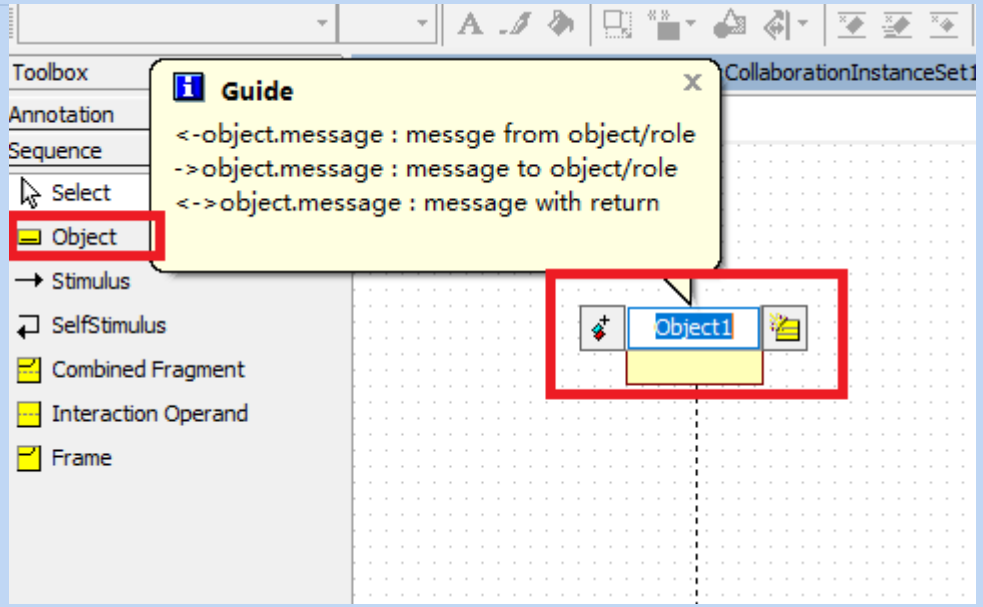
创建序列图

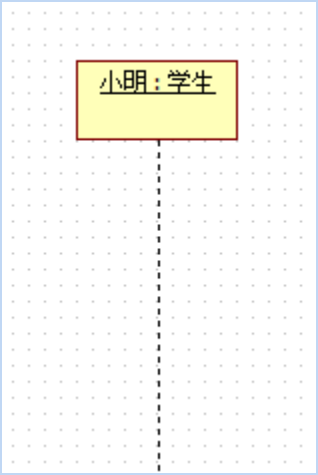
- 双击名为SequenceDiagram1的类图图标，会出现序列图的编辑工具栏和编辑区。

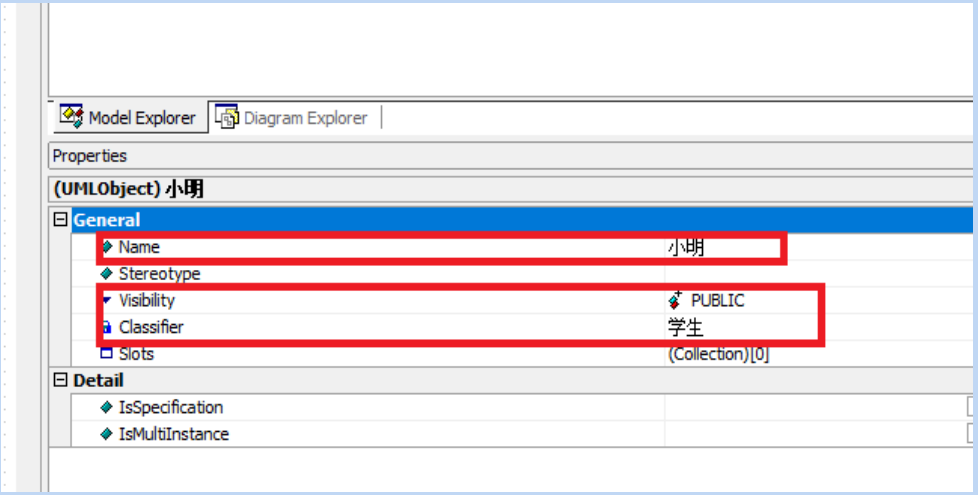


创建对象

- 单击左侧工具栏上的Object，设置左侧可见域，和对象名称，和右侧所属类的名称（可填可不填）。也可以在右方Properties里面修改相应的值

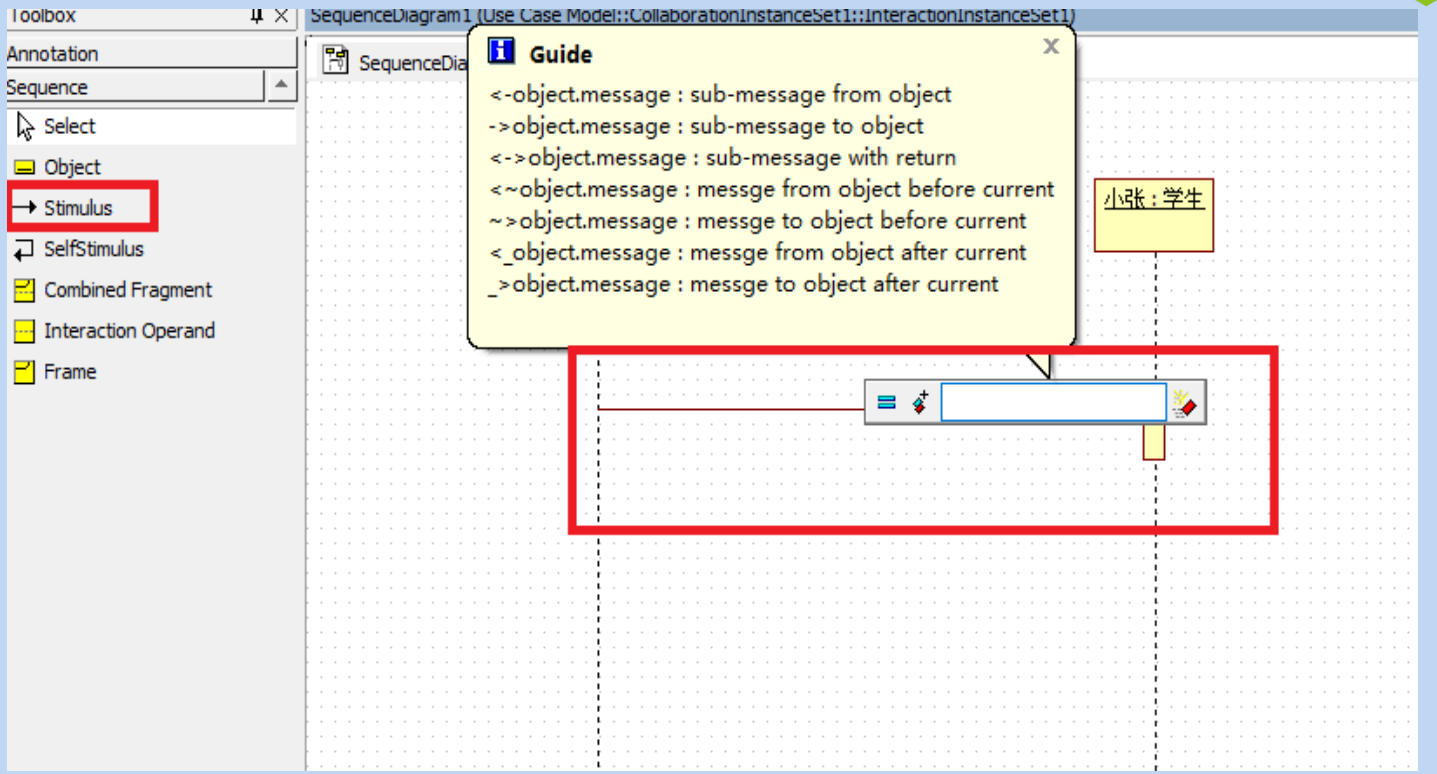


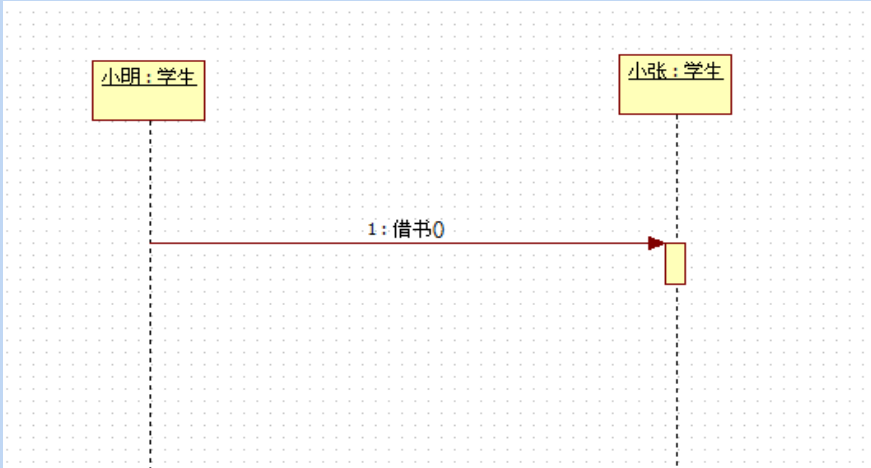




创建活动

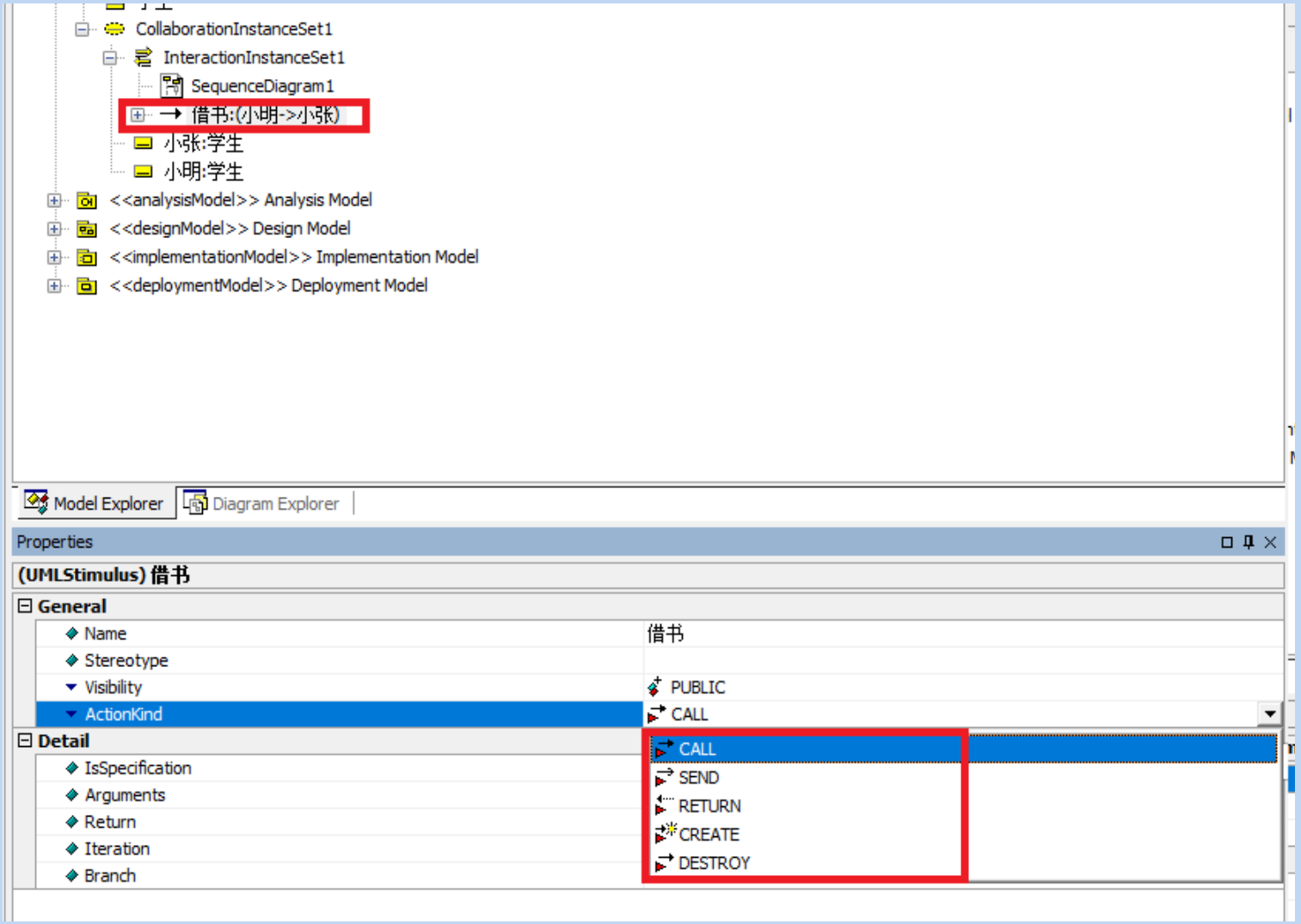
- 选中左侧关系，选中两个对象的生命线，填入活动名称





修改消息类型

- 可在Properties修改消息类型
- Call为同步消息
- Send为简单消息
- Return为返回消息
- Create为创建对象
- Destroy为撤销对象
- 其他类型消息在StarUML不做体现



05

Part five

课堂练习

课堂练习

图书馆借书流程绘制成序列图，需求描述如下：

- 读者在借书时，先由管理员把借书证上的读者编号扫描给系统，系统检查这个读者的借书证合法信息；如果不合法,则系统给出提示，如果正确系统返回读者的身份信息，以及读者的借阅信息。
- 如果读者借书数量没有超过借书的上限，则把要借书的图书编号输入系统，系统登记借书信息，并返回借书成功信息，借书过程完成

课堂练习-顺序图分析

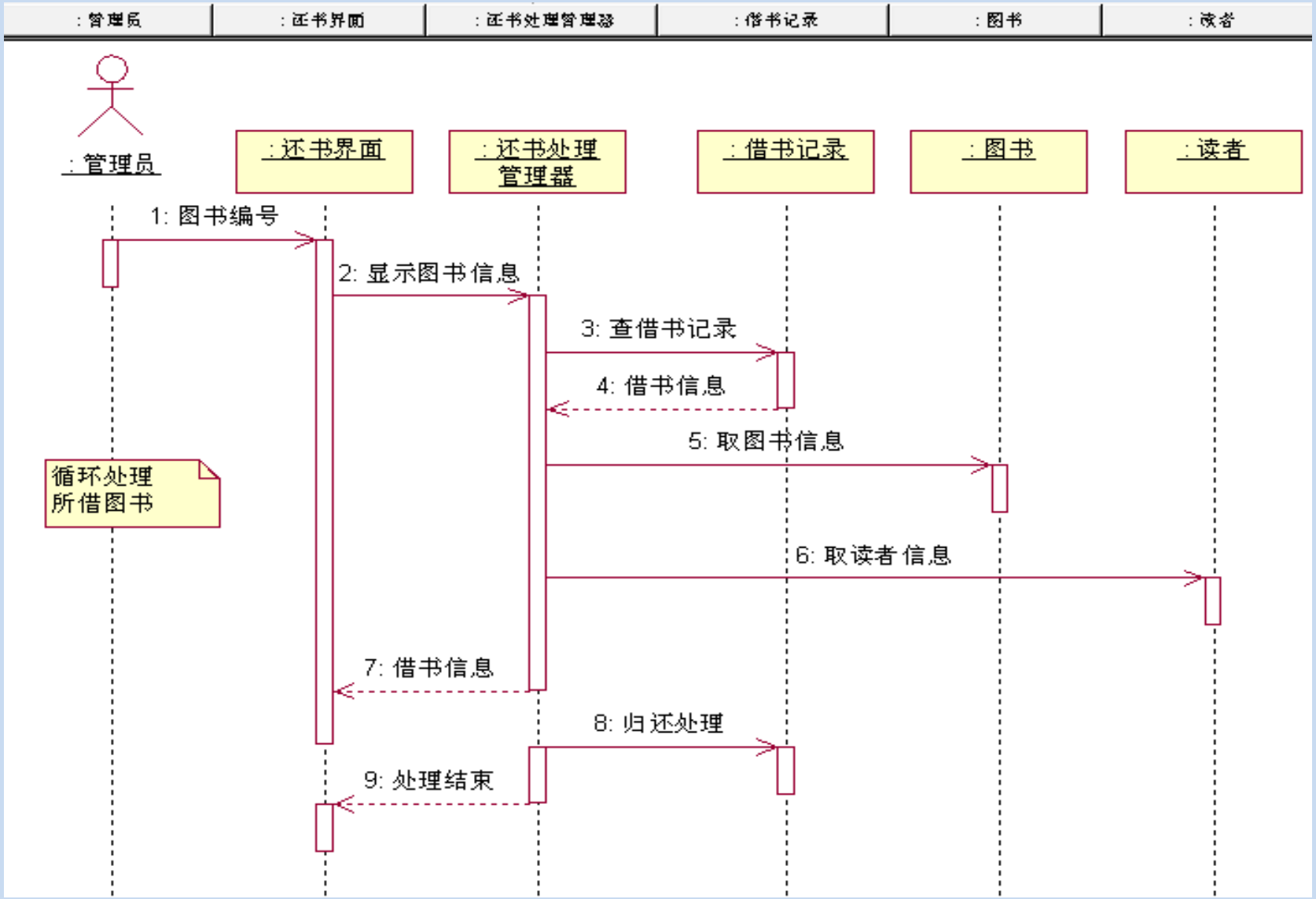
- 用例：借书
- 参与者：管理员,借阅者
- 基本事件流：

1. 借阅者带着借书证和书来到柜台。
2. 管员输入借阅者的借书证编码。
3. 系统检查借阅者的合法性。
4. 系统显示借阅者的信息。
5. 系统检查借阅者的借阅信息。
6. 管员输入所借图书的条码。
7. 系统显示所借图书的信息，并登记图书。
   管员重复6~7步，直到结束。
8. 系统登记借阅者的本次借阅信息，并显示。

- 扩展事件流：

3a. 借阅者身份非法，则提示，本次拒借。
5a. 该借阅者所借图书中有超期的, 则提示, 本次拒借。
5b. 该借阅者所借图书数量超过了上限, 则提示, 本次拒借。

课堂练习-参考答案


:管理员
:还书界面
:还书处理管理器
:借书记录
:图书
:读者
1: 图书编号
2: 显示图书信息
3: 查借书记录
4: 借书信息
5: 取图书信息
循环处理
所借图书
6: 取读者信息
7: 借书信息
8: 归还处理
9: 处理结束

感谢您的观看