

单项练习——顺序图

实验目的：

1. 掌握顺序图的涵义和内容
2. 掌握顺序图的绘制方法
3. 掌握顺序图的使用范围

实验内容：

画出下列描述的顺序图：

1. 某网络营销公司，负责人在有人申请入会时，将在会员登录画面上输入入会申请人的会员信息。请将“负责人”作为参与者、“会员登录画面”、“会员”、“会员列表”作为对象，画出能表现会员登录流程的顺序图。
2. 在商品销售系统中，客户可以多次对购物车中的一种商品创建订单，客户填写完订单内容后提交订单，消息发送到系统后由销售人员进行审核，审核通过订单完成，审核不通过订单取消。请画出客户购买多种商品流程的顺序图。

实验指导：

1. 顺序图的简介

顺序图是一种交互图，用来表示对象在运行的生命周期中，沿着时间使用代表消息的箭头从源生命线到目标生命线的交互。顺序图可以很好的表示一个对象和其他对象的交互，以及触发交互的消息。但是顺序图并不打算表示复杂的程序逻辑。

顺序图的组成部分包括：生命线、消息、复合片段等。下面将分别简单介绍这些概念。

生命线（Lifelines）：一条生命线代表着顺序图中的一个参与者。生命线经常有一个包含对象名的矩形。如果对象名为“self”，就暗示着这条生命线代表着拥有这个顺序图的分类器。有时顺序图会在开始处有带有生命线的 actor 元素。这通常用于顺序图属于一个 use case 的情况。边界元素、控制元素和实体元素也同样可以有生命线。见图 1。

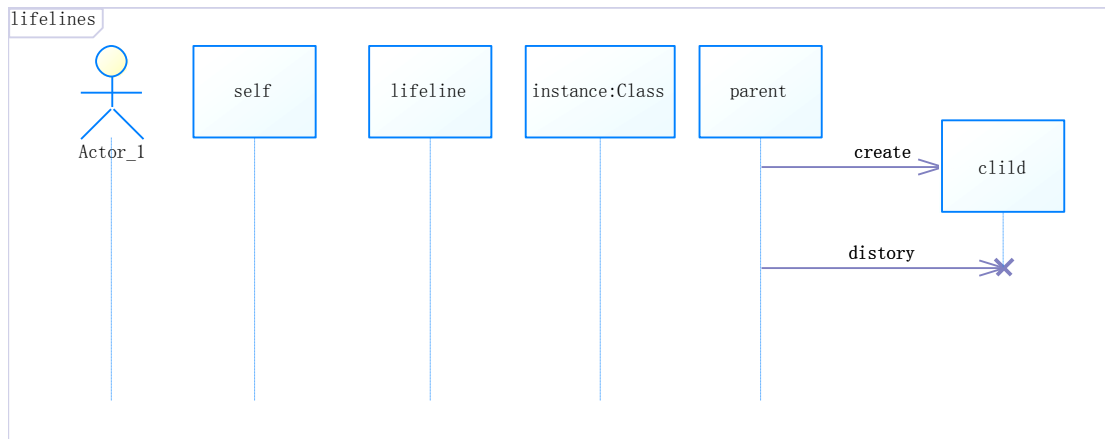


图 1 生命线

可以在顺序图的时间表中产生或者销毁一条生命线。销毁生命线时用一个结束符号叉来表示终止。产生生命线时，**parent** 对象处于比子对象高一级的位置上。如图 1 所示。

消息 (Messages): 消息用箭头表示。消息可能是完成 (**complete**)、丢失 (**lost**) 或者找到 (**found**);同步或者异步; 回调或者信号 (**call or single**)。图 2 中，第一条消息是同步消息 (用实箭头表示) 用一个固有的 **return** 消息完成。第二条消息是异步消息 (用线箭头表示)。第三条消息是异步消息并且返回消息 (用虚线表示)。发送给自己的消息代表着一个递归操作，或者一个方法调用另一个属于同一对象的方法。

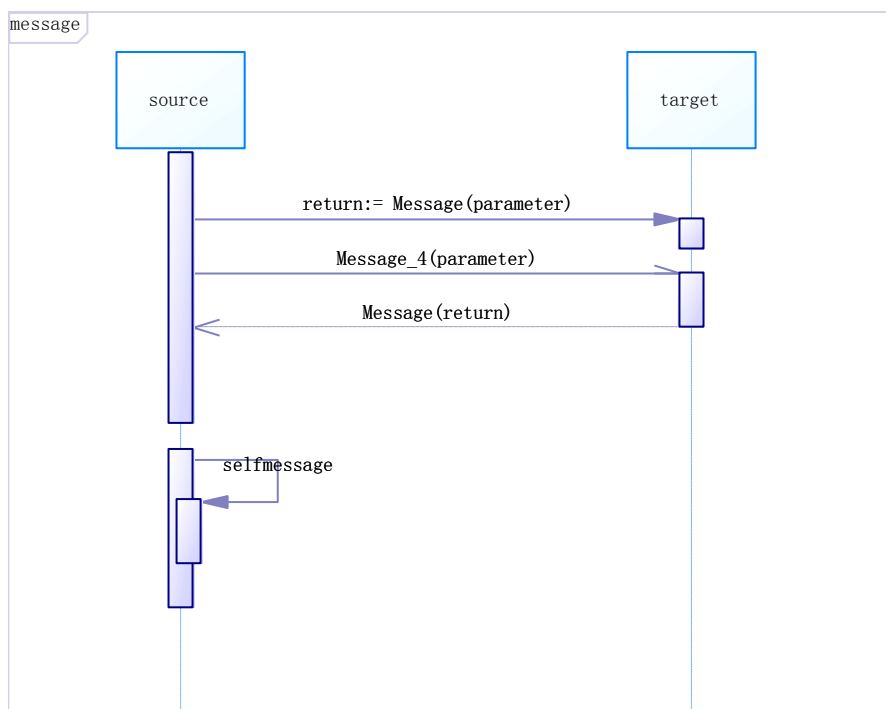


图 2: 消息

复合片段 (Combined Fragments): 使用复合片段机制可以为顺序图增加一定程度的处理逻辑，这些处理逻辑在复合片段下面。一个复合片段是一个或者多个封装在一个框架中并且一定的命名环境中执行的时序。现在可用的框架包括:

二选一片段 (符号 “**alt**”) 的模型为 **if...then...else**。

选择片段 (符号 “**opt**”) 的模型为 **switch**。

并行片段 (符号 “**par**”) 模型为并行处理。

弱时序片段（符号“seq”）封装了许多在接下来的片段要开始之前必须要处理的时序，但这些时序没有必要共享同一个生命线。

强时序片段（符号“strict”）封装了一系列的必须按给定顺序进行的消息。

否定片段（符号“neg”）封装了一系列无效的消息。

临界片段封装了一个临界会话。

忽略片段声明了一条在当前上下文中不感兴趣的消息。

断言片段（符号“assert”）指出了任何没有作为一个操作数的断言的时序都是无效的。

循环片段（符号“loop”）封装了一系列要重复执行的消息。

图 3 展示了一个循环片段。

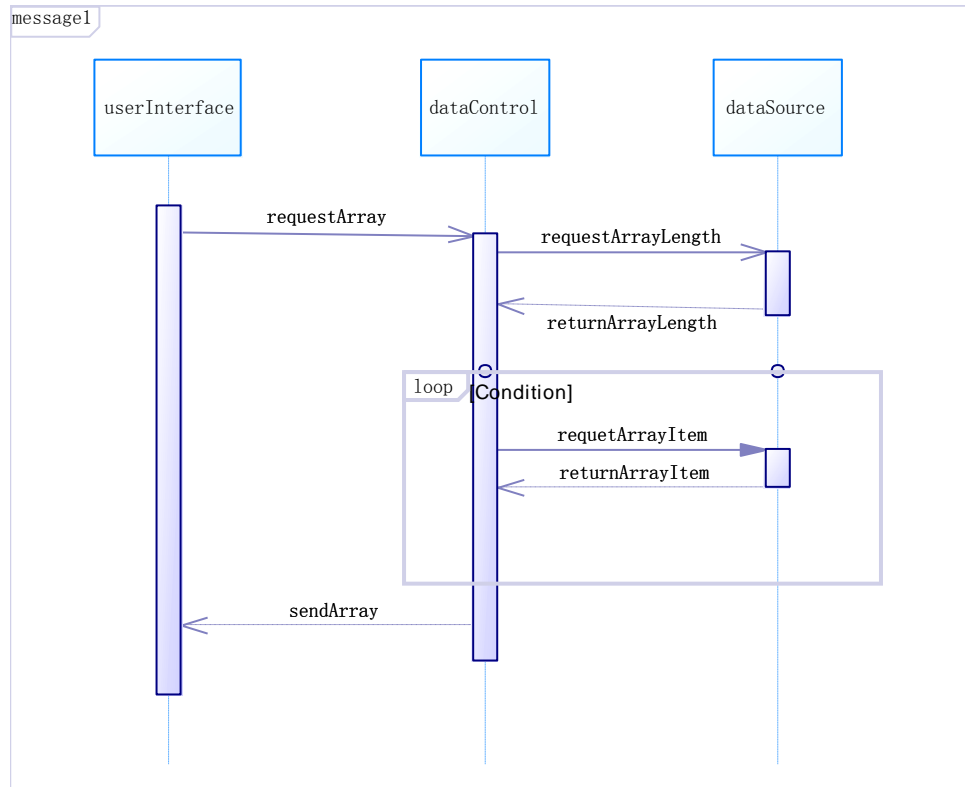


图 3 一个 loop 片段

2. 顺序图的示例

下面是一个顺序图的示例。示例的描述如下：

预定一个旅馆。从一个预定窗口开始.如图 4:

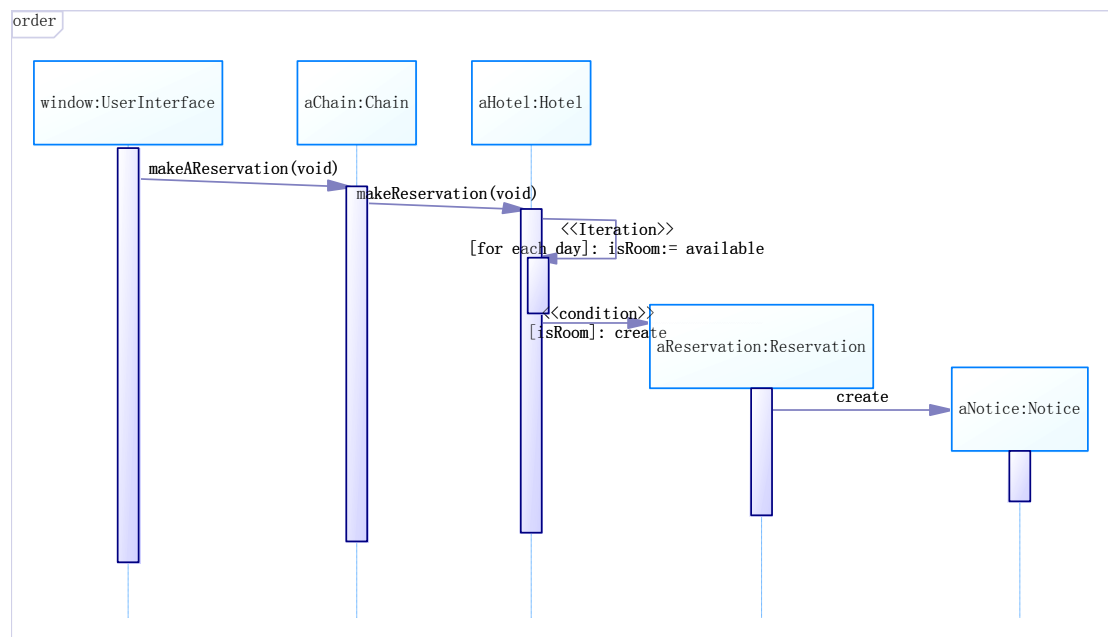


图 4 预定旅馆的顺序图