



Object Oriented Analysis & Design

面向对象分析与设计

Lecture_02 建模工具UML

主讲: 姜宁康 博士



■ 5、顺序图...

- 软件设计的强力武器!
- 是一种动态视图

5.1 问题的提出?

- **用例模型、活动图**

- 明确了用户的“需求”

- **类图（类模型）**

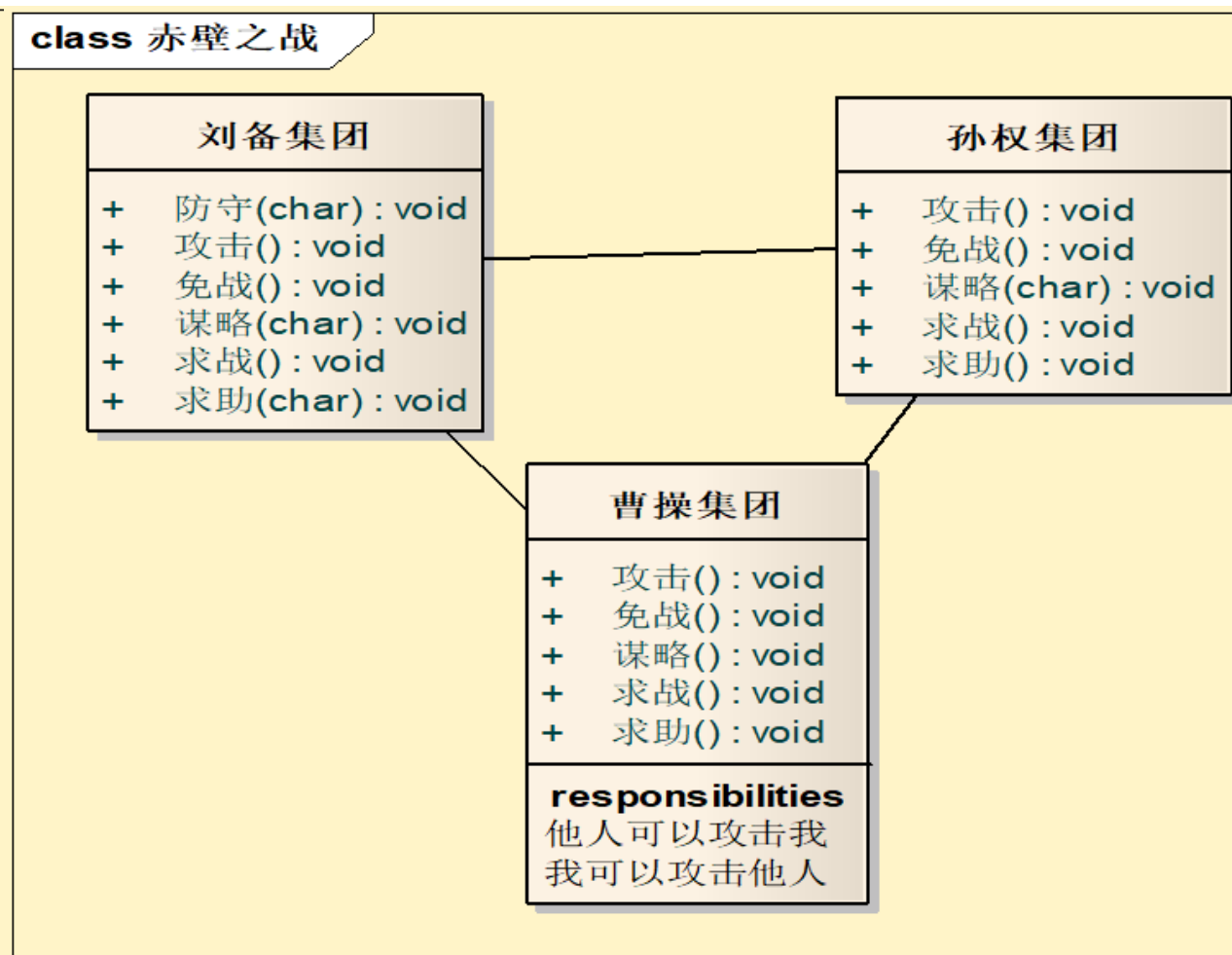
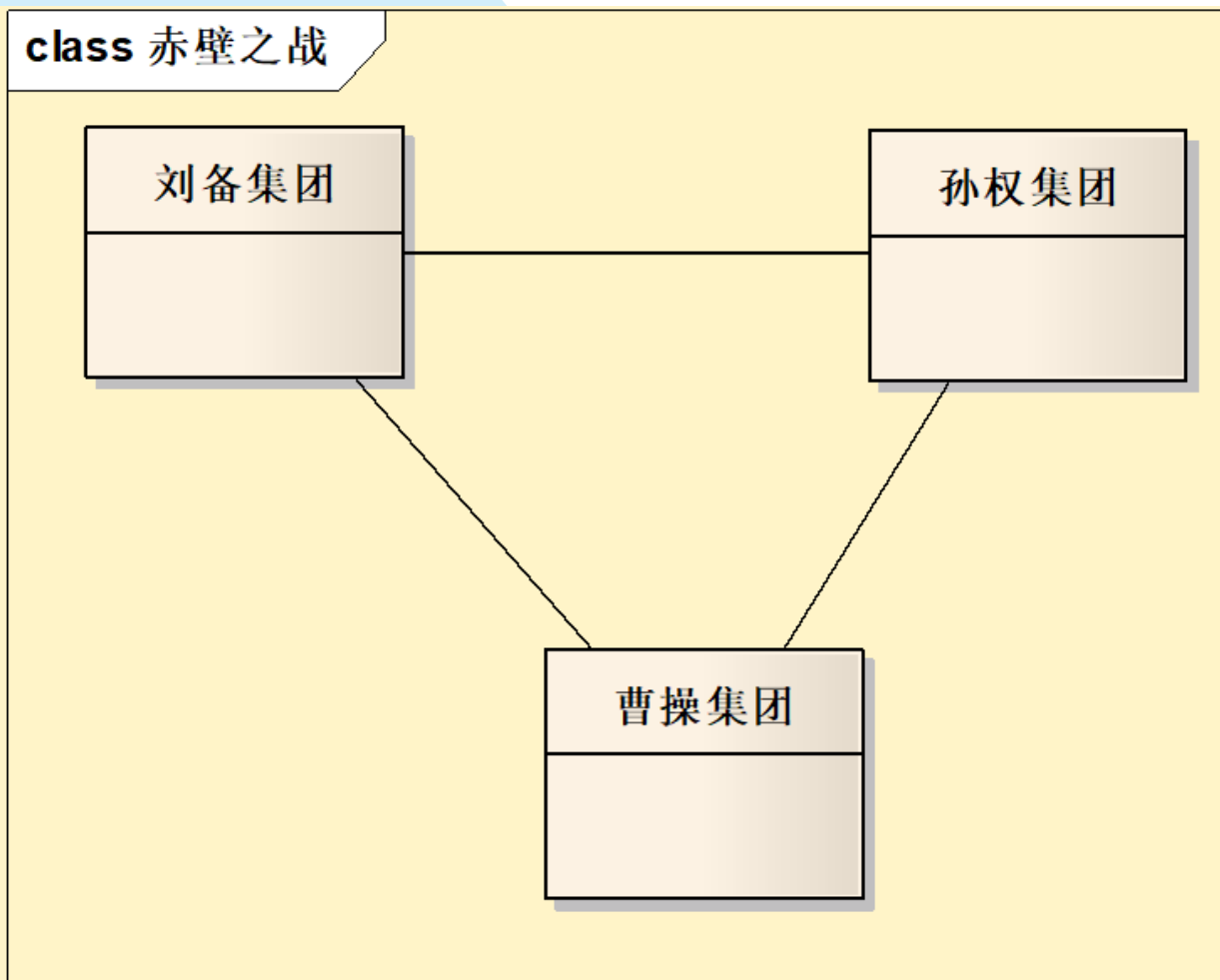
- 给出了解决问题的方案
- 但这个方案是静态的

- **问题**

- 如何在编写代码之前，检验一下上述方案是否能够实现所需要的功能？

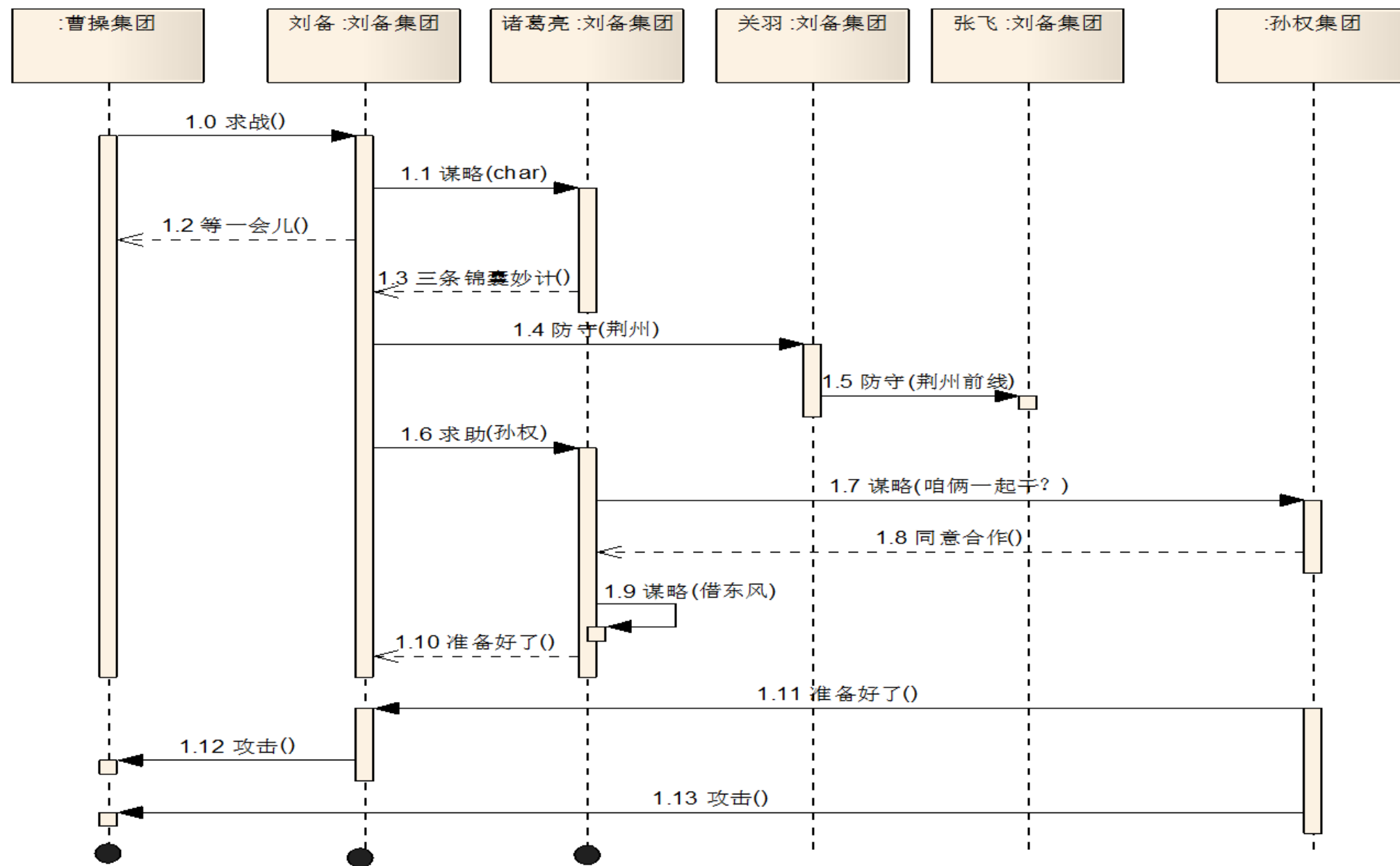
案例场景：赤壁之战

- “火烧赤壁”是如何发生的？
- 赤壁之战是指三国形成时期，孙权、刘备联军于汉献帝建安十三年（208）在长江赤壁（今湖北赤壁西北）一带大胜曹操军队，奠定三国鼎立基础的著名战役



Scenario: “火烧赤壁”

sd 赤壁之战_General



这就是
顺序图
交互图
的一种

5.2 相关概念

■ 交互

- 对象之间为实现某一功能而必须实施的**协作过程**、**动态行为**，称为交互

■ 消息

- 对象间的协作与交流表现为一个对象以某种方式启动另一个对象的活动，这种交流在UML里被定义为消息

■ 顺序图的建模元素

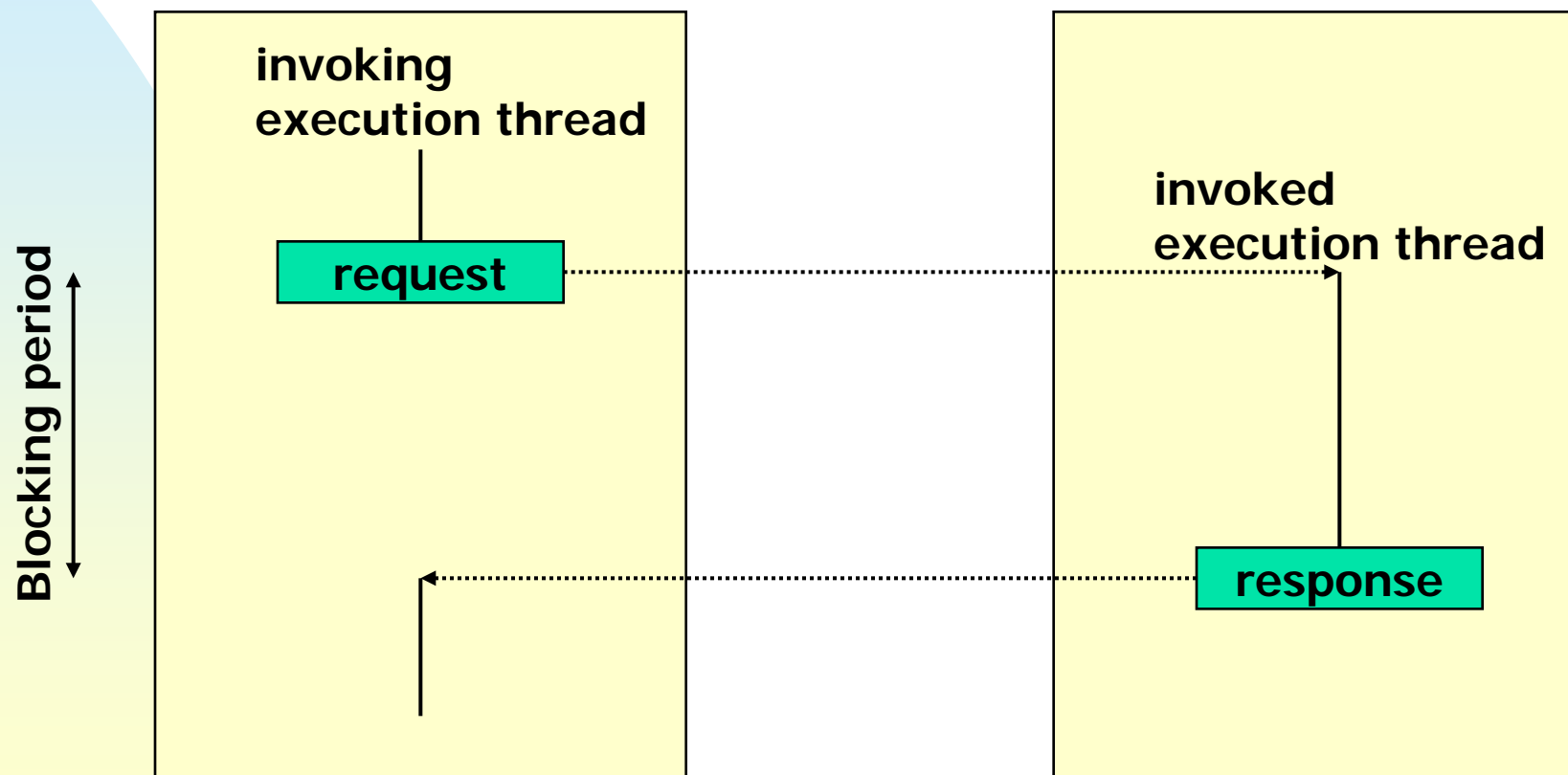
- 对象或角色(*role*)
- 参与者
- 消息

■ 交互图

- 交互的过程表现出来，就是交互图
- 顺序图、通信图，都是交互图

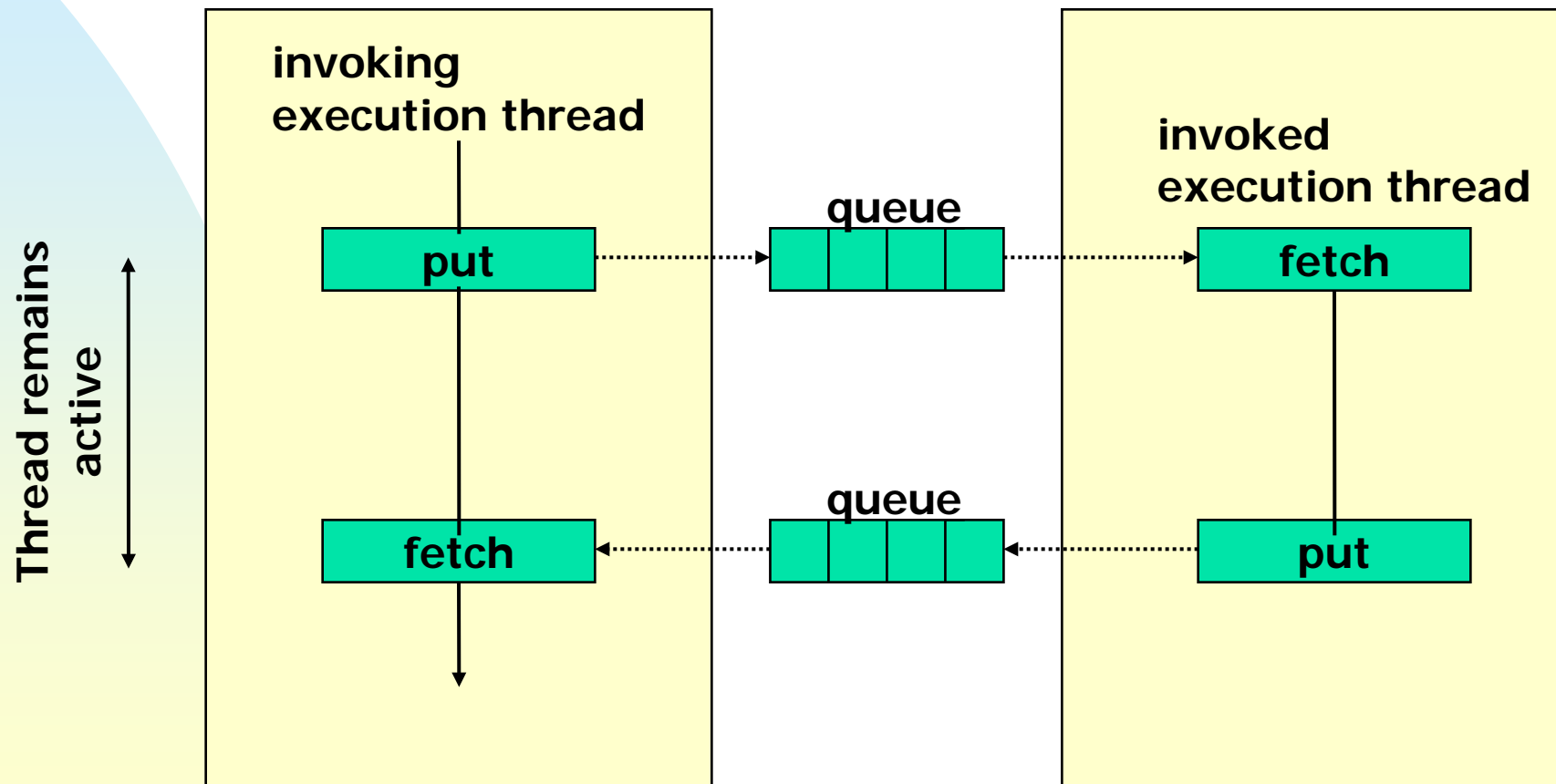
5.3 交互的基本概念 -- 消息

同步消息的逻辑意义



5.3 交互的基本概念 -- 消息

异步消息的逻辑意义

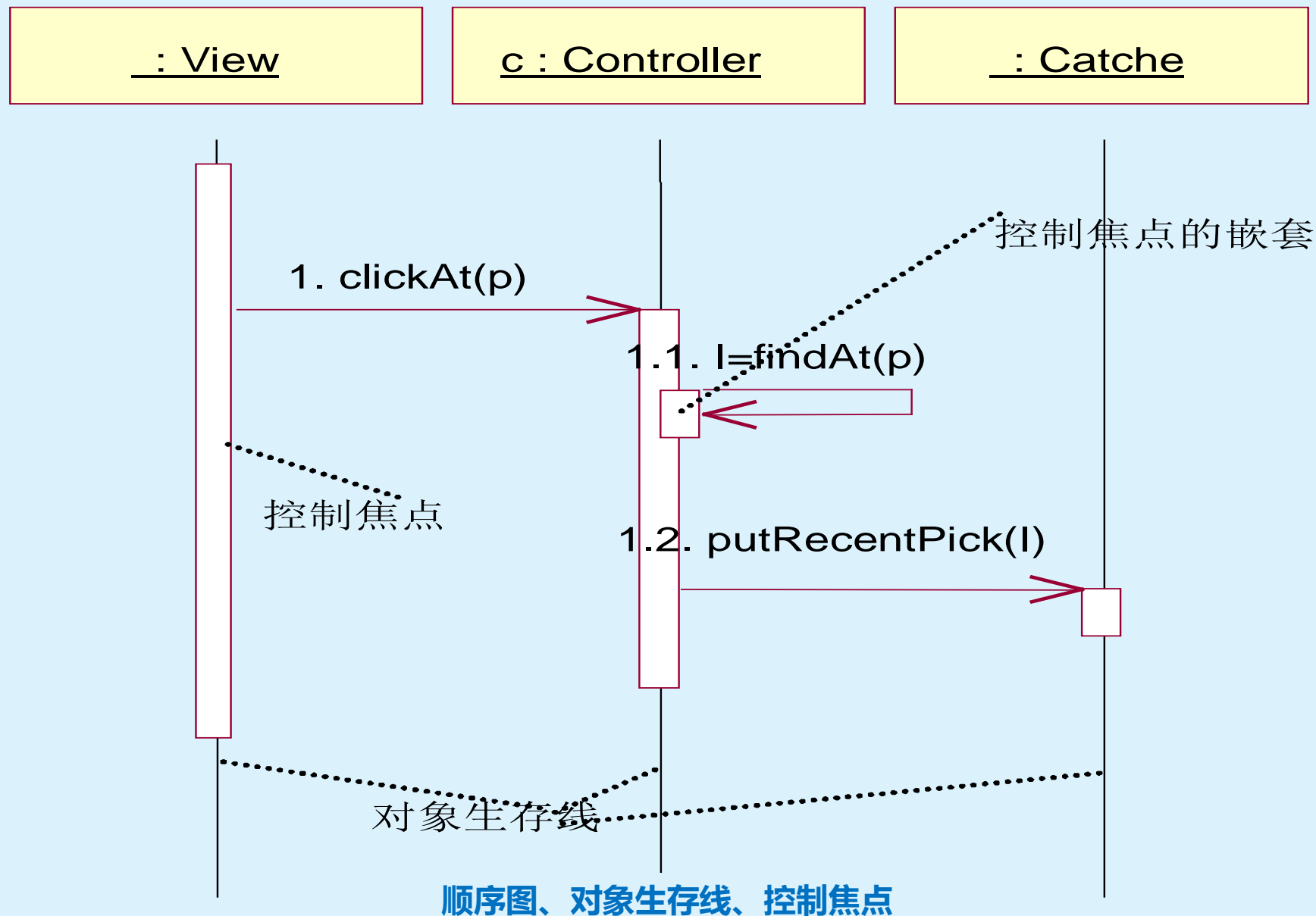


5.4 顺序图：交互图的一种

顺序图，基于类图，与类图相互校验类设计方案真的可行吗？

基本建模元素：

对象
生命期
控制焦点
消息 == 方法调用
消息序号（可选）



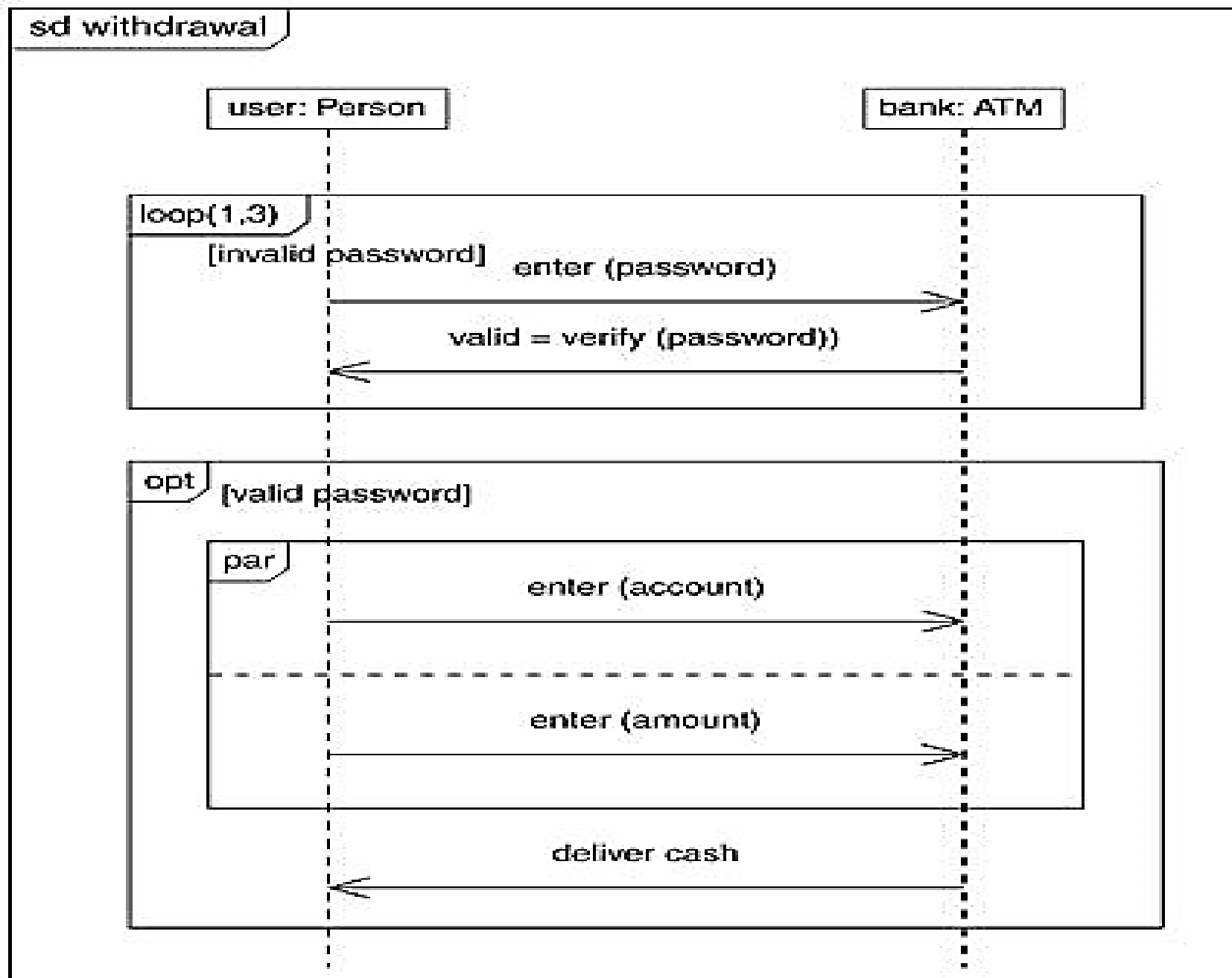
5.4 顺序图

- 顺序图中的结构化控制，又称：Frame, Fragment

- 类型

- 可选执行 (标签: opt)
- 条件执行 (标签: alt)
- 并行执行 (标签: par)
- 循环 (迭代) 执行 (标签: loop)

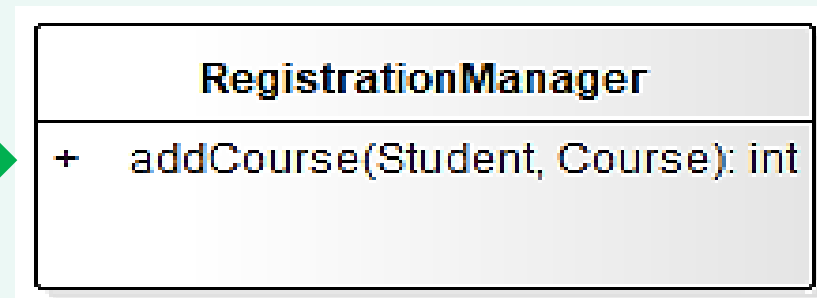
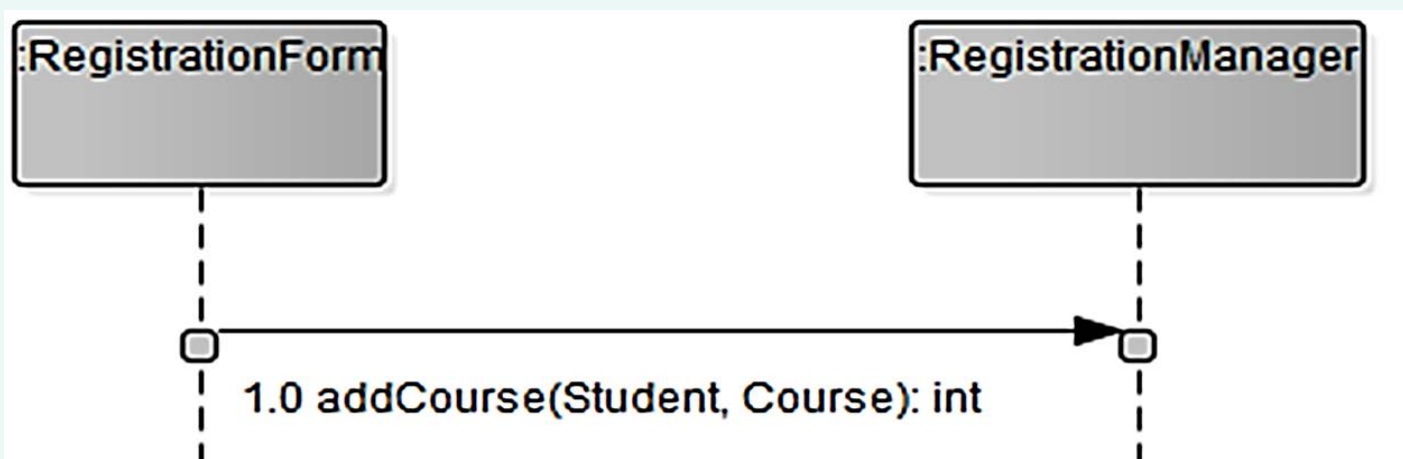
Figure 19-3. Structured Control Operators



5.4 顺序图

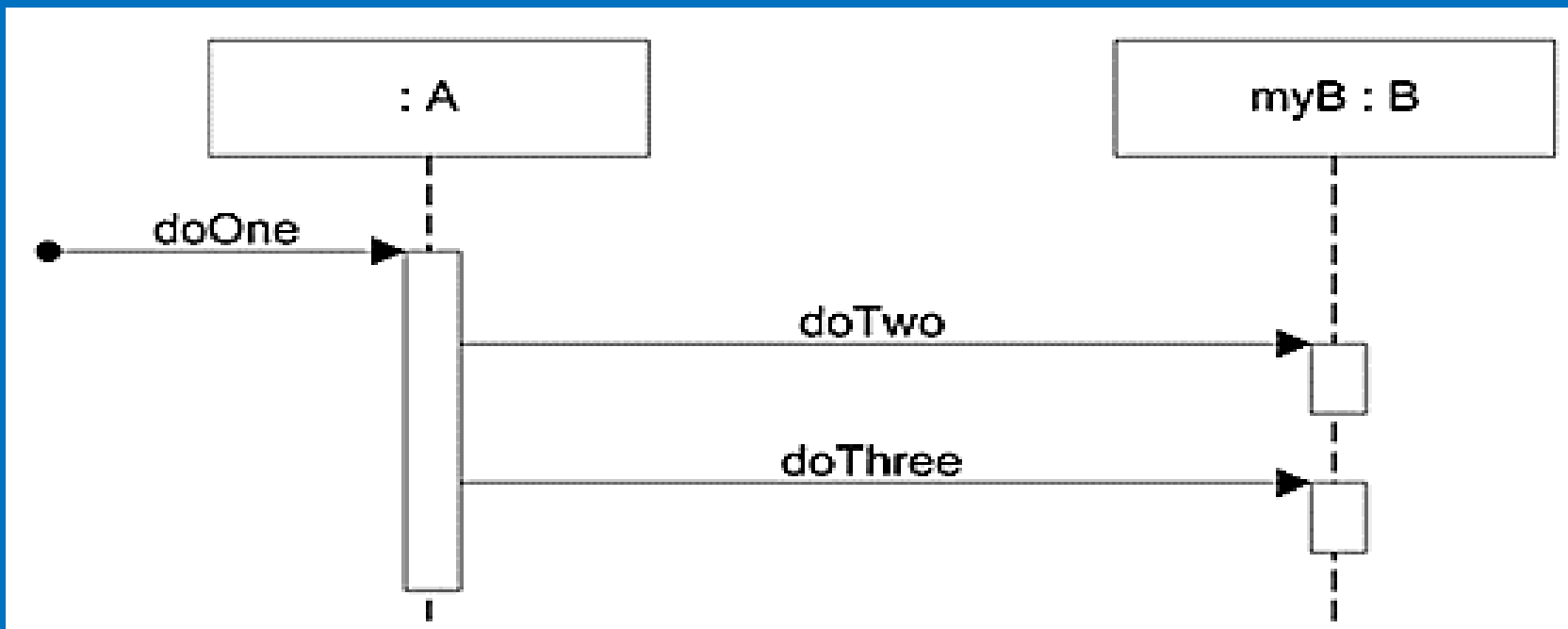
■ 强调一下：关于操作 (Operations)

- The behavior of a class is represented by its operations 一个类的行为/服务，是通过操作来表示的
- Operations may be found by examining interaction diagrams 在交互图里面可以看到对象在执行这些操作
- 先在类中定义操作，然后在顺序图中为对象选择操作！



问题： addCourse()操作是由哪个对象负责实施的？

5.5 代码和顺序图的映射



```
public class A {  
    private B myB = new B();  
    public void doOne() {  
        myB.doTwo();  
        myB.doThree();  
    } //  
    ...  
}
```

正向工程
Forward engineering:
从模型生成代码

逆向工程
Reverse engineering:
从代码生成模型

顺序图小节

- 顺序图可以动态验证类模型的可行性
- 顺序验证的某一功能，属于某个用例描述的功能中的一部分
 - 又被称为用例实现 “usecase realization”
- 顺序图从上到下，反映了个对象相互协作的时间顺序！





■ **本讲结束**