



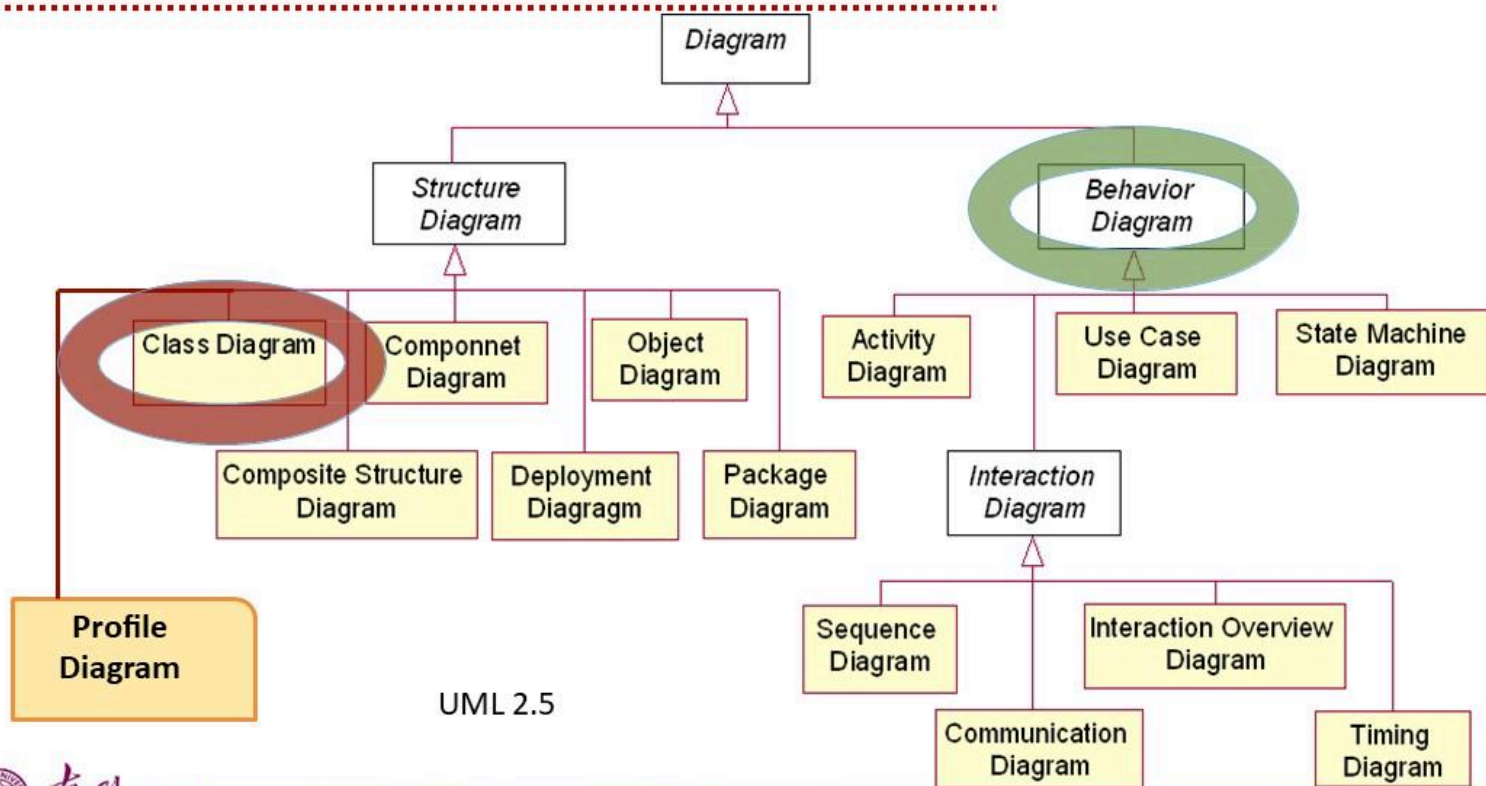
# 交互行为建模 Modelling Interactions

徐思涵  
南开大学

*Slides adapted from materials by Prof. Qiang Liu (Tsing Hua University)  
and Ivan Marsic (Rugers University)*



# UML 2.0中图的分类





# 顺序图

## Sequence Diagram

徐思涵  
南开大学

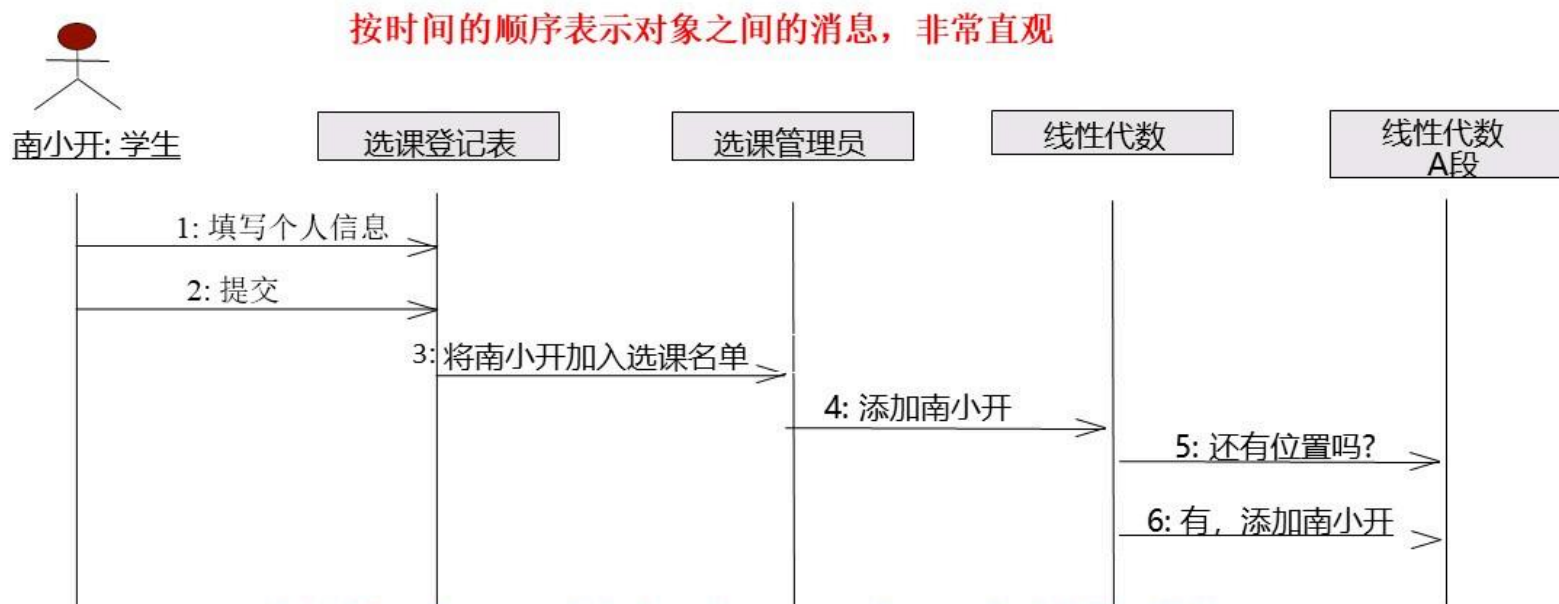
*Slides adapted from materials by Prof. Qiang Liu (Tsing Hua University)  
and Ivan Marsic (Rugers University)*



南开大学  
Nankai University

# 顺序图举例 (Sequence Diagram)

- 顺序图用来刻画系统实现某个功能的必要步骤



南开大学  
Nankai University

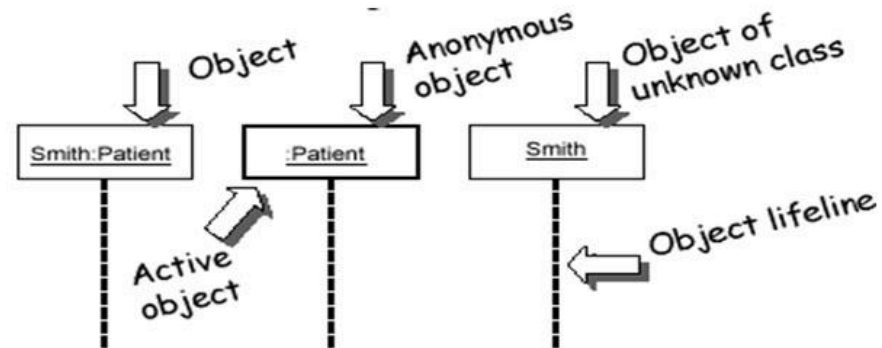
能够指出有哪些对象参与了交互，以及它们之间消息传递的序列

4

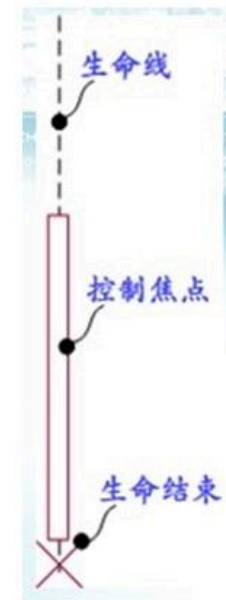
## 顺序图建模元素对象(Object)及其生命线(Lifeline)

**对象:** 以某种**角色**参与交互, 可以是人, 物, 其他系统或者子系统

**生命线:** 表示对象存在的时间



**Name syntax:** <objectname>:<classname>



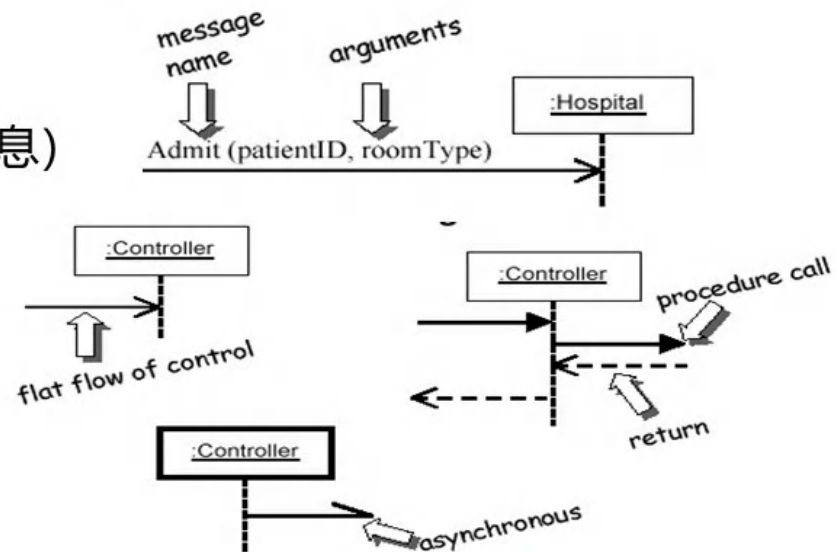
**控制焦点/激活期(Focus of Control/Activation):** 表示对象进行操作的时间片段

# 顺序图建模元素——消息(Message)

- 消息(Message)用于描述对象间的交互操作和值传递过程

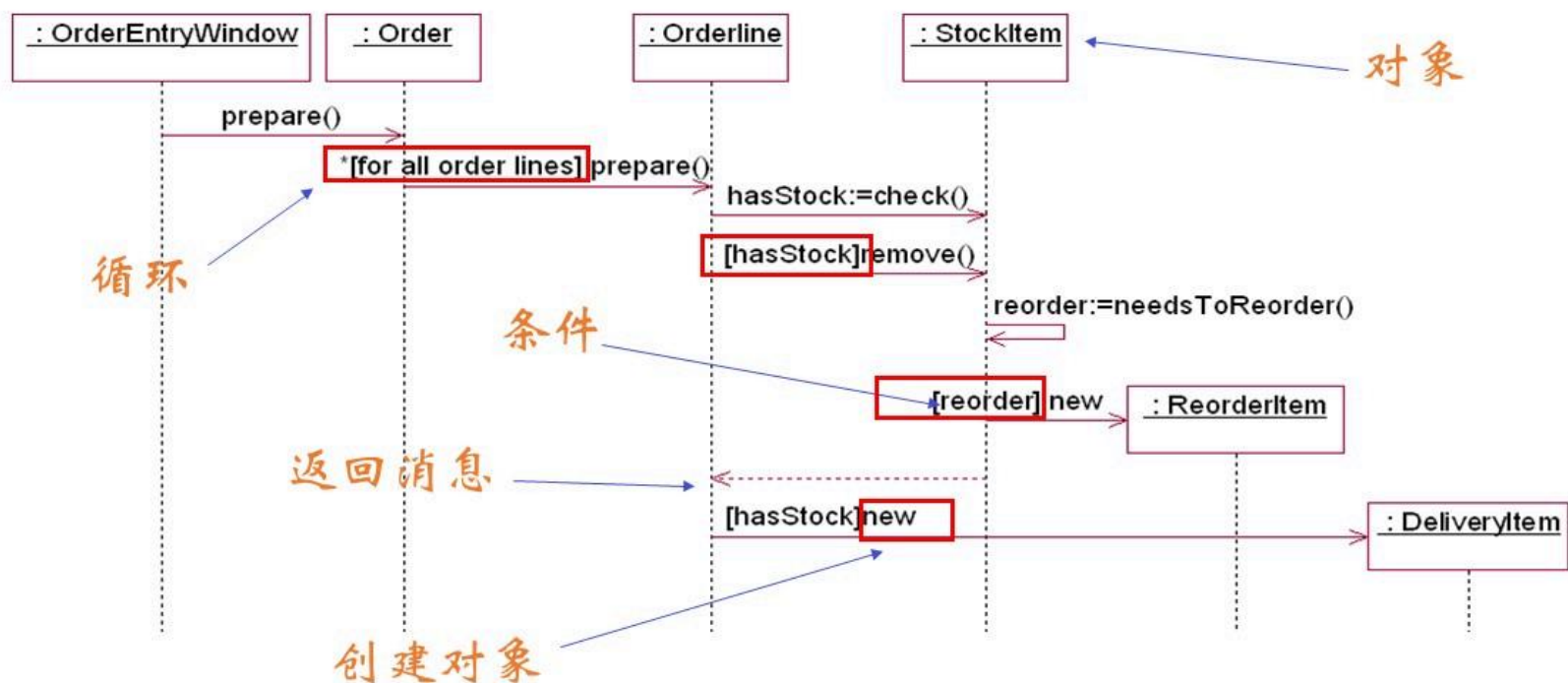
- 消息类型:

- Synchronous 同步消息 (调用消息)
- Return 返回消息
- Asynchronous 异步消息
- Self-message 自关联消息
- Time-out 超时等待
- Uncommitted / Balking 阻塞





# 顺序图中的基本结构



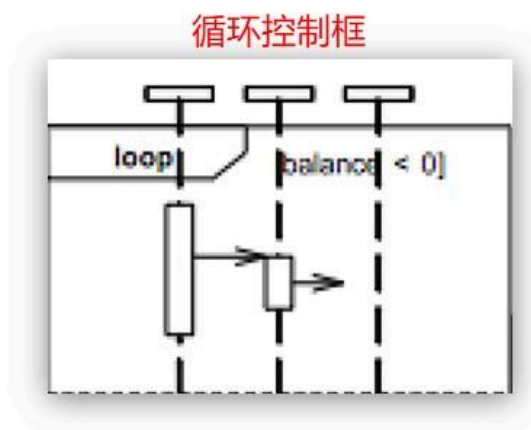
# 顺序图中消息的循环发送

- 在消息名字前加循环条件或添加循环控制框

例:

1.1 \*[ for all order lines]: message1()

2.1 \*[i:=1..n]: message2()

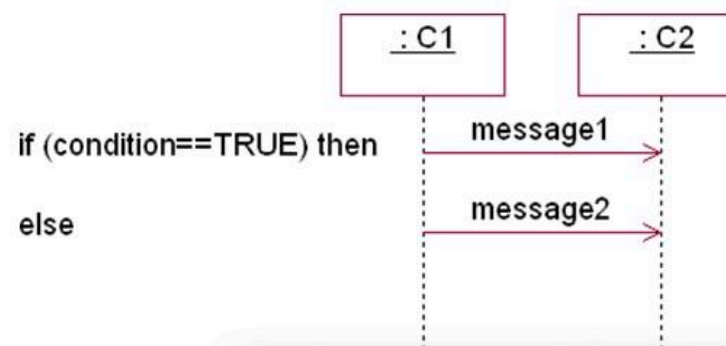





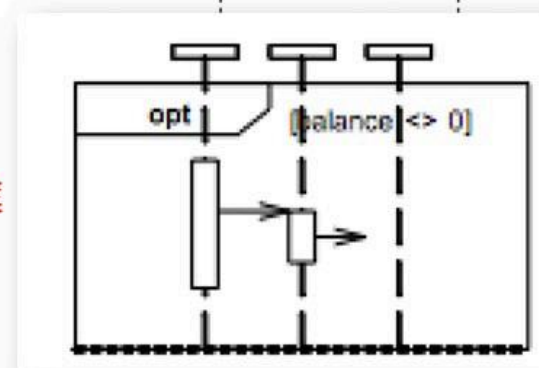
# 顺序图中带条件消息的发送

- 在消息名字前加条件子句;
- 使用文字说明;
- 添加条件控制框;
- 分成多个顺序图子图并关联

[ hasStock]: remove()



条件控制框



# 绘制顺序图

1. 在顺序图顶端绘制矩形框，定义参与交互的**类实例**（对象）名；
2. 在每个对象下面绘制竖直虚线，表示该对象的**生命线**；
3. 在对象间添加箭头表示各种类型的**消息**，跟踪对象间的控制流；
4. 生命线加竖直矩形定义对象**激活期**，表明对象正在执行某操作；
5. 根据需要添加**框**的组合与关联，表示复杂的控制结构。

# 顺序图建模过程

## • 参与者

顺序图中有关对象或者实体

## • 消息

参与对象之间的通信，通过箭头表示

- 顺序图的**起始**是一个没有发起对象的消息

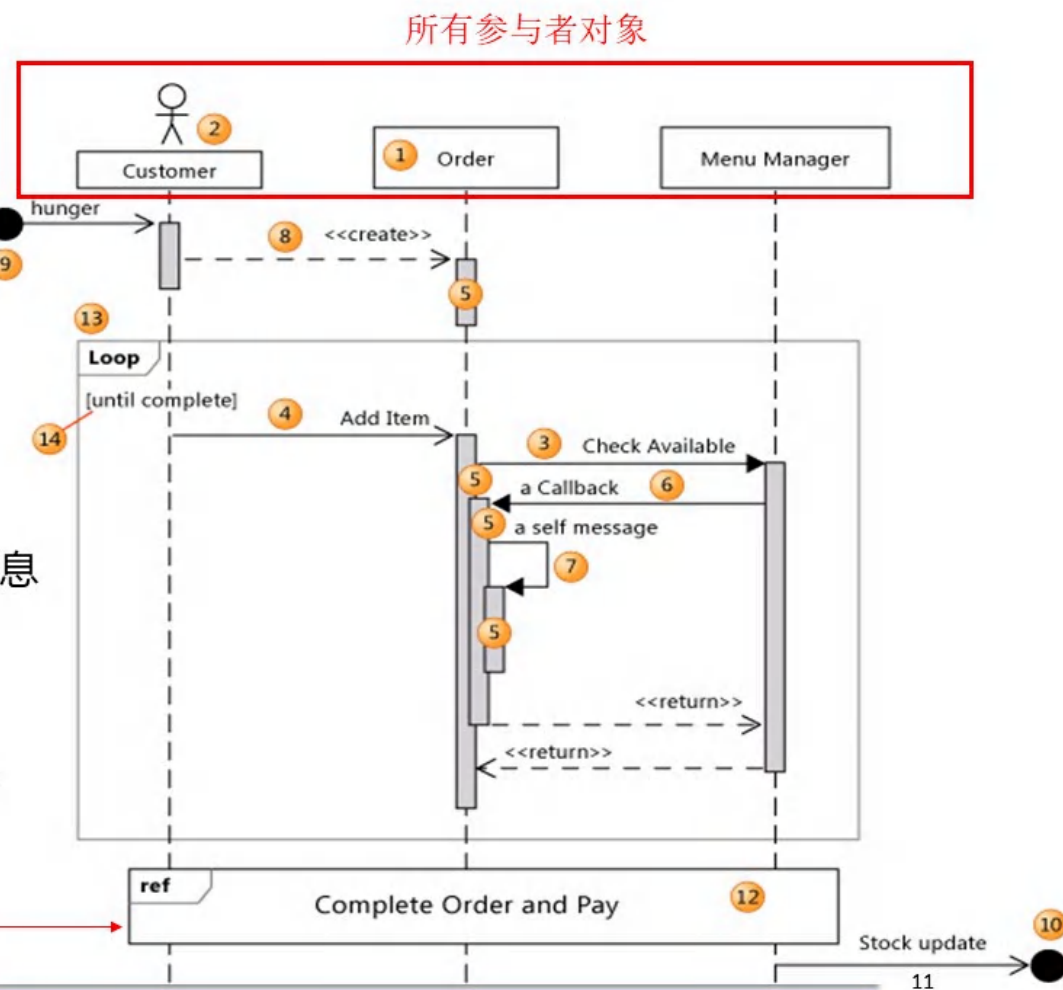
- **每个消息代表的操作属于消息接受方**

## • 轴

- 横轴: 表明正在进行操作的对象/参与者

- 纵轴: 时间 (向下表明时间的顺延)

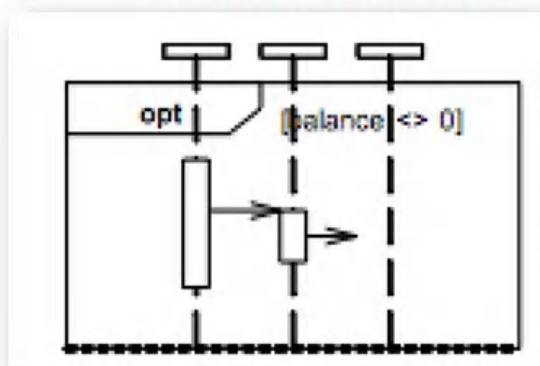
外部引用



## 组合框：复杂控制结构表示

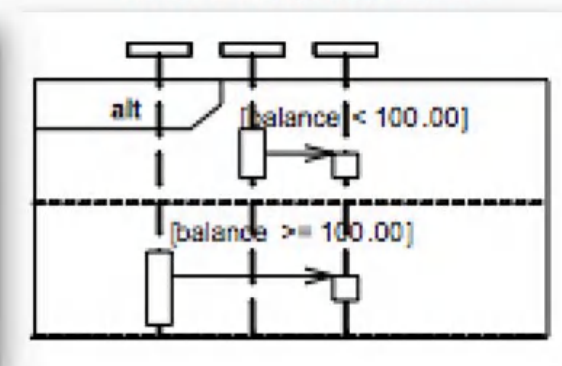
- 框 (Frame)：框中包含顺序图的部分结构，表示选择(selection)或者循环(loop)结构，左上角注明结构类型，[]中注明条件。

单一条件的选择结构



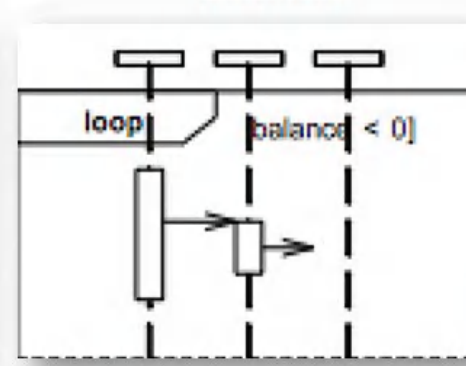
If ->  
(**opt**)[condition]

有两个分支的选择结构



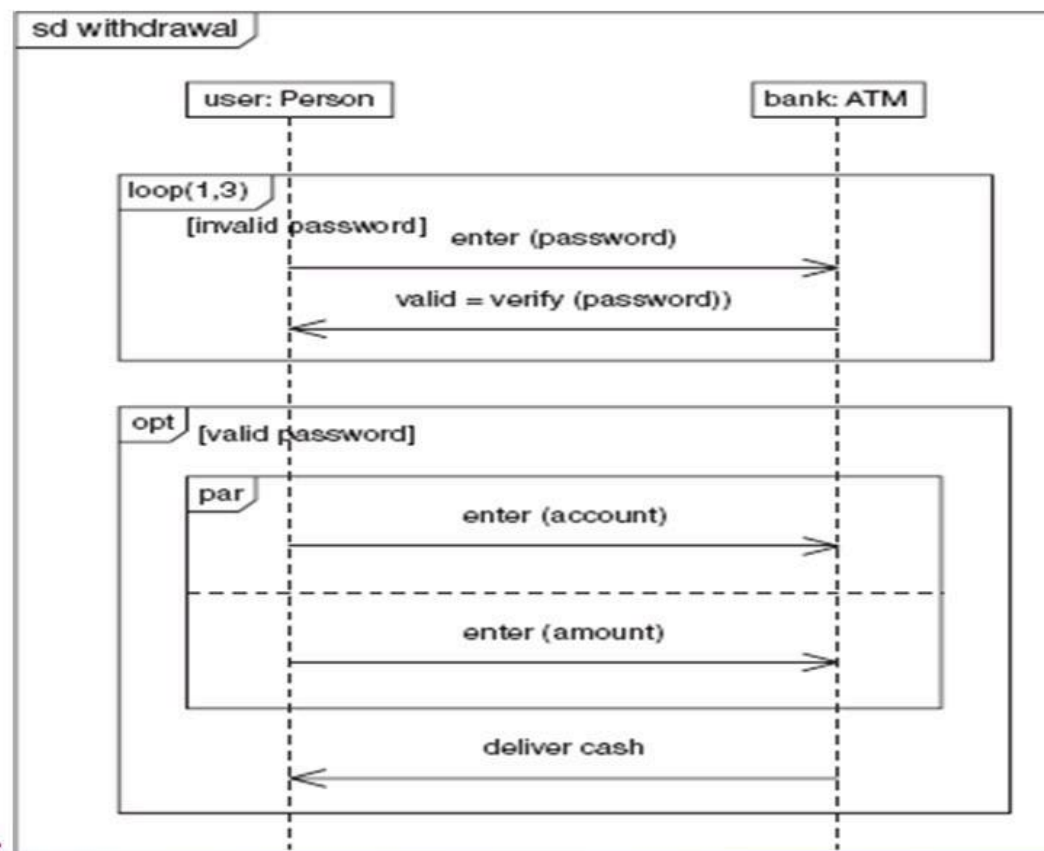
if/else  
(**alt**)[condition] ->  
通过水  
平虚线分割不同情形  
并发结构 -> (**par**)

循环结构

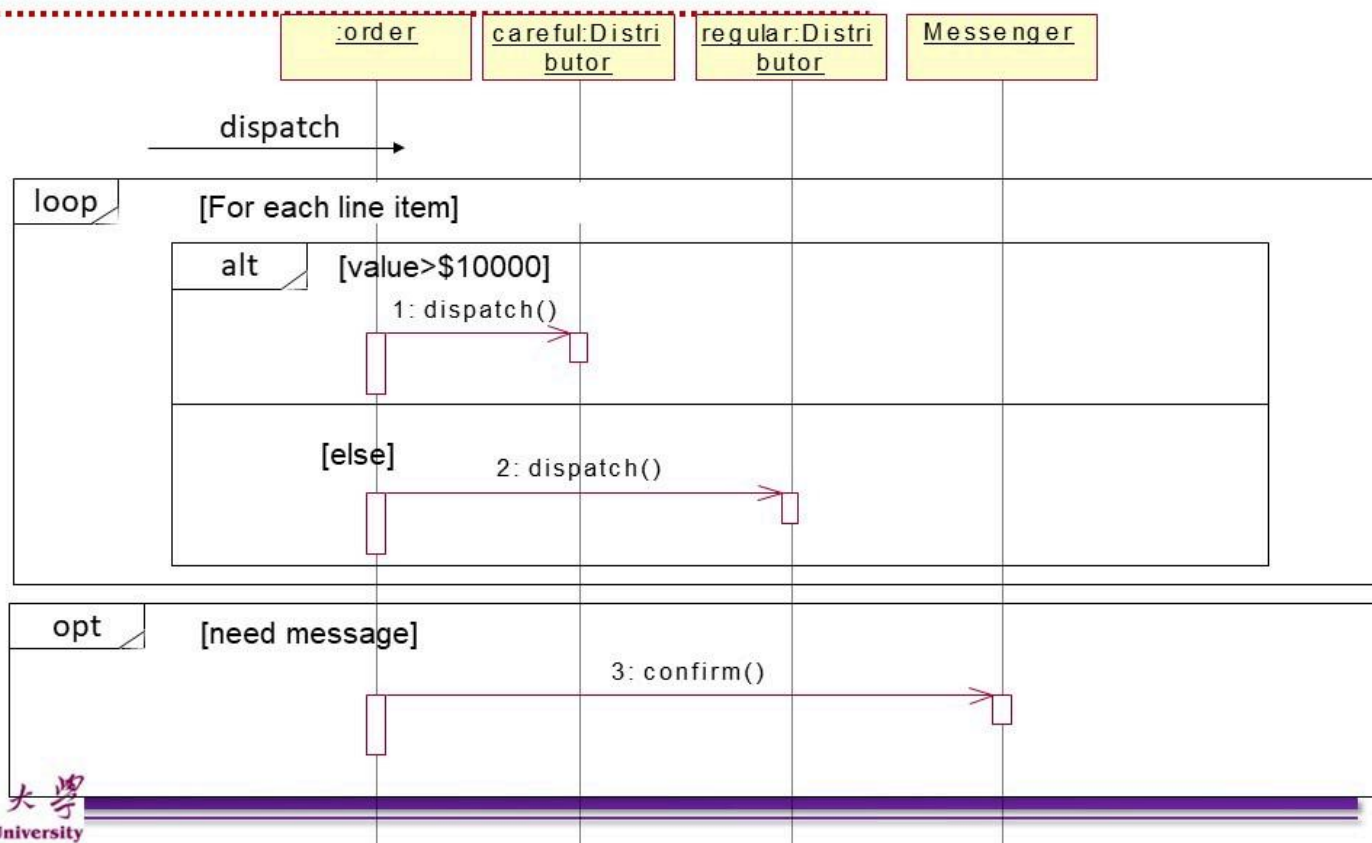


loop -> (**loop**)[condition or items to loop over]

## 控制框建模例子



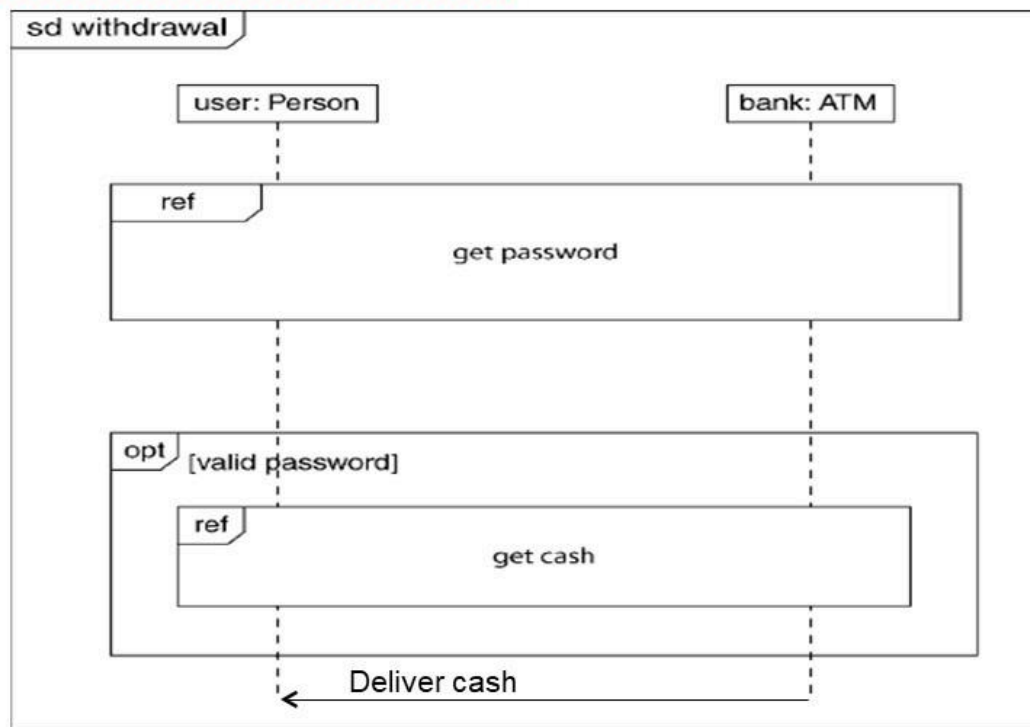
# 控制框建模例子

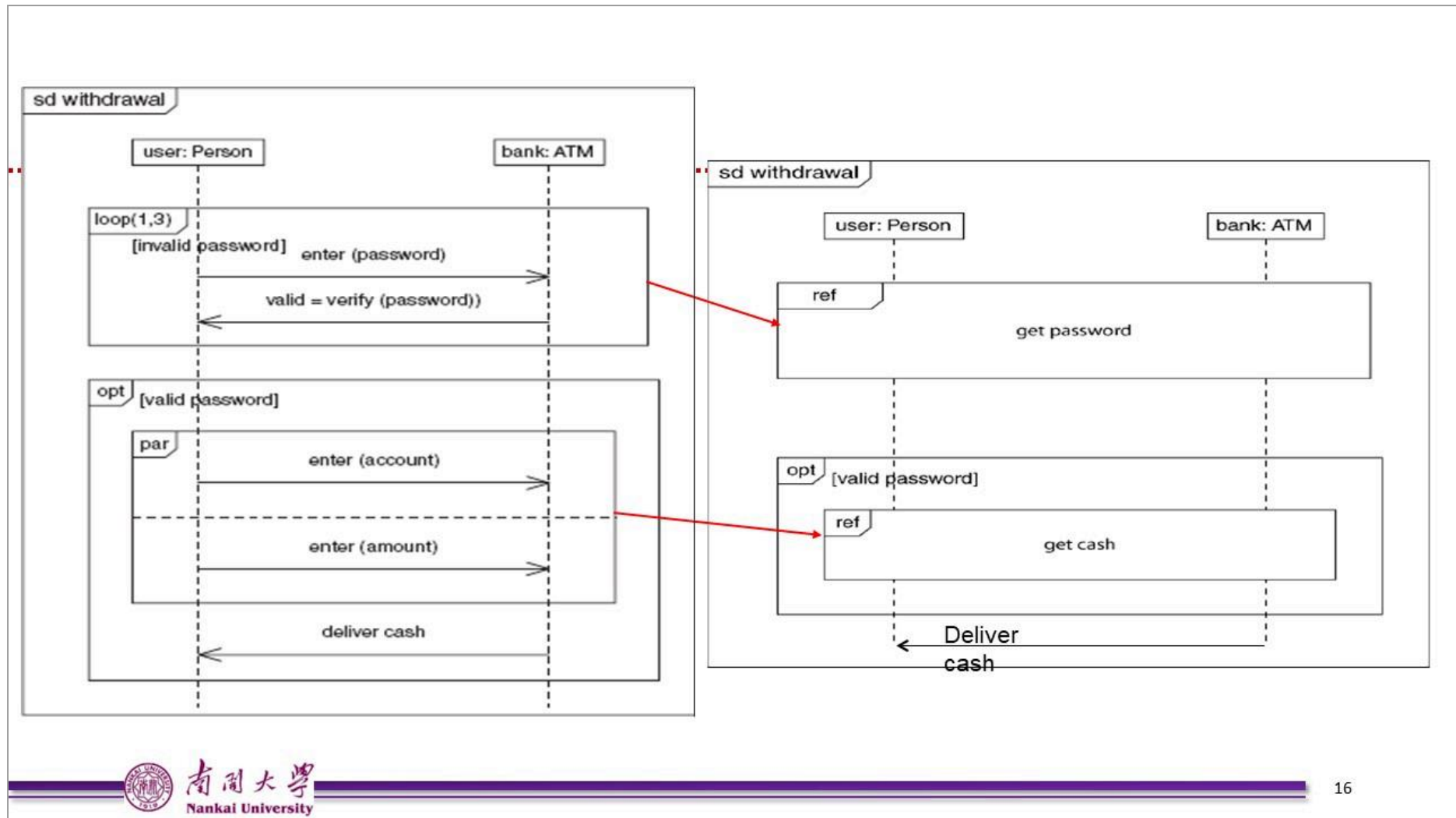




# 顺序图间的关联

- 当一个顺序图过大
- 需要引用其他图表时，  
选择下述表示：
  - 通过名为“ref”的  
框图引用相关图表

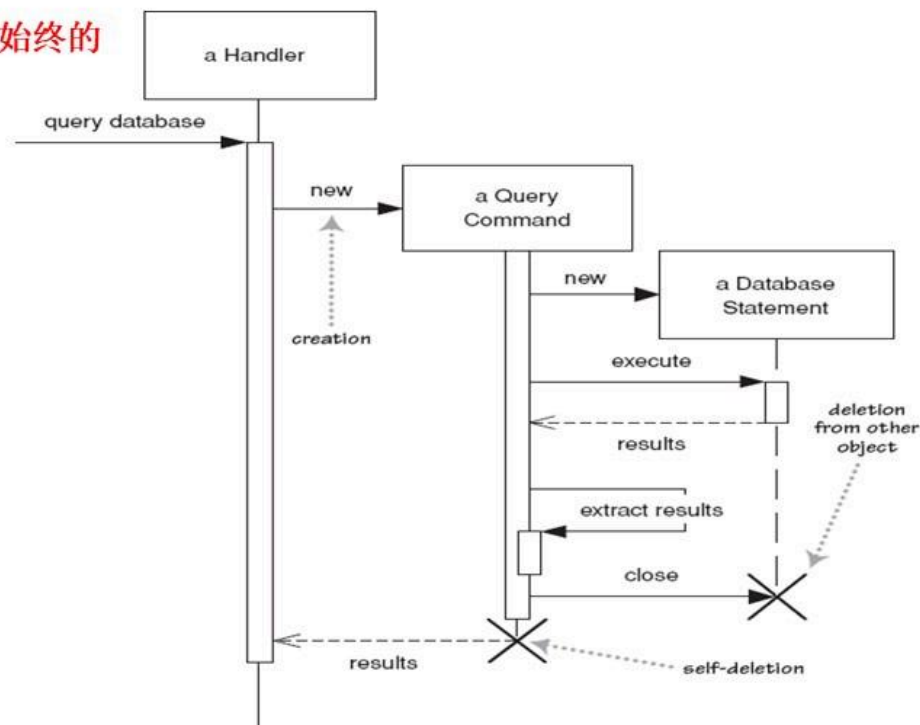




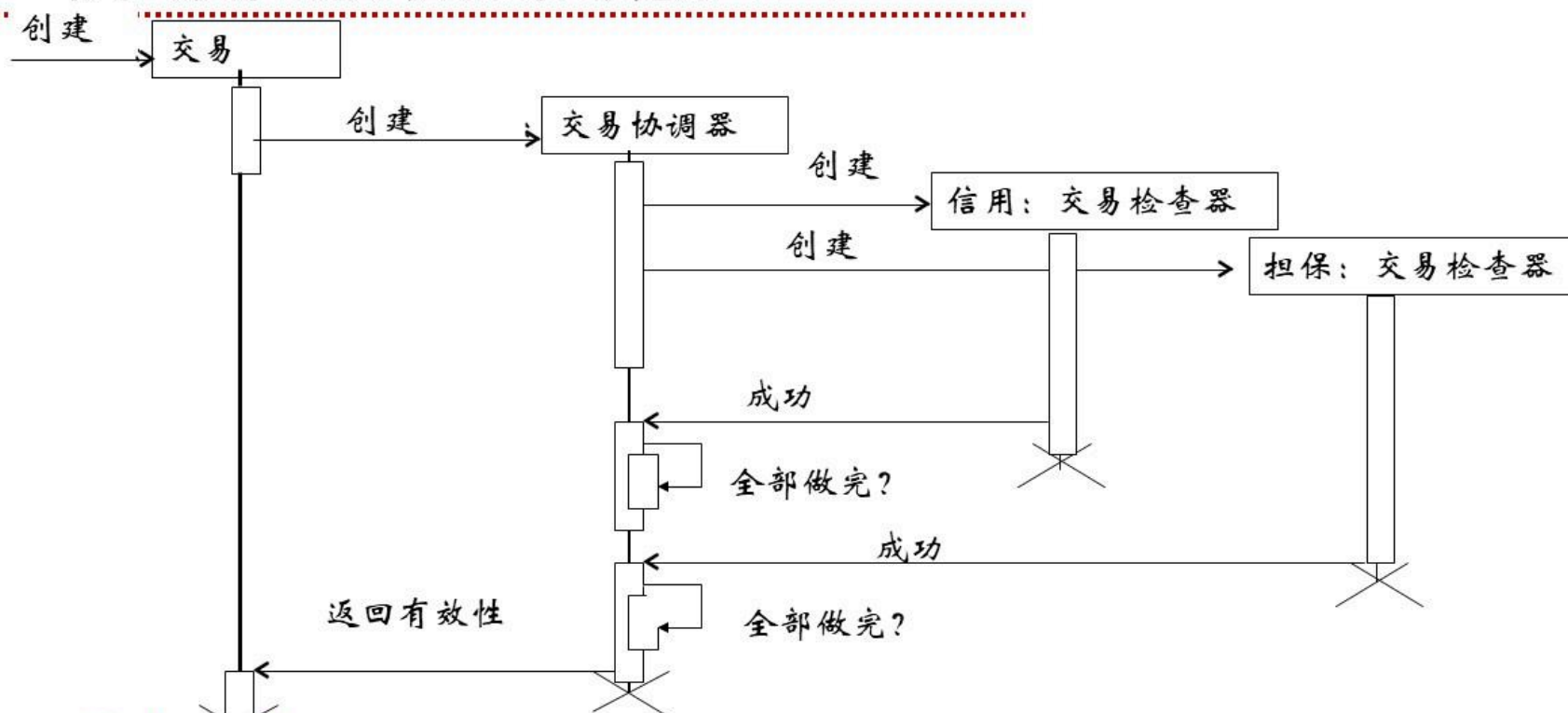
# 对象的创建与撤销

大多数情况下，对象的生命周期是贯穿一个场景执行的始终的

- 创建(creation): "new" 标明的箭头
  - 用例场景中新建的对象在图中的位置较低
- 撤销(deletion,destroy): 生命线底部的 "X"
  - 注: 在Java没有明确的销毁对象的操作, 通过垃圾回收机制处理

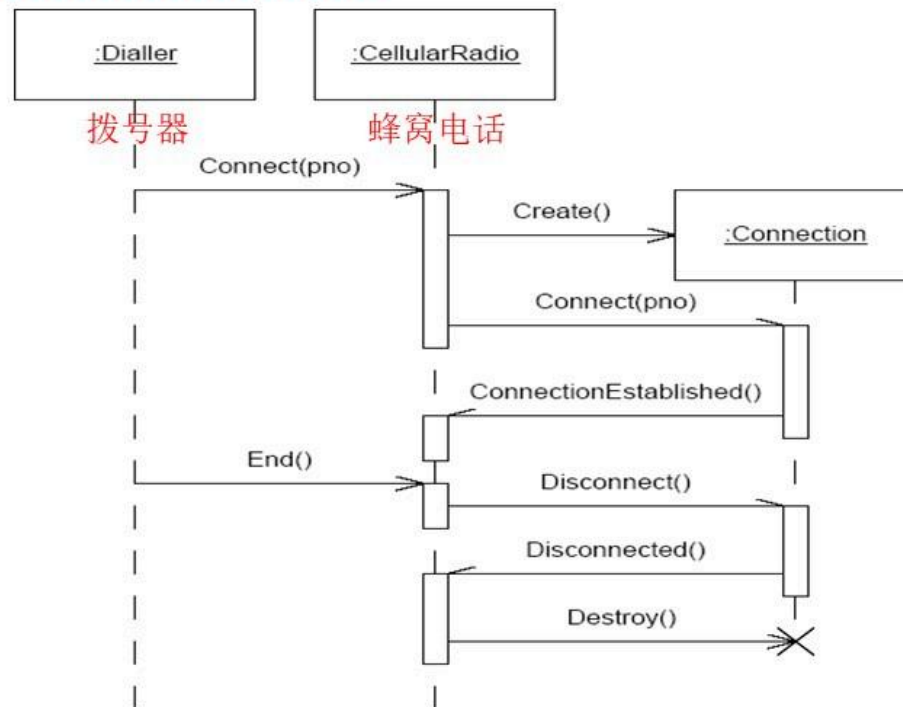


# 例:银行系统的交易验证

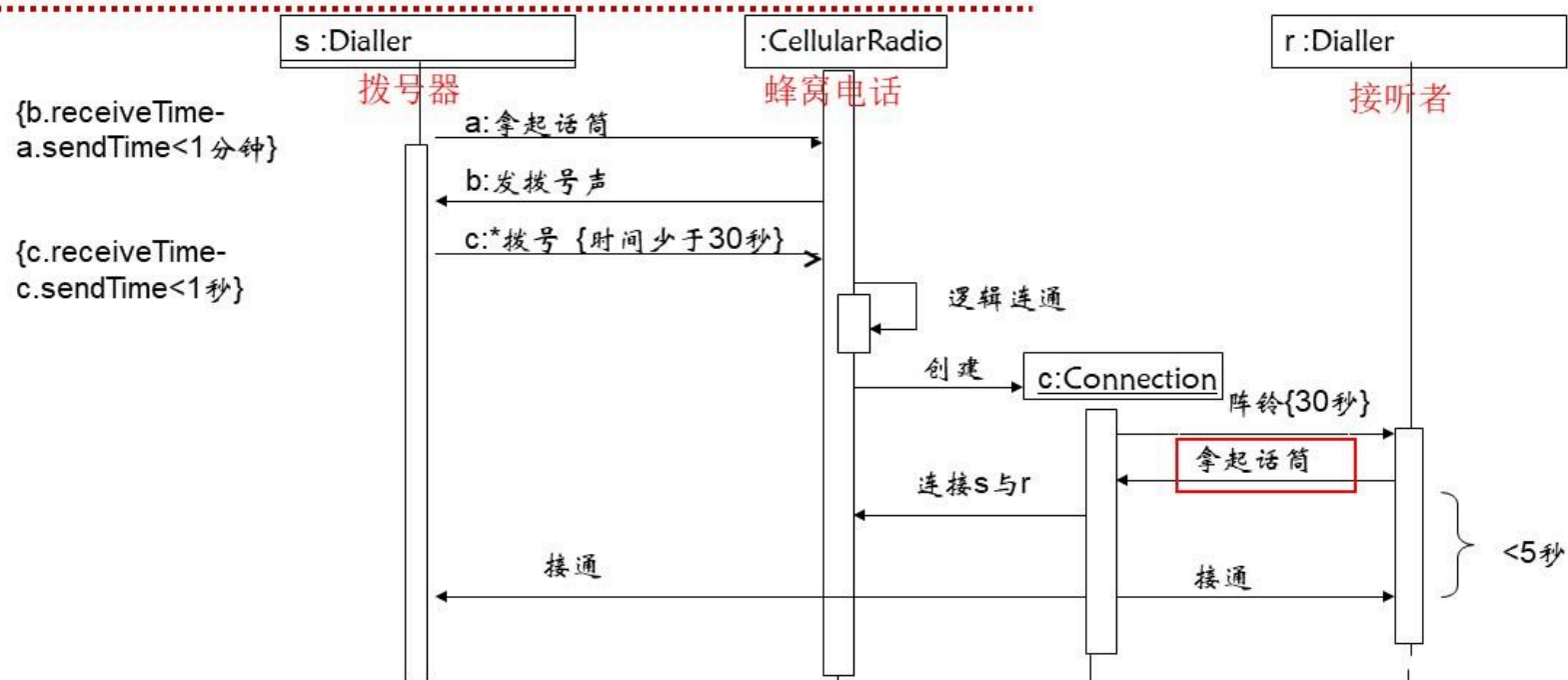


## 思考题：有缺陷的顺序图

- 这个顺序图有什么问题？  
(留意UML的语法以及场景的可行性)



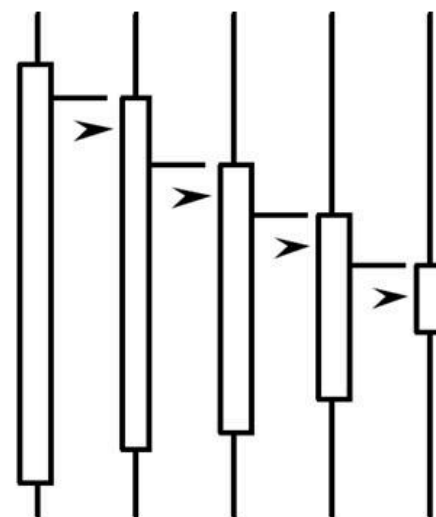
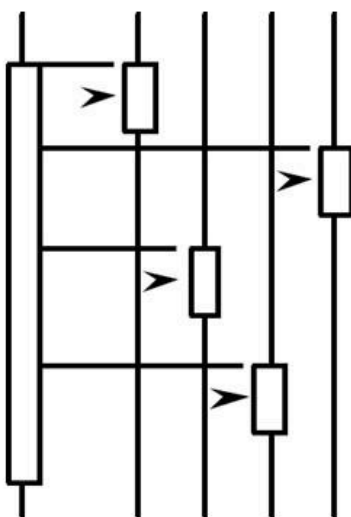
# 修改方案





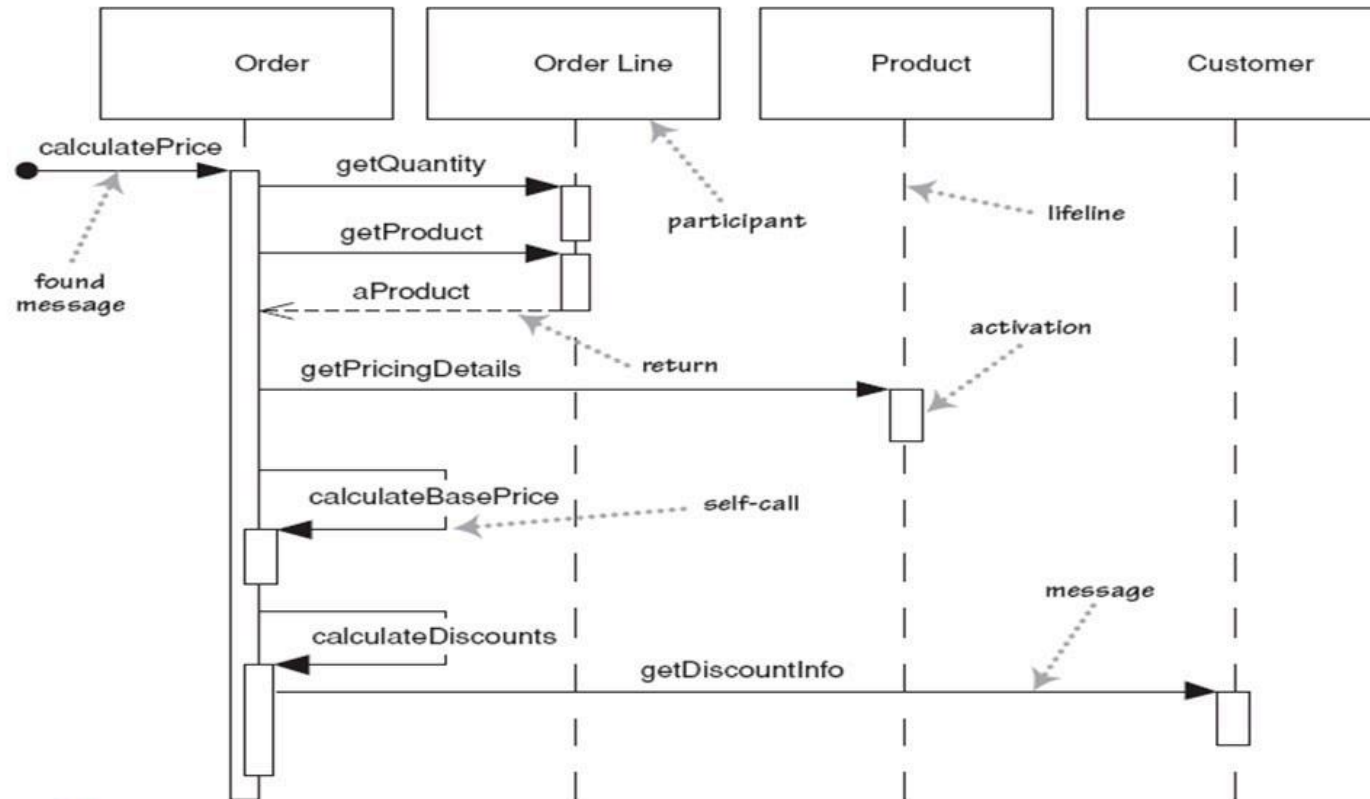
# 集中、分布控制

- 下述两种系统的控制流有什么特点?
  - 集中式的?
  - 分布式的?

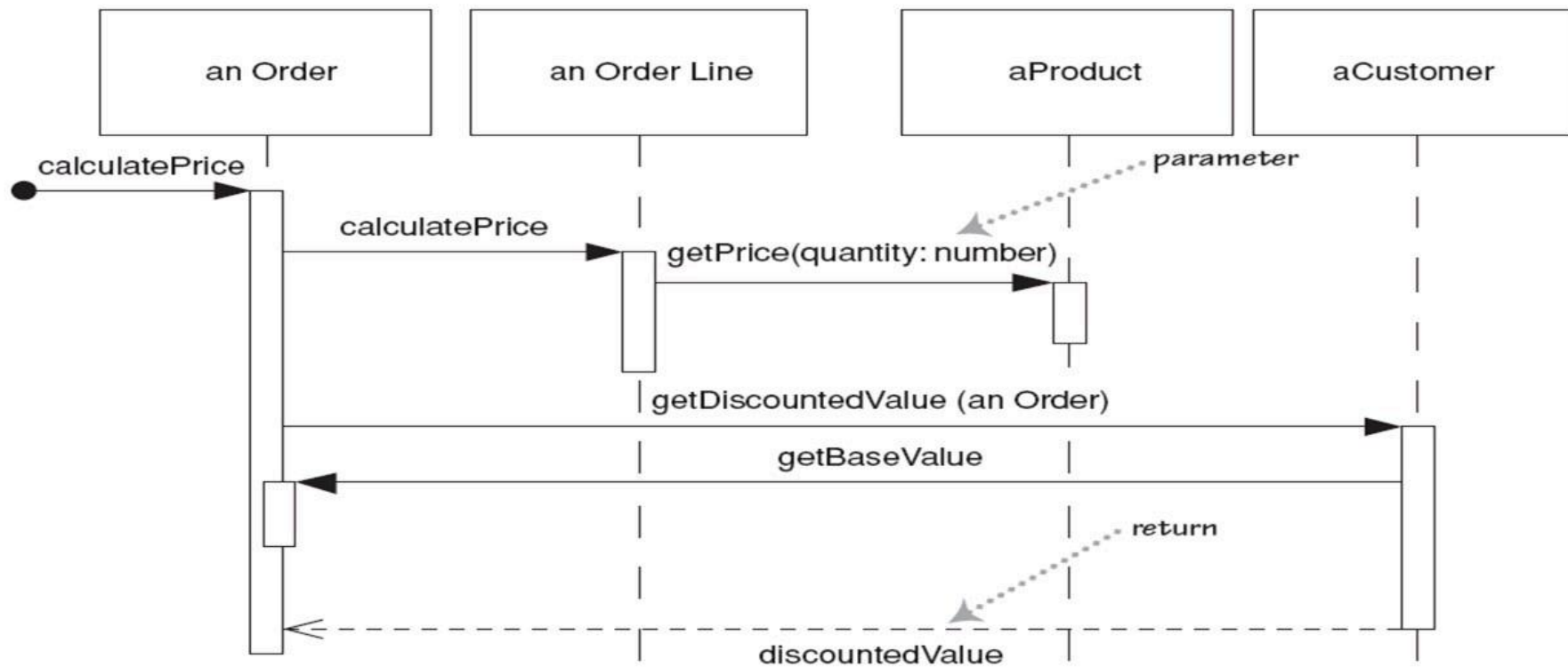


## 例1：集中控制的计价系统顺序图

几乎所有的控制命令都是从订单这个对象发出的



## 例2：分布控制的计价系统顺序图



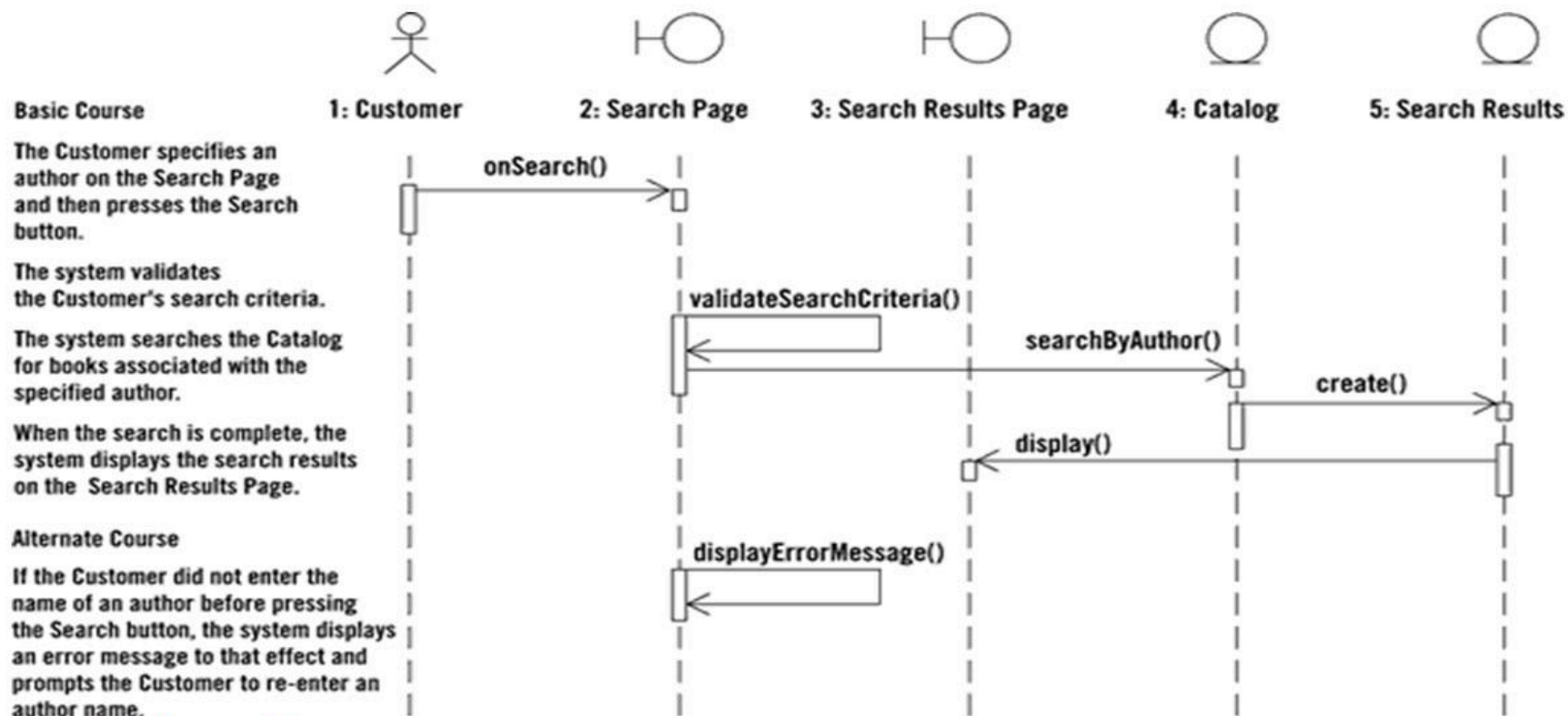
# 顺序图与用例的关系 I

- 顺序图表达**单个情景**实例的行为。
- 每个用例对应一个顺序图。
- 顺序图表达对象间如何**协作**完成用例所描述的功能。
- 顺序图用于表示为完成用例而在系统边界输入输出的**数据**以及**消息**
- 顺序图也用于表示系统内部对象间的消息传递。

## 顺序图与用例的关系 II

- 顺序图可帮助分析人员对用例图进行扩展、细化和补遗
- 顺序图可用于开发周期的不同阶段，服务于不同目的，描述不同粒度的行为
- 分析阶段的顺序图**不要**
  - 包含设计对象
  - 关注消息参数

## 从用例中抽取顺序图





# 顺序图建模意义

---

- 通过顺序图描述算法逻辑
- 高质量的顺序图是代码的抽象
- 顺序图是与语言无关的表示方式
- 可以绘制顺序图来描述业务逻辑
- 可以通过团队协作完成顺序图的绘制
- 可以在同一页浏览多个对象和类的行为

# 顺序图建模风格

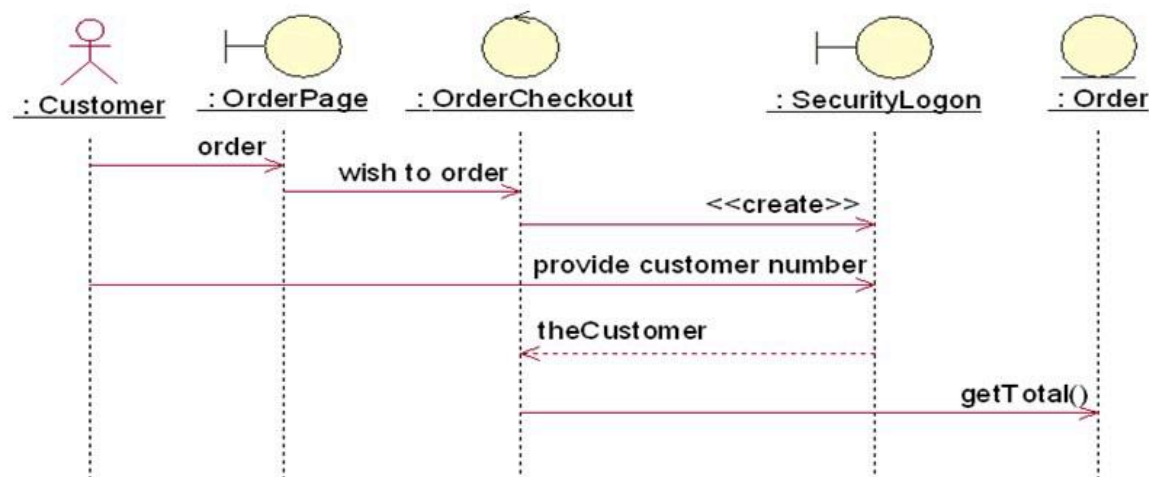
- 建模风格1：把注意力集中于关键的交互。
  - 创建模型时要把注意力集中于系统的关键方面，而不要包括无关的细节。  
例如：
  - 如果顺序图是用于描述业务逻辑的，就没必要包括对象和数据库之间的详细交互。

# 顺序图建模风格

- 建模风格2：对于参数，优先考虑使用参数名而不是参数类型。
  - 例如  
消息 `addDeposit(amount, target)` 比 `addDeposit(Currency, Account)` 传递了更多的信息
  - 在消息中只使用类型信息不能传递足够的信息
  - 参数的类型信息用UML类图表示更好

# 顺序图建模风格

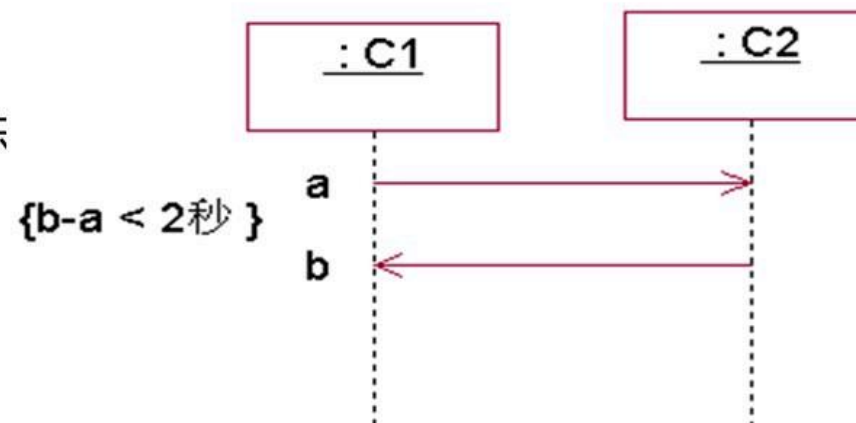
- 建模风格3：不要对明显的返回值建模。



- 建模风格4：可以把返回值建模为方法调用的一部分。

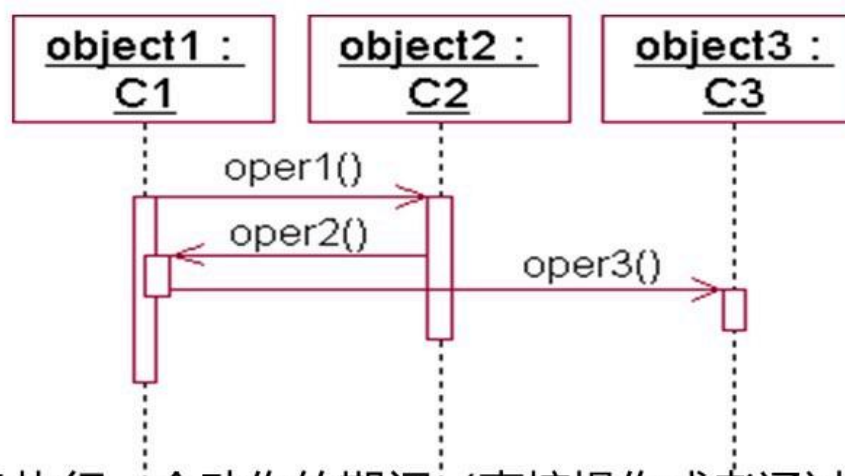
# 顺序图常见问题分析

- 顺序图中时间约束的表示
  - 用约束 (constraint)来表示



## 控制焦点(focus of control)的嵌套

- 嵌套的FOC可以更精确地说明消息的开始和结束位置。
- 图例：



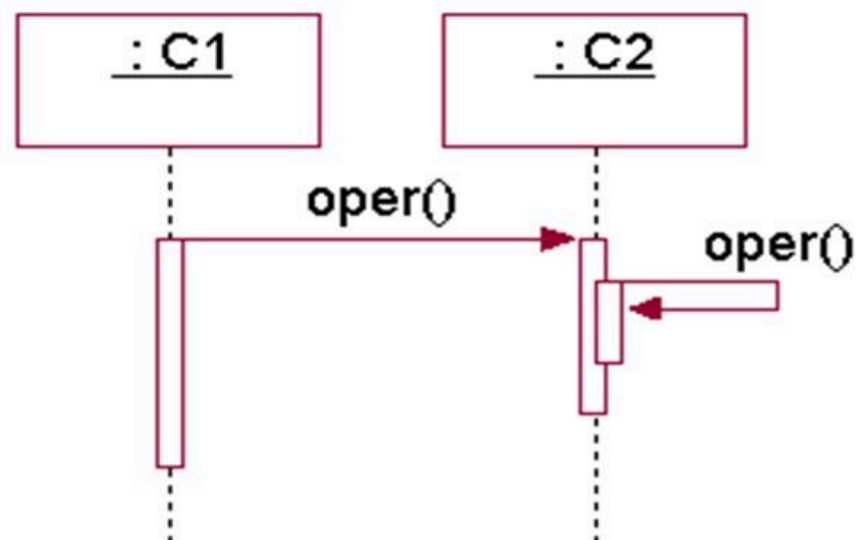
**激活期(activation)**：表示对象执行一个动作的期间（直接操作或者通过下级操作），也即对象激活的时间段。

**控制焦点和激活期**是同一个概念。



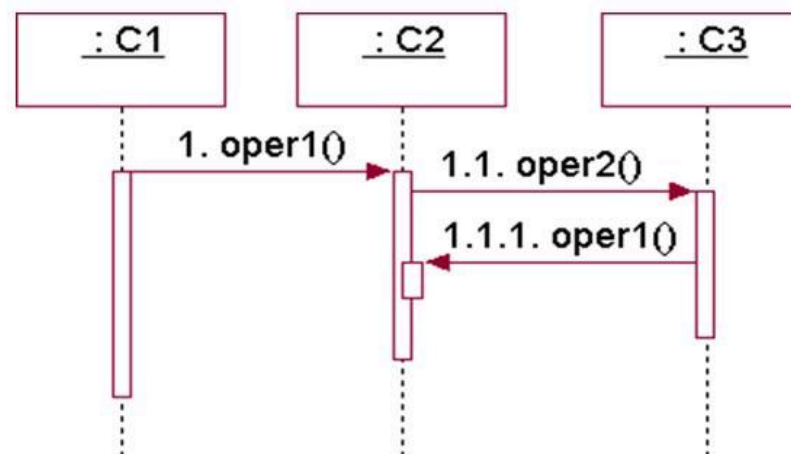
# 顺序图常见问题分析

- 顺序图中递归的表示
- 利用嵌套的FOC表示
- 例1. 单个对象自身的递归。



# 顺序图常见问题分析

- 顺序图中递归的表示
  - 利用嵌套的FOC表示
- 例2. 多个对象间相互递归调用的表示。



# 顺序图的作用

---

- 帮助分析人员对照检查用例中描述需求，是否已经落实给具体对象去实现
- 提醒分析人员去补充遗漏的对象类或操作
- 帮助分析人员识别哪些对象是主动对象
- 通过对一个特定的对象群体的动态方面建模，深入地理解对象之间的交互

单选题 1分

顺序图中带条件消息的发送，不能采用以下哪种方式进行？

- ☐ A 用文字说明，作为注释添加
- ☐ B 添加条件控制框
- ☒ C 分成多个顺序图子图来描述
- ☐ D 在消息名字前加条件子句

单选题 1分

顺序图的组合控制框 (Frame) 中, 用于表达多个分支选择关系的控制符是哪个?

- ☐ A opt
- ☒ B alt
- ☐ C ref
- ☐ D loop

单选题 1分

当一个顺序图过大时，最好的处理方法是？

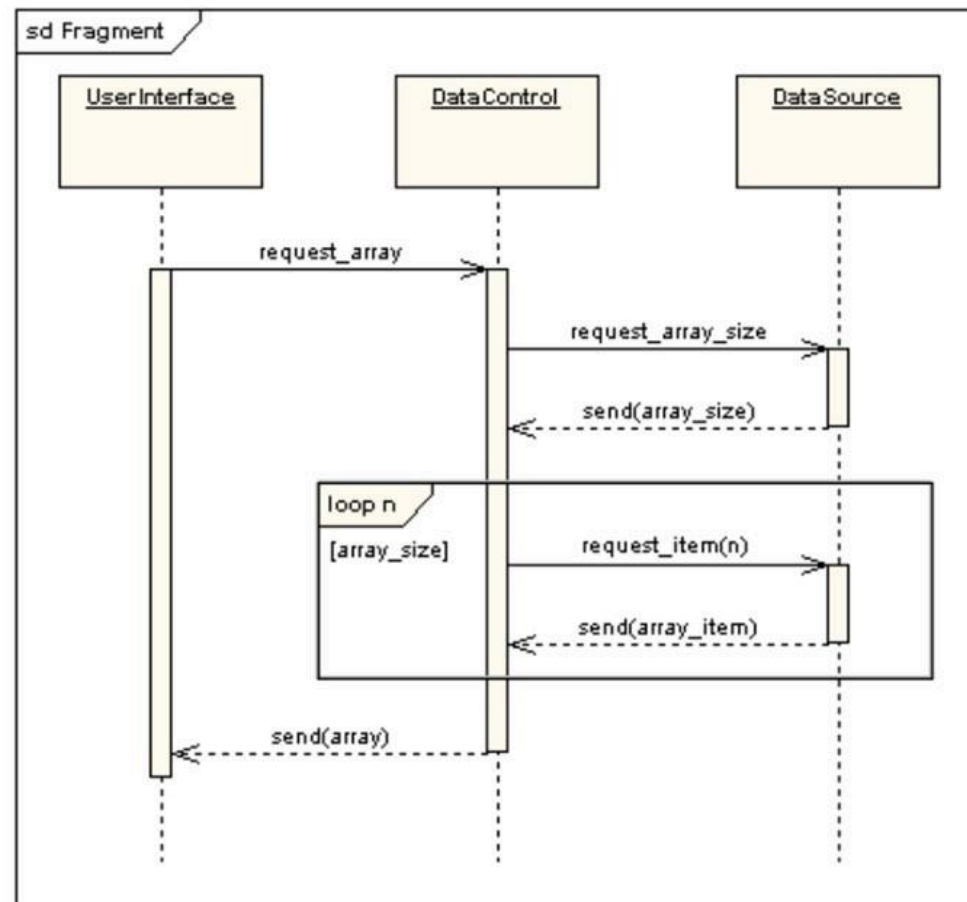
- ☐ A 添加注释，说明顺序图的各部分之间的接续关系
- ☐ B 用不完整的箭头，指明本页的顺序图未完待续
- ☐ C 去掉无关的细节，保持图的简洁性
- ☒ D 添加ref框，建立顺序图间的引用关系

单选题 1分

对顺序图与用例之间关系的阐述，错误的是：

- A** 顺序图表达单个情景实例的行为，每个用例对应一个顺序图
- B** 用例分析阶段的顺序图要包含设计对象，并关注消息参数
- C** 顺序图用于表示为完成用例而在系统边界输入输出的数据以及消息
- D** 顺序图可帮助分析人员对用例图进行扩展、细化和补遗





包含哪个结构？

它的控制逻辑  
是什么？

单选题 1分

该顺序图中包含哪个结构?

- ☐ A 分支选择结构
- ☐ B 并发结构
- ☐ C 单条件结构
- ☐ D 循环结构

单选题 1分

以下哪个说法是正确的？

- ☐ A 对于参数，优先考虑使用参数类型而不是参数名
- ☒ B 每个用例对应一个顺序图
- ☐ C 在顺序图过程中不能创建对象
- ☐ D 顺序图应该用于设计阶段而不是需求分析阶段