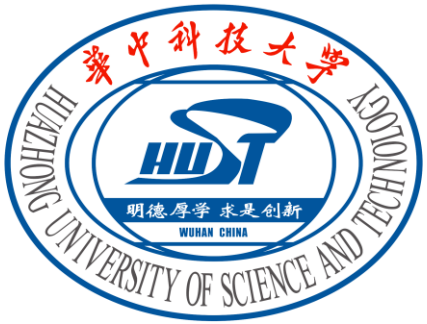
基于Java的面向对象程序设计

陈维亚

weiya_chen@hust.edu.cn

华中科技大学软件学院

第14讲：UML 1



目录

1. UML简介

❑ 目标

1997年，对象管理组织（Object Management Group， OMG）发布了统一建模语言（Unified Modeling Language, UML）。

UML的目的之一就是为开发团队提供标准、通用的面向对象设计语言。

通过使用UML，人们能够阅读和交流系统架构图和设计规划图，就像建筑的设计图一样。

1. UML简介

❑ 内容

UML采用一些标准图形元素来表示对象模型

可视化的面向对象建模语言

常见图类型：

静态模型：

用例图（Use Case Diagram）：从用户角度描述系统功能。

类图（Class Diagram）：描述对象模型中类与类之间的关系。

组件图（Component Diagram）：描述系统中各个组件之间的依赖关系，还可以描述组件的源代码组织结构。

部署图（Deployment Diagram）：定义系统中软硬件的物理体系结构。

1. UML简介

❑ 内容

UML采用一些标准图形元素来表示对象模型

可视化的面向对象建模语言

常见图类型：

动态模型：

时序图（Sequence Diagram）：显示对象之间的动态协作关系，强调对象之间消息发送的时间顺序。

活动图（Activity Diagram）：显示活动的顺序控制流。

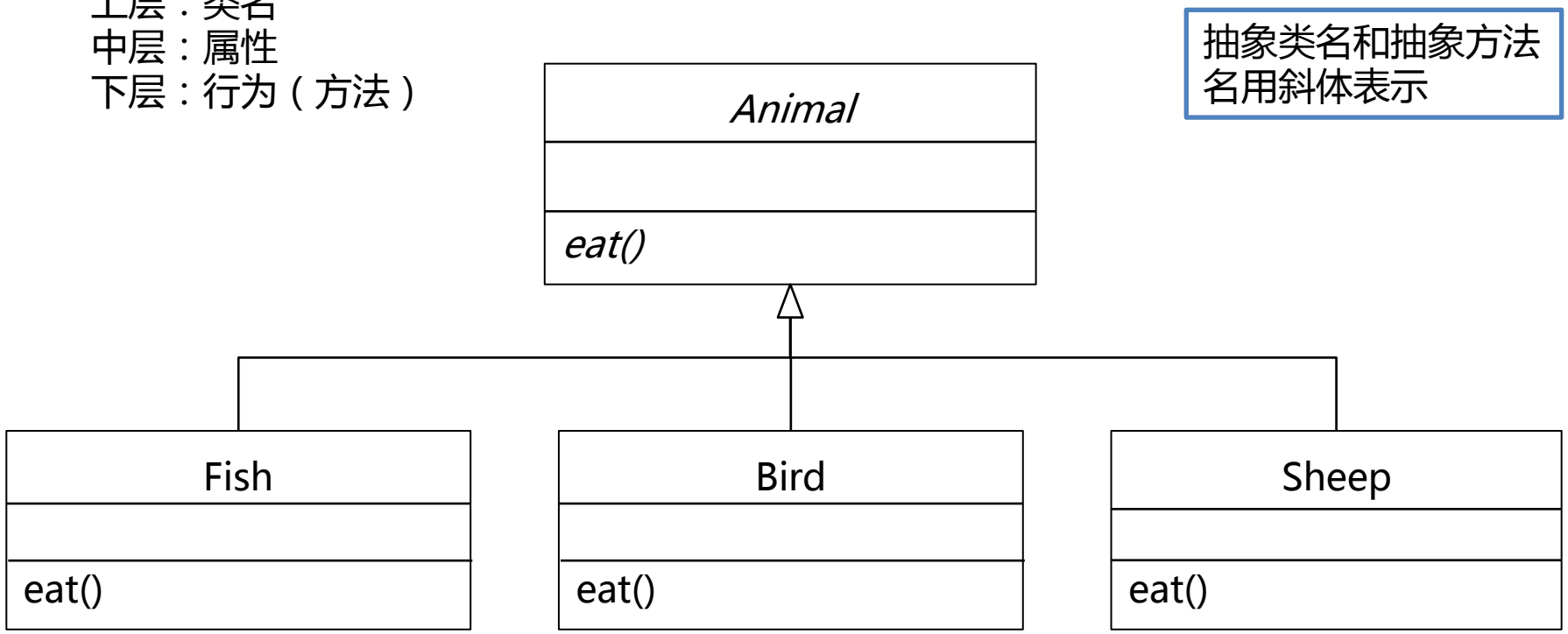
状态转换图（State Transition Diagram）：描述对象所有可能的状态，以及导致状态转换的条件。

2. 类图

❑ Class Diagram

类图显示了系统的静态结构，包含以下内容：

类：矩形框表示
上层：类名
中层：属性
下层：行为（方法）

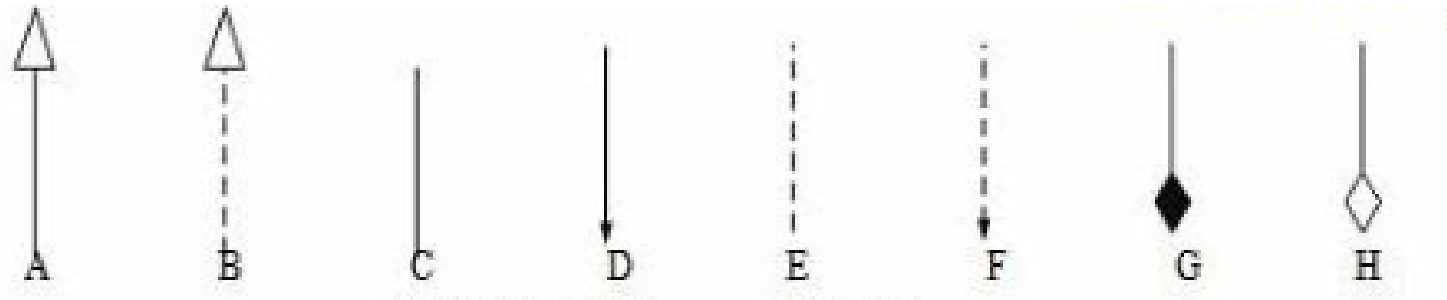


2. 类图

❑ Class Diagram

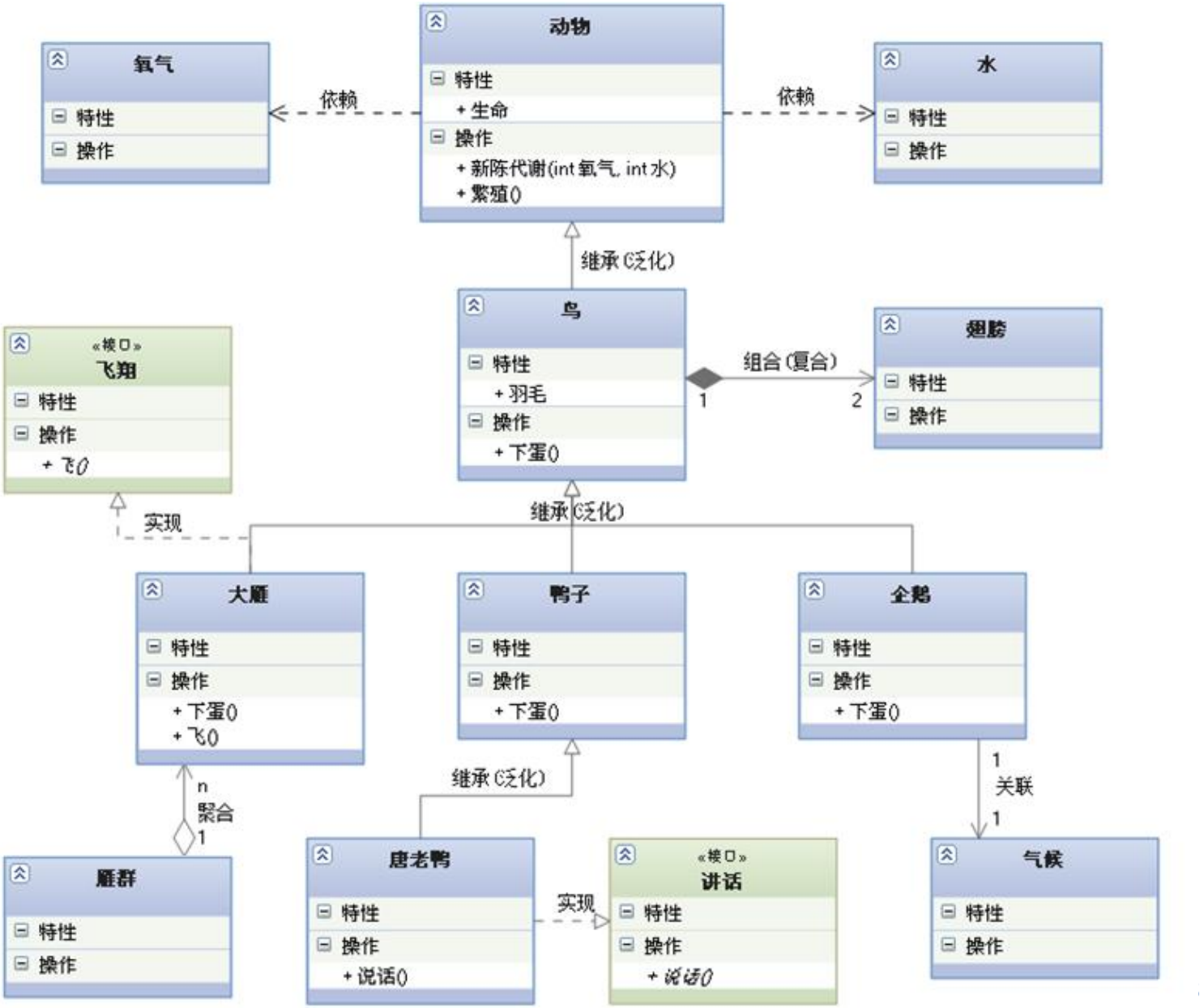
类图显示了系统的静态结构，包含以下内容：

类的关系：线段表示，包括：
继承（泛化）
关联、聚合、组合
依赖
实现（接口）



2. 类图

- 举例


氧气
特性
操作
依赖
动物
特性
+ 生命
操作
+ 新陈代谢(int 氧气, int 水)
+ 繁殖()
依赖
水
特性
操作
继承(泛化)
«接口»
飞翔
特性
操作
+ 飞()
鸟
特性
+ 羽毛
操作
+ 下蛋()
组合(复合)
1
2
翅膀
特性
操作
实现
继承(泛化)
大雁
特性
操作
+ 下蛋()
+ 飞()
鸭子
特性
操作
+ 下蛋()
企鹅
特性
操作
+ 下蛋()
n
聚合
1
继承(泛化)
1
关联
1
雁群
特性
操作
唐老鸭
特性
操作
+ 说话()
实现
«接口»
讲话
特性
操作
+ 说话()
气候
特性
操作

2. 类图

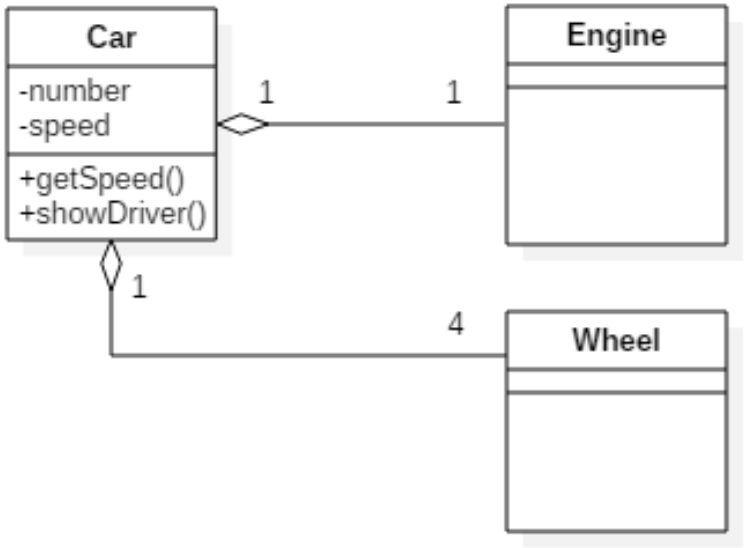
❑ 阅读类图

重点把握三项内容：类、关系、多重性

1）读出类并理解类的语意；

2）读出类之间的关系和多重性，从关系最复杂的类开始阅读；

3）理解类的属性和方法；



2. 类图

❑ 创建类图

重点把握三项内容：类、关系、多重性

1）寻找类和确定类（及属性），从需求分析和用例的描述中提取有意义的名词或名词短语；

2）明确每一个类的含义和职责，确定类的属性和功能（方法）。从需求分析和用例的描述中提取有意义的动词或动词短语；

3）找出类之间的关系，然后画出类图。

3. 用例图

❑ Use Case Diagram

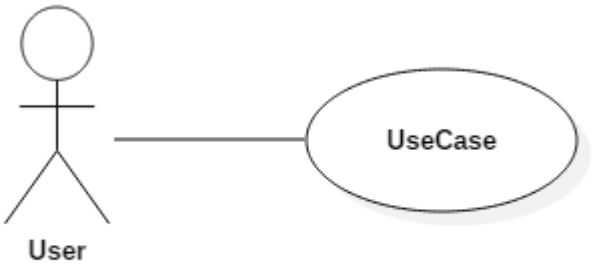
用例图显示了系统的功能，由参与者和用例构成

以图形化的方式表示了：

1）系统内部的功能

2）系统外部的参与者

3）以及它们之间的交互关系



3. 用例图

❑ 用例 Use Case

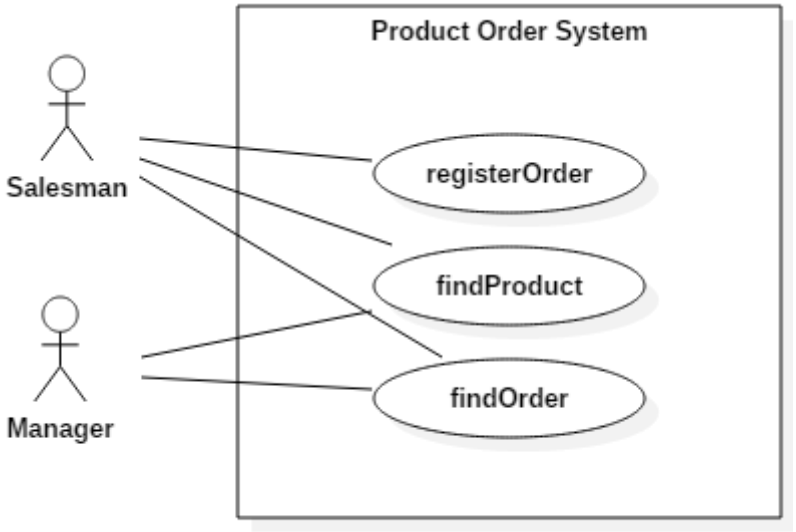
用例代表系统的一个功能，或定义系统参与者与系统的一次完整的交互。

用例不描述系统内部如何工作，只是定义功能，说明系统必须做什么。

一个用例由一系列动作或事件流组成。

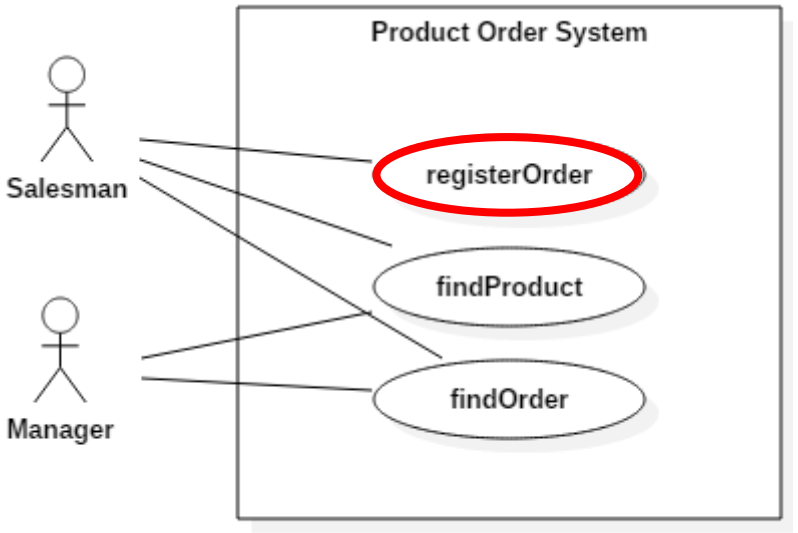
用例的一般描述分为以下几个方面：

- 简要说明
- 主事件流和其他事件流
- 前提条件
- 事后条件



3. 用例图

❑ 用例 Use Case



录入订单registerOrder用例如下：

- **简要说明**：Salesman根据客户的要求录入商品订单到系统中。
- **主事件流**：Salesman先查询有没有客户想订购的商品，如果有则输入客户姓名，查询该客户是不是新客户，如果是则创建客户资料，再为客户填写订单。
- **其他事件流**：无。
- **前提条件**：有客户要下订单。
- **事后条件**：无。

3. 用例图

❑ 创建用例的具体步骤

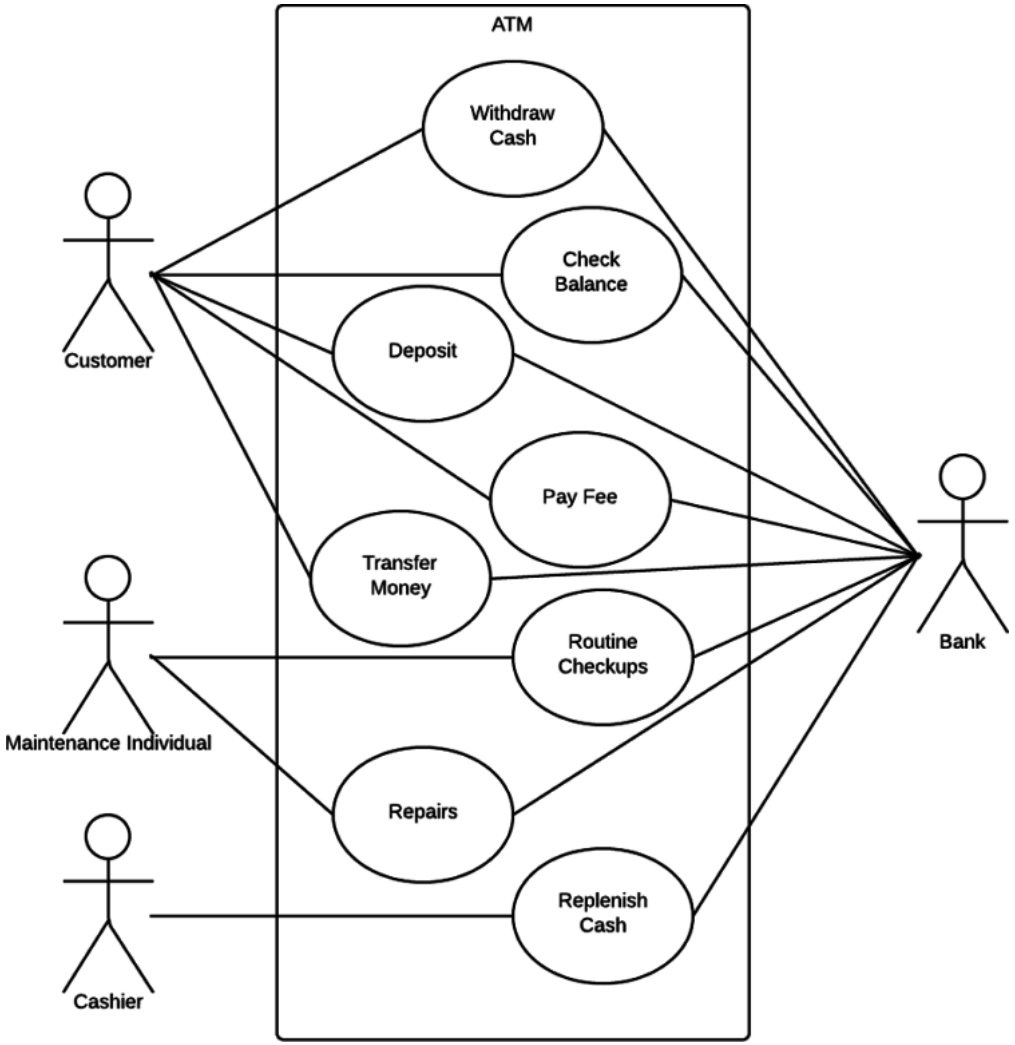
软件开发的需求分析阶段：

1）识别系统的参与者，以及他们对系统的需求；

2）从系统的需求描述中找出用例 - 以动词短语描述用户与系统交互的完整事件流；

3）对用例进行详细描述（主事件流、其他事件流、前提条件等）；

4）画出系统用例图。



练习1

请完成如下画板程序的设计：

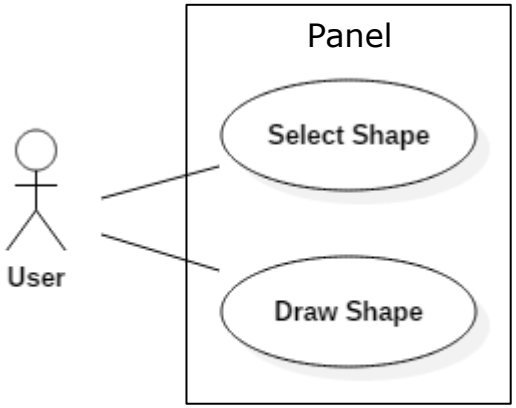
用户点击选中右边想画的图形后，便可在左边的空白处画出相应的形状。

练习1

请完成如下画板程序的设计：

1）画出系统的用例图

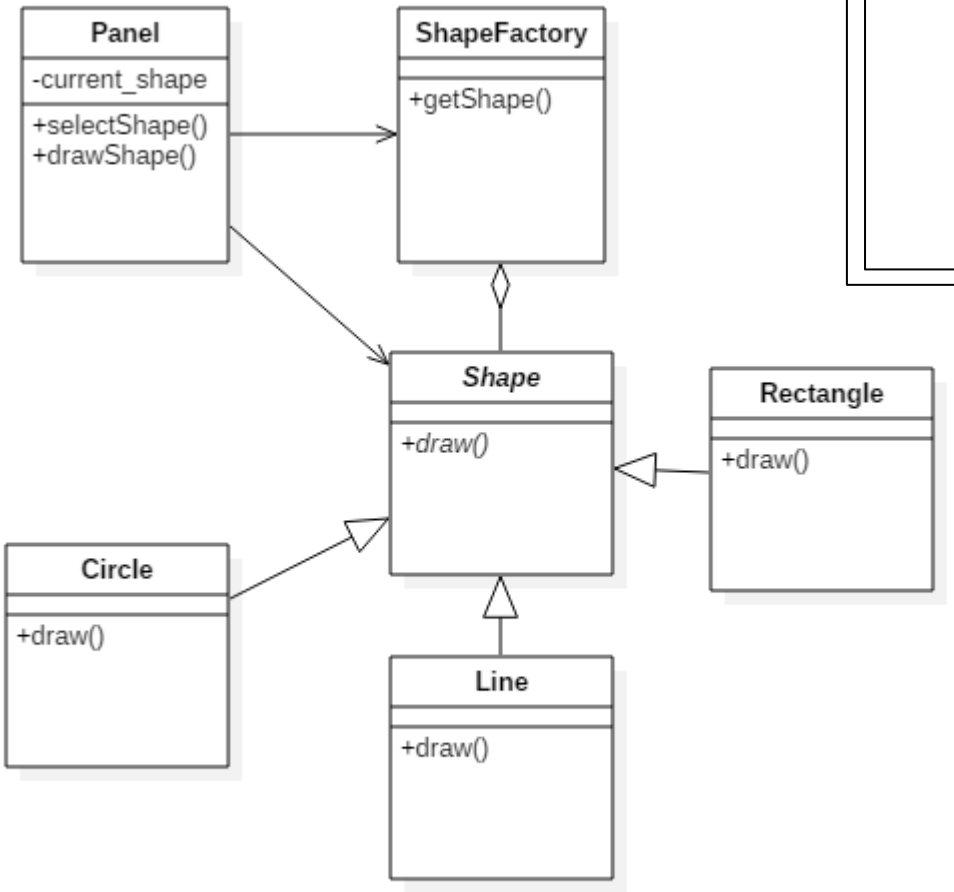
用户点击选中右边想画的图形后，便可在左边的空白处画出相应的形状。



练习1

请完成如下画板程序的设计：

2）画出系统的类图



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。

4. 其它静态模型

❑ 组件图 Component Diagram

作用

显示软件系统中组件之间的依赖关系，以及和第三方组件（比如类库）的依赖关系。

还能显示包含软件的源代码文件的物理组织结构。

组件

软件系统中的子系统，它由一组协作完成特定服务的类组成。

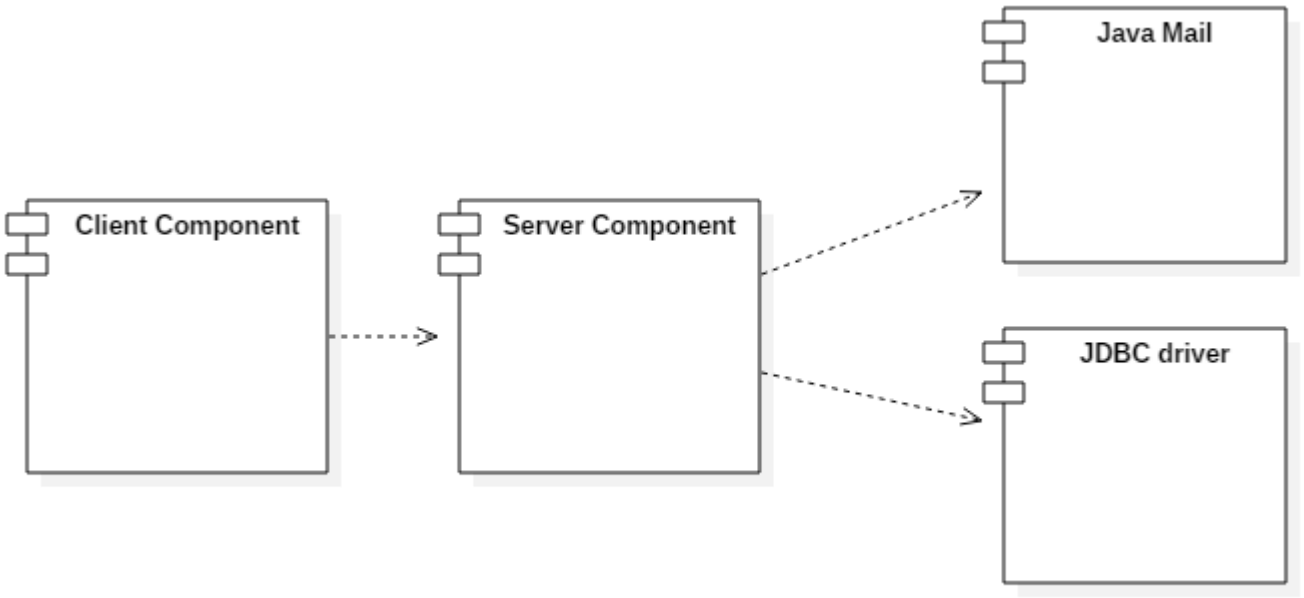
每个组件都封装实现的细节，对外公开接口。

组件之间仅存在依赖关系。

4. 其它静态模型

❑ 组件图 Component Diagram

客户端 - 服务器组件



4. 其它静态模型

❑ 部署图 Deployment Diagram

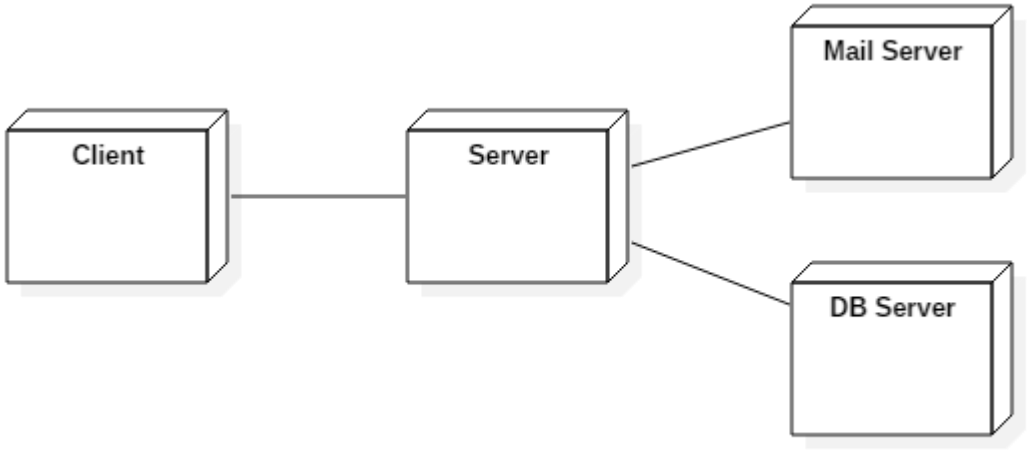
作用

表示软件系统如何部署到硬件环境中，能够展示系统中的组件在硬件环境中的物理布局。

节点

一个节点代表一台物理机器，或一个虚拟机器节点；

用三维立方体表示；



练习2

法国小镇上的面包店老板想使用一个简单的商店管理系统使自己的工作更轻松，我们来帮他设计一下。

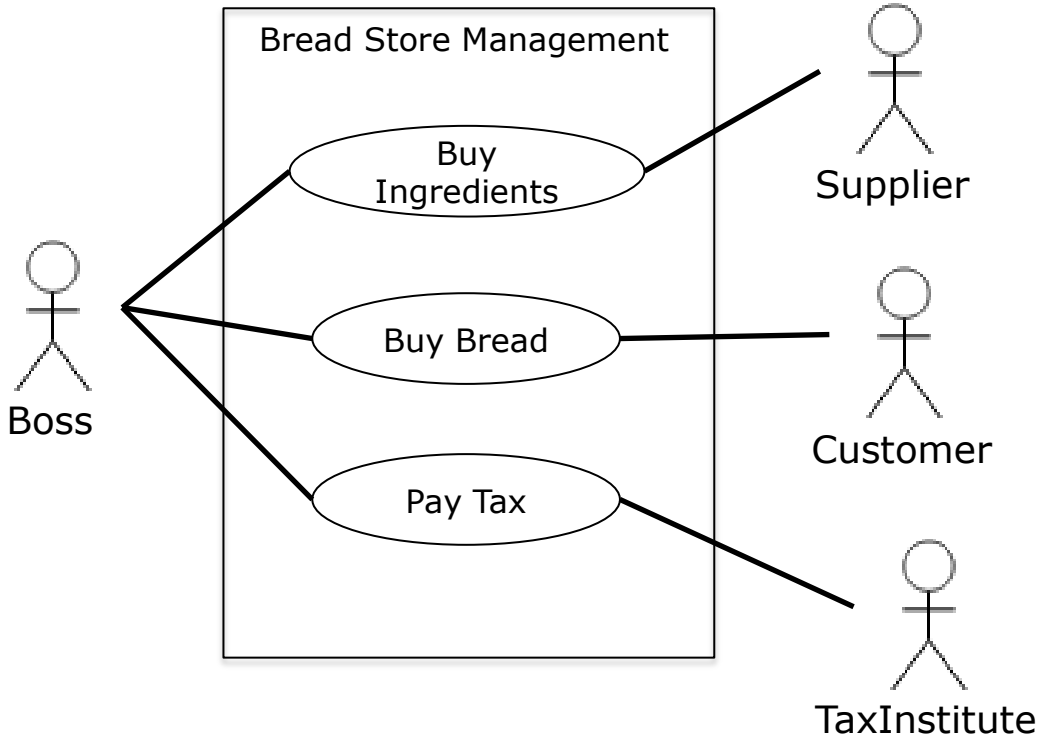
经过初步的讨论，系统涉及原材料进货、面包销售和税务报账3个模块。

1）请画出该系统的用例图。

练习2

经过初步的讨论，系统涉及原材料进货、面包销售和税务报账3个模块。

1）请画出该系统的**用例图**。



练习2

经过初步的讨论，系统涉及原材料进货、面包销售和税务报账3个模块。

2）请根据进一步的描述画出该系统的**类图**。

面包店有长棍面包Baguette，羊角面包Croissant，巧克力面包Chocolate，每种面包有不同的单价和原料用量，老板按照一周的生产计划（每种面包做多少个）给供应商提供一份采购清单，购买原材料。

(Baguette：100g 面粉
Croissant： 50g面粉，10g黄油，1g盐
Chocolate：50g面粉，10g巧克力)

每次顾客购买后，系统中都会添加一个订单，上面包含购买面包的种类和数量，总价格和购买日期。

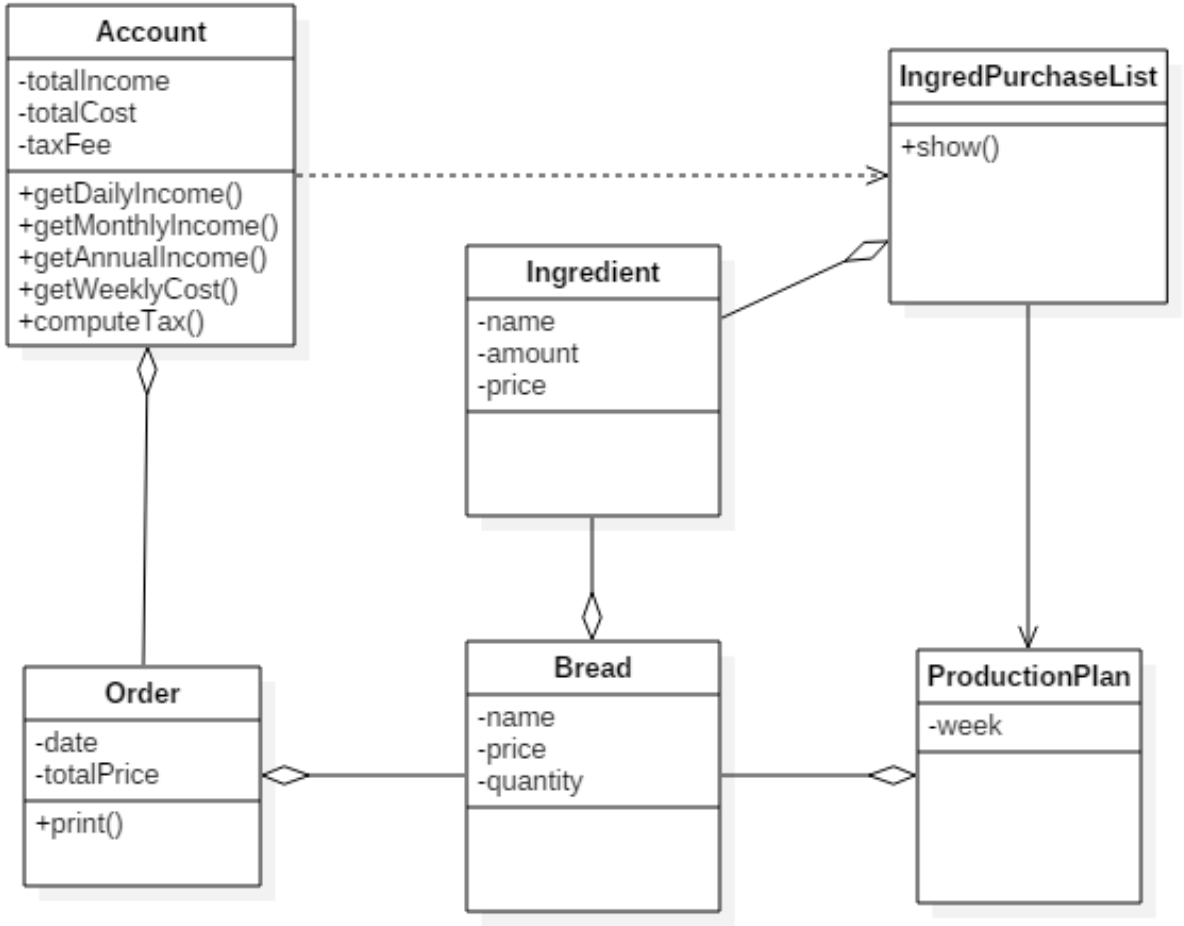
根据这些订单，老板可以算出每天的营业额和每个月的总营业额。

税务局的人每年会根据面包店的收入来征税，税款为一笔基础税费加上收益的一个百分比。

练习2

经过初步的讨论，系统涉及原材料进货、面包销售和税务报账3个模块。

2）请根据进一步的描述画出该系统的**类图**。



5. 总结

UML简介

静态模型

- 类图
- 用例图
- 组件图
- 部署图

下节预告

UML 第二部分