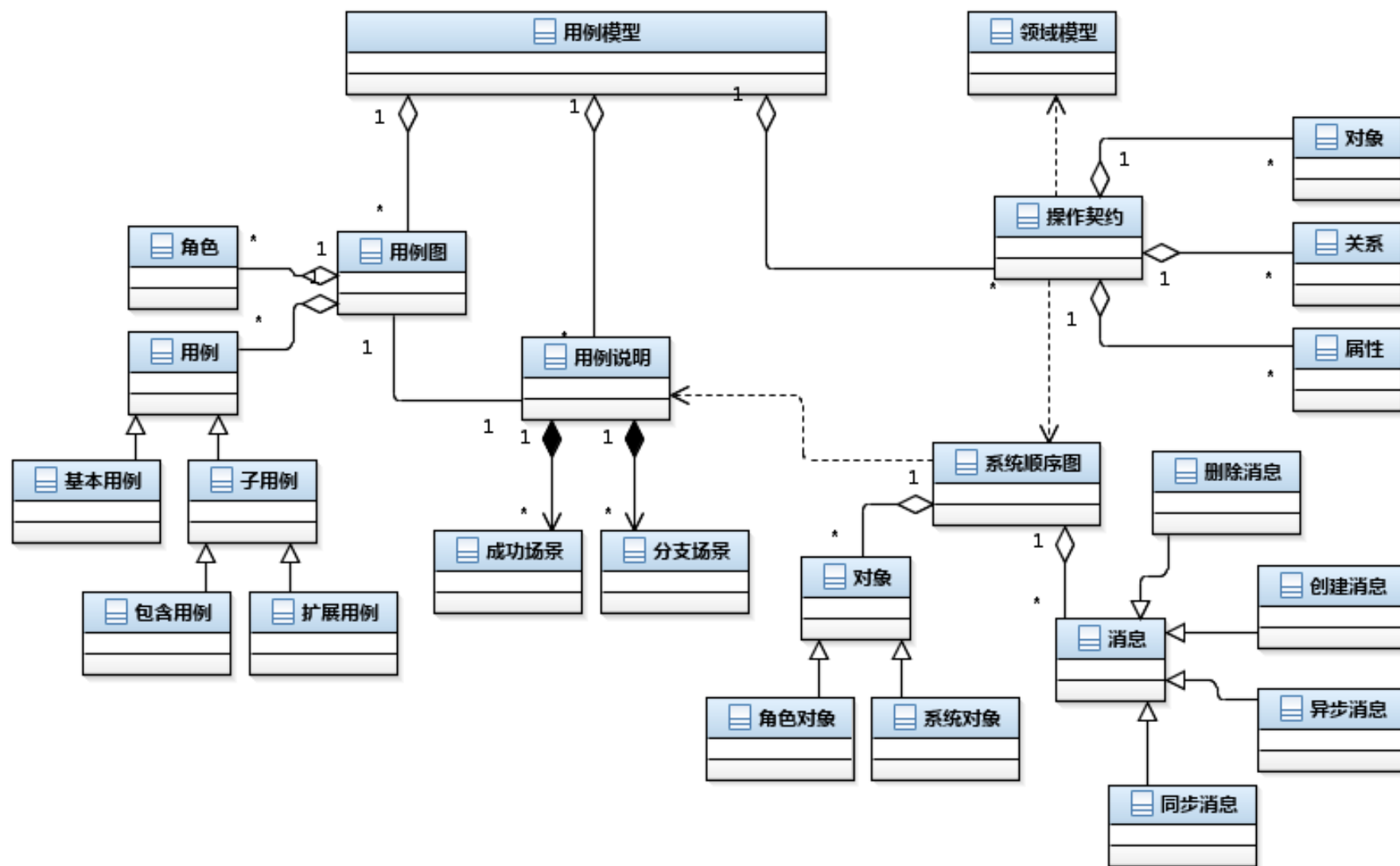


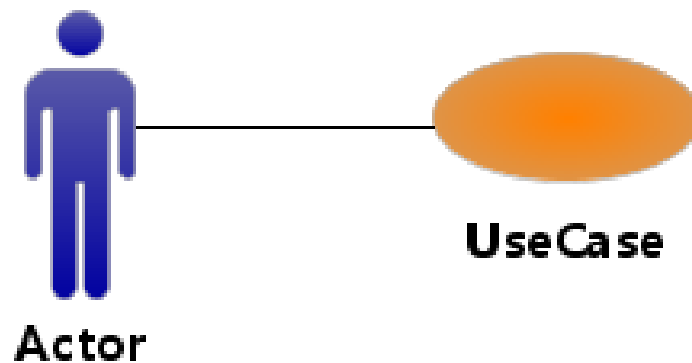
- 用例模型由以下四个部分组成：
 - 用例图；
 - 用例说明；
 - 系统顺序图（system sequence diagram, option）；
 - 操作契约（operation contract, option）；
- 以用例为核心从使用者的角度描述和解释待构建系统的功能需求

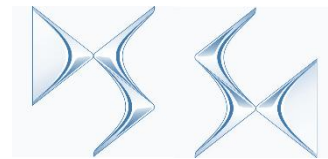
用例模型的基本结构



- 用例图由三个基本元素组成

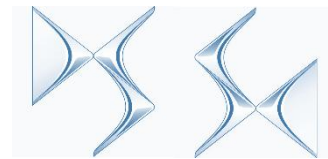
- Actor: 称为角色或者参与者, 表示使用系统的对象, 代表角色的不一定是人, 也可以是组织、系统或设备;
- Use_case: 称为用例, 描述角色如何使用系统功能实现需求目标的一组成功场景和一系列失败场景的集合;
- Association: 表示角色与用例之间的关系, 以及用例和子用例之间的关系;





案例-1：银行柜台取款场景

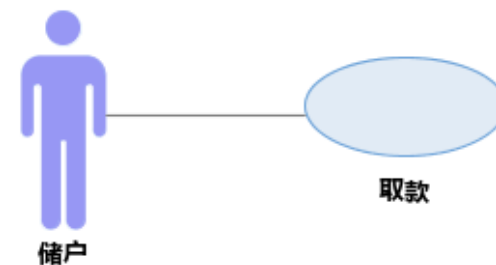
- 银行柜台取款的场景：
 1. 某储户到银行柜台取钱，人多排队；
 2. 等到柜台窗口时，将存折或银行卡交给窗口的营业员；
 3. 营业员通过刷卡器读入账号信息，并提示储户输入密码；
 4. 密码验证通过后营业员询问储户需提取多少钱？
 5. 储户告知具体的提款金额；
 6. 营业员通过系统输入取款金额，并打印本次取款的操作记录，交给储户签字认可；
 7. 营业员核对无误后将取款的现金和银行卡交给储户；
 8. 储户接收现金和银行卡，起身离开；
 9. 完成一次银行柜台取钱的交易。
- 问题：此时使用银行系统的角色是谁？
- 由于柜台人数限制，该银行希望开发ATM系统提供7*24小时的银行服务，并要求具有余额查询、取款、更改登录密码的功能。



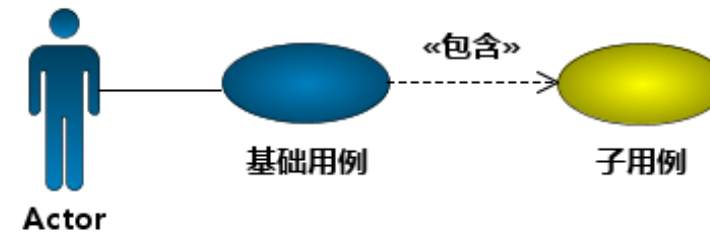
案例-1：ATM系统取款场景分析

- 根据银行柜台的取款场景，可以设计如下使用ATM取款的场景：
 1. 储户将银行卡插入ATM；
 2. 储户根据提示输入密码；
 3. 储户根据系统的操作提示选择取款；
 4. 储户根据提示确定取款金额；
 5. 储户从取款箱中取出现金；
 6. 储户选择是否打印操作凭据；
 7. 储户根据系统提示选择退出；
 8. ATM退出银行卡；
 9. 储户取卡离开；
 10. ATM完成一次取款交易。

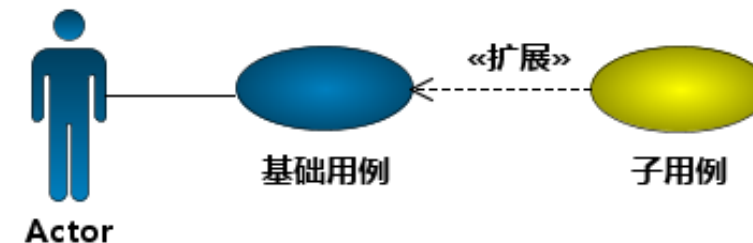
- 根据ATM取款的场景描述，有以下可以表示功能的动词：
 - 取款，插入，输入，选择，确定，取出，退出、取卡和完成；
 - 插入银行卡：表示验证银行卡的有效性 功能-1；
 - 输入密码：表示进行身份验证 功能-2；
 - 选择操作：表示储户本次操作的的目的性，取款 功能-3；
 - 确定金额：表示目的性的具体数值 功能-4；
 - 取出现金：表示本次操作是否正确 功能-5；
 - 退出银行卡：表示系统已经确认本次操作的结束 功能-6；
 - 储户取卡：表示本次交易结束 功能-7；
- 代表该角色：储户 目的性 功能的用例：取款
- 请大家练习描述查询余额以及更改密码的场景！



- 基本用例：与角色直接相关的用例，表示系统的功能需求；
- 子用例：通过场景描述分析归纳出的用例，也表示了系统的功能，是基本用例的一个组成部分；
 - 包含子用例：多个基本用例中的某个与角色交互的场景具有相同的操作，且这些场景都是基本用例中必须执行的步骤，可以将其抽取出来作为基本用例的子用例；

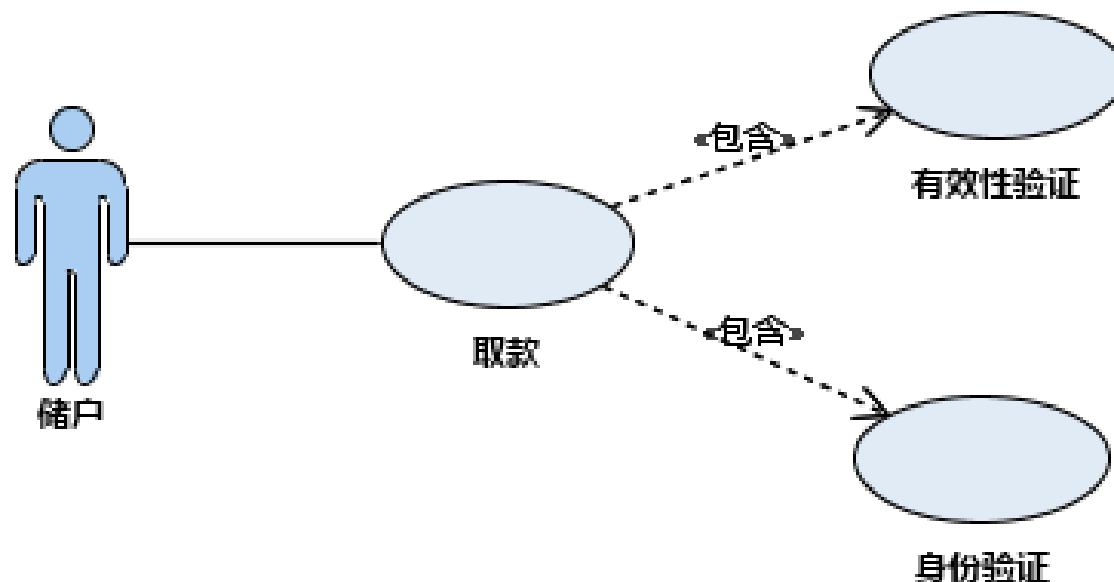


- 扩展子用例：（多个）基本用例中的某些场景存在相同的条件判断的情况，可以将其抽取出来作为基本用例的子用例；



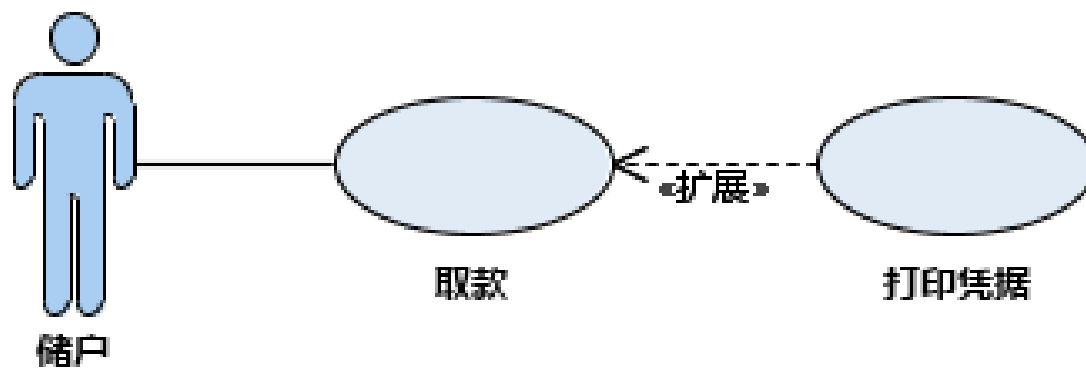
案例-1：包含子用例

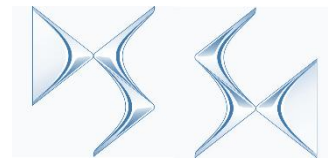
- 在ATM的取款、查询余额、更改密码的操作场景中，经分析都存在一段验证银行卡有效性及身份验证的场景，而且都是基本用例必须执行的操作，为此可以将其抽取出来作为基本用例的包含子用例。



案例-1：扩展子用例

- 在取款和查询余额的操作场景中，存在一段是否打印操作凭据的情景，经分析该功能可以成为一个子用例，且符合在某种条件下可以执行的用例，为此该子用例应为基本用例的扩展子用例。





- 基于已经找到的用例和子用例，并参考之前的需求定义以及场景描述的内容，将用例交互的成功场景和失败场景以标准的格式归纳描述。

用例编号:

每一个用例一个唯一的编号，方便在文档中索引。

用例名称:

（状语 + ）动词 + （定语 + ）宾语，体现参与者的目标。

范围:

应用的软件系统范围

级别:

用例/子用例

参与者:

参与者的名称

项目相关人员及其兴趣:

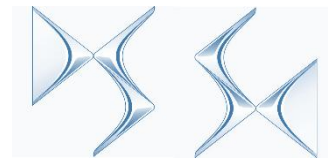
用户应包含满足所有相关人员兴趣的内容

前置条件:

规定了在用例中的一个场景开始之前必须为“真”的条件。

后置条件:

规定了用例成功结束后必须为“真”的条件。



主要成功场景:描述能够满足项目相关人员兴趣的一个典型的成功路径。不包括条件和分支

1.

.....

..... 包含子用例的名称 或者 扩展子用例的名称

n.

扩展（或替代流程）：（备选路径）说明了基本路径以外的所有其他场景或分支

a. 描述任何一个步骤都有可能发生的条件，前边加

5a. 对基本路径中某个步骤的扩展描述，前边加基本路径编号

特殊需求： 与用例相关的非功能性需求

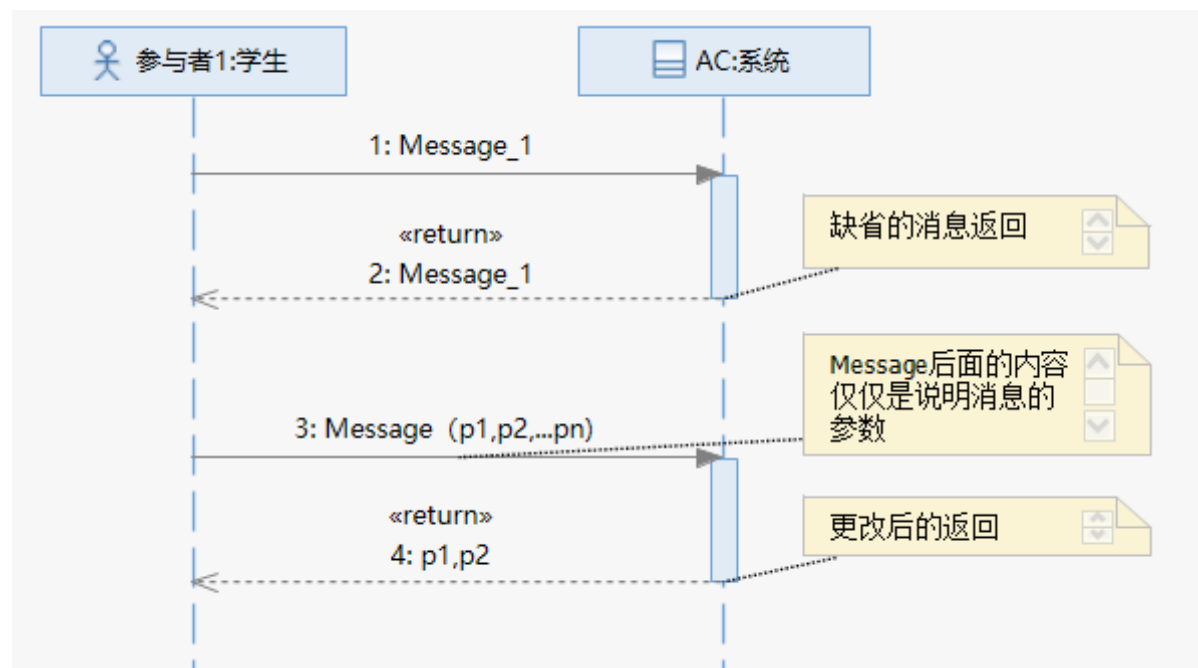
技术与数据的变化列表： 输入输出方式上的变化以及数据格式的变化。

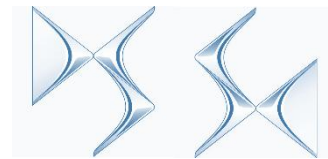
发生频率： 用例执行的频率。

待解决的问题： 不清楚的、尚待解决的问题可集中在此进行罗列

课后练习：请给出使用ATM一次取款和一次存款的用例说明，并给出所有可能的分支场景

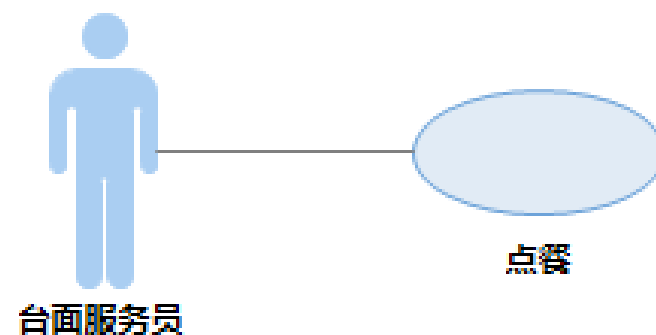
- 使用UML的sequence diagram描述角色与系统之间的交互场景实例；
- 在用例描述的基础上需进一步确定角色与系统之间的交互信息，并以可编程的方式将其命名；
- 系统顺序图中“一般”只需要三个UML的符号元素
 - 顺序图中的对象图标：<类名：对象名>
 - 角色，类的特殊标识；
 - 代表软件系统的对象，一般使用system或者系统命名；
 - 角色与system之间的交互信息，简称消息或操作；
 - 同步消息：请求必须有应答才能发送后续消息；
 - 异步消息：无须等待消息应答就可发送其他消息；
 - 创建消息：创建一个对象实例的消息，SSD中不需要；
 - 删除消息：删除一个对象实例的消息。





案例-2：餐馆点餐场景

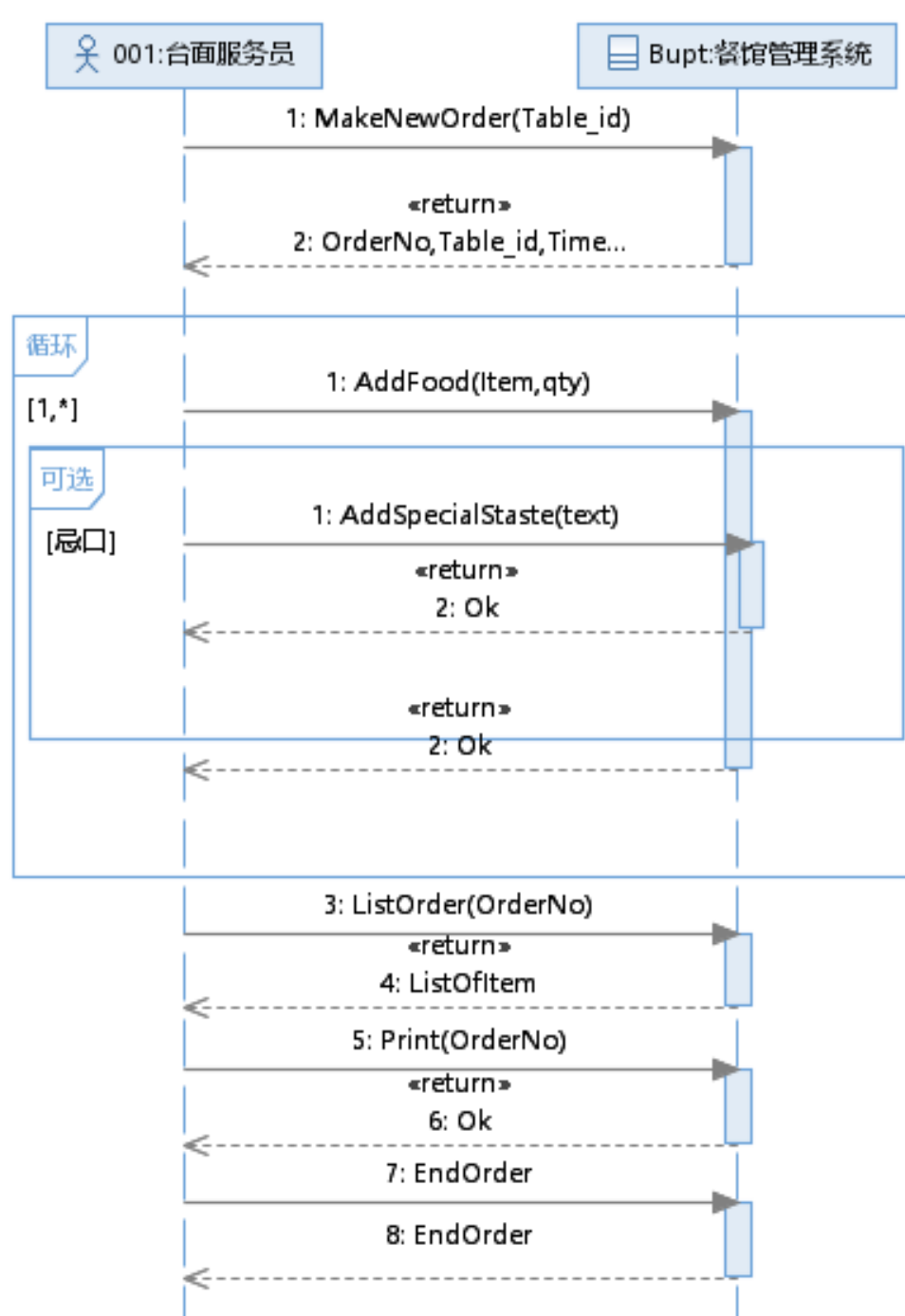
- 角色：台面服务员，负责台面客人的点菜，生成订单；
 1. 客人进入餐馆，安排就坐；
 2. 服务员将菜单递给客人阅览；
 3. 客人呼叫台面服务员点菜；
 4. 服务员手特点菜POS机（或者点菜小本）开始记录菜品信息；
 5. 同时记录菜品的忌口信息；
 6. 直到客人表示完成点菜，生成一张订单并打印；
 7. 完成一次点菜。
- 考虑该场景中的领域模型中的概念类



系统顺序图 (SSD)

注意:

- 1.SSD是用于替代用例说明文本的一种方式;
- 2.图中只有两个对象,表示角色对象与系统对象;
- 3.图中的消息名称及参数要求以可编程的方式命名;
- 4.消息名称和参数可以通过一个列表使用中文说明具体含义;
- 5.用例图中的每个用例都应该对应一张SSD;
- 6.角色发给系统的指令(系统事件)是操作契约关注的元素



- 系统操作：即角色发给系统对象的请求，也称为系统事件；
- 操作契约：契约的含义表示该系统接收到该系统请求后，为了能够得到按照约定必须返回的结果，**该系统内规定必须有哪些业务对象（参考领域模型中的概念类）必须被激活、关联（协同）并记录哪些必须持久保存的数据（概念类属性值的改变）**。它是后续软件设计活动中确定软件对象的参考依据。

操作：

操作以及参数的名称

交叉引用：

（可选择）可能发生此操作的用例

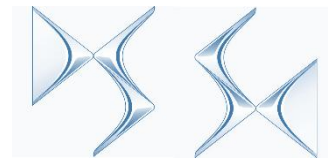
前置条件：

执行该操作之前系统或领域模型对象的状态

后置条件：

操作完成后领域模型中对象的状态：

- 1、对象的创建和删除；
- 2、对象之间“关联”的建立或消除；
- 3、对象属性值的修改；



- 创建操作契约的指导原则如下：
 - 根据系统顺序图**识别**进入到系统内的所有系统事件，即操作；
 - 针对每一个系统操作结合对应的领域模型，找到与此操作相关的概念类；
 - 定义概念类响应该操作的以下三项内容；
 - **对象实例创建和删除；**
 - **对象关联形成和断开；**
 - **对象属性修改。**
- 后置条件中至少有一项存在，该操作才有存在的必要性！
- 后置条件的陈述应该是声明性的，以强调系统状态所发生的变化，无需考虑如何设计和实现的。

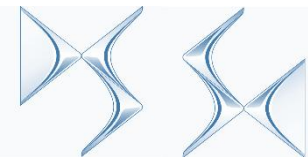
- MakeNewOrder (Table_id) : 开始一个点菜服务请求;
- AddFood (Item_id, quantity) : 添加菜品;
- AddSpecialTaste (text) : 添加特殊忌口信息;
- ListOrder (OrderNo) : 列出所选菜品清单;
- Print (OrderNo) : 打印所选菜品订单;
- EndOrder () : 结束一次点菜服务;



MakeNewOrder (Table_id)

- 参考领域模型分析并说明：开始一次点菜服务，说明要创建一个新的订单。
 - 根据领域模型，该订单必须与台面相关联；
 - 还必须与管理该台面的服务员相关联；
 - 除此之外，该订单流水号、订单创建时间必须被动态创建；
 - 为了能够记录多项菜品信息，还必须初始化一个“集合”项来添加多个菜品；

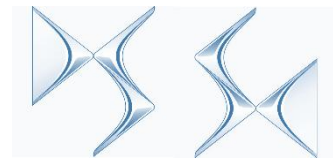
系统事件	MakeNewOrder (TableId)
交叉引用	订单处理
前置条件	服务员身份验证通过，开始订单处理
后置条件	<div>1. 一个新的（概念类）订单被创建； 2. 订单与（概念类）台面建立关联； 3. 订单与（概念类）台面服务员建立关联； 4. 订单的属性初始化：订单流水号、订单时间、存储菜品的数组等</div>



AddFood (Item_id, quantity)

- 参考领域模型分析并说明：在记录顾客所点菜品的同时，说明菜品的对象被创建；
 - 同时隐含说明订单与菜品之间的关联被创建；
 - 除此之外，为了能显示该菜品的详细信息和价格，订单还必须通过菜品Id与菜品描述概念类建立关联；
 - 顾客所点菜品的数量，比如5碗米饭等，将被赋值；

系统事件	AddFoodItem (ItemId, quantity)
交叉引用	订单处理
前置条件	服务员正在处理订单
后置条件	<div>1. 一个新的（概念类）菜品被创建； 2. 菜品与订单建立关联； 3. 订单与菜品描述建立关联； 4. 菜品属性被修改：quantity;</div>



课后练习



课后练习1：请各位同学给出以下系统事件的操作契约

`AddSpecialTaste(text);`

`ListOrder(OrderNo);`

`PrintOrder(OrderNo);`

课后练习2：给出餐桌二维码自助点餐的SSD及操作契约

- 面向对象的需求分析结果：需求分析规格说明书，由以下两个部分组成
- 领域模型由以下两个部分构成：
 - 业务背景知识：概念类及概念类之间关系构成的类图，UML class diagram
 - 业务流程：由UML活动图表示的业务对象之间为了完成某个活动所执行的一系列子活动和动作序列，参见附录附录三
- 用例模型由以下四个部分构成：
 - 用例图，UML use-case diagram
 - 用例说明，use-case descriptions
 - 系统顺序图，UML sequence diagram
 - 操作契约，Operation Contract