

1. GIt
2. 23种设计模式
3. 基本概念：危机、要素、分类、本质
4. 软件过程：生命周期、过程流、过程模式、**模型**、PSP（个人软件过程）、**敏捷开发/建模、极限编程**、人员与团队
5. 实践原则：指导过程的原则、Hooker原则、（需求、设计）建模原则、部署/测试原则
6. 需求：用例图、需求建模、UML活动/泳道图、基于类建模、描述属性、CRC模型（类-职责-协作者）、导航/功能/行为模型/建模、状态表达、系统状态、内容建模、数据树、顺序图、交互模型、状态图、
7. 软件设计：

软件设计内容/过程/目标/原则/评估/方法/概念、软件体系结构、模块化、切面、重构、面向对象的设计、设计类特征、设计模型、数据设计/对象、体系结构/接口设计元素、

体系结构、架构模式分类、构件、OCP（开放封闭原则）、LSP（替换原则）、DIP（依赖倒置原则）、ISP（接口分离原则）、内聚与耦合、构件级设计、协作图、WebApp构件、基于构件的开发/流程、SE/CBSE、构件合格性检查、OMG/CORBA（公共对象请求代理）/ORB/COM（构件对象模型）、JavaBean、构件分类、

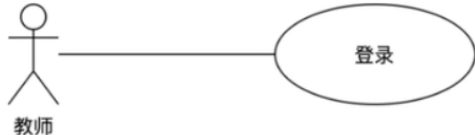
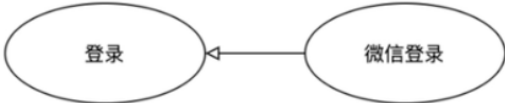
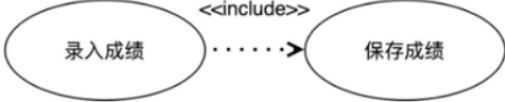
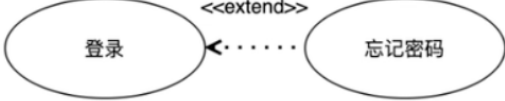
界面设计（原则）、任务分析和建模、工作流分析（泳道图）、
8. 面向对象的设计模式：分类优缺点、6大原则、以及各种设计模式（工厂等）
9. 体系结构：软件模式层次、体系结构作用/目标、分类（2处）、“混沌到结构”/Layers/Pipes&Filters/黑板 Blackboard/分布式系统/代理Broker/交互系统/MVC/PAC/"适应性系统"/微内核/反射**模式**、C-S/层次消息总线/HMB/异构结构/SIS/SASIS/OMA**风格**、webService架构、SOA面向服务的体系结构、P2P、DSSA、三层结构
10. 软件质量：SQA
11. 软件测试：方法、回归/冒烟/通用/面向对象/webApp/移动应用/高阶测试、白盒/基本路径/控制结构/黑盒测试、流图和图矩阵、基于图的测试方法、等价类、边界值分析、基于模型的测试、oo测试、面向对象测试方法、随机测试、划分测试、类级别/类间测试、webApp测试、测试过程
12. 软件配置管理SCM：中心存储库、SCM过程、对象标识、版本控制、变更控制、配置审核、状态报告、WebApp、配置对象、内容管理
13. 产品度量：度量框架、需求模型/功能的度量、规格说明/设计模型/体系结构的度量、面向类的度量、构件设计/界面/webapp的度量、
14. 项目管理：团队、“毒点”、敏捷团队、合并产品以及过程、W5HH原则、过程/项目测量/度量、面向对象/规模度量、项目估算、进度安排、进度跟踪
15. UML示例：
16. UML：面向对象/过程方法、统一过程RUP、用例驱动、视图、对象分析方法、UML核心元素：用例、业务实体、包、类、关系、组件、节点。UML核心视图：静态视图（用例、业务、概念、系统、概念层/说明层/实现层类）、包图（领域包图、层次包图）、动态视图（用例/对象活动图、泳道、状态图）、时序图、协作图、概念/系统用例模型、领域模型、分析模型、架构
17. UML：
18. UML示例：

用例图

参与者：小人；

用例：椭圆；

系统边界：方形

关系类型	说明	表示符号	
关联	参与者与用例的关系		
	泛化	参与者之间或父用例和子用例之间的关系	
	包含	用例之间的关系（包含用例一定会执行）	
	扩展	用例之间的关系（扩展用例不一定会执行，满足某种条件才会执行）	

类图

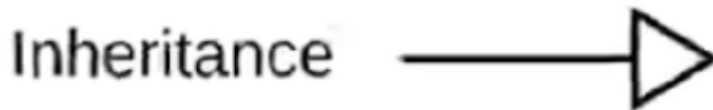
Zoo System



Visibility

- private
- + public
- # protected
- ~ package/default

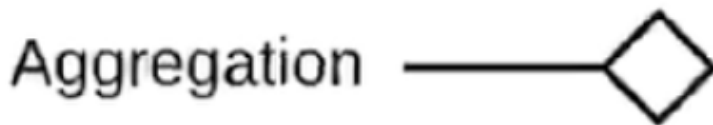
- 抽象类可以用斜体或者<<>>包裹



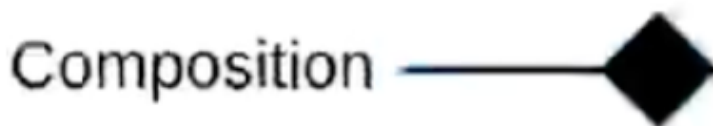
继承



关联：如狗吃肉



聚合：一群乌龟的一群
一个可以存在于之外

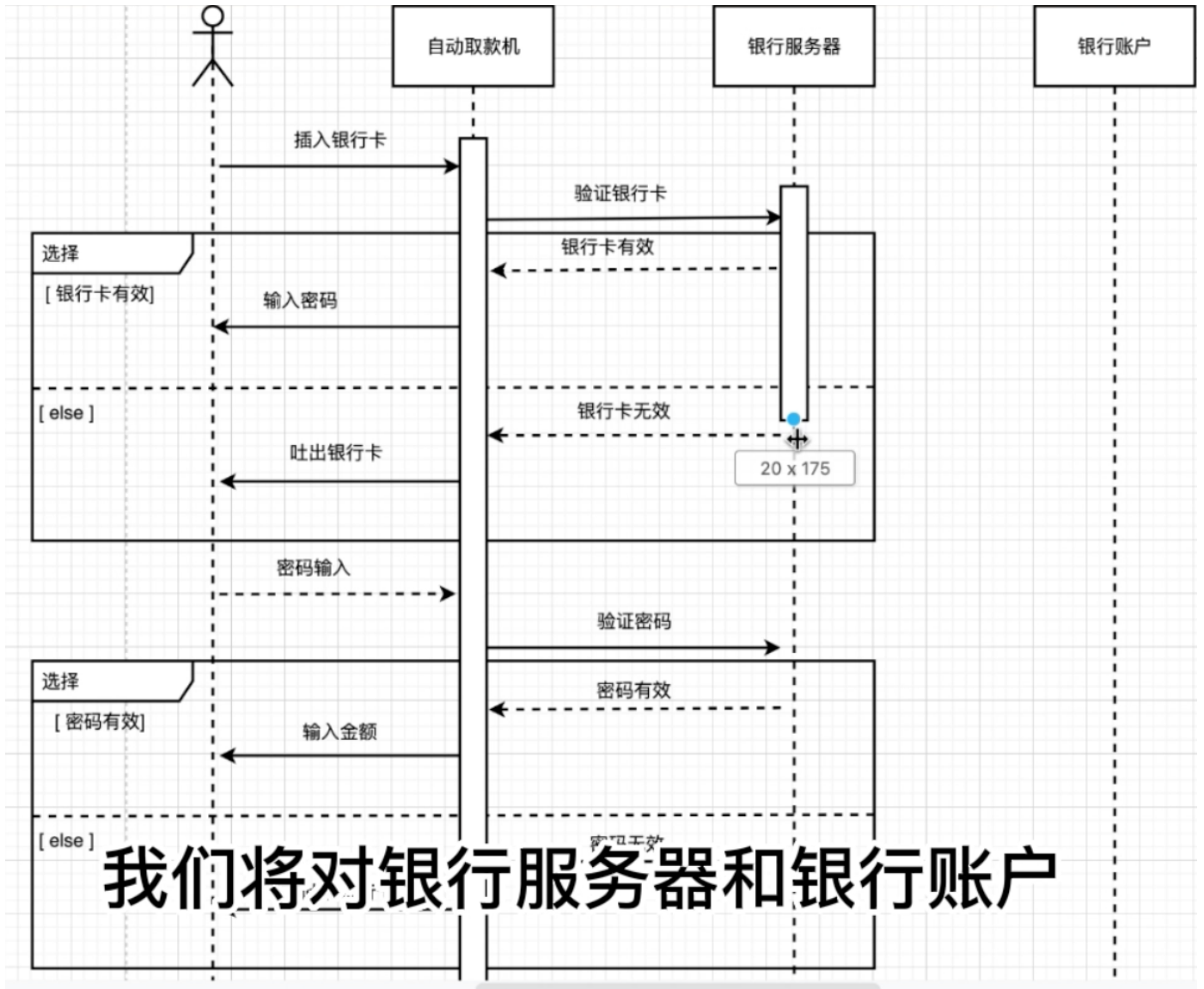


组合：游客中心有大堂
和卫生间。拆中心大堂和
卫生间也没了

- 其中组合可以用 n 0..* 1..n 表示对应关系

时序图

- 虚线表示响应的消息（请求是实线）
- 一个大框框内可以表示if-else（选择）

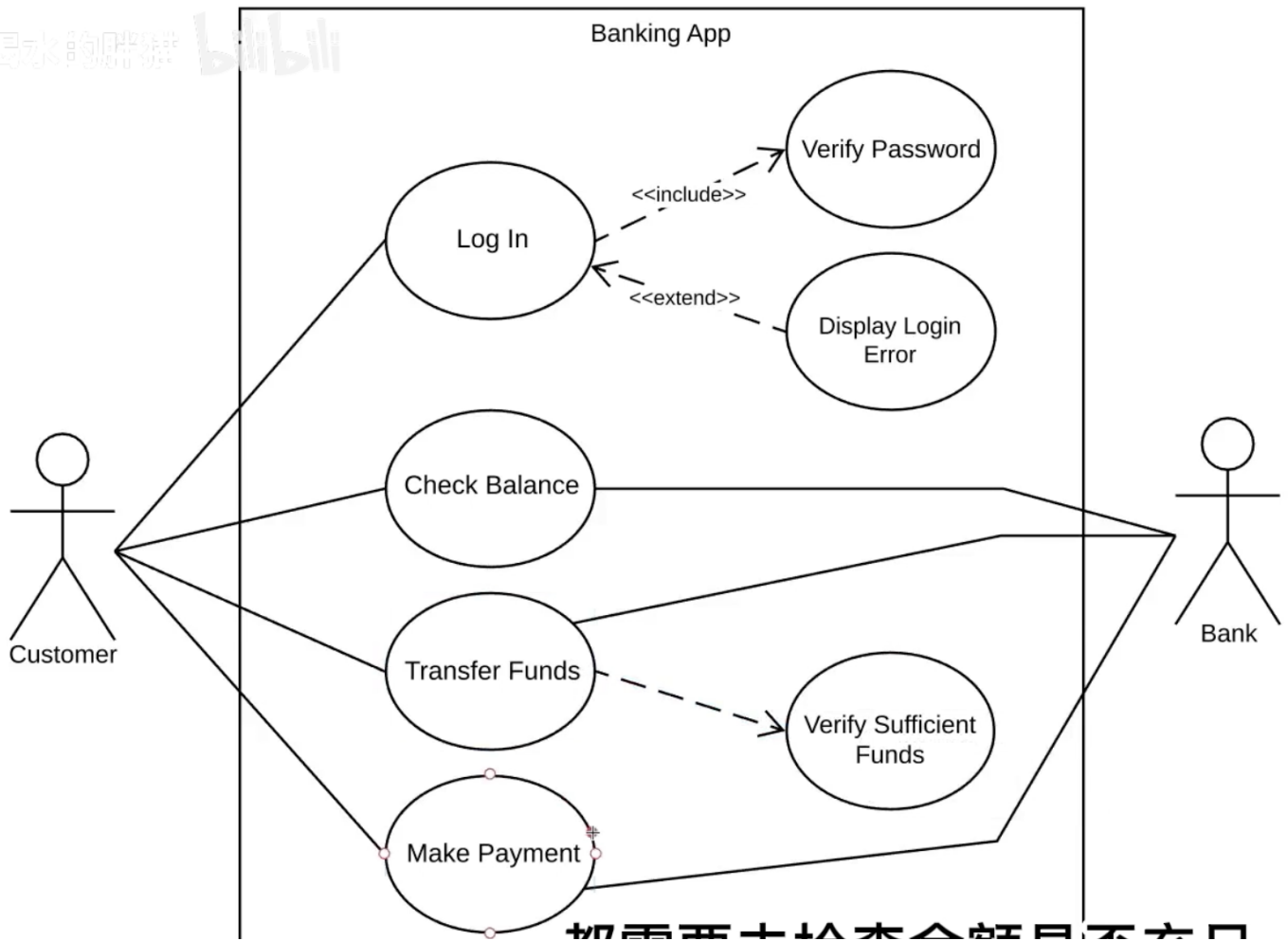


我们将来对银行服务器和银行账户

- 最后把生命线（垂直虚线）变为激活框（从第一条消息到最后一条消息）

用例图

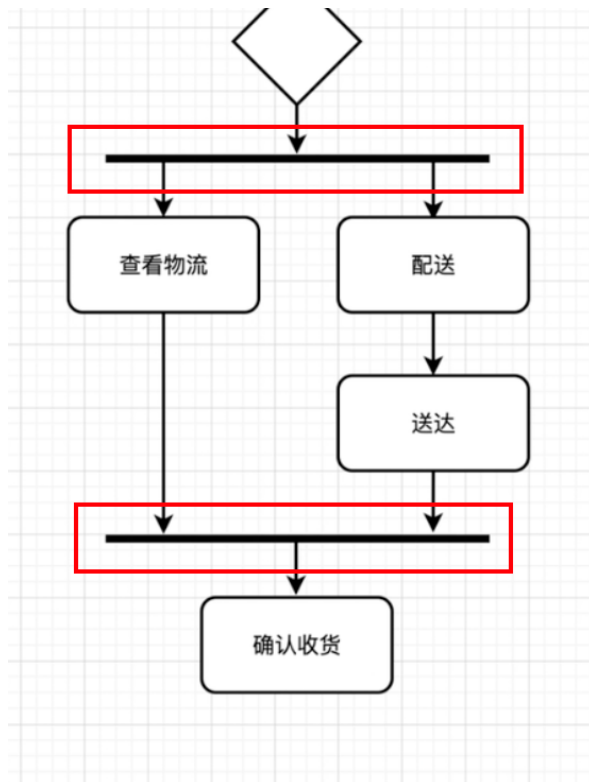
1. 系统：矩形
2. 小人：参与者：主要参与者（系统/矩形 左边）、次要参与者（右边）
主要参与者作出动作后，次要参与者响应
3. 用例：椭圆形状。在系统（矩形）内部。
 1. 包含关系：带箭头虚线。基本用例执行时包含用例也执行（登陆--->密码验证）
 2. 扩展关系：带箭头虚线，满足条件时才触发。注意是扩展用例指向基本用例（错误提醒--->登陆）
 3. 泛化/继承关系：带箭头实线，子用例指向父用例（从支票账户付款—>付款）



活动图

- 开始/结束节点：实心圆。开始只能有一个，结束可以多个
- 动作：圆角矩形，用文字描述任务
- 决策：菱形
- 合并节点：菱形

- 并行进行以及等待：



泳道图

将活动图中不同角色的任务分开来，更加清晰

