判断10

单选15

填空10

名词10

简答25

应用40

1. 系统思想
2. 系统的特性 p4 七点特性及其内涵
3. 系统工程方法p13 六个方面 霍尔三维结构每方面七点

第二章

1信息的定义及其性质 p22 八点基本性质

2 管理中的信息 p25 根据管理层次可分成三类，各类的基本内涵

3 信息系统的基本功能 p28 有六点 信息识别有三种方法、信息传输6个部分

4 信息系统的结构 p30 信息系统的概念结构4部分、管理职能逻辑结构七个子系统、

信息系统的物理结构2类，各自的概述，分布式有四种

第三章

1 UML的主要内容 p60 五种视图 八种图 各自的基本内涵

2 信息系统的生命周期p62 五个阶段、各阶段的含义、以及产生的文件

3 基于生命周期的开发方法共五种 各自的优缺点p64

4 结构化开发方法解百纳思想、面向对象开发方法（oop）基本思想有四点 p71

5 若兰模型六个阶段p73

第四章

1 系统规划的任务p83 三个系统规划的特点4点，系统规划的原则五点

2 系统规划的技术和方法

1. 战略目标转移法2个基本概念，信息系统规划过程三个步骤
2. 企业系统规划法四个基本步骤，第二步三种方法，第三步2种方法，UC方法
3. 关键成功因素发四个步骤

3 可行分析论证的内容3各方面p97

第五章系统分析概述

1 需求分析p106

用户需求与系统需求的 2 方面需求、

功能性高求和非功能性需求、用什么模型可以表达

需求分析的方法有4个

2 系统说明书的内容

说明书 3 方面内容。

项目概述 5 部分、实施计划 3 部分、说明书的 6 点品质

第六章流程建模

1 绘制业务流程图的注意事項 7点p113

2 数据流的注意事项p130

层次划分 3 点标准、正确性检查 4 点、易理解性 3 方面

3 数据字典的作用

数据字典的 6个条目，数据元素 5 点属性、数据结构 3 种特殊

数据流 5.种属性、判定表、判定树

4 举例

跨职能流程图 P112

活动图 P112

第7章用例建模

1 用例描述p150

有5 方面内容、事件流的书写准则 5个、非功能性需求

2 建立用例的关系p154

3个关系、含义以及其表示方法

包含关系|

扩展关系

泛化关系

用例分组：分包

包含关系和扩展关系的异同：

相同点：都是一个用例(基本用例)的植为包含了另一个用例(包含用例或扩展用例)，也就是说都是基本用例的行为的一部分。不同点：在基本用例的每一次执行时，包含用例都一定会执行，而扩展用例只是偶尔被执行。

3 用户故事

4 举例

绘制用例图 P148

书写用例规约（主事件流)，采用双列格式P152

第8 章领域对象建模

1 主要概念:

O对象p163

对象是一些属性及专用服务的封装体，它是问题城中一些事物的抽象。

O类

是指有相同屬性和服务的一组对象的抽象概念（concept)。

类是静态描述

O实例

一个具体的对象（object）就是该对象所在类的一个实例

对象是动态的

O抽象类（abstract class）

用来表征我们在对问题领域进行分析、设计中得出的抽象概念。

如“形状”就是一个抽象概念，圈是具体概念。

O接口（interface）

是抽象类的变体，仅仅代表共同行为的抽象，其它类需要实现某个接口时才对这个接口的所有行为进行定义。

如很多物体都有“开”和“关”的操作，

O封装（Encapsulation）

封装即信息隐藏，它保证对象或部件具有较好的模块性。

类是更高一个级别的封装体，它把数据和服务封装于一个内在的整体。

O消息

是指向对象发出的服务请求

一个消息应当包含以下信息：消息名、接收消息对象的标识、服务类型、输入信息、回答消息

O继承

是指对特殊类的对象拥有其一般类的全部属性和服务

O多态

指相同的操作可作用于多种类型的对象并获得不同的结果

覆盖 重写，运行时多态，父类和子类之间的多态性

重载 静态多态 参数不同，编译时多态，一个类中的多态性

O关系

两个类之间，或类和接口之间的

继承/泛化 ：对象分类的层次关系

实现：对接口进行实现

对象之间

关联：多个对象实例之间的一种长期关系

依赖：对象实例之间的一种短期关系

2识别领域对象

O类图

对象及其关系，用于描述系统静态结构

O状态图

对象的状态及转换，用于描述基于事件响应的对象动态行为和状态之间的关系

O领域对象

是问题域中有意义的概念类，他们是显示系统中的事物或事件

O识别领域对象的方法

名词短语策略：摘取用例的详细文档中的名词（术语或名词短语），然后进行分析

使用概念类别列表，七种概念类p170

业务表格分析法（表格项即事物属性，类聚抽象出事物）

3 识别对象属性

发现属性策略，保持属性的简单性p172

4 识别对象的关联

描述关联的要素关联名称、角色、多重性、导向性

整体-部分关联p176

组合聚集

共享聚集

5 识别泛化关系

6 类图的画法p180

状态机图，三种状态p184

7 举例

名词短语p180

业务表格

第9 章系统设计概述！

1 系统设计的目标，6 个方面衡量

基本设计方法p189

2 系统设计的内容 p190

应用架构设计、接口层设计、业务逻辑设计、数据层设计、技术架构设计

3 软件设计方法

（1）面向过程的设计方法

模块、模块结构图、

模块的联系：耦合 3 方面因素（联系方式、来往信息的作用、模块间来往信息数量)，模块内聚 7 类内聚 p196，扇入扇出

模块设计原则：低耦合、高内聚，如何理解

（2）面向对象的设计方法

类图p197例9.4

类图——Java代码

(3）面向服务的设计方法

（4）事件驱动的设计方法

4 系统设计说明书，2 种形式

第 10 章系统应用架构设计

1 概念

架构、结构、应用架构、架构模式、软件框架

2 分层应用架构设计

三层、五层次架构模式。

3 MVC 架构模式，3 个基本部件、职责 p213，与经典三层架构的对应关系

4 Spring框架

第 11 章接口层设计

1 人机交互设计原则，以人为本，5 点p231

2 软件接口设计

构件图 p246

第12章面向对象的业务逻辑层设计

1 类的属性

属性类型和初值

属性的可见性

公有（public）“+”

受保护（protected）“'#”

私有（private）“-”

包（packnge）“~”

2 类的关系

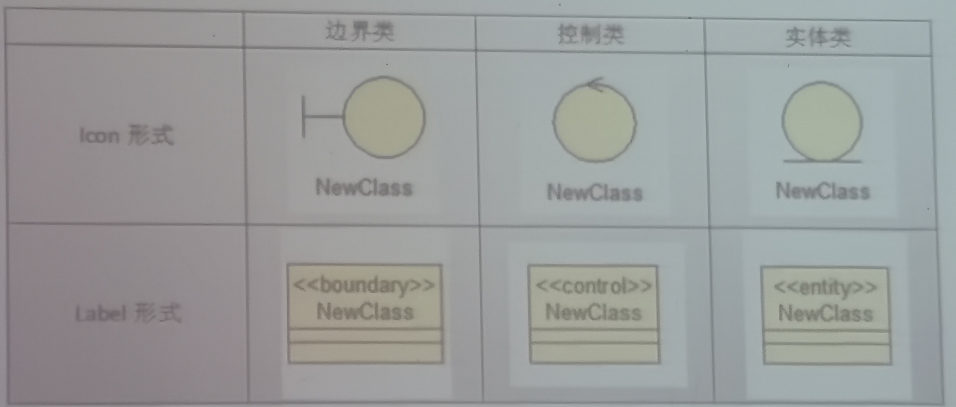
泛化关系（类与类)

关联关系（对象间）

实现关系（类与接口）

依赖关系（对象间）

3 根据应用架构设计类 3种



实体类

边界类

控制类

4 包的含义、分包的两种原则

共同封闭原则

共同复用原则

5 顺序图(Sequence Dingram)

顺序图的基本元素有5 个 p260

绘制順序图 p263

绘制协作图/通信图 p264

协作图与顺序图异同:

相同点：顺序图和协作图都属于交互模型，都用于描述系统中对象之间的动态关系，

两者可以相互转换。

不同点：两者强调的重点不同，顺序图强调的是消息的时间顺序，而协作图强调的是参与交互的对象的组织(即空间关系或结构关系)：在两个图所使用的建模元素上，顺序图中有生命线和控制焦点，协作图中没有；协作图中有路径，顺序图中没有：协作图中的消总必须要有消息顺序号，但顺序图中可以有也可以没有。

第13章 面向服务设计

定义系统操作

定义一组软件服务

定义服务API和协作方式

第14章数据层设计

1 领域类映射到表

2 对象关系映射（object relationnl mpping，ORM）就是为了解决面向对象与关系数据库之间数据转換的一种技术

ORM 框架

Hibanute 框架

Spring Boot 的 JPA

输入作者，查询该作者的图书。使用 Spring Boot JPA

MyBatis框架|

3 NoSQL泛指非关系型数据库，不仅仅是”NoSQL”，还可以理解为Not Only SQL”。

第 15 章系统实施

1 实施阶段的主要活动 p305

硬件、软件、数据、人员 4 个方面

4 种实现策略、版本划分需要考虑 5 个方面

2系统集成与构建

3系统测试p315

概念、3 种手段检验软件、测试有 5 种级别、测试有4 个原则

测试用例

一个测试用例就是为了测试某个目标（模块、功能、性能）而准备的一份输入数据及其预期结果

测试技术：白盒测试、黑盒测试

4 系统部署、系统迁移

第16章系统运维与管理

1 运维管理

流程管理、制度管理、人员管理

2 开发运维一体化 DevOps

(Development 和 Operations 的组合词，开发运维一体化）

3 运维自动化

AIOps 智能运维

画图

1、类图（、对象图）

类的属性、类的操作，

类之间的关系（关联、聚合、泛化、依赖）

类版型（实体类 entity、边界类 boundary、控制类 control） P152

用 UML 画类图

在软件开发的不同阶段使用的类图具有不同的抽象层次。

**领域模型**——从面向对象的视角看待现实世界-主要工作是找出相关类-然后明确它们的关系-必要时加入一些多重性描述和业务规则-不涉及具体语言

**分析模型**——从领域模型将得到实体类-对软件系统进行分析-可以得到边界类:描述的是软件的接口-不是软件的实现-最利于开发者使用和交流的类图

**设计模型**-加入了抽象类-接口等设计元素-加入了设计模式等-描述了类的实现细节-可

以直接映射到可执行代码-因此-涉及具体语言和设计模式等

**4-13.根据下面的陈述画出类图。**

1. 学生包括本科生-研究生两种：2)研究生的一部分利用课余时间担任助教；3)教师包括助教、讲师和教授三种： 4)一名助教可以为一位讲师或一位教授助课，一位讲师只能有一名助教，一位教授可以有 5 名助教。



2、用例图（相当于功能模型）

概念、组成元素、用例规约（主事件流）

用例之间的关系(包含、扩展、泛化)

建模步骤、用 UML 绘制用例图

3、泳道流程图（相当于业务流程图）与活动图

4 顺序图与协作图（通信图）p264

【02】 ATM 自动取款机系统1 .ppt

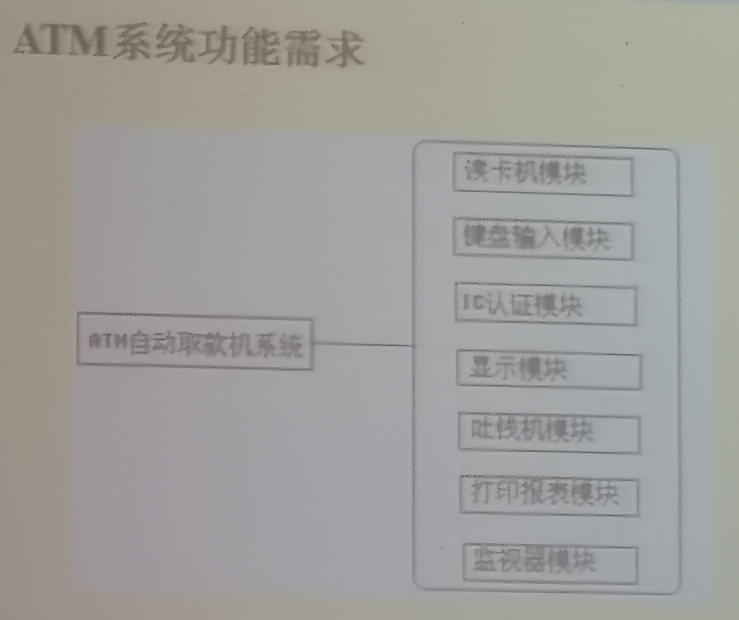
案例-志愿者服务管理系统:pdf

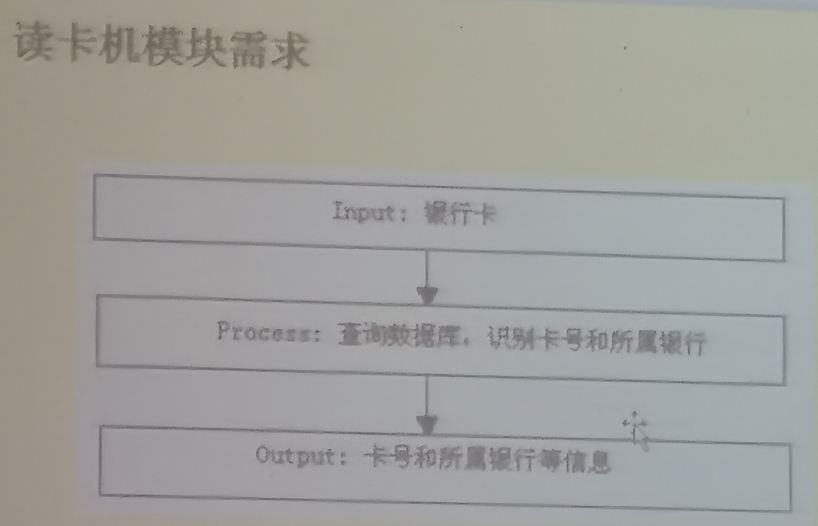
一、需求分析

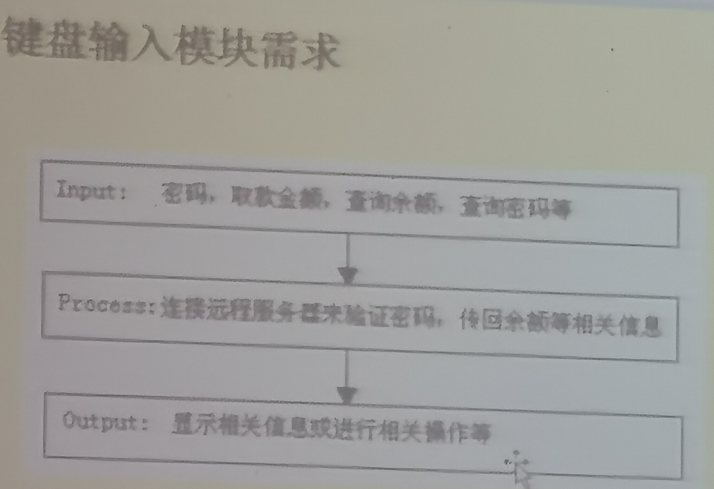
**系统总体功能需求**

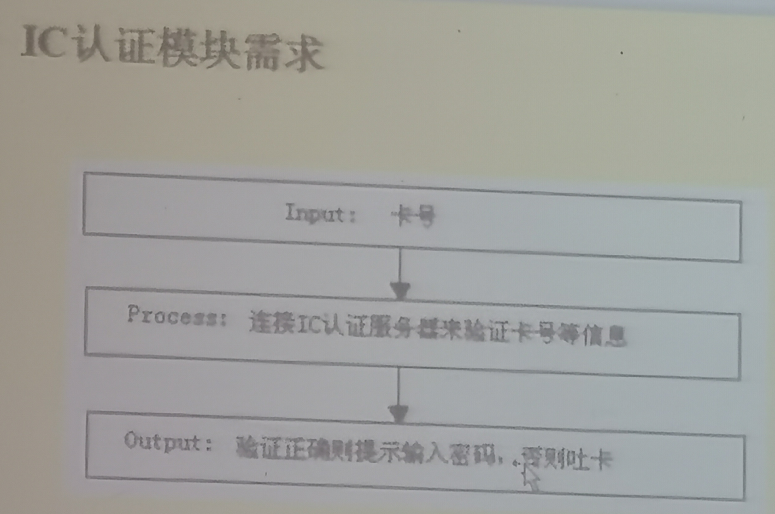
**一个功能完全的ATM系统，必须包括以下的几个模块**

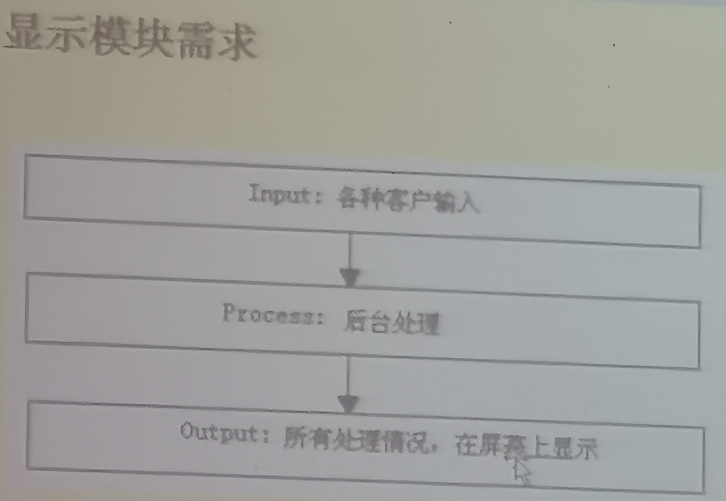
1. 读卡机模块
2. 键盘输入模块
3. IC认证模块
4. 显示模块
5. 吐钱机模块
6. 打印机报表模块
7. 监视器模块

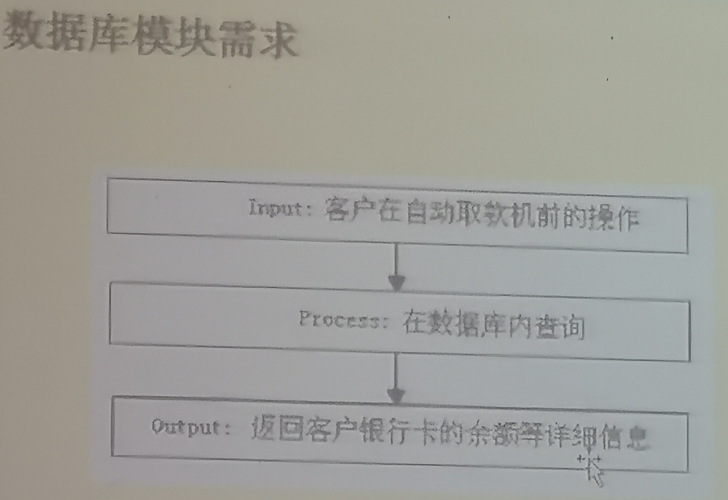
****

****

****

****

****

****

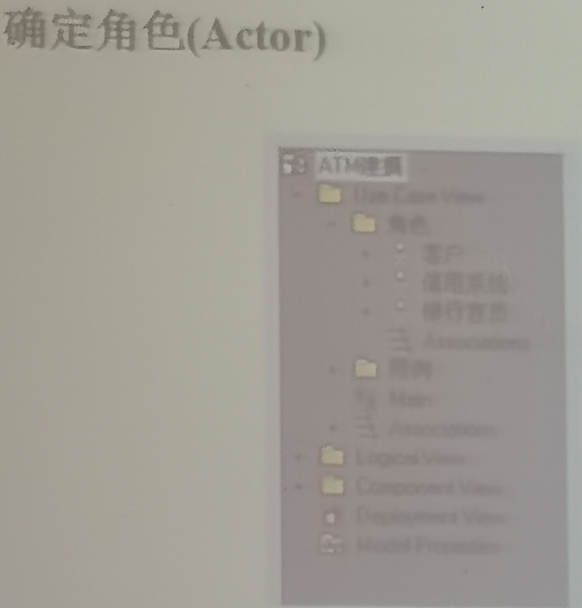
**二、系统用例模型**

建立用例视图分为以下几个步骤

确定角色（Actors）

创建用例（Use Case）

创建角色（Use Cases） ——用例（Use Case）关系图



**创建用例**

用例是角色启动的，基于这样的考虑，ATM系统根据业务流程大概可以分为以下几个用例：

客户取钱

客户存钱

客户查询余额

客户转账

客户更改密码

客户通过信用系统付款

银行官员改变密码

银行官员为ATM添加现金

银行官员维护ATM硬件

信用信用启动来自客户的付款

**创建角色——用例关系图**

客户的角色——用例关系图一共有六个用例

转账

查询余额

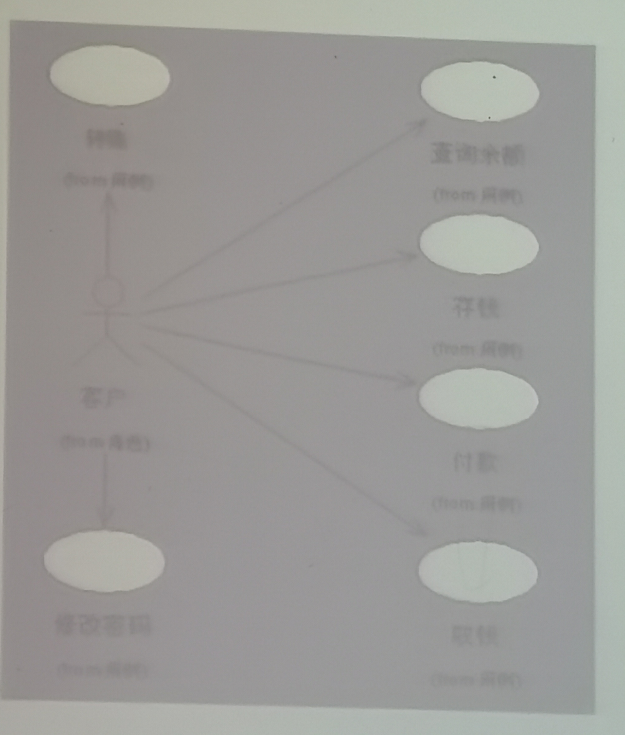
存钱

付款

取钱

修改密码

**客户的用例关系图**



**用例描述(Use Case Specification)**

错误示例1

Use Case: Withdraw Cash

参与者：Customer

主要事件流：

（1）储户插入 ATM 卡，并输人密码。

（2）储户按Withdraw 按钮，并输人取款数目。

（3）储户取走现金、ATM 卡并拿走收据。

（4) 储户离开。

改进：

Use Case: Withdraw Cash

参与者：Customer

主要事件流：

（1）通过读卡机，储户插入ATM卡

（2）ATM系统从卡上读取银行ID，账号，加密密码，并用主银行系统验证银行ID和账号

（3）储户输入密码，ATM系统根据上面读取出的卡上加密密码，对密码进行验证

（4) ATM系统进入设置取款金额页面，储户输入取款数额，取款金额应是100或50的倍数

（5）ATM系统通知银行系统，传递储户账号和取款金额，并接受返回的确认信息和储户账户金额

（6）ATM系统输出现金，ATM卡显示账户余额的收据

（7）ATM系统记录事务到日志文件

动态模型包括许多框图：活动框图、序列图、协作图等

创建这些框图目的是为了更好地了解业务流程

“开户”的活动图

