

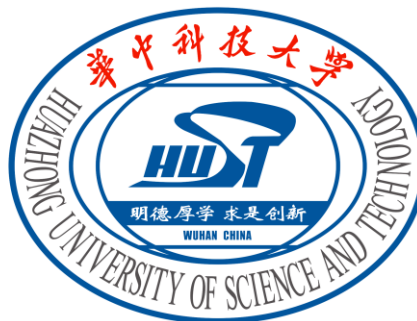
# 基于Java的面向对象程序设计

陈维亚

*weiya\_chen@hust.edu.cn*

华中科技大学软件学院

## 第15讲：UML 2



1. 时序图
2. 活动图
3. 状态转换图
4. 总结

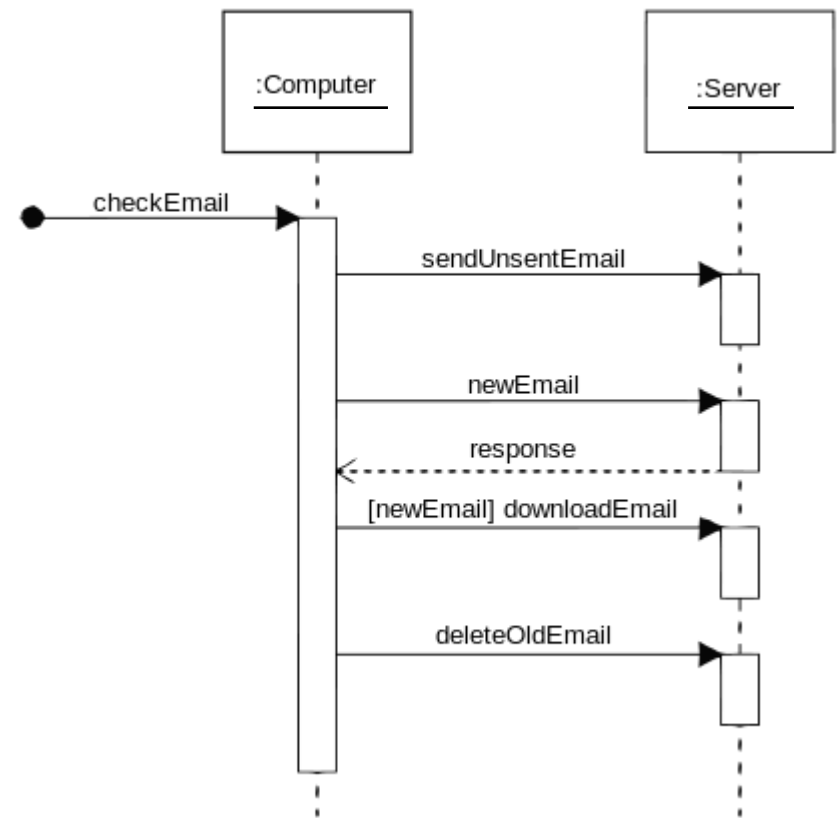
# 1. 时序图

## □ Sequence Diagram

主要用来显示具体用例的详细动作或事件流程

时序图为二维图形

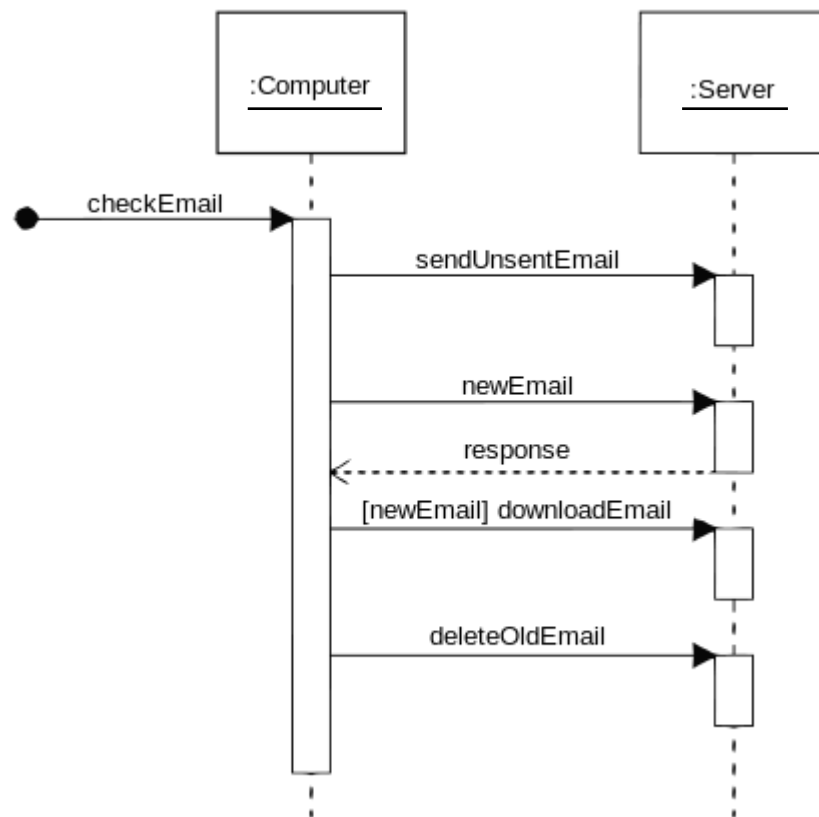
- 水平维度：显示对象之间发送消息的时间顺序；
- 垂直维度：显示发送消息的时间顺序；



## □ Sequence Diagram

### 时序图的构成：

- 对象生命线（垂直的虚线）；
- 对象激活框（细长矩形）；
- 消息（带箭头的水平线 + 消息名）；
  - 调用：实线段，实心箭头
  - 返回：虚线段，枝状箭头
- 顶部方框表示类的对象或类，有3种写法
  - 对象名和类名间用冒号隔开，如 c : Car
  - 省略对象名称，: Car
  - 只保留对象名，c

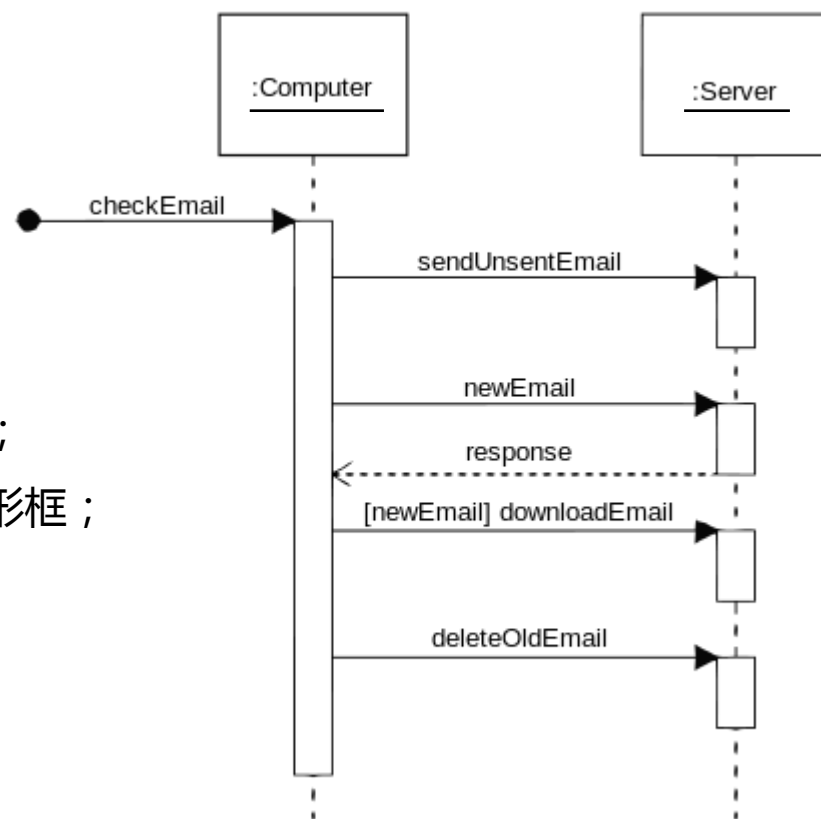


# 1. 时序图

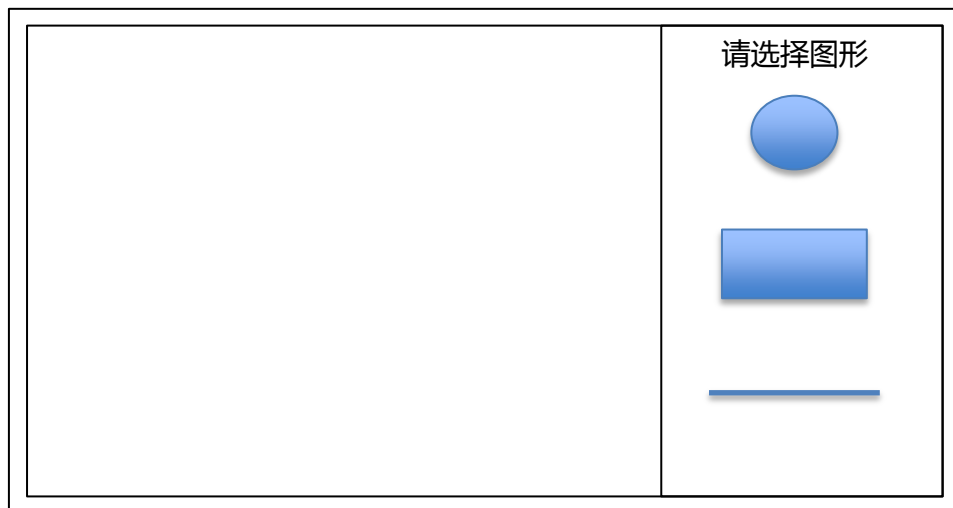
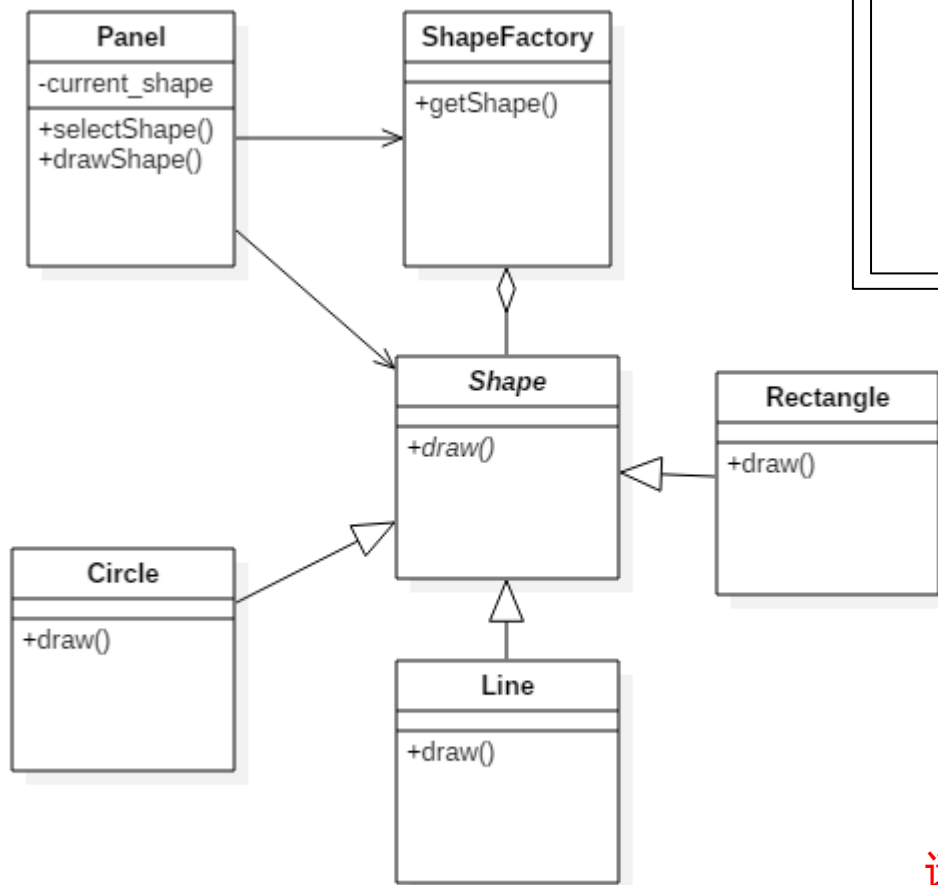
## □ Sequence Diagram

绘制时序图：

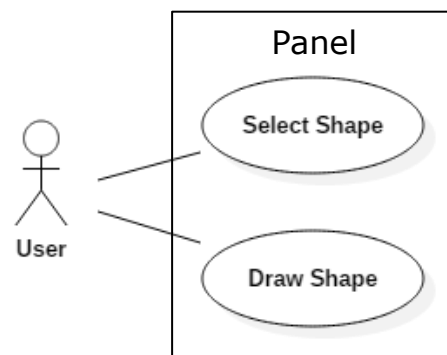
- 1) 确定所描述用例的事件流程；
- 2) 确定参与事件流程的对象，放置在图的顶部；
- 3) 添加各对象的生命线、发送的消息、激活矩形框；



请完成如下画板程序的设计：

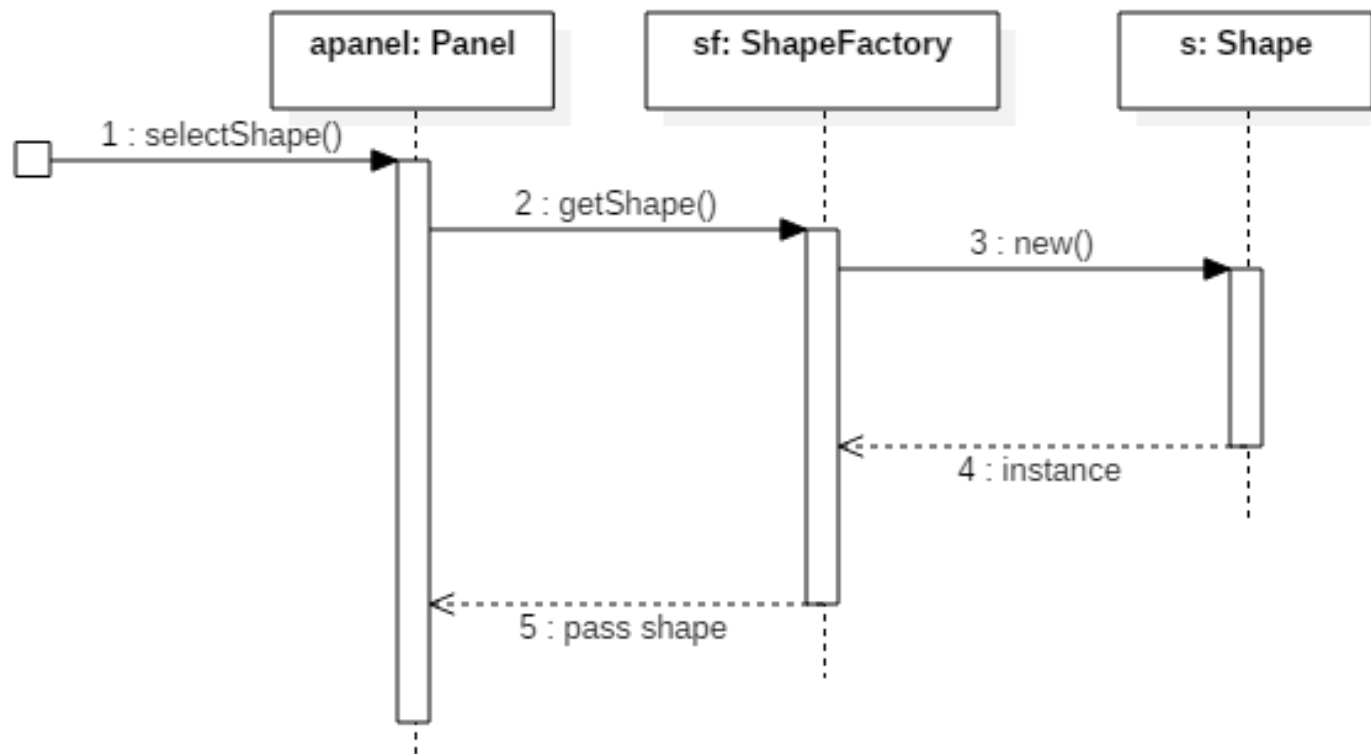


用户点击选中右边想画的图形后，便可在左边的空白处画出相应的形状。

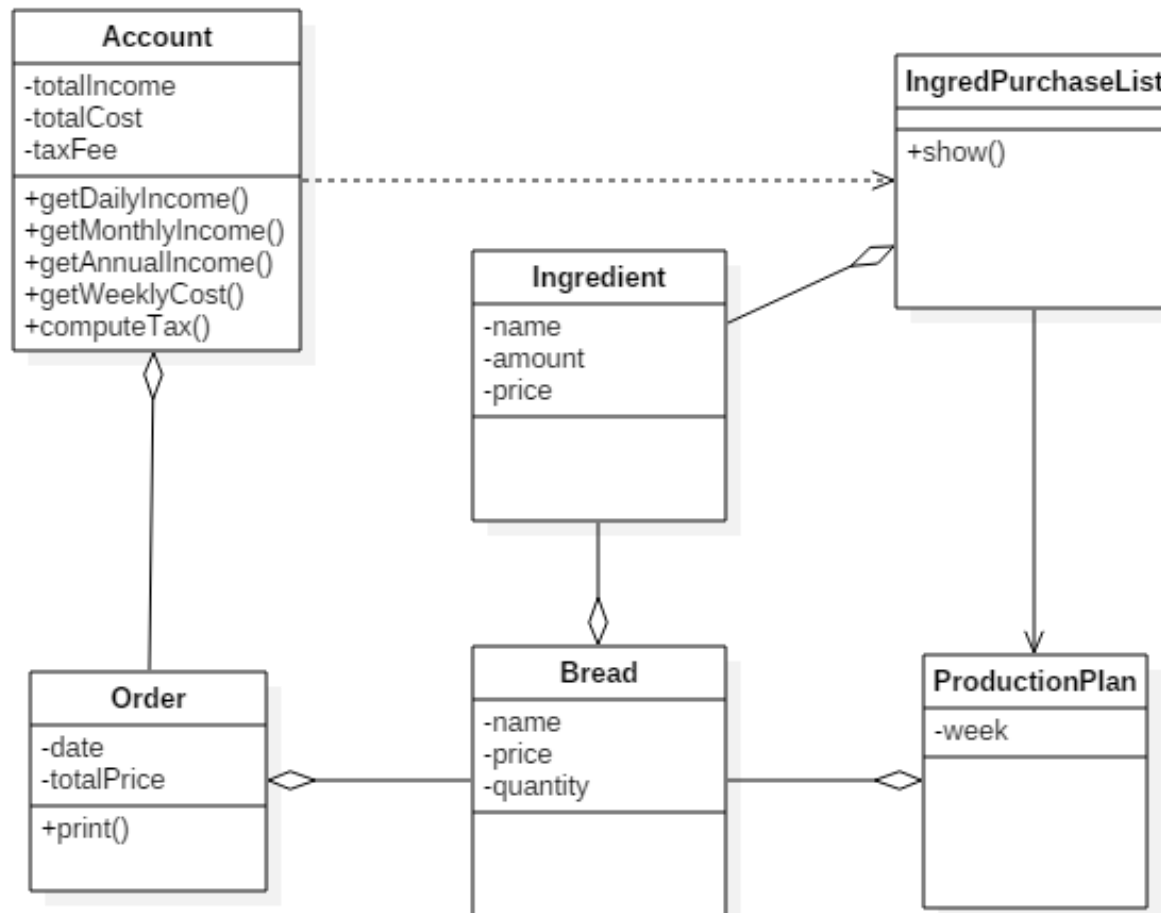


请画出Select Shape用例的时序图

Select Shape用例的时序图：



( 课后 ) 请画出面包店例子中顾客买面包Buy Bread用例的时序图：





## 2. 活动图

### □ Activity Diagram

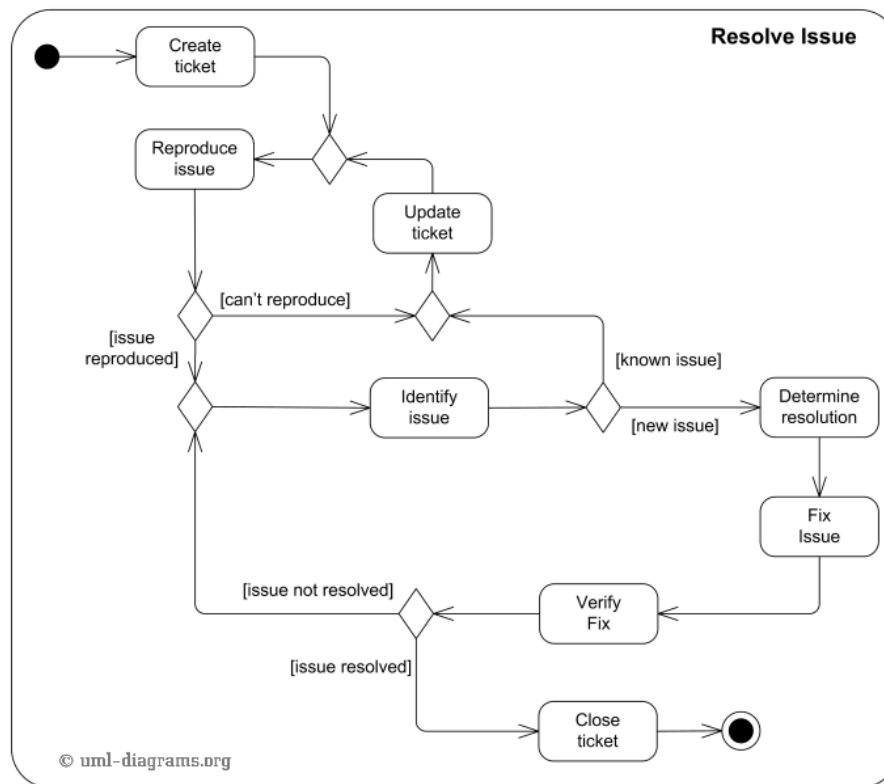
描述活动的顺序，展现从一个活动到另一个活动的控制流

#### 主要元素：

- 活动起点、活动终点
- 活动
- 分叉、汇合（并发）
- 分支、合并（选择）

#### 应用场景：

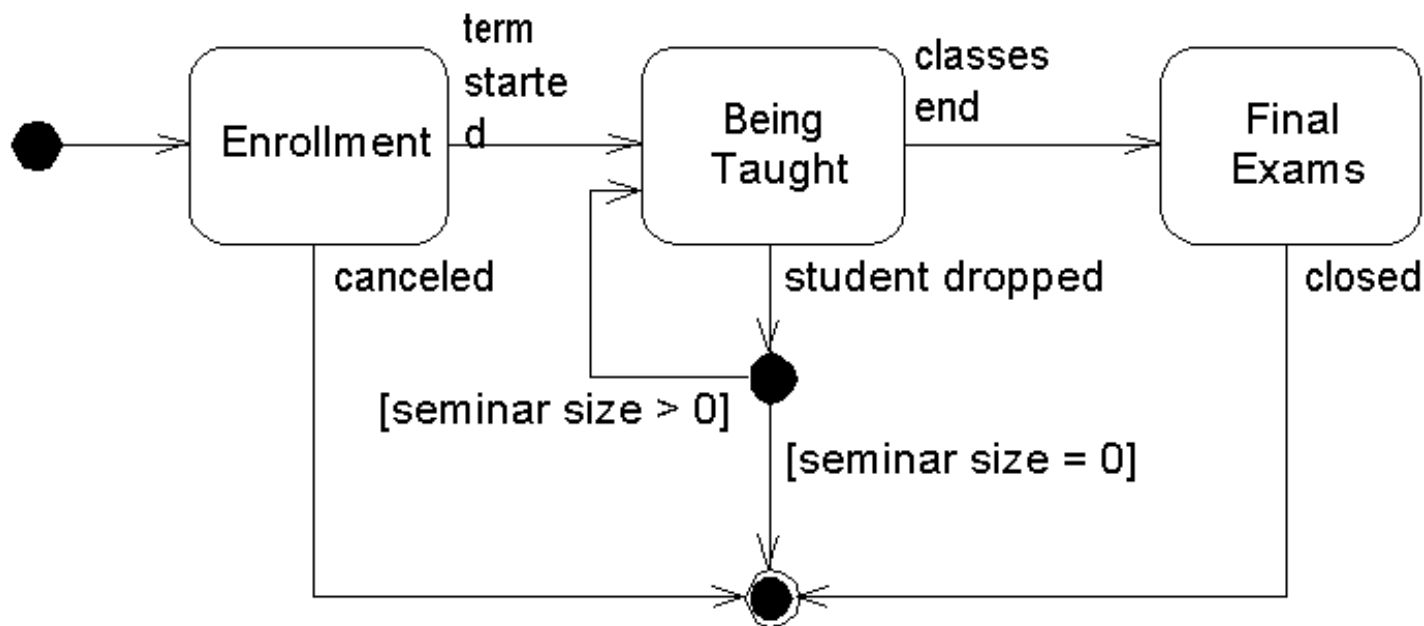
- 用户和系统之间的业务流程或工作流
- 某一用例中执行的步骤
- 软件协议，组件级的交互
- 软件算法流程



### 3. 状态转换图

#### □ Statechart Diagram

描述一个对象的生命周期中所处的不同状态，以及状态之间的转换过程。



#### 主要元素：

- 初始点、终止点
- 状态
- 状态的转换

一个研讨会Seminar对象的状态转换图

### 动态模型

- 时序图
- 活动图
- 状态转换图

## 类的关联综合练习