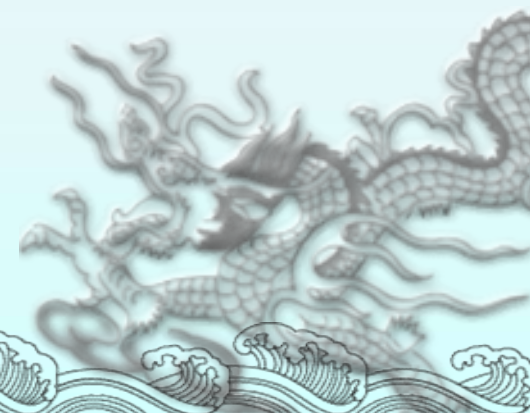


协作图和顺序图



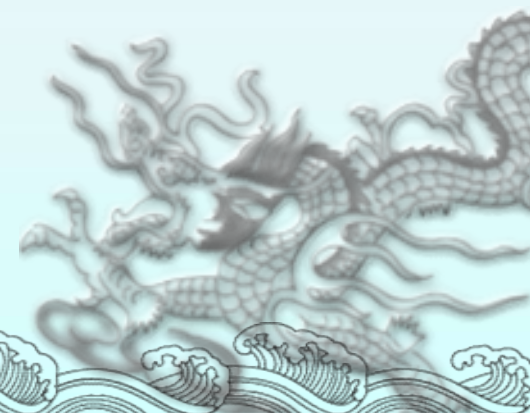
内容提纲

- 交互图
- 协作图
 - 相关概念
 - 实例讲解
- 顺序图
 - 相关概念
 - 实例讲解



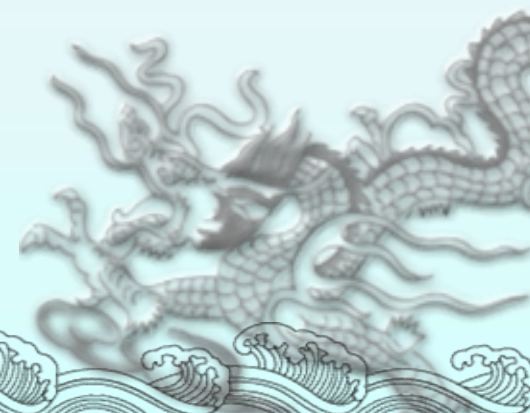
交互图

- ◆ 在一个待开发的系统中，任何对象都不是孤立存在的，如果对象独立则失去存在的意义，系统中对象之间通过消息的传递来进行交互。因此在设计系统时，需要使用交互图对对象之间的交互进行建模。



交互图

- ◆ 一次交互是指在特定语境下，为了实现某一个目标，而在一组对象之间进行交换的一组消息所表示的行为。通常，消息的传递通过方法的调用来实现。



交互图

- ◆ 交互图(Interactive Diagram)是表示各对象如何依据某种行为进行协作的模型。
- ◆ 通常可以使用一个交互图来表示和说明一个用例的行为。
- ◆ 在 UML 中，协作图、顺序图等属于交互图。

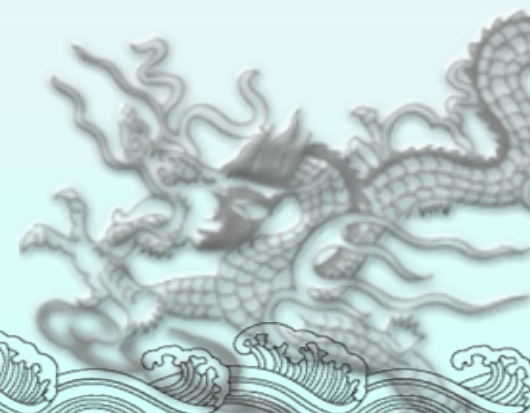


交互图

- ◆ 协作图（通信图）：强调收发信息的对象的结构组织的交互图，通信图展示了一组角色、这些角色间的连接件以及由扮演这些角色的实例所收发发的消息。
- ◆ 顺序图（序列图）：强调消息的时间次序的交互图，顺序图展示了一组角色和由扮演这些角色的实例发送和接收的消息。
- ◆ 定时图：展现消息跨越不同对象或角色的实际时间，而不仅仅关心消息的相对顺序。
- ◆ 交互概览图：活动图和顺序图的混合物，可以对活动图中的一些关键的、复杂度不高的活动节点进行细化，用顺序图来表示其对象间的控制流。

内容提纲

- 交互图
- 协作图
 - 相关概念
 - 实例讲解
- 顺序图
 - 相关概念
 - 实例讲解


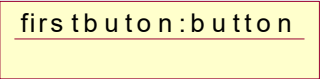



协作图

1 概要


- ◆ 协作图是一种交互图，强调的是发送和接收消息的对象之间的组织结构，使用协作图来说明系统的动态情况。
- ◆ 协作图主要描述协作对象间的交互和链接，显示对象、对象间的链接以及对象间如何发送消息。
- ◆ 协作图可以表示类操作的实现。

2 协作图中的事物及解释

事物名称	解释	图
参与者	发出主动操作的对象，负责发送初始消息，启动一个操作。	
对象	对象是类的实例，负责发送和接收消息，与顺序图中的符号相同，冒号前为对象名，冒号后为类名。	
消息流 (由箭头和标签组成)	箭头指示消息的流向，从消息的发出者指向接收者。标签对消息作说明，其中，顺序号指出消息的发生顺序，并且指明了消息的嵌套关系；冒号后面是消息的名字。	

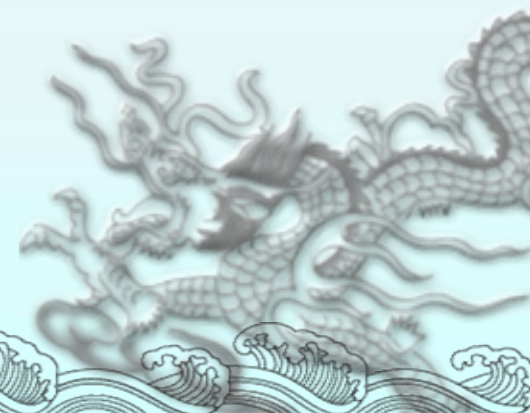
协作图

3 协作图中的关系及解释

关系名称	解释	关系实例
链接	用线条来表示链接，链接表示两个对象共享一个消息，位于对象之间或参与者与对象之间	

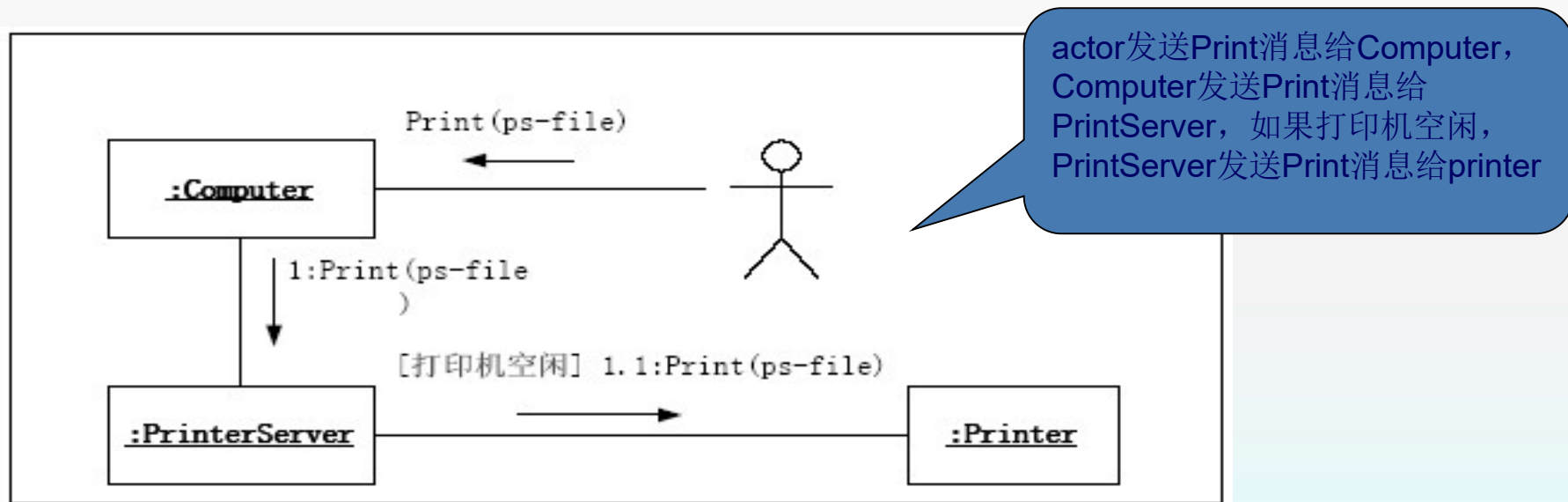
内容提纲

- 交互图
- 协作图
 - 相关概念
 - 实例讲解
- 顺序图
 - 相关概念
 - 实例讲解



协作图例子

1. 打印操作的协作图



协作图例子

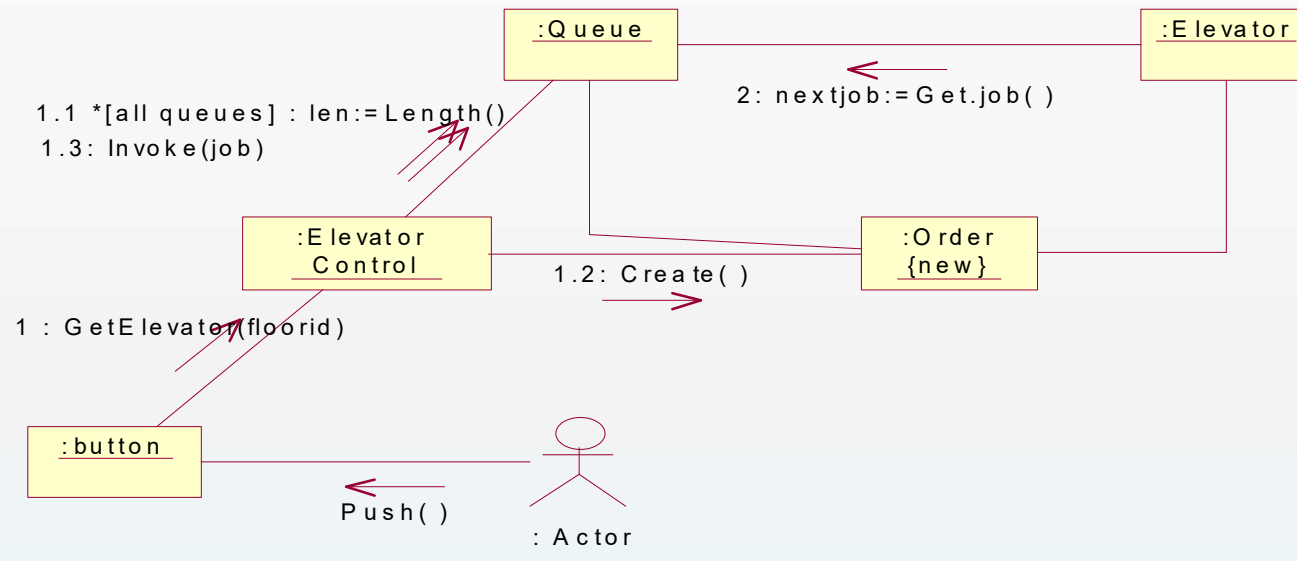
2. 乘坐电梯的协作图

参与者需要乘坐电梯，他从系统外部按下按钮，让电梯到达他想去的楼层。此时，电梯系统的操作被启动，电梯控制对象以循环的方式检查所有的电梯，从中选择一个工作队列长度最短的。

然后，它创建一个作业命令，并将该命令放入对应电梯的工作队列，接着激活队列。电梯对象并发运行，从它的队列中选择一个作业并执行。电梯是一个活动对象，它与它的控制线程并发执行。

协作图例子

2. 乘坐电梯的协作图



图中存在的事物有：
参与者

按钮对象
电梯控制对象
命令对象
工作队列
电梯对象

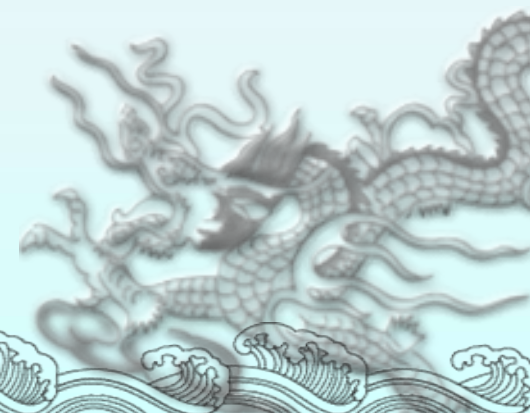
图中存在的关系有：
链接

参与者需要乘坐电梯，他从系统外部按下按钮，让电梯到达他想去的楼层。此时，电梯系统的操作被启动，电梯控制对象以循环的方式检查所有的电梯，从中选择一个工作队列长度最短的。

然后，它创建一个作业命令，并将该命令放入对应电梯的工作队列，接着激活队列。电梯对象并发运行，从它的队列中选择一个作业并执行。电梯是一个活动对象，它与它的控制线程并发执行。

内容提纲

- 交互图
- 协作图
 - 相关概念
 - 实例讲解
- 顺序图
 - 相关概念
 - 实例讲解



顺序图

1 概要

- ◇顺序图用来表示用例中的行为顺序。当执行一个用例行为时，顺序图中的每条消息对应了一个类操作或状态机中引起转换的事件。
- ◇顺序图展示对象之间的交互，这些交互是指在场景或用例的事件流中发生的。顺序图属于动态建模。
- ◇顺序图的重点在消息序列上，也就是说，描述消息是如何在对象间发送和接收的。表示了对对象之间传送消息的时间顺序。
- ◇浏览顺序图的方法是：从上到下查看对象间交换的消息。

顺序图

2 顺序图中的事物及解释

事物名称	解释	图
参与者	与系统、子系统或类发生交互作用的外部用户(参见用例图定义)。	
对象	顺序图的横轴上是与序列有关的对象。对象的表示方法是：矩形框中写有对象或类名，且名字下面有下划线。	
生命线	坐标轴纵向的虚线表示对象在序列中的执行情况(即发送和接收的消息，对象的活动)这条虚线称为对象的“生命线”。	
消息符号	消息用从一个对象的生命线到另一个对象生命线的箭头表示。箭头以时间顺序在图中从上到下排列。	

顺序图

◆ 消息符号

- ◆ **同步消息**：一个对象向另一个对象发出同步消息后，将处于阻塞状态，一直等到另一个对象的回应



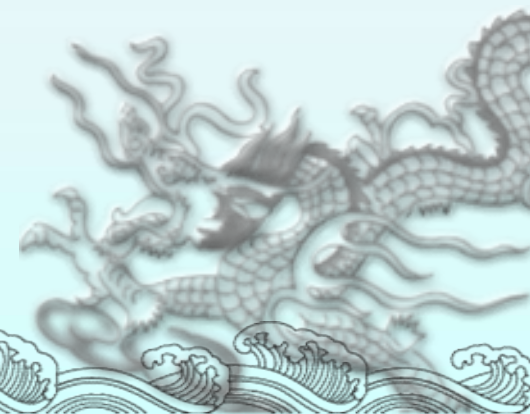
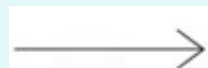
- ◆ **异步消息**：一个对象向另一个对象发出异步消息后，这个对象可以进行其他的操作，不需要等到另一个对象的响应



- ◆ **返回消息**：表示从过程调用返回



- ◆ **简单消息**：不区分同步或异步



顺序图

◆ 顺序图定义

- ◆ 顺序图是一种**强调信息时序**的交互图，又称为顺序图或序列图。顺序图供不同类型的用户使用：用户可以从顺序图中看到业务过程的细节；分析人员可以从顺序图中看到处理流程；开发人员可以看到需要开发的对象和对这些对象的操作；质量保证工程师可以看到过程的细节，并根据这个过程开发测试案例，顺序图对项目的各方面人员都有用。
- ◆ 顺序图由执行者（actor）、对象（object）、消息（message）、生命线（lifeline）、激活（activation）等组成。顺序图描述对象之间的动态交互关系，**着重体现对象间消息传递的时间顺序**。

顺序图

◆ 顺序图定义

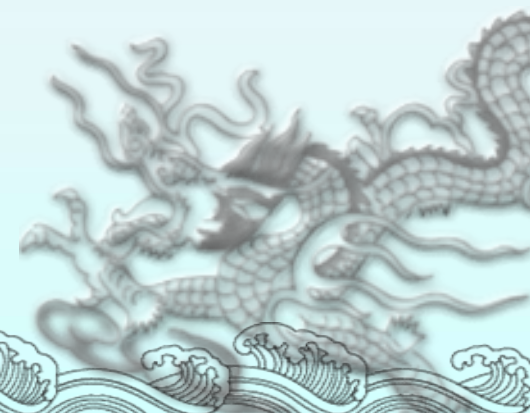
- ◆ UML顺序图一般用于确认和丰富一个使用情境的逻辑。
- ◆ 一个使用情境的逻辑或是一个用例的一部分；或是一条扩展路径；或是一个贯穿单个用例的完整路径，例如动作基本过程的逻辑描述；或是动作的基本过程的一部分再加上一个或多个的备用情境的逻辑描述；或是包含在几个用例中的路径。



顺序图

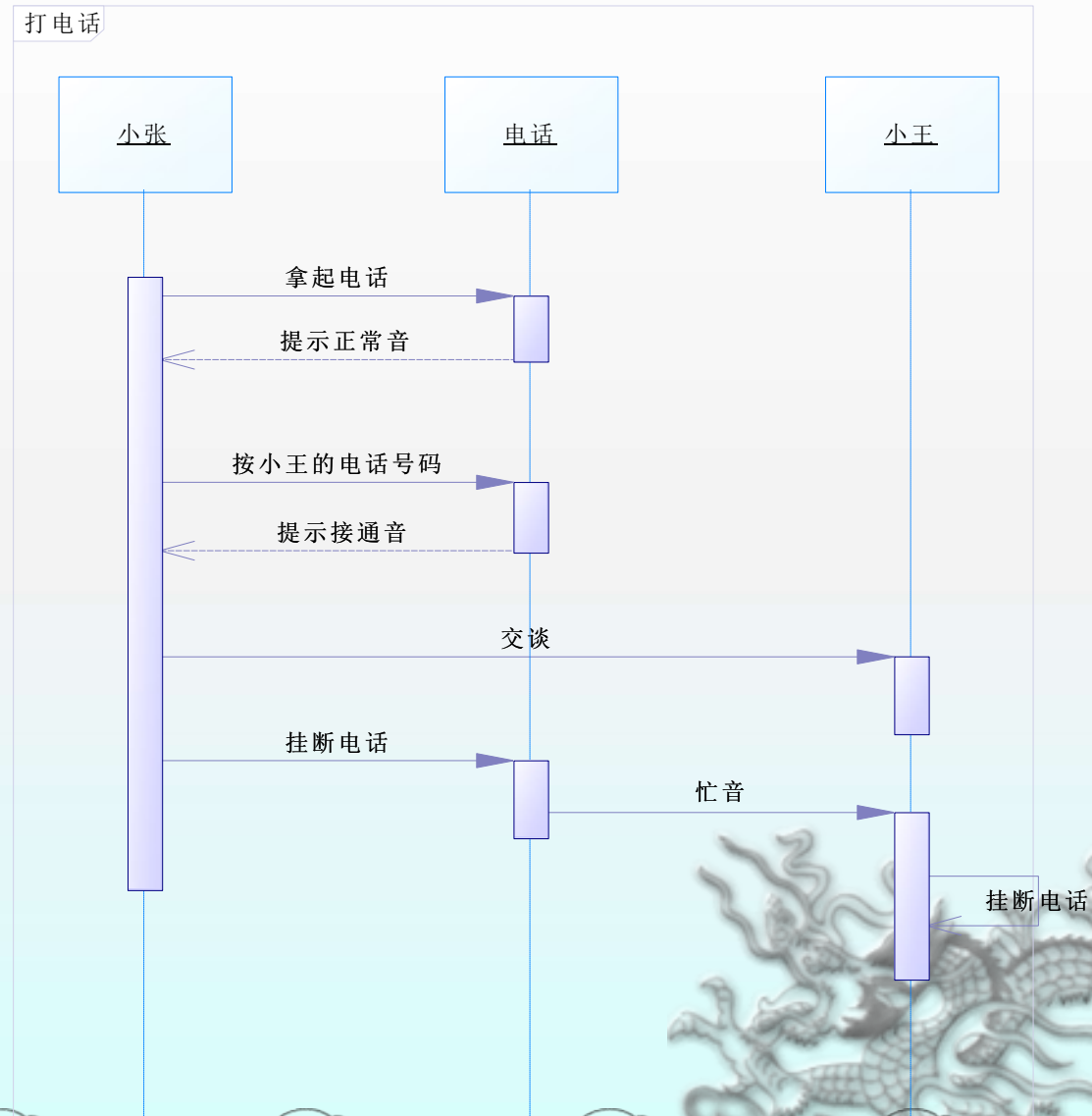
◆ 顺序图定义

- ◆ 顺序图将交互关系表现为一个二维图，纵向是时间轴，时间沿竖线向下延伸。横向轴代表了在协作中各独立对象的类元角色，类元角色的活动用生命线表示。



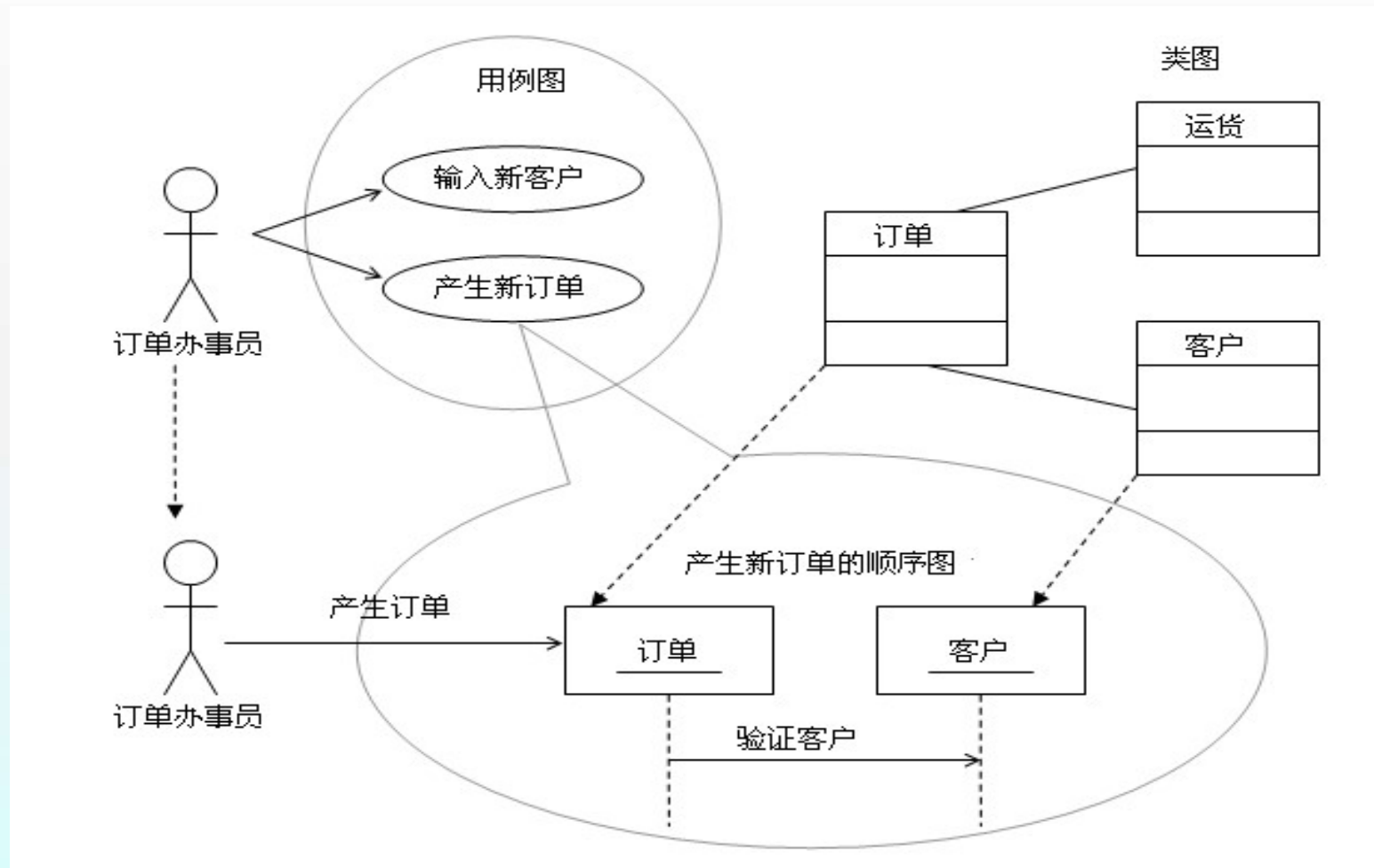
顺序图

◇ 顺序图示例



顺序图

顺序图与用例图和类图的关系



顺序图

- ◆ 顺序图组成元素

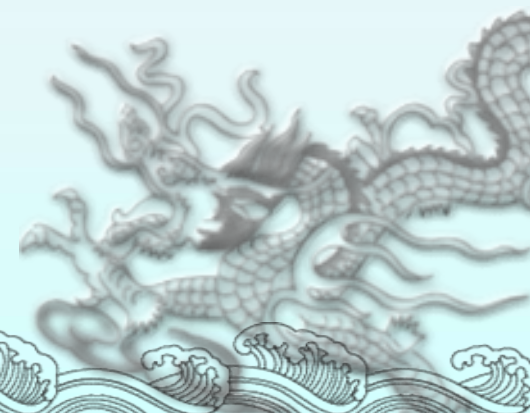
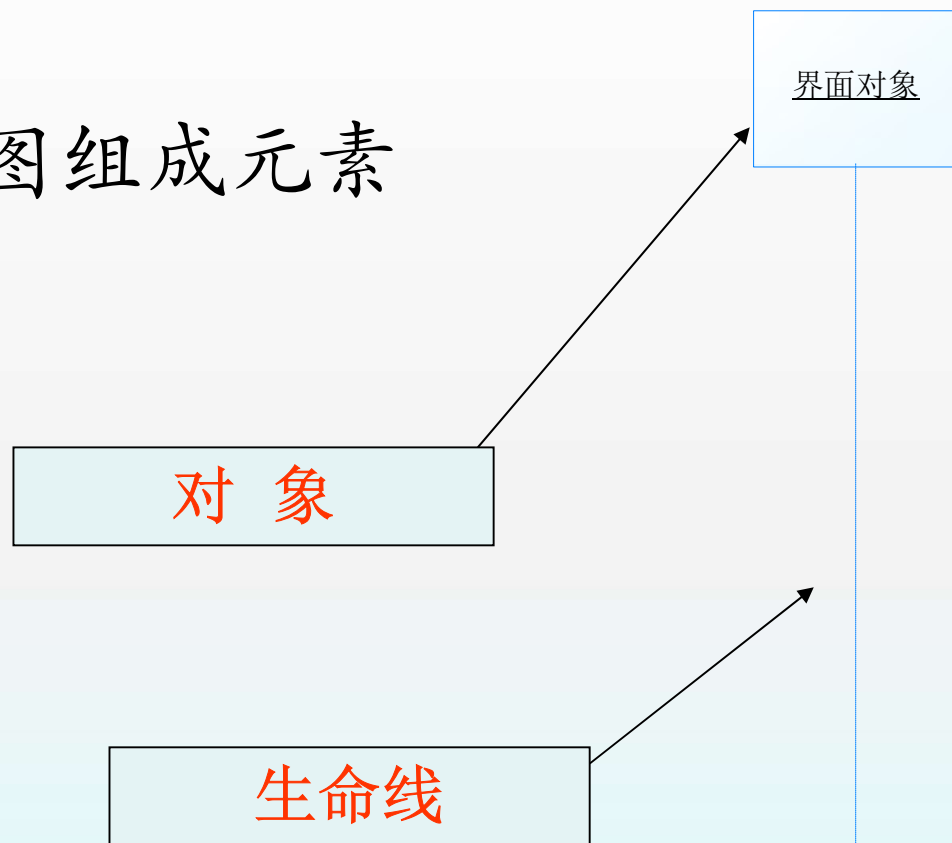
- ◆ 生命线用一条纵向虚线表示。

- ◆ 在UML中，对象表示为一个矩形，其中对象名称标有下划线。



顺序图

◆ 顺序图组成元素



顺序图

◇ 顺序图组成元素

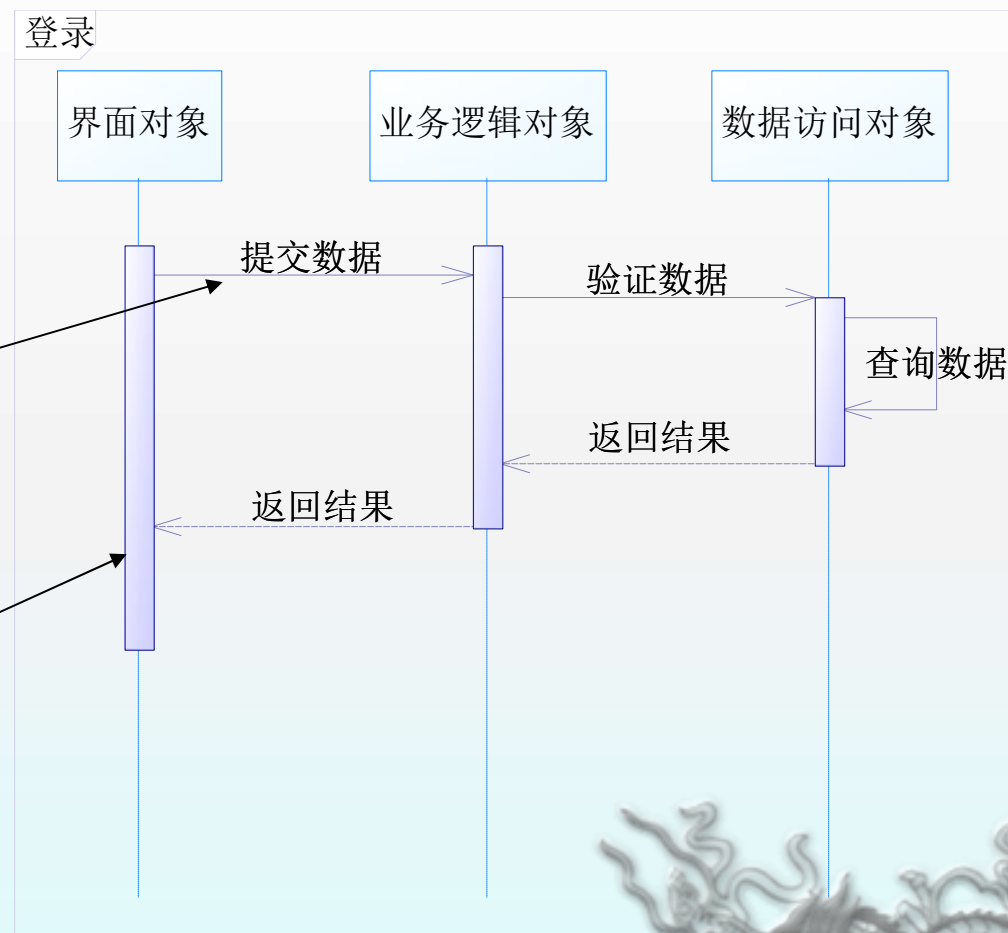
- ◇ 激活是过程的执行，包括等待过程执行的时间。在顺序图中激活部分替换生命线，使用长条的矩形表示。
- ◇ 消息是对象之间的通信，是两个对象之间的单路通信，是从发送者到接收者之间的控制信息流。消息在顺序图中由有标记的箭头表示，箭头从一个对象的生命线指向另一个对象的生命线，消息按时间顺序在图中从上到下排列。

顺序图

◆ 顺序图组成元素

消息

激活



顺序图

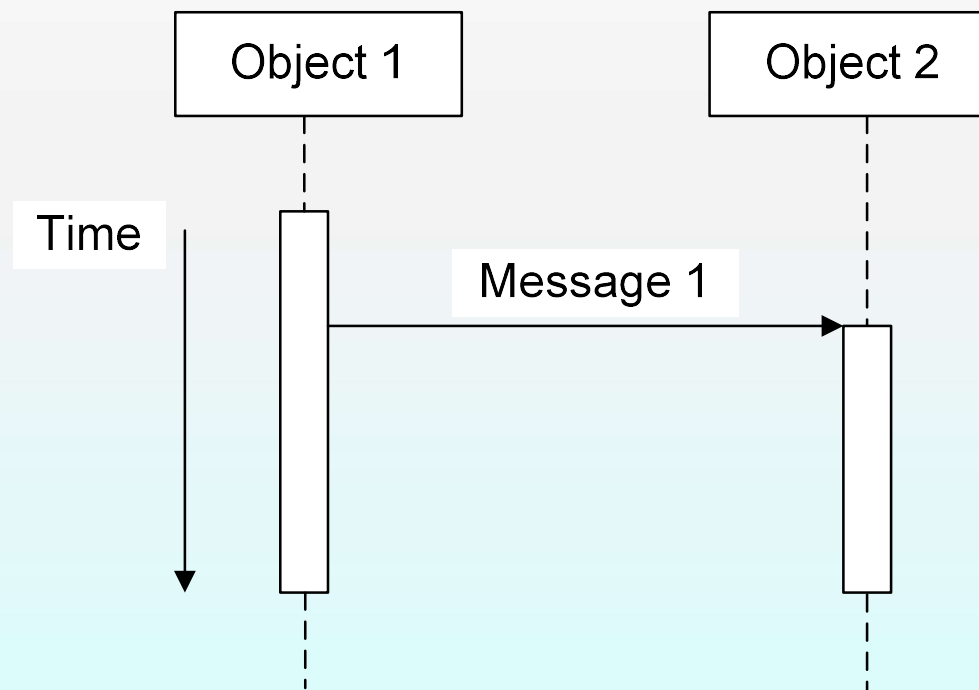
◆ 顺序图组成元素

- ◆ 在顺序图中，对象安排在X轴。启动交互的对象放在最左边，随后放入消息的对象放在启动交互对象的右边。交互中对象发送和接收的消息沿着Y轴以时间增加的次序放置。在顺序图中，有的消息对应于激活，表示它将会激活一个对象，这种消息称为调用消息(Call Message)；如果消息没有对应激活框，表示它不是一个调用消息，不会引发其他对象的活动，这种消息称为发送消息(Send Message)。

顺序图

◆ 顺序图组成元素

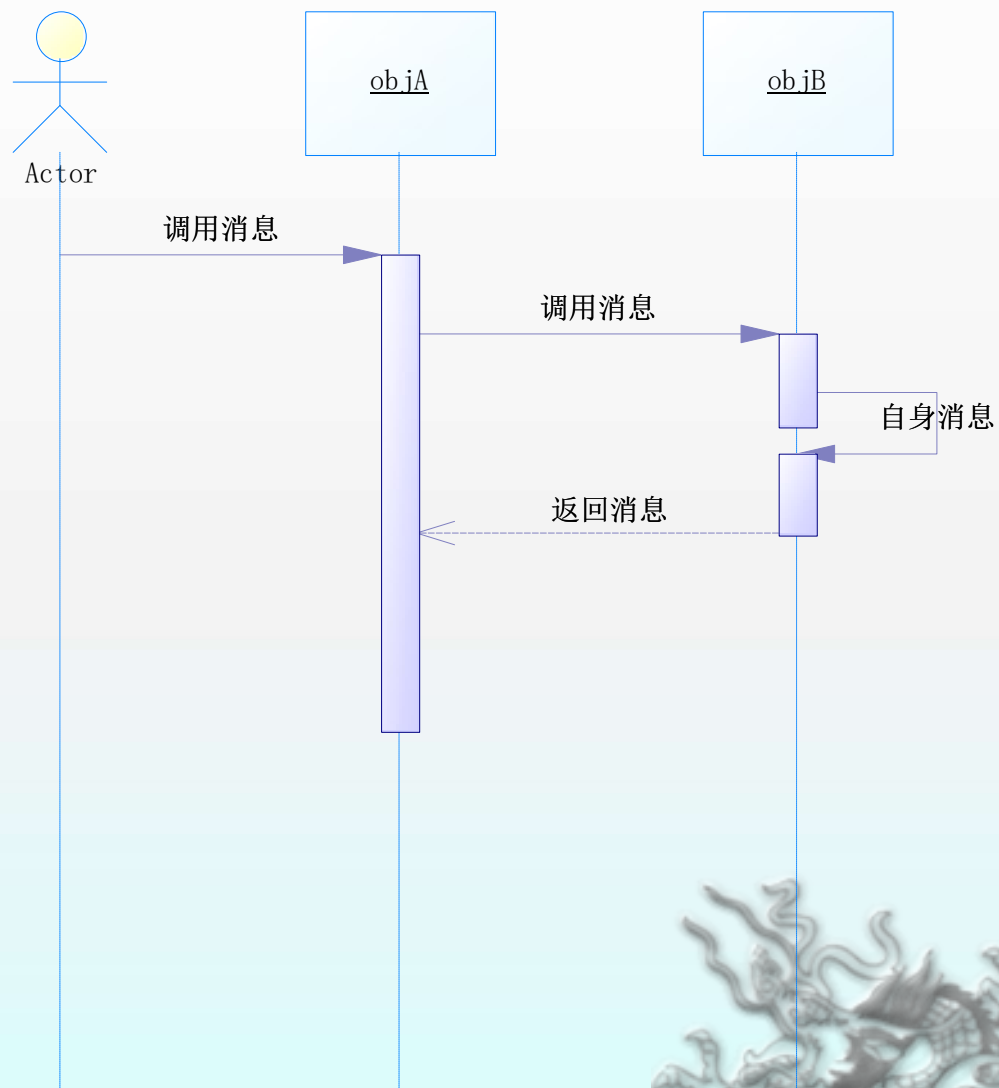
◆ 下图显示顺序图中对象和消息的安排：



顺序图

◆ 消息

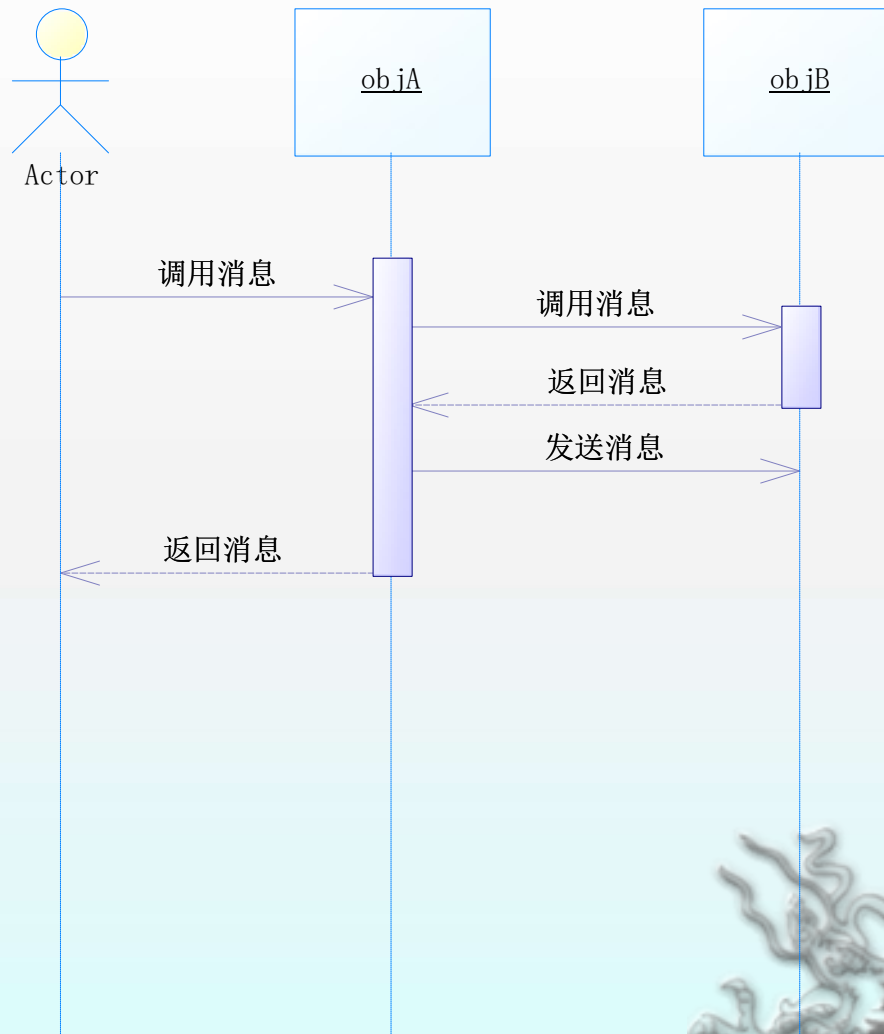
- ◆ 调用消息
- ◆ 返回消息
- ◆ 自身消息



顺序图

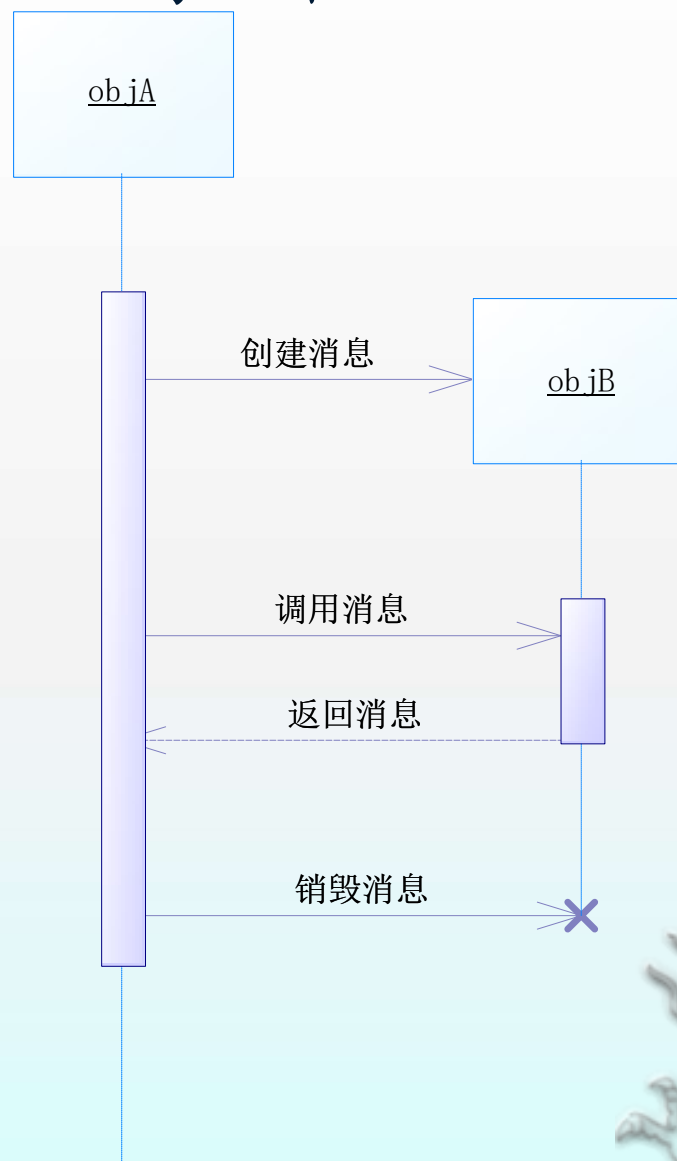
◆ 消息

- ◆ 调用消息
- ◆ 发送消息



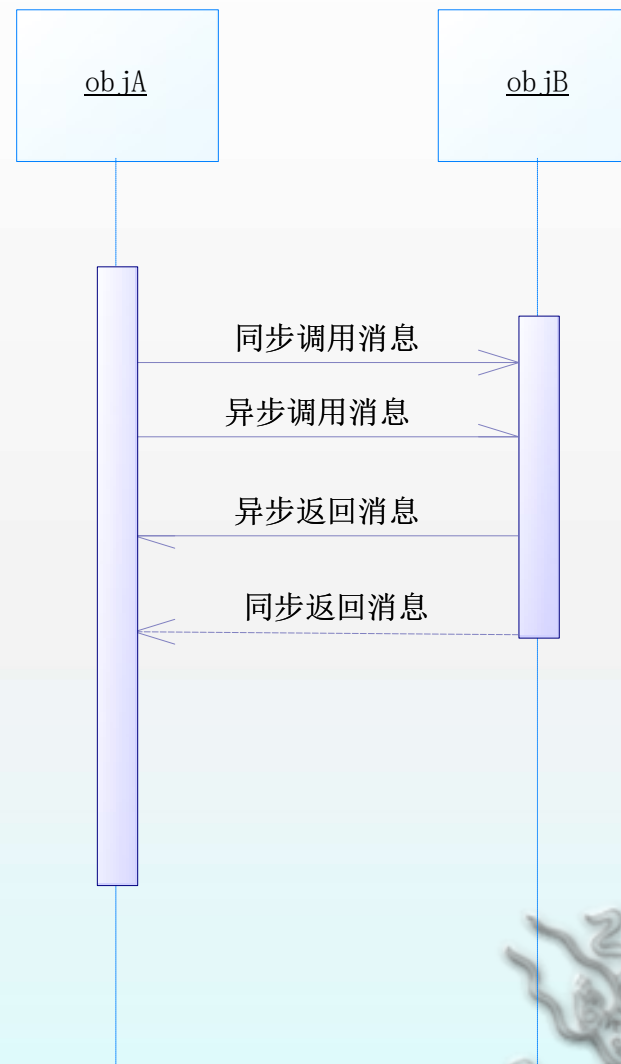
顺序图

- ◆ 消息
 - ◆ 创建消息
 - ◆ 销毁消息



顺序图

- ◆ 消息
 - ◆ 同步消息
 - ◆ 异步消息

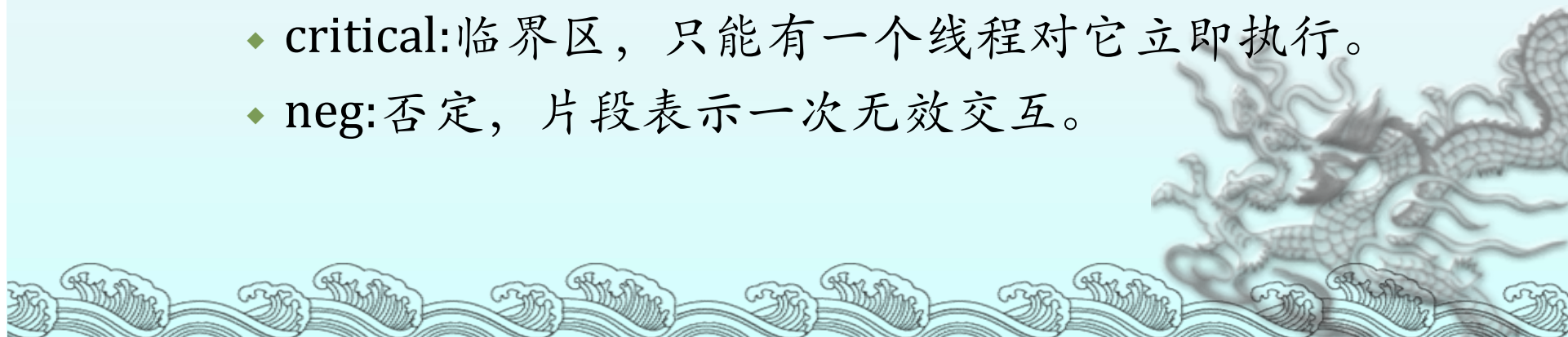


顺序图

◆ 交互片段

◆ 一个复杂的顺序图可以划分为几个小块，每一个小块称为一个交互片段。每个交互片段由一个大方框包围，其名称显示在方框左上角的间隔区内，表示该顺序图的信息。常用操作符如下：

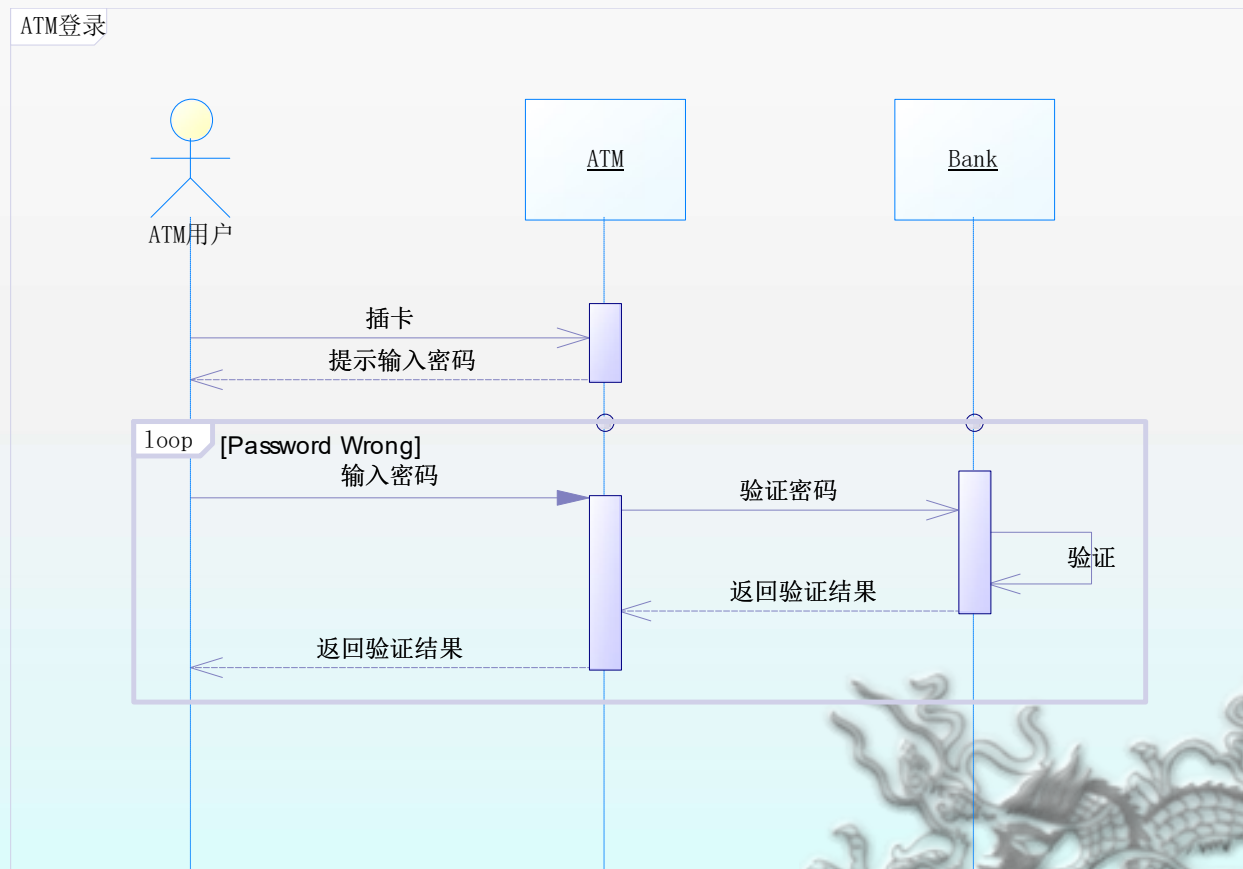
- ◆ alt: 多条路径，条件为真时执行。
- ◆ opt: 任选，仅当条件为真时执行。
- ◆ par: 并行，每一片段都并发执行。
- ◆ loop: 循环，片段可多次执行。
- ◆ critical: 临界区，只能有一个线程对它立即执行。
- ◆ neg: 否定，片段表示一次无效交互。



顺序图

交互片段

ATM登录用例



顺序图

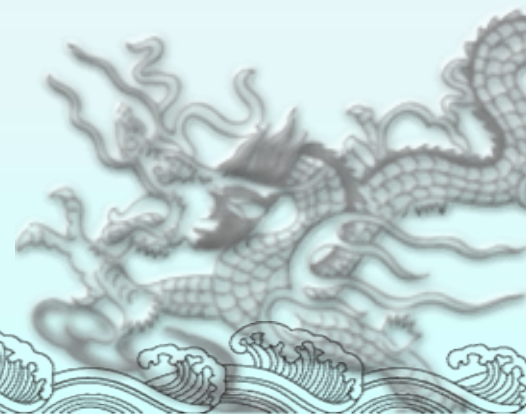
◆ 顺序图作用

- ◆ 对于**业务人员**，顺序图可显示不同的业务对象如何交互，对于交流当前业务如何进行很有用。除记录组织的当前事件外，一个业务级的顺序图能被当作一个需求文件使用，为实现一个未来系统传递需求。
- ◆ 对于**需求分析人员**，顺序图能通过提供一个深层次的表达，把用例带入下一层次。通常用例被细化为一个或者更多的顺序图。顺序图的主要用途之一，是把用例表达的需求，转化为进一步、更深层次的精细表达。
- ◆ 对于**技术人员**，顺序图在记录一个未来系统的行为应该如何表现时非常有用。在设计阶段，架构师和开发者能使用顺序图挖掘出系统对象间的交互，进一步完善整个系统的设计。

顺序图

◆ 顺序图绘制技巧

- ◆ **以用例为单位创建顺序图**，针对每个用例，考察为完成它所描述的功能需要哪些对象的操作参与执行，并且进一步考察这些操作的执行需要通过消息而引起其他哪些对象操作的执行。把这些对象以及参与交互的参与者组织到一个顺序图中。
- ◆ **理论上需要为每一个用例创建一个顺序图**，但是如果一个用例的交互对象很简单可以不需要创建顺序图。

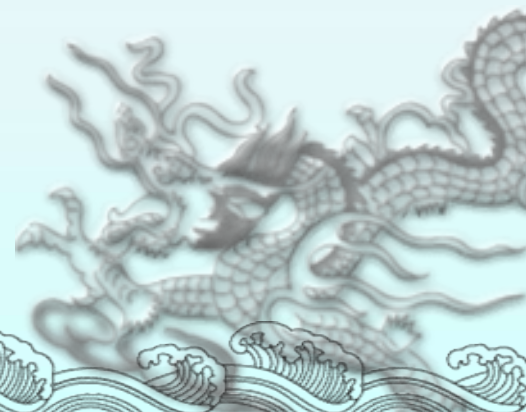


顺序图

◆ 顺序图绘制技巧

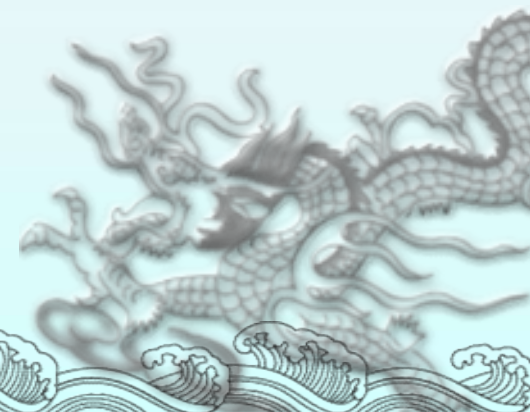
◆ 顺序图绘制步骤:

- ◆ 确定参与交互的执行者
- ◆ 确定与执行者直接交互的对象
- ◆ 确定与交互相关的全部对象（顺藤摸瓜）



顺序图

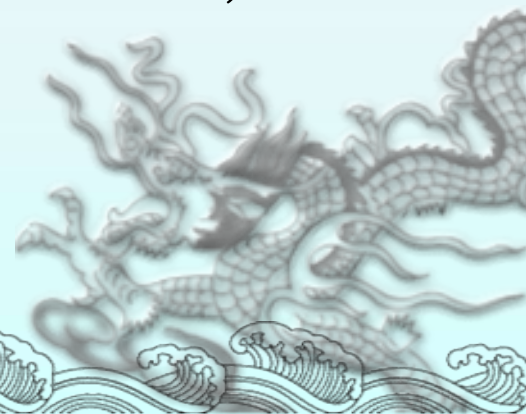
- ◆ 顺序图绘制技巧
 - ◆ 如果需要考察单个用例内部多个对象的行为就应该使用顺序图；
 - ◆ 如果需要考察单个对象的行为就需要使用状态图；
 - ◆ 如果需要考察跨用例或者跨线程的行为就需要考虑使用活动图。



顺序图

◆ 顺序图绘制技巧

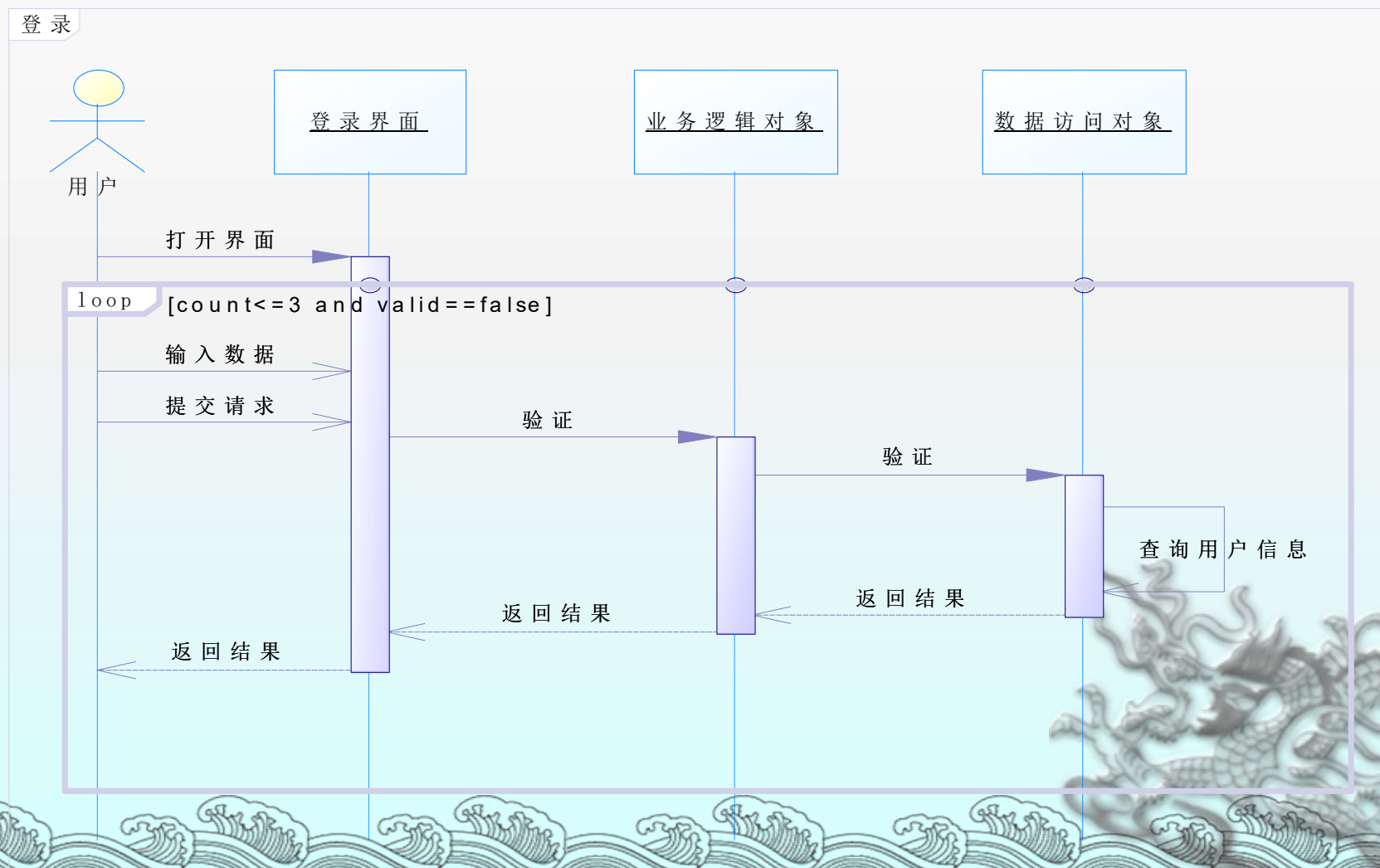
- ◆ **需求分析阶段的顺序图**：主要用于描述用例中对象之间的交互，可以使用自然语言来绘制，用于细化需求。从业务的角度进行建模，用描述性的文字叙述消息的内容。
- ◆ **系统设计阶段的顺序图**：确切表示系统设计中对象之间的交互，考虑到具体的系统实现，对象之间通过方法调用传递消息。



顺序图

◆ 顺序图绘制技巧

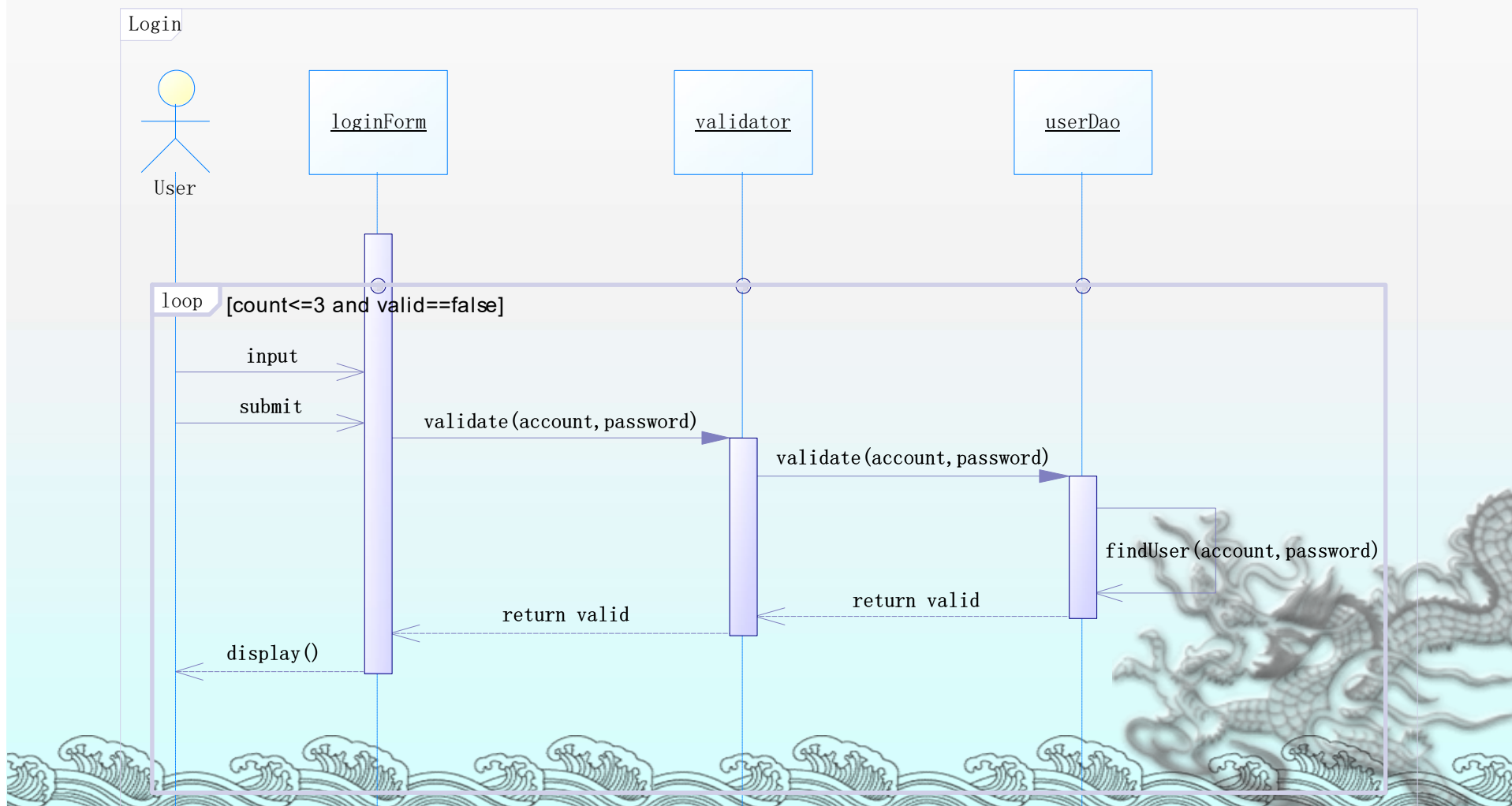
◆ 需求分析阶段的顺序图



顺序图

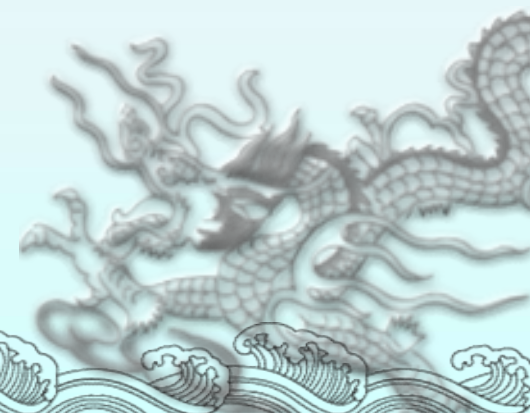
◇ 顺序图绘制技巧

◇ 系统设计阶段的顺序图



内容提纲

- 交互图
- 协作图
 - 相关概念
 - 实例讲解
- 顺序图
 - 相关概念
 - 实例讲解



顺序图

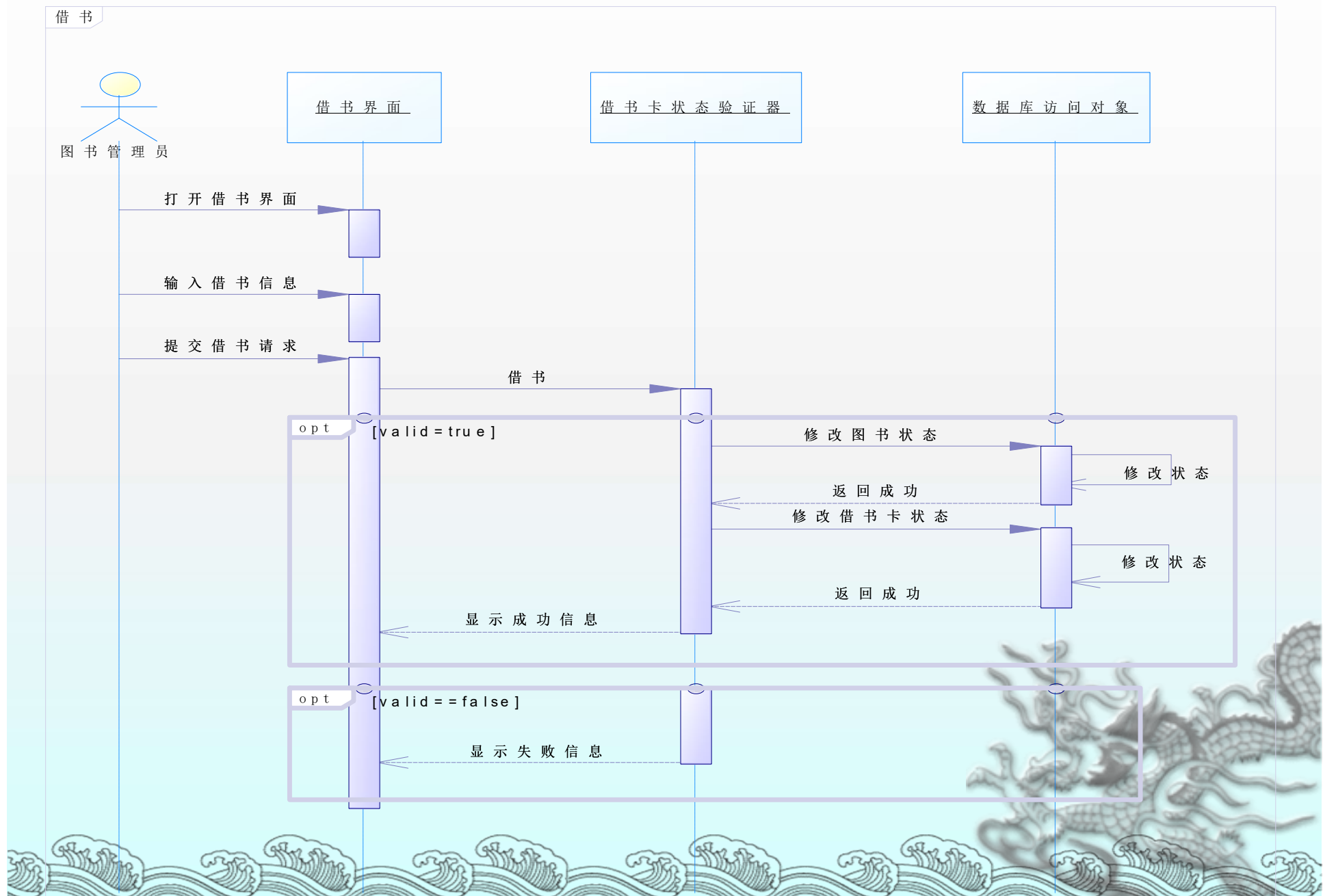
◆ 顺序图实例分析

◆ 绘制图书管理系统“借书”用例的顺序图（业务模型）。

- ◆ 图书管理员打开借书界面，输入借书信息并提交借书请求；系统验证借书卡状态，如果借书卡未借书则记录借书信息且修改图书状态和借书卡状态，并提示借书成功；否则提示借书失败。



◆ 顺序图实例分析



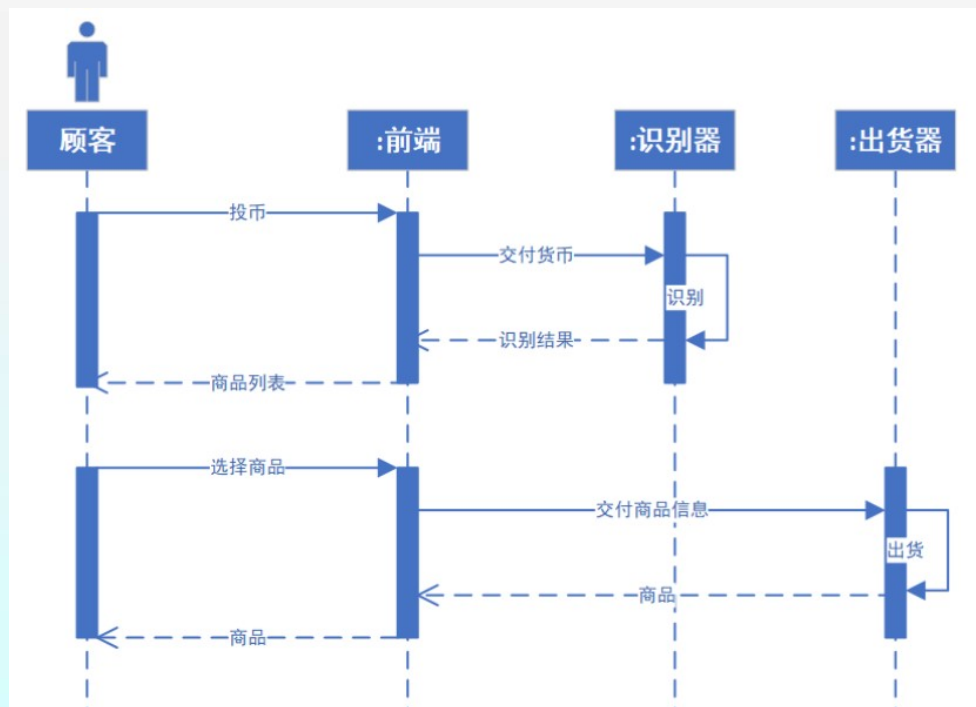
练习

1. 根据下面的叙述，绘制一幅关于顾客从自动售货机中购买物品的顺序图。
 - (1) 顾客 (User) 先向自动售货机的前端 (Front) 投币；
 - (2) 售货机的识别器 (Register) 识别钱币；
 - (3) 售货机前端 (Front) 根据Register的识别结果产生商品列表；
 - (4) 顾客选择商品；
 - (5) 识别器控制的出货器 (Dispenser) 将所选商品送至前端 (Front)



练习

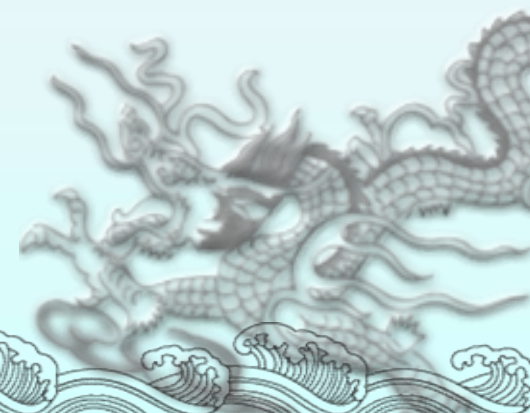
1. 根据下面的叙述，绘制一幅关于顾客从自动售货机中购买物品的顺序图。
 - (1) 顾客 (User) 先向自动售货机的前端 (Front) 投币;
 - (2) 售货机的识别器 (Register) 识别钱币;
 - (3) 售货机前端 (Front) 根据Register的识别结果产生商品列表;
 - (4) 顾客选择商品;
 - (5) 识别器控制的出货器 (Dispenser) 将所选商品送至前端 (Front)



练习

2. 用顺序图描述学生选课过程：

学生通过选课系统的前端页面查询课程信息并选择相应课程，前端在接收到学生的选课信息后提交给后台，后台判断该课程选课人数是否已满，若选课人数已满，则在前端页面提示学生重新选课，若人数未滿，则给出选课成功信息。



练习

2. 用顺序图描述学生选课过程：

学生通过选课系统的前端页面查询课程信息并选择相应课程，前端在接收到学生的选课信息后提交给后台，后台判断该课程选课人数是否已满，若选课人数已满，则在前端页面提示学生重新选课，若人数未滿，则给出选课成功信息。

