



软件工程

第6章 面向对象分析

重庆理工大学
计算机科学与工程学院 李梁


qq:1255214405
liliang@cqut.edu.cn

智慧树课号: K3388095
智慧树网址: www.zhihuishu.com

课程号: K3388095




18:01:00 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



第6章 面向对象分析

- 1 面向对象分析概述
- 2 用例驱动分析: 建立功能模型
- 3 对象模型基础
- 4 建立静态模型(对象类模型)
- 5 建立动态模型(行为建模)
- 6 通过用例图建立领域模型实践
- 7 小结

2 18:01:00 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



1 面向对象分析概述

结构化分析方法: 面向功能


软件结构严重依赖于功能, 而功能是软件开发中最不稳定的因素。
数据和操作相分离

OOSD (Object-Oriented Software Development) 是一种把面向对象的思想应用于软件开发过程, 指导开发活动的系统方法。

面向对象 = 对象 + 类 + 继承 + 通信

Object-Oriented objects classes inheritance communication

1 18:01:00 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



1 面向对象分析概述

OOSD由三部分组成

OOA(Object-Oriented Analysis) 面向对象的分析	定义系统中的对象特征和行为, 建立系统的三类模型(对象、动态、功能)。
OOD(Object-Oriented Design) 面向对象的设计	与OOA密切配合顺序实现对现实世界的进一步建模。
OOP (Object-Oriented Program) 面向对象的编码	使用面向对象的程序设计语言进行编码。

2 18:01:00 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



1 面向对象分析概述

对象 (Object) 是将一组数据 (属性) 和对该数据的操作或过程封装在一起的实体。


案例 某部智能手机可以看作是一个对象。

数据 (属性): 品牌、厂商、价格、处理器、运存、闪存、屏幕、电池、摄像头等

操作 (过程): 通讯、联网、拍照、导航、娱乐、支付、社交、语音识别、家居控制等

数据封装 并行性 模块独立性 通信性

1 18:01:01 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)




1 面向对象分析概述

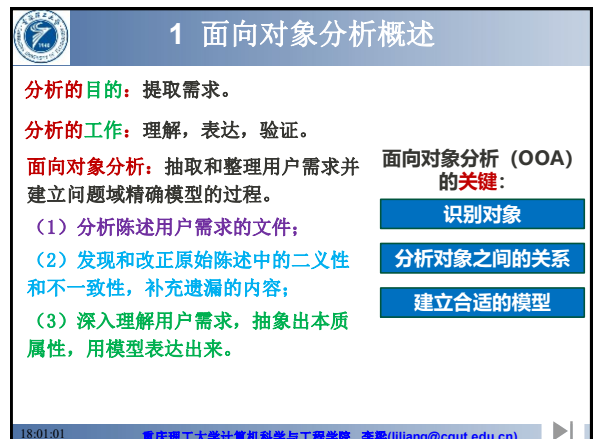
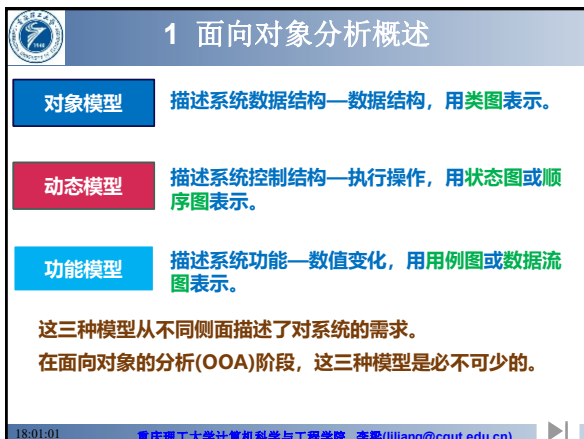
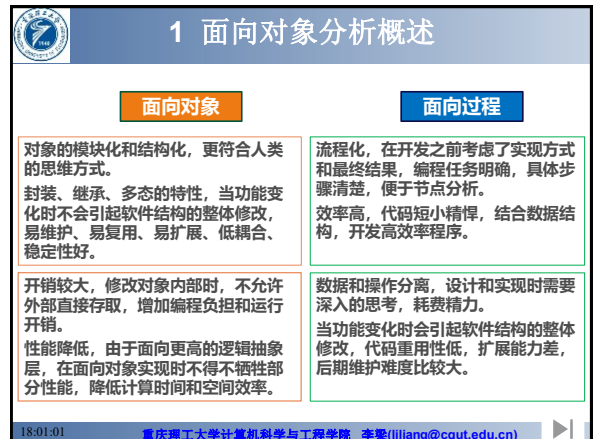
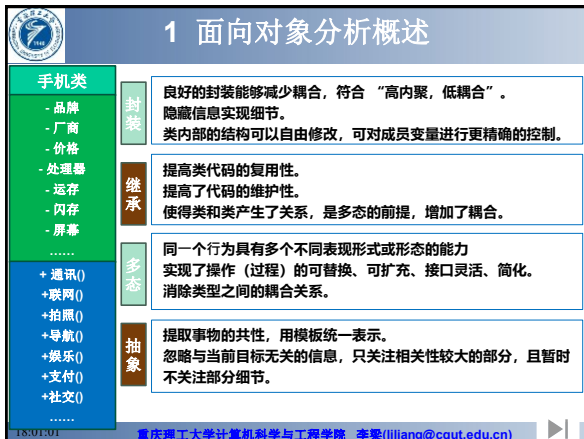
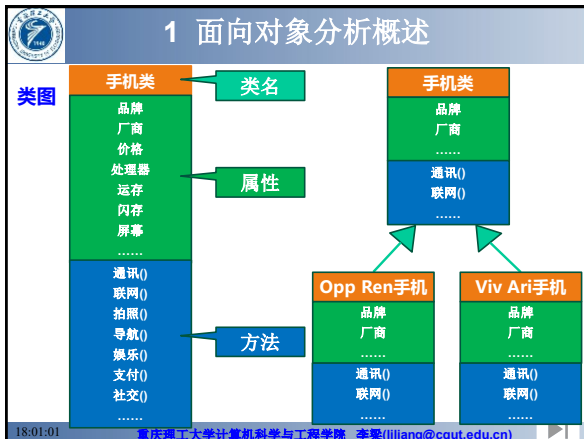
类 (Class) 是对具有相同属性和行为的一组对象的抽象。模式、模板

对象是类的实例化

- * 使用手机配置 (类) 可生产 (派生) 出多台相同的手机 (对象)
- * 根据手机配置 (类) 进行继承、修改, 可生产 (继承, 多态) 出多台不同的手机 (对象)



1 18:01:01 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



1 面向对象分析概述

面向对象分析对相关的**所有类及其有关系的行为**进行定义，可通过**对象模型、功能模型、动态行为模型和物理实现模型**，来表达分析结果。

通常需要执行下述**步骤**：

- ①在**客户和软件工程师之间对基本用户需求进行交流**。
- ②**定义类（包括属性和方法）**
- ③**定义类的层次结构**
- ④**定义类与类之间的关系**
- ⑤**为对象行为和物理实现建模**
- ⑥**重复上述步骤直到模型完成**

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

13

1 面向对象分析概述

3个子模型

5个层次及对应的活动

服务层	定义服务
属性层	定义属性
结构层	识别结构
类与对象层	找出类与对象
主题层	识别主题

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

14

1 面向对象分析概述

面向对象分析的步骤：

- ①**需求获取及分析问题域，用例模型。**（第2节）
- ②**发现和定义对象和类。结构模型（类图）**（第3-4节）
- ③**识别对象的内部特征。**
- ④**识别对象的外部联系。**
- ⑤**识别对象之间的交互。动态模型（顺序图、状态图第5节）**

建立Class

建立Use Case

发现对象

定义属性与服务

建立结构与连接

定义：顺序图、协作图、状态图

详细说明

分析过程中各个步骤不要求按固定顺序进行。所以，面向对象的分析步骤经常被叫做“活动”。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

15

2 用例驱动分析-用例模型

用例驱动分析过程

- ①绘制用例图
- ②用例描述
- ③开发活动图
- ④开发泳道图

用例模型(Use case model)：用例模型描述的是**外部执行者(Actor)**所理解的**系统功能**。一个用例模型由若干个用例图描述。

用例模型描述系统只描述做什么，不描述怎么做，它**内部如何工作、用例如何实现**，对用例模型并不重要。

用例图：用来图示化描述用户的需求，即用户希望系统具备的完成一定**功能的动作**，通俗理解用例就是软件的功能**模块**，这些模块之间的协作调用关系。

用例：用例是在一个系统中所进行的一连串的**活动场景**，该活动要能够满足系统**外部的执行者**对系统的**预期**。就是用户对于产品或系统的某一个完整的预期。从另一个角度来说，用例也代表着一个具体的业务场景。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

16

2 用例驱动分析-用例图

用例图包括：参与者、用例、关联和边界四个要素。

- ✓**参与者**：用小人形表示
- ✓**用例**：用椭圆表示
- ✓**关联**：用直线表示说明参与者驱动某个用例
- ✓**边界**：用矩形框表示，说明系统关注点。

参与者

箭头

用例

系统边界

家教信息处理

网站公告发布

家教信息登记

负责人

家教管理系统后台管理用例图

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

17

2 用例驱动分析-用例

用例是系统中可感受到的一个完整的服务（功能）。

- ✓系统完成有价值**结果**的一系列动作，动作的结果能被特定的**参与者**察觉到。
- ✓这些动作除了完成系统内部的计算和工作外，还包括与一些行为者的**通信联系**。关联指出一个用例与哪些行为者交互，这种交互是双向的。
- ✓用例一般使用**动词短语**表示，用例名通常由2~3个单词组成：**动词+名词+形容词**。从参与者而非系统角度来命名，如“**购买物品**”而不是“**出售物品**”

用例具有下述特征：

- ✓用户可见的功能，实现一个具体的用户目标
- ✓是被参与者**启动**的，并向参与者提供确切的**值**
- ✓可大可小，相对完整的，用例之间都是**独立、并列**的，用例之间不存在**从属关系**。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

18

2 用例驱动分析-参与者

- **参与者**：与系统的一次或多次交互中扮演角色的人、物、组织或外部系统。
 - ✓ 是与系统交互的人或其他系统 代表系统外部的实体
 - ✓ 代表一种角色 不是某个具体的人或物
- **关联**（连接参与者与用例直线）：参与者和用例之间的联系，两者间交换（交互）信息。箭头表示关联初始调用方向。
 - ✓ 参与者触发（激活）用例，并与用例交换信息。单个参与者可与多个用例联系；一个用例也可与多个参与者联系。

学生信息注册 家教信息查看 教师信息注册
学生资料修改 家教信息搜索 教师资料修改
学生 教师

重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 19

2 用例驱动分析-用例建模步骤

- 一个用例模型由若干幅用例图组成，创建用例模型步骤：
 - ✓ 定义系统边界
 - ✓ 寻找参与者和用例
 - ✓ 描述用例
 - ✓ 反复细化用例，确定用例间的关系（包含、扩展、泛化）
- 1、寻找参与者：通过用户回答一些问题的办法来发现参与者：
 - ✓ 谁将使用系统的主要功能？谁需要借助于系统完成日常工作？
 - ✓ 谁来维护和管理系统？
 - ✓ 系统控制哪些硬件设备？
 - ✓ 系统需要与哪些其他系统交互？
 - ✓ 哪些人或物对系统产生的结果感兴趣？

重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 20

2 用例驱动分析-寻找用例

2、寻找用例：请每个参与者回答下述问题来获取用例：

- ✓ 参与者需要系统提供哪些功能？
- ✓ 参与者是否需要读取、创建、删除、修改或存储系统中的某类信息？
- ✓ 系统中发生的事件需要通知参与者吗？参与者需要给系统输入某些信息吗？
- ✓ 参与者的日常工作是否因为有了系统的新功能而被简化或提高了效率？

下述的针对整个系统的问题，也能帮助建模者发现用例：

- ✓ 系统需要哪些输入输出？输入来自何处？输出到哪里？
- ✓ 当前使用的系统的主要问题是什么？

重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 21

2 用例驱动分析

图书馆系统
图书借阅者 期刊借阅者 浏览者 图书管理员
预约图书 借出图书 归还图书 续借图书 浏览 更新目录
借出期刊 归还期刊

重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 22

2 用例驱动分析-寻找用例

- 用例建模要点
 - ✓ 用例是可以不断改变的，不是一成不变的
 - ✓ 用例之间没有按照时间排序
 - ✓ 一个参与者可能涉及多个用例
 - ✓ 用例不是功能，而是一项有价值的服务
 - ✓ 关联线上最好无箭头，箭头会使人困惑，无价值
 - ✓ 图不要太大
 - ✓ 每个参与者至少涉及一个用例，一个用例至少设计一个参与者
 - ✓ 关联只存在于参与者和用例之间

重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 23

2 用例驱动分析-寻找用例

用例只描述做什么，而不应描述怎么做。

做什么 怎么做
成绩统计 数据检索 计算 排序
提取现金 插入卡 输入密码 输入金额

重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 24

2 用例驱动分析-用例描述

3、用例描述:

用例使用非正式的描述性风格编写，也可以使用某个结构化的格式编写，有些格式更强调描述的直观性。

用例不同部分	说明
用例名称	以动词开始描述用例名称
范围	要设计的系统
级别	“用户目标”或者是“子功能”
主要参与者	调用系统，使之交付服务
涉众及其关注点	关注该用例的人，及其需要
前置条件	开始前必须为真的条件
成功保证	成功完成必须满足的条件
主成功场景	典型的、无条件的、理想方式的成功场景
扩展	成功或失败的替代场景
特殊需求	相关的非功能性需求
技术和数据变元素	不同的I/O方法和数据格式
发生频率	影响对实现的调查、测试和时间安排
杂项	未决问题等

POS机系统用例描述-例子

- POS机系统是电子收款机系统，通过计算机化用于处理销售和支付，记录销售信息。该系统包括计算机、条码扫描仪、现金抽屉等硬件，以及使系统运转的软件和为不同服务的应用程序提供接口。
- POS机系统为例说明用例建模分析过程。
- 收银员可以记录销售商品信息，系统计算总价。
- 收银员能够通过系统处理支持，包括现金支付、信用卡支付和支票支付。
- 经理还能处理顾客退货。
- 系统要求具有一定的容错性，即如果远程服务（如库存系统）暂时中断，系统必须仍然能够获取销售信息并且至少能够处理现金付款。
- POS机必须支持日益增多的各种的客户终端和接口，比如多种形式的用户图形界面、触摸屏输入、无线PDA等。
- 系统需要一种机制提供灵活的处理不同客户独特的业务逻辑规则和定制能力。

POS机系统部分用例图

POS机系统中处理销售的场景

- 用例名称: 处理销售
- 范围: POS机应用
- 级别: 用户目标
- 主要参与者: 收银员
- 涉众及其关注点:
 - ✓ 收银员: 希望能够准确、快速地输入，而且没有支付错误，因为如果少收货款，将从其薪水扣除。
 - ✓ 售货员: 希望自动更新销售提成
 - ✓ 顾客: 希望以最小代价完成购买活动并得到快速服务。希望便捷、清晰地看到所输入的商品项目和价格。希望得到购买凭证，以便退货。
 - ✓ 公司: 希望准确地记录交易，满足顾客要求。希望确保记录了支付授权服务的支付票据。希望有一定的容错性，即便在某些服务器构件不可用时（如远程信用卡验证），也能够完成销售。希望能够自动、快速地更新帐户和库存信息。
 - ✓ 经理: 希望能够快速执行超控操作，并易于更正收银员的不当操作。
- 前置条件: 收银员必须经过确认和认证。
- 成功保证（或后置条件）: 存储销售信息，更新帐户和库存信息，记录提成，生成票据，记录支付授权的批准。

POS机系统中处理销售的主要成功场景

- ① 顾客携带所购商品或服务到收银台通过POS机付款。
- ② 收银员开始一次新的销售交易。
- ③ 收银员输入商品条码。
- ④ 系统逐步记录出售的商品，并显示该商品的描述、价格和累计金额。价格通过一组价格规则来计算。收银员重复3~4步，直到输入结束。
- ⑤ 系统显示总额和计算折扣。
- ⑥ 收银员告知顾客总额，并请顾客付款。
- ⑦ 顾客付款，系统处理支付。
- ⑧ 系统记录完整的销售信息，并将销售和支付信息发送到外部的账务系统（进行账务处理和提成）和库存系统（更新库存）。
- ⑨ 系统打印票据。
- ⑩ 顾客携带商品和票据离开。

扩展或替代流程场景见教材P147

2 用例驱动分析-用例关系

4、寻找用例间关系

(1) 包含（include）关系（使用关系）: 表示一个用例包含另一个用例，即要完成包含用例就一定要执行被包含用例。

✓ 有时当某用例的事件流过于复杂时，为了简化用例的描述，我们可以把某一段事件流抽象为一个包含的用例；如果若干个子过程，然后从主程序中调用这一子过程。

2 用例驱动分析-用例关系

例如：业务中，总是存在维护某某信息的功能，如果将它作为一个用例。那么新建、编辑以及修改都要在用例详述。用例理清：不想。

没有了包含用例，基本用例的行为不完整

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 31

2 用例驱动分析-用例关系

(2)扩展 (extend) 关系：一个用例扩展到另一个用例，即将基本用例中一段相对独立的并且可选的动作，用扩展用例加以封装，从而使基本用例行为更简练和目标更集中。

对于一个扩展用例，可以在基本用例上有一个或多个扩展点。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 32

2 用例驱动分析-用例关系

例如：系统中允许用户对查询的结果进行导出、打印。对于查询而言，能不能导出、打印查询都是一样的，导出、打印是不可见的。导入、打印和查询相对独立，而且为查询添加了新行为。因此可以

没有扩展用例，基本用例的行为依然完整

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 33

2 用例驱动分析-理解用例关系

在执行“取款”用例的过程必定要执行“银行卡验证”用例。将来在画“取款”的活动图的时候，一定要把“银行卡验证”作为一个步骤加入进去。

扩展用例“还书”有一个“扩展点”即“过期”。当且仅当还书过期的时候，“还书”用例的活动图中才需要加入扩展用例“罚款”，即“罚款”用例是可选的

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 34

2 用例驱动分析-用例关系

(3)泛化 (generalization) 关系：子用例和父用例相似，但表现出更特别的行为；子用例将继承父用例的所有结构、行为和关系。父用例通常是抽象的，实际应用中很少使用泛化关系。用例泛化过程是将不同的用例之间的可合并部分抽象成独立的父用例，并将不可合并部分单独成各自的子用例

例如：业务中可能存在许多需要部门领导审核的事情，但是领导审核的过程很相似，就可以做成泛化关系表示：

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 35

2 用例驱动分析-用例关系

(4)三种关系的区别

包含、扩展与泛化类似，但三者关系的优化侧重点不同的。

- 1、泛化侧重表示子用例之间的互斥性，描述用例间的继承关系
- 2、包含侧重表示包含用例对参与者提供服务的间接性，描述用例间具有的重复行为
- 3、扩展侧重表示扩展用例的出发不定性；描述用例间可选的公用行为
- 4、扩展与使用都是从几个用例中抽取那些公共的行为并放入一个单独用例中，而这个用例被其他几个用例使用或扩展，但目的不同。

按照用例发生条件可分为如下两种情况：

- 1、无条件发生：肯定发生的
- 2、有条件发生：未必发生，发生与否取决于系统状态

泛化与包含用例属于无条件发生用例，扩展是有条件发生用例。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 36

2 用例驱动分析-捕获用例的原则

- 1) 一个用例只描述一个功能，功能不能太笼统。
- 2) 一个用例是在一个相对完整的时间段中发生的。
- 3) 一个参与者可以对应多个用例，一个用例也可对应多个参与
- 4) 一个用例可以对应多个界面，一个界面也可能对应多个用例。

企业信息系统

参与者与用例

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cquf.edu.cn) 37

2 用例驱动分析-捕获用例的原则

- 4) 一个用例可以对应多个界面，一个界面也可能对应多个用例。

订单处理系统用例

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cquf.edu.cn) 38

2 用例驱动分析-从登录看用例中包含、扩展

学生在登录后，有两个功能是被反复使用的。两个功能是登录主程序的从属功能，都是必须要执行的功能。从处理逻辑上看有问题。

学生可选择两个功能，当进入每个功能时，都必须登录。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cquf.edu.cn) 39

2 用例驱动分析-从登录看用例中包含、扩展

学生有三个独立的功能，这些功能之间没有直接的关系，以后增加功能（如修改或添加），对其它的功能没有直接影响。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cquf.edu.cn) 40

2 用例驱动分析-从登录看用例中包含、扩展

条件:
if 选择“选课” then 选课
If 选择“查学分” then 查学分

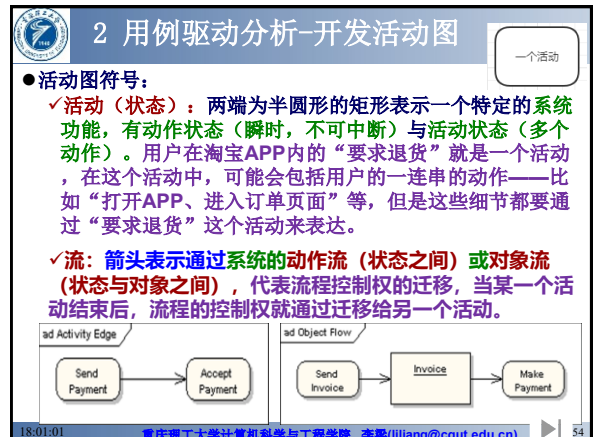
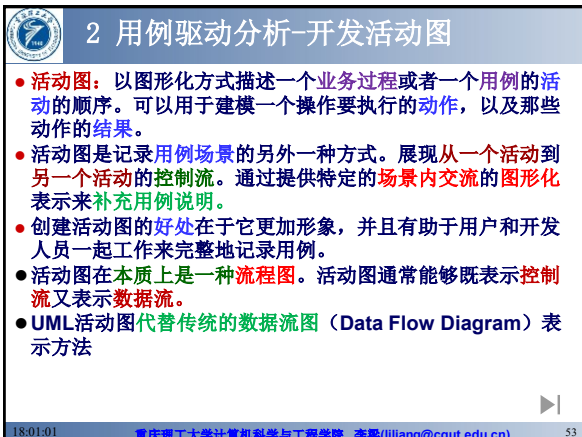
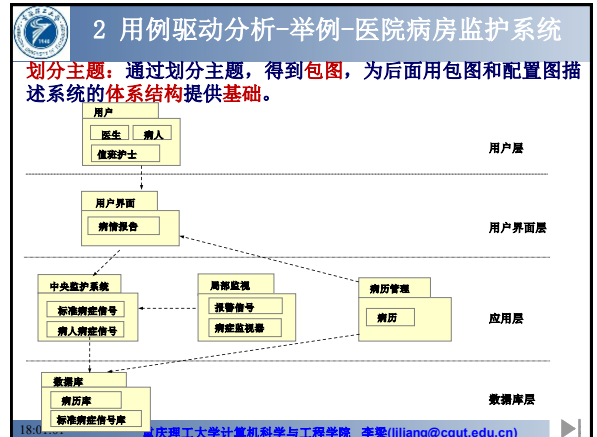
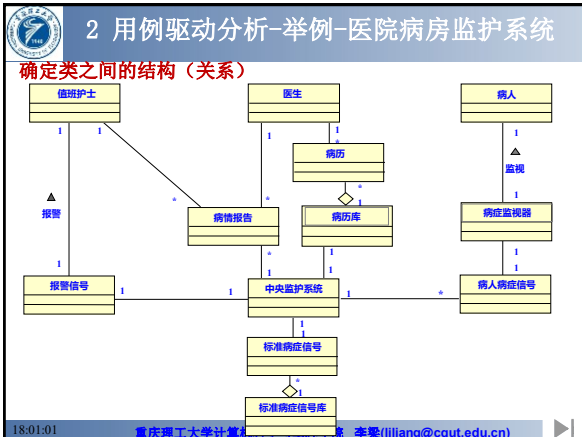
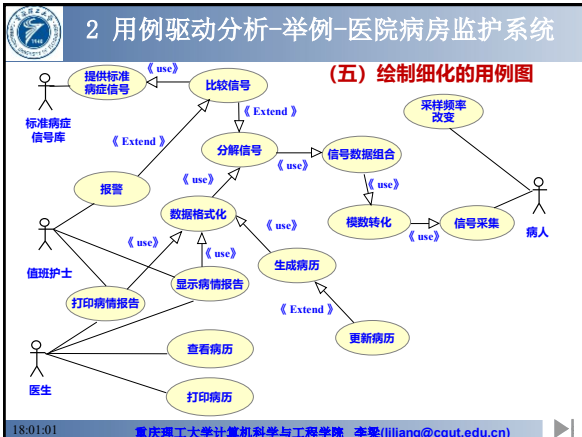
学生在登录后，选择执行“选课”或“查学分”。若增加其它功能（如修改密码），必须修改登录的条件和判断逻辑。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cquf.edu.cn) 41

2 用例驱动分析-举例-家庭安全系统

问题: 1. 系主在和系统电话报警 2. 系主在系统电话报警 3. 系主在系统电话报警

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cquf.edu.cn) 42



2 用例驱动分析-开发活动图

●活动图符号:

- ✓开始节点 (Initial Node): 实心圆点, 活动的开始
- ✓终止节点 (Final Node): 空心圆点, 活动或流程的结束

✓分支与合并: 用菱形表示判定分支和分支合并

18:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 56

2 用例驱动分析-开发活动图

●活动图符号:

- ✓分叉与汇合 (Fork and Join Nodes): 表示并发活动对象在运行时可能会存在两个或多个并发运行的控制流。分叉用于将动作流分为两个或多个并发运行的分支, 而汇合则用于同步这些并发分支, 以达到共同完成一项事务的目的。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 56

2 用例驱动分析-开发活动图

●活动图符号:

- ✓对象节点: 方框表示活动作用的对象实体 (包括普通对象和数据存储对象)。活动可以有多个输入和输出, 对象可以是成为活动的输入。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 57

2 用例驱动分析-如何画活动图

第一步, 定义活动图的范围
活动图是描述行为操作的辅助图, 可以单个或多个用例的场景或场景的部分活动、商务流程描述、类的方法 (服务操作) 等

第二步, 添加起始和结束点
每个活动图有一个起始点和一个或多个结束点。

第三步, 添加活动
对用例建模: 对每个参与者所发出的主要步骤引入一个活动 (可能包括起始步骤, 加上对起始步骤系统响应的任何步骤)。
对商务流程建模: 对每个主要流程引入一个活动, 通常为为一个用例或用例包。

第四步, 添加活动间的转变: 连接各活动
一旦一个活动有多个转变时, 必需对每个转变加以相应标示。

第五步, 添加判断点: 需要检查某些事务或比较某些事务。

第六步, 找出可并行活动: 当两个活动间没有直接的联系且都必需在第三个活动开始前结束, 那它们是可以并行运行的。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 58

产品经理工作的活动图

18:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 59

2 用例驱动分析-活动图例子

网络外卖

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 60

3 对象模型基础

3.1 实体和对象

3.1 对象的基本特征：封装、继承、多态服务和消息

3.3 类的定义

3.4 类的关系

3.5 类图

3.6 对象图

3.7 小结

13:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 67

3.1 实体和对象

- **实体**：客观存在的事物。
 - ✓ 客观世界中的问题都是由实体及其相互之间的关系构成的
- **对象**：系统分析和系统构造中，对象是对客观世界事物的一种抽象，是由数据(属性)及在其上的操作(行为)所组成的封装体
 - ✓ 用来描述客观事物的一个实体，它是实体的抽象，是构成系统的一个基本单位
 - ✓ 由一组属性和对这组属性进行处理的一组服务(操作)组成
 - ✓ 对象可以是具体实体也可以是抽象实体
 - 具体实体：学生、书等
 - 抽象实体：规则、算法、事件等

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 68

3.1 实体和对象

- **服务(操作、方法)**：为完成某任务，一个对象所提供的并体现其责任的操作称为服务。同一类的所有对象共享相同的服务。
- **面向对象技术的封装机制使对象是相互独立的，并且彼此之间信息是隐藏的。消息是对象之间相互请求或相互协作的唯一途径。**
 - ✓ 对象间的联系只能通过传递消息来进行
 - ✓ 对象也只能在收到消息时才被激活
- **消息**：用来请求对象执行某个处理或回答某些信息的要求，是连接对象的纽带(对提供服务的对象发出的服务请求)。或者说，一个对象为实现其责任而与其它对象的通信称为消息。
 - ✓ 接收消息的对象：对象标识
 - ✓ 消息选择符(也称为消息名)：服务标识
 - ✓ 零个或多个变元：输入信息和应答信息

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 69

3.2 对象的基本特征：封装、继承、多态

- **封装**：将某事物封闭成为一个对象，可能是一段代码，也可能是一个设计好的界面；
 - ✓ 对象与对象之间通过消息进行信息的传递，对象中**对内**的方法与属性是私有的，**对外**的方法与属性是**公有**的，只有公有的方法与属性才能与外部进行交互。
- **继承**：一个对象可以获得另一个对象的属性和方法，被继承的对象一般称之为**基类**，而那个继承它的对象就称之为**子类**；
 - ✓ 儿子(子类)继承父亲(父类)，那么儿子就拥有了父亲的所有财产(父类的方法和属性)，**儿子可以直接用**。
 - ✓ 但有的东西**儿子继承来了不一定有用**，需要根据儿子的特点改造，如：鸟(父类)有飞行的能力，但有的鸟飞行方式有所不同，像蜂鸟。蜂鸟就可以**重写鸟类的飞行方法**，以实现自己独特的飞行方式。
- **继承是不是复制粘贴？继承-基类改变影响子类 复制不影响**

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 70

3.2 对象的基本特征：封装、继承、多态

- **多态**：每一个子类的**实际属性和方法**都会有一些不同，就像中国人是人类的子类，中国人既有全部人类都有的特点，又有独属于自己的特点。
 - ✓ 继承可以让儿子拥有父亲的财产，那么父亲为什么不可以用儿子的财产呢？这就是**向后兼容**，让父亲(父类)指向儿子(子类)，实现儿子(子类)的方法。
 - ✓ 这种**向后兼容**，父类指向子类并调用子类的方法就是**多态**。多态实现了父类指向子类，并可调用子类的方法，这样有利于提高程序的**扩展性和可维护性**。
- 通过面向对象的方式，让大的系统开发变为可能，在系统中将**对象抽象成类**，再通过继承与信息交互**建立起类与类的关系**，增强了代码的**重用性**，减少了重复性劳动。将复杂的问题封装起来，对外只需要暴露出**必要的接口**，减少信息的交互，增强对象内部的**内聚性**，使系统分割清晰，简单明了。通过**多态性**，让对象增加了**更多的可能性**，增强对象的扩展性及维护性。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 71

3.3 类的定义

1. 类的定义：类是一组具有相同数据结构和相同操作(行为)和关系的对象的集合。类的一个实例就是**对象**，对象是具有明确语义边界并封装了状态和行为的**实体**，是构成系统的**基本单位**，由一组属性和作用在这组属性上的一组操作构成。

类所创建的对象，根据语境的不同可以有多种不同的行为，叫做“**角色**”。

例如：某人是经理，但他在读硕士学位，在学校中他是学生。

缩略表示

实体类 界面类 控制类

简化表示

13:07:42 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 72

3.3 类的定义

2. 类的类型

①**边界类**：边界类位于系统与外界的交界处，承担系统与外界的信息功能。窗口、窗体、窗幕、通信接口、打印机接口、传感器、终端以及API

边界类处在用例图中,参与者与用例的关联处，可以根据用例图发现边界类。

边界类(图标)

<<boundary>>
边界类(标签)

边界类(装饰)

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqut.edu.cn)

73

3.3 类的定义

2. 类的类型

②**实体类**：实体类对应着现实中的客观实物，用来保存信息，一般对应着数据表、文件等。

实体类可以从现实中的客观事物，以及需要持久存放的信息两方面来发现。

实体类(图标)

<<entity>>
实体类(标签)

实体类(装饰)

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqut.edu.cn)

74

3.3 类的定义

2. 类的类型

③**控制类**：控制类承担着事务处理，控制调控的控制作用。

一个用例中最少会有一个控制类，用来控制用例中的事件顺序，也可以在多个用例之间协调用例之间的联系。

控制类(图标)

<<control>>
控制类(标签)

控制类(装饰)

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqut.edu.cn)

75

3.3 类的定义

一个例子：订货系统

订货管理

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqut.edu.cn)

76

3.3 类的定义

一个例子：订货系统

控制类

实体类

边界类

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqut.edu.cn)

77

3.4 类之间的关系-关联

类之间的关系：关联、聚集和组合、泛化、依赖

●**关联**的通俗含义：表示事物之间的一种牵连式的固有联系

●**关联**代表一组存在于两个或多个对象之间的、具有相同结构和含义的具体连接。关联可以是物理的，也可能是逻辑的

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqut.edu.cn)

78

3.4 类之间的关系-关联

关联的种类

① 一元(自返)关联

② 二元关联

③ 多元关联

三元关联

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 79

3.4 类之间的关系-聚集与组合

3.4.1 聚集和组合

1. 聚集(aggregation): 表示类之间一种松散的整体与部分的组成关系, 是一种特殊的关联。处于部分方向的对象可以同时参与多个处于整体方对象的构成

表示

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 80

3.4 类之间的关系-聚集与组合

3.4.1 聚集和组合

2. 组合(composition): 表示类之间一种紧密的整体与部分的组成关系, 也是一种特殊的关联。部分类对象完全隶属于整体类对象, 部分类对象与整体类对象共存, 整体类对象不存在了部分类对象也将随之消失(或失去存在价值了)

表示

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 81

3.4 类之间的关系-聚集与组合

3. 聚集与组合的区别

- ① 聚集松散, 组合紧密;
- ② 一个部分事物对象可以属于多个聚集对象, 但一个部分事物对象仅能属于一个组合对象;
- ③ 聚集的对象生命周期可以不同, 但组合对象则是同存同亡。

问题:

- 1、计算机由CPU、内存、软盘、硬盘、显示器、键盘、鼠标组成, 用类图表示这种关系。
- 2、举出一个具有聚合关系的类图的例子。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 82

3.4 类之间的关系-聚集与组合

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 83

3.4 类之间的关系-泛化

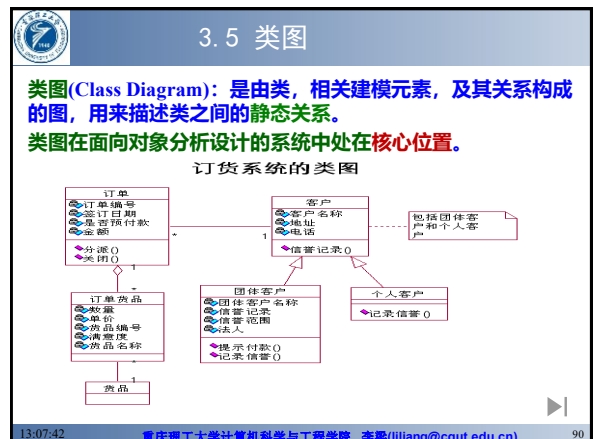
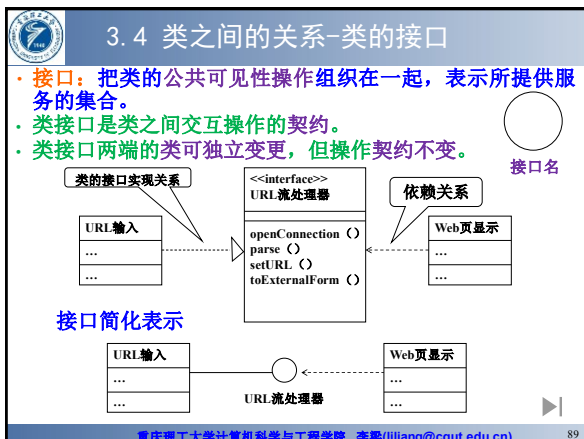
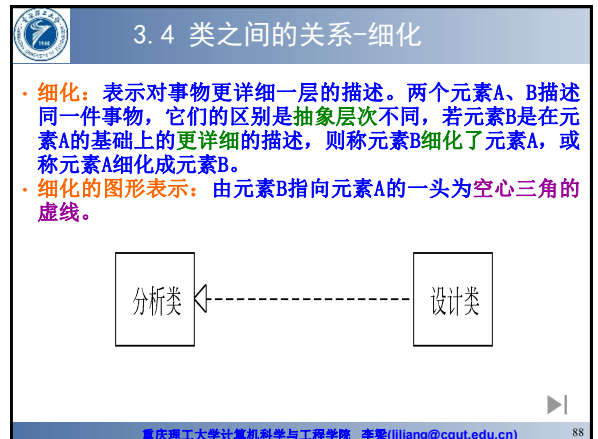
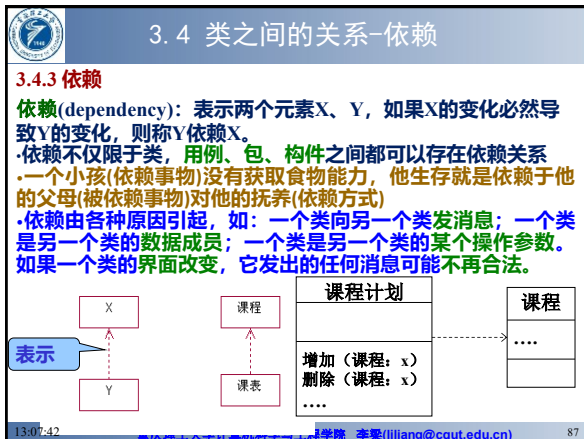
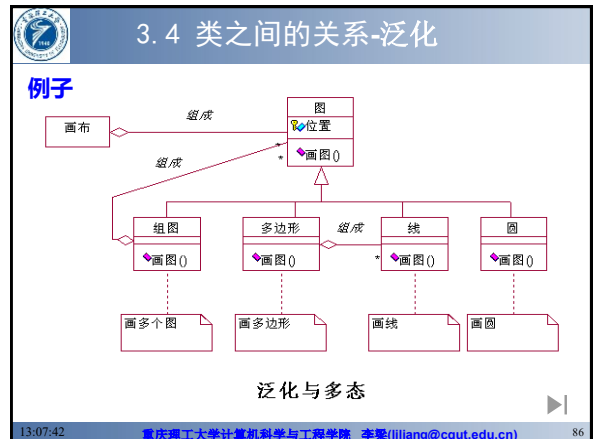
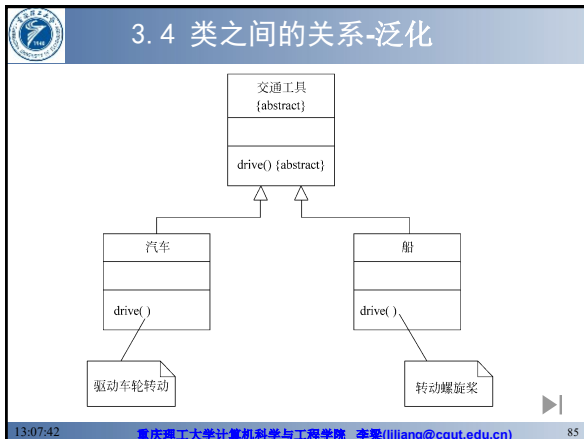
3.4.2 泛化

泛化(generalization): 表示事物之间的一般与特殊的关系。也可以称为继承关系。是通用类(基类)和具体类(派生类)之间的关系, 具体类完全拥有通用类的数据和操作, 并且还可以补充一些数据或操作。

泛化关系还可以表示参与者、用例、包、构件、接口等建模元素之间的关系

表示

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn) 84



3.5 类图-编辑器页面设计例子

无标题 - 记事本

文件 编辑 查看

编辑

放大

Ctrl+加号

状态栏

缩小

Ctrl+减号

自动换行

还原默认缩放

Ctrl+0

文本编辑器

1

1

1

1

标题栏

菜单栏

工具栏

编辑

1

1

1

1

下拉菜单

按钮

滚动条

13:07:42 91 cqut.edu.cn

3.5 类图-抽象层次

类图的抽象层次：在系统的不同开发阶段，类图可以具有不同的抽象程度。随着开发的深入，类图应该越来越详细、具体。可以分为：概念层（应用领域中的概念，需求分析阶段），逻辑层（接口部分，概要设计阶段），实现层（系统实现阶段）。

Circle

center: float

radius: float

area()

move()

scale()

Circle

center: float

radius: float

area(): float

move(location: Point): void

scale(ratio: float): void

概念层 逻辑层 实现层

13:07:42 92 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn)

3.5 类图-抽象层次

人

借方

*

抵押

本金

利率

到期

房屋

*

资产

金融机构

债权人

*

信贷银行

银行

概念层类图

13:07:42 93 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn)

3.5 类图-抽象层次

动物

特征

生命

呼吸

移动

繁殖

植物

特征

光合作用

生长

开花

结果

概念层类图

13:07:42 94 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn)

3.6 对象图

1、对象图的概念：对象图表示一组对象及其它们之间的联系。描述的是参与交互的各个对象在交互过程中某一时刻的状态。

对象图可以被看作是类图在某一时刻的实例。是系统的详细状态在某一时刻的快照，通常用来表示复杂类图的一个实例。

2、说明：对象图并不在任何时间必须

13:07:42 95 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn)

3.6 对象图

公司系统类图

公司

部门

员工

对象图

公司

部门

员工

13:07:42 96 重庆理工大学计算机科学与工程学院 李黎(lililang@cqut.edu.cn)

4.1 建立静态模型-领域模型概述

需求捕获

业务建模

领域模型

分析

分析模型

设计

设计模型

相关元素

1. 类的主要职责
2. 类团关系
3. 重要的多重性关系
4. 部分业务规则性约束

相关元素

1. 边界/控制/实体类
2. 从系统实现角度整理
3. 进一步细化类的职责
4. 进一步细化类的关系
5. 进一步细化业务规则

相关元素

1. 用构造型细化类团关系
2. 模板/主调/关联/嵌套类
3. 引用对象、值对象
4. 抽象类与接口、OCL
5. 设计类(框架与设计模式)

• **领域模型**是描述业务实体的静态结构, 包括业务实体、各个业务实体所具有的业务属性及业务操作、业务实体之间具有的关系。

• **分析模型**和领域模型是很相近的, 主要是针对软件系统的分析, 领域模型则更多是偏重对业务领域的分析

• **设计模型**则是在分析模型的基础上添加设计元素的结果。与分析模型相比, 设计模型中的类的属性集更趋完善

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

4.1 建立静态模型-概述-模型步骤

建立领域模型 (类图) 的一般步骤:

- ① 研究分析问题领域, 确定系统需求;
- ② 识别和筛选对象, 从对象抽象类, 明确类的含义和职责, 确定类的属性和操作;
- ③ 确定类之间的关系。关联, 泛化, 聚集, 组合, 依赖;
- ④ 调整和细化类及其关系, 解决重复和冲突;
- ⑤ 绘制类图, 并增加相应说明。
- ⑥ 建立顺序图、协作图和状态图的活动, 放在对象识别之后, 并与基本模型活动交错进行。
- ⑦ 详细说明应分散在各项活动之中进行, 最后做一次审查和补充。

13:07:42 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

4.1 建立静态模型-概述-建议

- **不要试图使用所有的符号。**从简单的开始, 例如, 类、关联、属性和继承等概念。有些符号仅用于特殊的场合和方法中, 只有需要时才去使用。
- **根据项目开发的不同阶段, 用正确的观点来画类图。**分析阶段, 画概念层类图; 软件设计时, 画逻辑层类图; 考察某个特定的实现技术时, 应画实现层类图。
- **不要为每个事物都画一个模型, 应该把精力放在关键的领域。**最好只画几张较为关键的图, 经常使用并不断更新修改。
- **绘制类图的最大危险是过早地陷入实现细节。**应该将重点放在概念层和说明层。
- **模型和模型中的元素是否有清楚的目的和职责(系统功能最终是分配到每个类的操作上实现的, 这个机制叫职责分配)。**
- **模型和模型元素的大小是否适中。**过于复杂的模型和模型元素是很难生存的, 应将其分解成几个相互合作的部分。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

4.2 建立静态模型-发现和定义对象与类

- 在对象识别中最关键的是正确运用抽象原则, 对象映射问题域中的事物, 但并不是问题域中的所有事物都需要映射。
- **舍弃**与系统责任无关的事物及与系统责任无关的特征。
- 判断事物极其其特征是否与系统责任相关准则是: 该事物是否向系统提供了一些服务或需要系统描述它的某些行为。

主要的策略是: 从问题域、系统边界和系统责任这三方面寻找。

- ① 在问题域方面: 人员、组织、物品、设备、事件、表格、结构等等。
- ② 在系统边界方面: 人员、设备和外部系统, 一些系统与外部活动所进行的交互, 并处理系统对外接口的对象。
- ③ 对系统责任的分析: 对象识别的遗漏的考虑, 对照系统责任所要求的每一项功能查看是否可以由已找出的对象来完成该功能, 在不能满足要求时增加相应的对象, 可以是系统分析员尽可能完全的找出所需要的各种对象。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

4.2 建立静态模型-发现和定义对象与类

- **对系统开发的用例或用例场景叙述进行“语法分析”, 可以开始业务类的识别。**
- ✓ **外部物理实体、角色、组织单元:** 客观存在的信息, 使用系统、与之交互、必须保留信息的人与组织、机构。如**顾客、收银员、某某系统**。
- ✓ **事物:** 问题信息域的一部分。
- ✓ **发生或事件:** 系统必须观测、记忆的与时间有关的事件。
- ✓ **场地:** 系统的工作的环境和场所。
- ✓ **结构:** 定义了**对象的类**或与**对象相关的类**。
- ✓ **地点:** 系统需了解掌握的物理位置、办公地点等。
- ✓ **性能说明**
- ✓ **外部实体:** 与系统有关的外部实体
- ✓ **操作过程:** 系统必须记忆、且不在问题域约束中的顺序操作过程(为了指导人机交互)。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

4.2 建立静态模型-家庭安全系统

问题陈述:

- 1 系统由房主在安装时配置, 通过**系统控制窗口**和**键盘功能**与房主交互进行。系统可以监控所有连接的传感器。
- 2 在安装过程中, 每个传感器可以设置编号和类型, 系统的**启动和关闭**必须有主人**密码控制**; 传感器事件发生时, 软件激活系统上的**警报器**, 在设定的时间延时后, 软件**自动拨打设定的电话号码**, 并提供位置信息和事件性质, 电话号码将每隔20秒重拨一次, 直至电话接通。
- 3 系统运行时, 系统控制窗口显示提示信息和系统状态。通过键盘可以控制系统运行。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

4.2 建立静态模型-家庭安全系统

Step1 提取名词、代词和名词短语

Step2 整理归并

系统、房主、系统控制窗口、键盘功能键、提示信息、系统状态、安装过程、配置系统、系统运行、系统连接的传感器、编号、类型、位置信息、事件性质、系统启动、系统关闭、主人密码、传感器事件、电话、系统上的警报器、设定的时间延时、设定的电话号码

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

103

4.2 建立静态模型-家庭安全系统

根据家庭安全系统用例

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

104

4.2 建立静态模型-分析筛选对象策略

策略1 舍弃无用的对象

1) 舍弃对应于对象信息（属性）的词：

某事物是对象所必须保存和管理的信息，它们可能是属性。

例如：传感器的编号和类型，可以表明所发生事件的位置和性质。

2) 舍弃对应于对象功能（操作）的词：

某对象提供的一些有用的操作，直接或间接地反映了用户所需要的功能，它们可能是操作。

例如：系统的安装过程和拨打电话过程。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

105

4.2 建立静态模型-分析筛选对象策略

策略2 精简对象

1) 如果对象只有一个属性，但没有操作，应考虑它被哪些别的对象引用，能否合并到那些对象中。

例：在家庭安全系统中，如果设立主人对象，只有“密码”一个属性，在启动和关闭系统时通过引用主人对象来执行其必要的操作，如果把“主人密码”作为一个属性，增加到系统对象中，会更明确和简便，没有必要设立主人对象。

2) 如果一个对象只有一个操作，而没有属性，并且系统中只有一个类的对象请求这个操作，可以考虑把该对象合并到它的请求对象中。

例如：格式转换器没有属性，只有一个操作“文件格式转换”，并且这个操作只有输出设备对象使用，因此可以不设格式转换器类，把它的操作放到输出设备对象中。

注：没有属性而有操作的对象类是经常需要的，如超类。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

106

4.2 建立静态模型-分析筛选对象策略

策略3 将对象推迟到设计阶段考虑

系统责任要求的某些功能与实现环境有关，应该把这样的功能所涉及的对象，推迟到设计阶段去考虑，OOA模型应独立于具体的实现环境。

如：为系统控制窗口对象而设定的具体对象，包括窗口对象、对话框对象、下拉菜单对象、按钮对象等等，应该到设计时再考虑。

可能的操作：

可能的属性：

可能有用的对象：

安装过程

配置系统

系统运行

系统启动

系统关闭

提示信息

系统状态显示

传感器编号

传感器类型

位置信息

事件性质

时间延时

电话号码

系统状态

系统

控制窗口

键盘功能键

传感器

传感器事件

警报器

电话

不必要的类：

房主

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

107

4.2 建立静态模型-从对象抽象类

依据对象和类的定义，使用概括来寻找两个或多个具有相同属性和操作的对象，概括这些共同的方面以形成类。

系统

显示窗口

控制功能键

传感器

警报器

电话

传感器事件

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

108

4.2 建立静态模型-从对象抽象类

对类进行调整

- 1) 如果类的属性或操作不适合该类的全部对象，则应重新划分。
例如：如果在家庭安全系统中设置控制台类，则显示窗口和控制键，其中的属性不适用于窗口和控制键。
- 2) 如果属性及操作都相同，即使不是同类，应该考虑重新划分。
例如：计算机和吸尘器，虽然不是相同的类，但在商店销售系统中，它们的属性和操作并无差别。
- 3) 如果不同类有很多相同的属性及操作，则应考虑提取超类(继承)
例如：烟雾传感器、热敏传感器和触点传感器，其中很多属性和操作是相同的，可以提取传感器作为超类，用继承作为类之间的联系。
- 4) 如两个类属性有重复且有附属关系，则可考虑合并为一个主要类
例1：工作证类和职员类，如果在特定的问题中，工作证除了“编号”以外的属性，其他属性都与职员类属性一致，则可以考虑将“编号”合并到职员类中，去掉工作证类。
例2：在家庭安全系统中，如果不需要保留每次传感器发生事件的记录，则可以考虑将传感器事件类合并到传感器类中。

18:01:01
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
109

4.2 建立静态模型-属性和操作

➤ 对象的**内部特性**包括对象的属性和服务。问题域中，事物的特征可以区分为静态特征和动态特征，静态特征通过一组数据来表示，而动态特征则可以通过一系列操作来表达。

➤ 对象的属性和服务描述了对对象的内部细节，只有给出了对象的属性和服务才能说对于该对象有了明确的认识和定义。

可以从以下角度确定对象应具有的属性：

- ① 按照一般常识，对象应该具有哪些属性；
如传感器属性，包括编号、类型、临界值、.....
- ② 在当前问题域中，对象应具有哪些属性；
如传感器属性，需要安装地点、性质。同样传感器属性，在设备管理系统中，需要数量、购置时间
- ③ 根据系统责任的要求，对象应具有哪些属性；
如报警器的属性，有自动拨打电话的责任，因此需要延时时间属性。

18:01:01
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
110

4.2 建立静态模型-属性和操作

可以从以下角度确定对象应具有的属性：

- ① 按照一般常识，对象应该具有哪些属性；
- ② 在当前问题域中，对象应具有哪些属性；
- ③ 根据系统责任的要求，对象应具有哪些属性；
- ④ 建立该对象是为了保存和管理哪些信息；
- ⑤ 对象为了在服务中实现其功能，需要增设哪些属性；
- ⑥ 是否需要增设属性来区别对象的不同状态；
如传感器属性，有被设置在线和撤消的状态，则需要有在线状态属性。
- ⑦ 问题陈述中定语用的词汇，可以帮助确定类的属性如：“红色的按钮”，可以确定按钮类有颜色属性。
- ⑧ 用什么属性来表示对象的整体-部分联系和实例连接。如某学生的指导教师，学生类和教师类之间具有关联关系，控制台的输出显示窗口，控制台类与输出显示窗口类具有聚合关系。

18:01:01
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
111

4.2 建立静态模型-属性和操作

➤ **对象属性的审查和筛选：**对于找到的对象属性，还应进行严格的审查和筛选，才能最终确定对象应具有的属性，在审查和筛选中，应考虑的问题有：

- ① 该属性是否体现了以系统责任为目标的抽象；
- ② 该属性是否描述了该对象本身的特性；课程类有主讲教师属性。即使限定一个课程只有一个主讲，一个教师只能主讲一门课程，把主讲教师的电话、住址等作为课程的属性也是不行的，它会造成概念上的混乱。
- ③ 该属性是否破坏了对对象特征的“原子性”；如：门窗、服装等，都应该具体分为门和窗，上衣和裤子。
- ④ 该属性是否已通过类的继承得到；
- ⑤ 该属性是否可以其他属性推导得到。有出生年月属性，不必保留年龄属性。有个项税率属性，也应有总税额属性。

18:01:01
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
112

4.2 建立静态模型-属性和操作

系统	显示窗口	控制键	传感器
启停状态	坐标位置	按键编号	编号
主人密码	大小	按键功能	类型
			位置
			性质
			限值

报警器	电话	传感器事件
启停状态	电话号码	事件时间
延时限制		

18:01:01
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
113

4.2 建立静态模型-属性和操作

- **定义对象和类的服务：**在系统需求中分检出相应的动词，作为类中可能使用的服务，通过这种方法，我们能够发现类对象类的一些服务，进而发现对象类的操作。
- 操作定义了某个对象的行为。
- 操作可以分为四种类型：
 - ✓ 以某种方式操纵数据，例如：添加、删除、选择、更新等
 - ✓ 执行计算的操纵，例如：销售中的计算总价。
 - ✓ 请求某个对象状态的操作。
 - ✓ 监视某个对象发生某个控制事件的操作。
- 操作的构造需要交互图和场景描述等手段多次反复分析才能获取。在研究语法分析并分离动词作为候选的操作。推荐的一个方法是使用CRC（类-职责-协作）技术。

18:01:01
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
114

4.2 建立静态模型-属性和操作

系统	显示窗口	控制键	传感器
启停状态	坐标位置	按键编号	编号
主人密码	大小	按键功能	类型
系统启动	按坐标位置显示信息	功能键操作处理1	位置
密码设置	按大小打开窗口	功能键操作处理2	性质
.....	限值
			判断限值

警报器	电话	传感器事件
启停状态	电话号码	事件时间
延时限制	自动拨号处理	事件查询
根据延时限制报警
.....		

4.2 建立静态模型-家庭安全系统

建立的系统领域类图

4.2 建立静态模型-家庭安全系统

4.3 建立静态模型-分析类

- **分析类**（在进行面向对象分析过程中所使用的类）侧重于处理功能性需求，通过较高的、非形式化层次的职责类定义某行为（分析类不是具体OOD中的类）也有称之为设计类
- 分析类封装了信息和行为，它是具有相同属性、方法和关系的对象集合的总称。
- 在系统中，每个类都具有一定的职责，职责是指类所担任的任务。一个类可以有多种职责，设计得好的类一般至少有一种职责，在定义类时，将类的职责分解为类的属性和方法，其中属性用于封装数据，方法用于封装行为。分析类时OOD中最重要的组成部分，也是最复杂和最耗时的部分。

4.3 建立静态模型-分析类

- 分析类根据其功能和作用分三种基本构造型：
 - ✓ **实体类**：表示系统存储和管理的永久信息的类。实体类是直接从业务对象模型中相应的业务实体得到的。
 - ✓ **边界类**：边界类用于建立系统与其参与者之间交互的类（UI），经常代表对窗口、窗幕、通信及打印接口、传感器、终端以及API等的抽象。每个边界类至少与一个参与者有关。
 - ✓ **控制类**：在运行过程中的业务控制逻辑。负责协调边界类和实体类，在现实世界中没有对应的事物，负责接收边界类的信息，并将其分发给实体类。如协调、排序、事务处理以及其他对象的控制，常用于封装与某个具体用例有关的控制。

4.3 建立分析模型-分析类

- **实体类**映射需求中的每个**实体**，实体类保存需要永久存储在存储体中的**信息**。实体类通常都是永久性的，它们所具有的属性和关系是长期的，有时甚至在系统的整个生存周期都需要。
- **实体类**是直接从业务对象模型中相应的**业务实体**得到的。
- **实体类**是对用户来说最有意义的类，通常采用**业务领域术语命名**，一般来说是一个名词，在用例模型向领域模型转化中，一个参与者一般对应于**实体类**。通常可以从数据库表（需要持久存储）对应的名词着手来寻找实体类。通常情况下，实体类一定具有属性，但不一定有操作。

销售实体类：保存了完成一次销售的信息（数据）

边界类

常用边界类：窗口、通信协议及接口、传感器、打印机接口、传感器、终端、报表以及API

：处理销售用户界面 ：销售

边界类

> **边界类**是从那些系统和外界进行交互的对象中归纳、抽象出来的，它是系统内的对象和系统外的参与者的联系媒介，外界的消息只有通过边界类的对象实例才能发送给系统。

> **边界类**用于封装在用例内、外流动的信息或数据流。每个参与者和用例交互至少要有一个边界类

> **边界类**可以是用户接口、系统接口和设备接口。对于用户接口来说，边界类集中描述了用户与系统的交互信息，而不是描述用户接口的显示形式，如按钮等；对于系统接口和设备接口来说，边界类集中描述所定义的通信或交换协议，而不是说明协议如何实现的。边界类使系统与系统的外部变化隔离开来。参与者只许通过边界类和系统通信。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 121

控制类

● **控制类**类似于设计模型中的控制器类，其目的是UI层之上的第一个对象，主要负责接收和处理系统操作消息。

> **控制类**是实体类和边界类之间的润滑剂，用于协调边界类和实体类之间的交互。

> **控制类**与用例存在着密切的关系，它在使用例开始执行时创建，在使用例结束时取消。一般来说，一个用例对应一个控制类。

● **控制类**是用于控制用例工作的类，是由动宾结构短语（“动词+名词”或“名词+动词”）表示，用例“身份验证”可以对应于一个控制类“身份验证器”，它提供了与身份验证相关的所有操作控制，控制类用于对一个或几个用例所特有的控制行为建模，控制对象（控制类的实例）通常控制其他对象，他们的行为具有协调性。

● **控制类**将用例的特有行为进行封装，控制对象的行为与特定用例的实现密切相关，当系统执行用例的时候，就产生了一个控制对象，控制对象经常在其对应的用例执行完后消亡。通常情况下，**控制类没有属性，但一定有方法。**

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 122

控制类举例

通过控制类将系统请求和输入信息转发给下面的实体类进行处理（销售、支付）

执行活动

处理销售

委派

：收银员 ：处理销售用户界面 ：ProcessSaleHandler ：销售 ：支付

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 123

4.6 开发业务类图（用例实现）

● **用例实现(开发业务类图)**分析是分析模型内部的一种协作，主要描述了如何根据分析类及其交互的分析对象来实现和执行一个具体的用例。

● 用例实现包括事件流的文本描述、反映参与者用例实现的分析的类图以及按照分析对象的交互作用描述特定流实现或用例脚本的交互图。

● 用例实现侧重于功能性需求。

类图：一个分析类及其对象经常参与多个用例实现，而一个具体类的某些职责、属性及关联只与一个用例实现相关

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 124

处理销售类图

处理销售

：处理销售用户界面 ：ProcessSaleHandler ：销售 ：支付 ：商品 ：商品列表 ：商品描述

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 125

4.4 对象及其类与外部的关系

● **关联：**需求陈述中使用的描述性动词或动词词组

● **一般-特殊关系（即继承关系）：**即对象之间的分类关系。

① 按照问题域已有的分类找出与之相对应的一些一般-特殊结构；

② 按照尝试考虑事物的分类，从而发现其中的一般-特殊关系；

③ 考察类的属性与服务，若一个类的属性与服务只能适合该类的一部分对象，则从类中划分出一些特殊类；若两个或两个以上的类含有一些共同的属性和服务，则可将共同的属性和服务提取出来，构成一个在概念上包含原先那些类的一般类；

④ 考虑领域范围内的复用，在更高水平上运用一般-特殊结构，并能开发贡献一些可复用性更强的类的构件。

● **整体-部分关系：**即对象之间的组成关系。

● **静态连接关系（即关联关系）：**即通过对象属性所反映出来的联系，用实例连接表示；

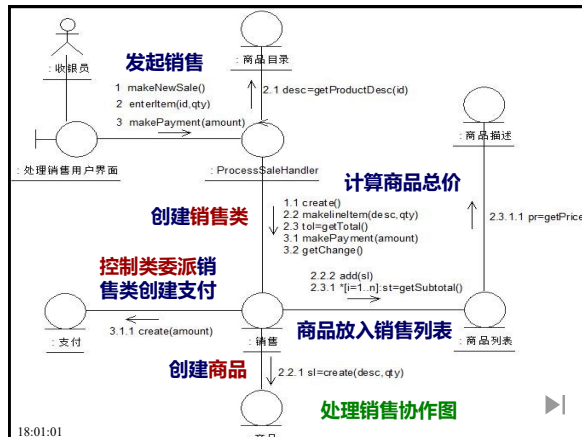
● **动态连接关系（即依赖关系）：**即对象行为之间的依赖关系，用消息连接表示。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 126

4.5 识别对象之间的交互：开发交互图

- 用例实现分析是分析模型内部的一种协作，主要描述了如何根据分析类及其交互的分析对象来实现和执行一个具体的用例。
- 用例实现侧重于**功能性需求**。
- 当**参与者**向系统发送某种形式的消息而激活用例时，开始执行该用例中的动作序列。
- 边界类对象**将接收来自参与者的消息。
- 边界对象**向其他对象发送一个消息，并使有关对象与之交互从而实现该用例。
- 在分析阶段，通常使用**协作图**类描述用例的实现。
- 协作图**又称为**通信图**，是以图或网络格式描述**对象交互**，其中对象可以置于图中任何位置。
- 协作图**通过在对象之间建立链接并在上面附加信息来表明对象之间的交互，消息名称反映了在与被引用对象交互时引用对象的意图。不同对象有不同的生命周期；边界对象和实体对象通常不需要专用于一个用例实现；控制对象通常与具体用例有关的控制进行封装。

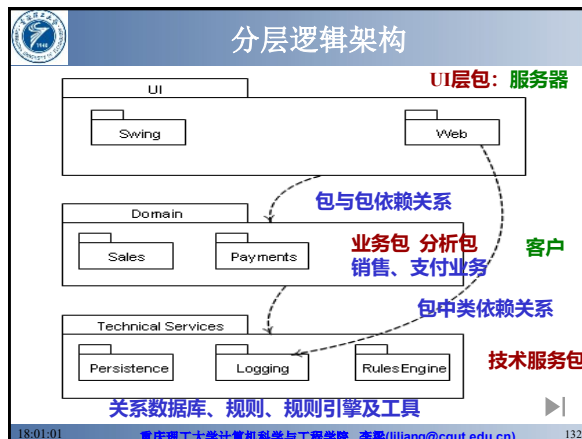
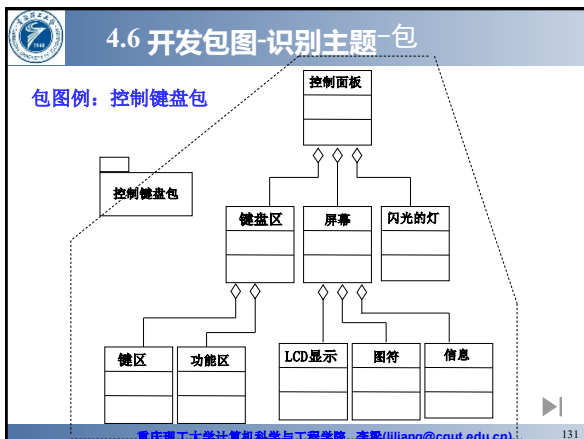
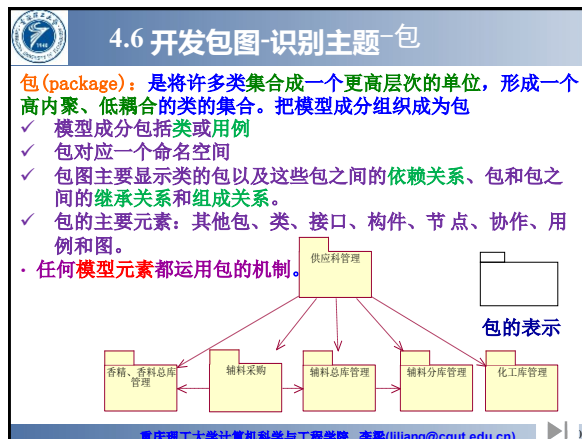
18:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

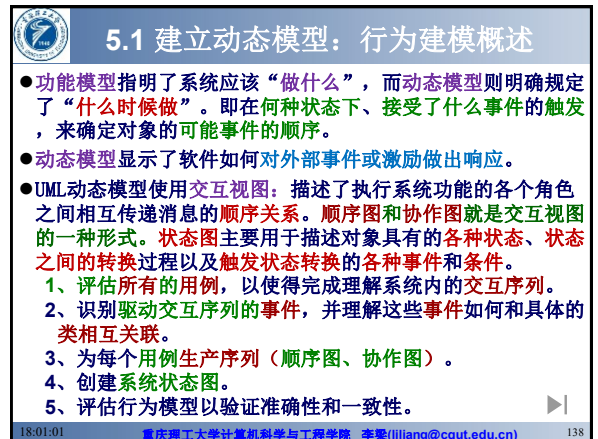
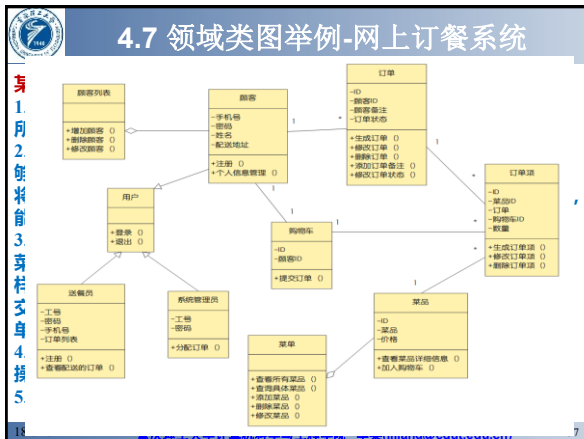
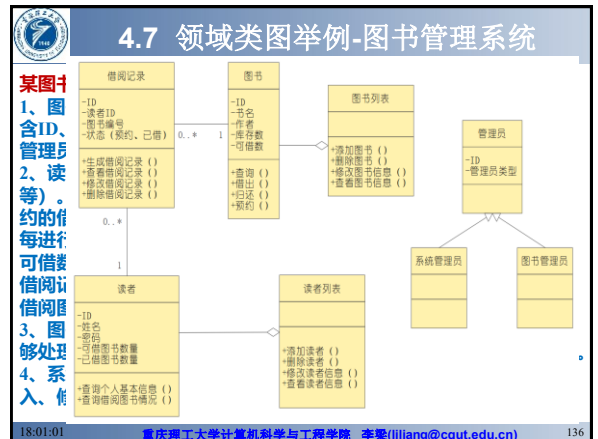
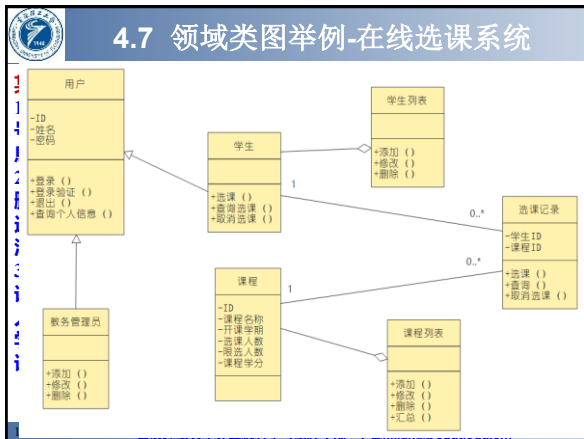
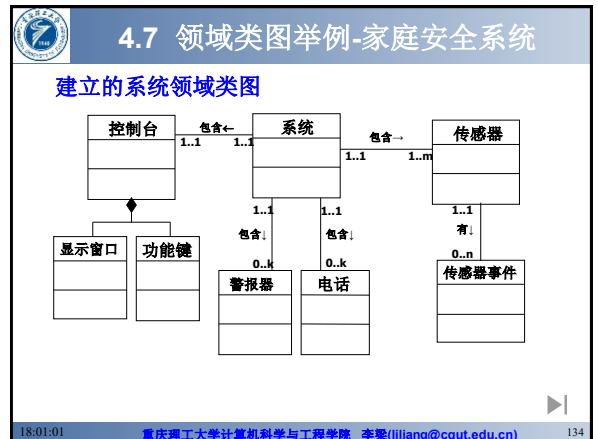
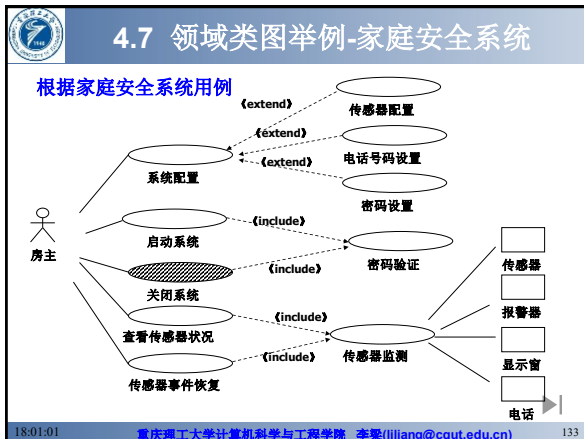


4.6 开发包图-识别主题

- 主题**是一种指导开发者或用户研究大型复杂模型的机制。
- 主题**是把一组具有较强联系的类组织在一起而得到的类的集合。
- 主题**是一种手段，有助于分解大型项目以便分组承担任务。
- 主题**还可以给出面向对象分析和设计的模型总体概貌。
- 主题**所依据的原理是**整体——部分关系的扩充**。
- 一个系统模型可以包含多个**主题**。
- 主题的特点：**
 - 一个主题内部的对象类应具有某种意义上的**内在联系**
 - 描述系统中相对独立的组成部分（如一个子系统）
 - 描述系统中某一方面的事物（如人员、设备）
 - 解决系统中某一方面的问题（如输入输出）
 - 主题的划分有一定的灵活性和随意性

18:01:01 重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn) 129





5.1 建立动态模型：行为建模概述

交互图

状态机

顺序图（时序图） 用来表示在用例的一个特定场景中，外部参与者产生的事件，事件的顺序及系统之间的交互事项。

通信图（协作图） 描述对象之间的关联及它们彼此之间的消息通信。

区别： 顺序图强调消息发送的时间顺序，而通信图则强调接收和发送消息对象的组织结构。

状态图 使用于描述状态和动作的顺序，可以展现一个对象拥有的状态，还可以说明事件如何随着时间的推移来响应这些状态。

活动图 用来描述过程（业务过程、工作流、事件流等）中活动及其迁移，活动图是面向对象的流程图。

区别： 活动图强调从活动到活动控制流，状态图强调对象潜在的状态和这些状态之间的转移。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

9

5.2 建立动态模型-顺序图

1、顺序图的定义

- 顺序图用来描述对象之间动态的交互关系，着重体现对象间信息传递的时间顺序。
- 顺序图是对对象之间传送消息的时间顺序的可视化表示。顺序图的主要用途是把用例表达的需求，转化为进一步、更加正式层次的精细表达。用例常常被细化为一个或者更多的顺序图。同时顺序图更有效地描述如何分配各个类的职责以及各类具有相应职责的原因。
- 顺序图存在两个轴：水平轴表示不同的对象，垂直轴表示时间。

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

140

5.2 建立动态模型-顺序图

- 表现系统行为方式的一种方式是UML的顺序图和协作图。
- 顺序图和协作图的作用相同，但顺序图强调事件的时间关系。
- 顺序图表现了导致行为从一个类流动到另一个类的关键类和事件
- 顺序图的主要元素有：
 - 对象：参与交互的类的实例，对象之间可以发送事件和接收事件。
 - 参与者：描述本次交互的发起者，即用例的驱动者。用小人形状表示。
 - 生命线：表示一个类的实例，用虚线表示。
 - 消息：表示对象间的每个事件，用带箭头的实线表示。
 - 执行规格条：表示控制焦点的控制期，也称为激活条。
 - 消息标签：指明消息的名称。消息可以有两种方式返回结果：
 - 使用消息语法 `return var=message (parameter)`;
 - 在执行规格条末端使用应答消息线（带箭头虚线）。

18:01:01

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

141

5.2 建立动态模型-顺序图

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

143

5.2 建立动态模型-顺序图

医院病房监护系统顺序图

顺序图有两个主要的标记符：对象和这些对象之间的通信消息。对象是系统中扮演角色的对象，对象之间发送的消息是顺序图的关键。对象拖出的长虚线称为生命线，生命线说明按照时间顺序对象所发生的事件。消息用从对象生命线到接收对象生命线的箭头表示。

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

142

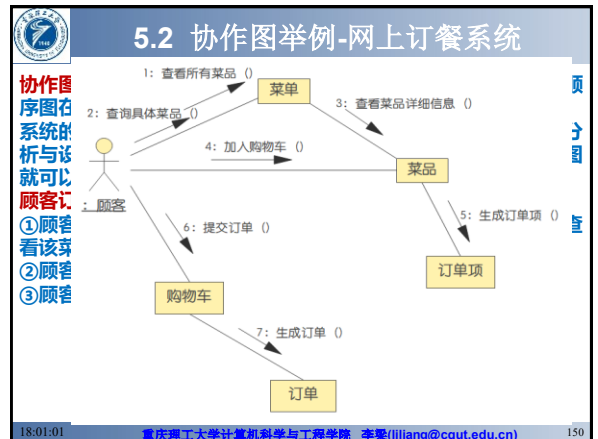
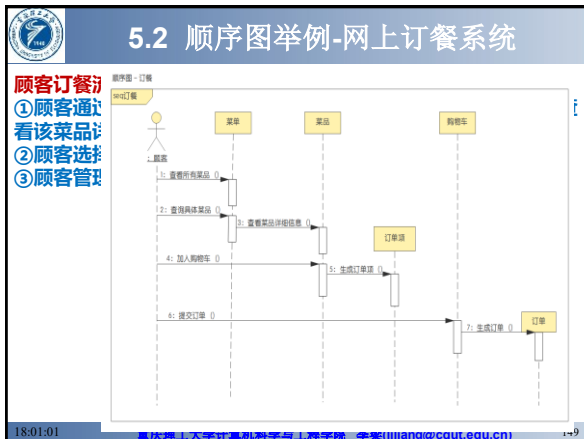
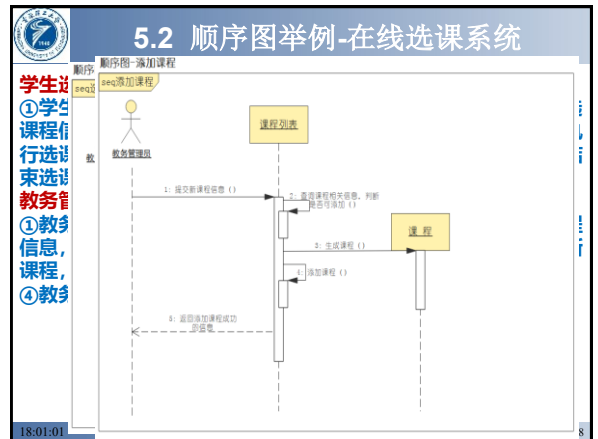
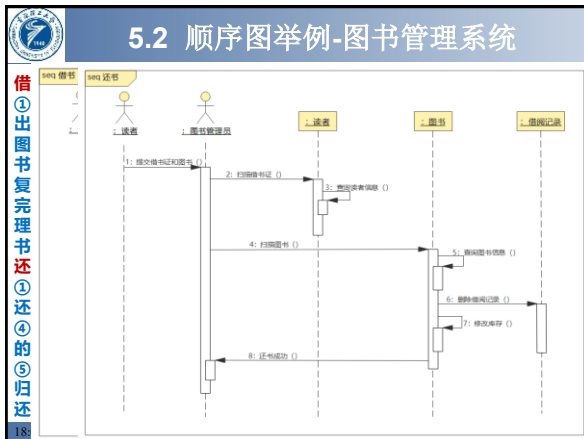
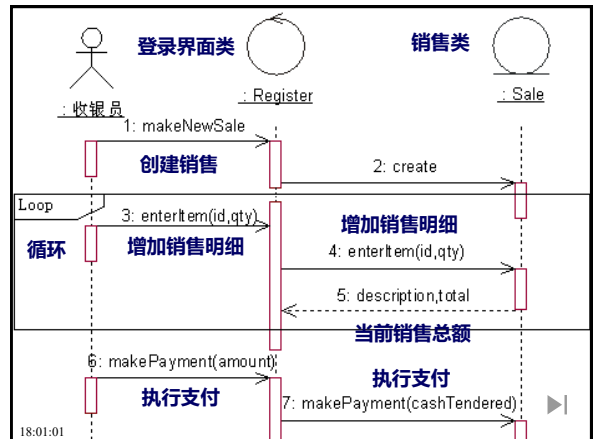
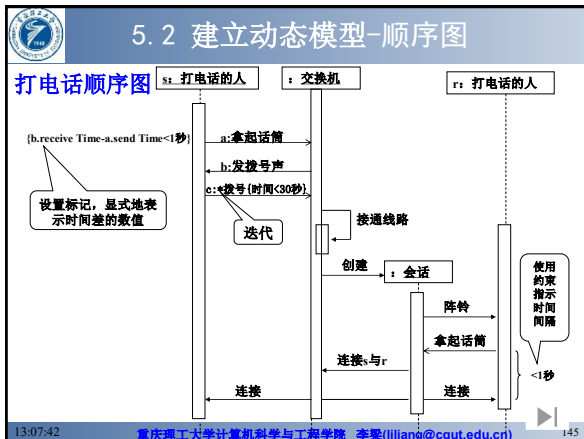
5.2 建立动态模型-顺序图

家庭安全关闭系统顺序图

13:07:42

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

144





5.3 建立动态模型-状态图

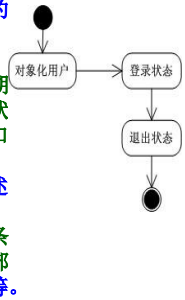
状态图主要用于描述对象具有的各种状态、状态之间的转换过程以及触发状态转换的各种事件和条件。

1. 状态图的组成

状态：主要用于描述一个对象在生命周期内的一个时间段。状态图中的状态包括状态名、内部活动、内部转换、入口和出口动作等部分。

事件：是对一个可观察的事情的类型描述。这种事情的发生可以引发状态的转换。

转换：转换是指状态之间在某种事件或条件的驱动下的切换的过程。转换分为外部转换、内部转换、完成转换、复合转换等。



13:07:42

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)

151



5.3 建立动态模型-状态图

建立状态图步骤

- 1) 考虑某对象在特定语境中的交互行为;
- 2) 建立初始状态和终止状态;
- 3) 从属性值的范围和条件考虑对象所在的稳定状态;
- 4) 从对象的生命期开始, 确定状态偏序;
- 5) 决定对象可能响应的事件;
- 6) 用事件连接这些状态, 并给这些事件添加名称、条件和动作。
- 7) 描绘各状态进入或退出的动作及保持状态的活动。
- 8) 必要的时候, 从对象高端状态开始, 描绘状态中可能的子状态。

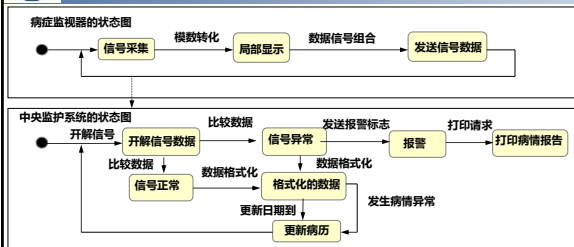
13:07:42

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)

152



医院病房监护系统的状态图



状态图的组成包括：状态、转换、初始状态、终结状态、判定。

结点表示状态，一般是给定类对象中的一组属性值。

箭头直线表示状态之间转换，线上标注触发事件。

监护条件是触发转换必须满足的条件。监护条件只会触发一次，而触发事件是连续检查的。

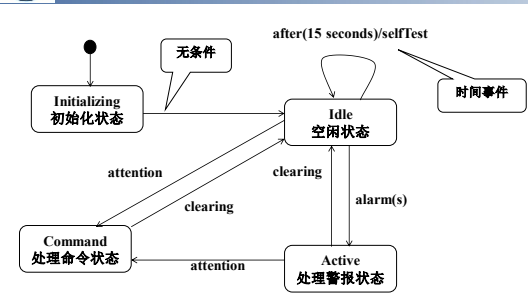
18:01:01

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)

153



监视传感器的控制器类的状态图



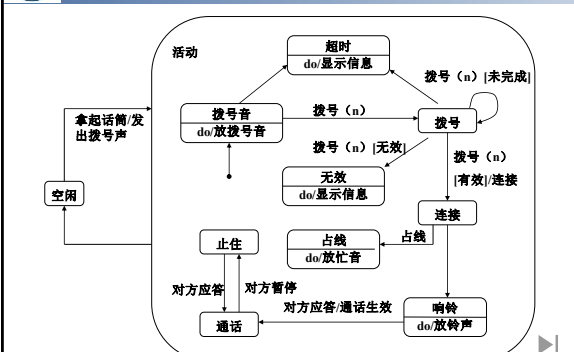
18:01:01

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)

154



电话连接对象的状态图



18:01:01

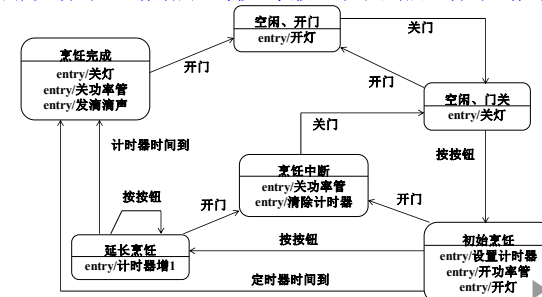
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)

155



简易微波炉状态图

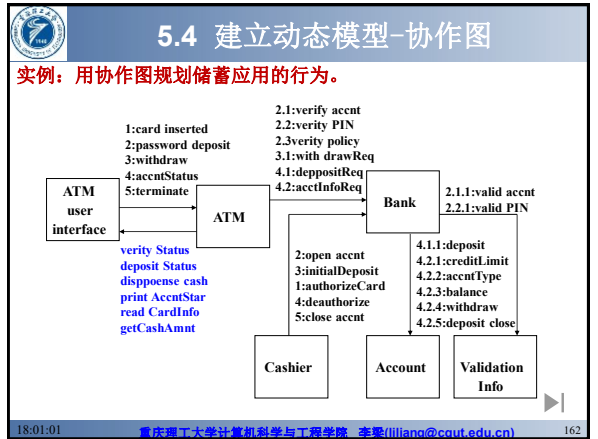
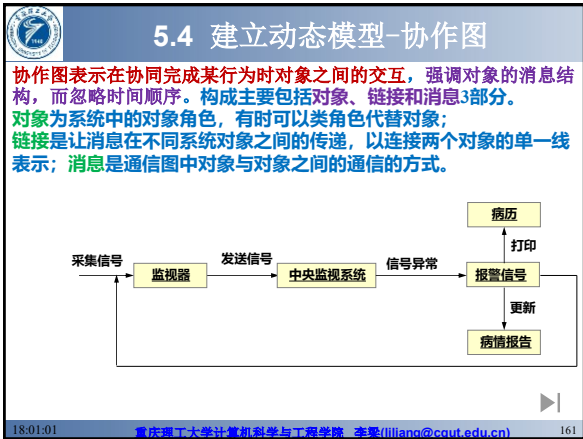
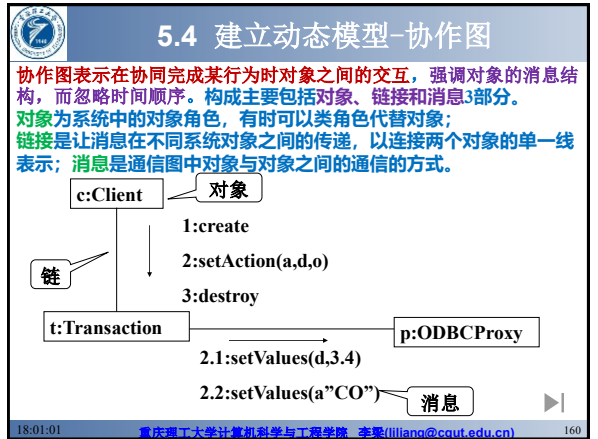
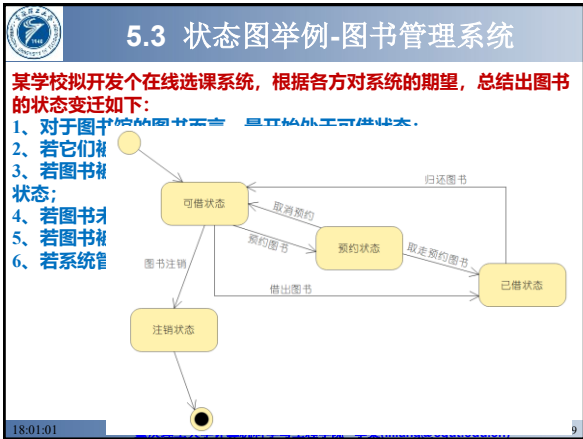
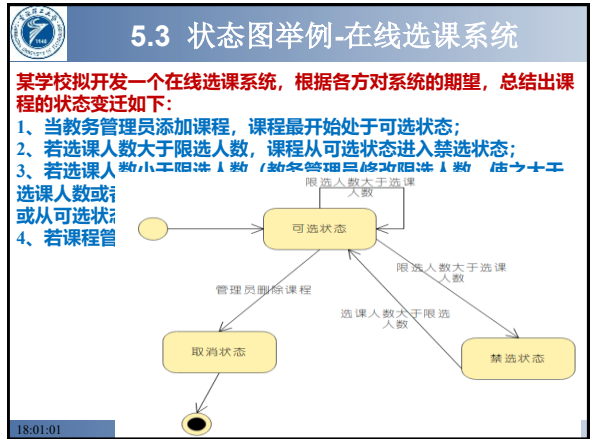
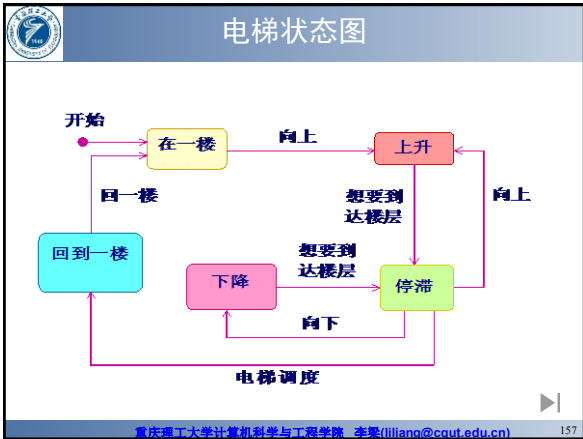
Note: 只有一个按钮的简易微波炉，按一下按钮开始工作，工作时间为一分钟，工作期间，每按一次按钮计时器增加一分钟工作时间。



18:01:01

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)

156





5.5 建立动态模型-简易饮料自动售货机

问题陈述:

- 1、饮料自动售货机用于顾客自助购买饮料。不同的饮料有不同的价格，顾客自助购买饮料的过程：
- 2、饮料自动售货机接受投入的硬币，判断硬币是否有效及面值；
- 3、金额计算器负责累加金额，待累加到饮料售价时，该饮料的选择键就会亮起来；
- 4、顾客可以按下选择键，购买该饮料；
- 5、顾客在按下选择键前，若按饮料售货机上的退币键，则将所有金额退还顾客。



18:01:01

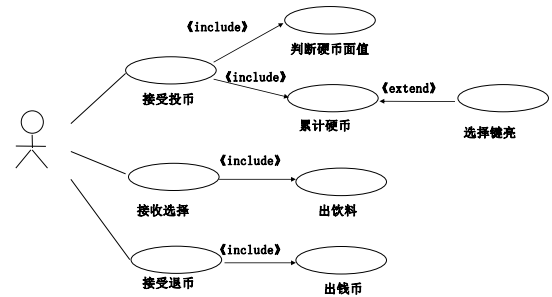
重庆理工大学计算机科学与工程学院 李强(liliang@cqu.edu.cn)

163



5.5 建立动态模型-简易饮料自动售货机

建立系统用例图:



18:01:01

重庆理工大学计算机科学与工程学院 李强(liliang@cqu.edu.cn)

164



5.5 建立动态模型-简易饮料自动售货机

分析和筛选有关名词:

饮料自动售货机	系统主体，有必要的操作
顾客	系统外部的操作者
饮料	系统有用的信息，有必要的属性
价格	饮料的属性
硬币	系统有用的信息，有必要的属性
面值	硬币的属性
计算器	系统的实体，有必要操作
金额	计算器上的属性
售价	与价格同义的词
选择键	系统的实体，有必要操作
退币键	系统的实体，有必要操作



18:01:01

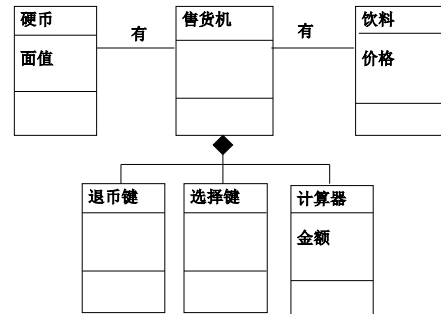
重庆理工大学计算机科学与工程学院 李强(liliang@cqu.edu.cn)

165



5.5 建立动态模型-简易饮料自动售货机

建立类图(未完善):



18:01:01

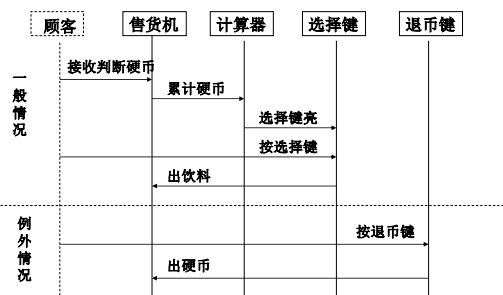
重庆理工大学计算机科学与工程学院 李强(liliang@cqu.edu.cn)

166



5.5 建立动态模型-简易饮料自动售货机

建立顺序图:



18:01:01

重庆理工大学计算机科学与工程学院 李强(liliang@cqu.edu.cn)

167



5.5 建立动态模型-简易饮料自动售货机


完善类图:



18:01:01

重庆理工大学计算机科学与工程学院 李强(liliang@cqu.edu.cn)

168



6 通过用例图建立领域模型实践


1、用例中寻找业务场景：一般通过动词来寻找，比如招聘系统中，一个应聘者投递一个职位就是一次应聘，应聘就是一个业务场景；一个学生参加某门课的考试，那么考试就是一个业务场景；一个学生去图书馆借书，那么借书就是一个业务场景。

2、寻找场景参与者：针对每个业务场景分析出有哪些场景参与者，哪些参与者以对象的形式参与，哪些参与者以服务的形式参与（对象形式参与为领域类图做准备）

- 为什么要区分对象还是服务。是因为有时候我们不关心参与者是哪个，而只关心参与者是什么。
- 一般服务在系统中我们只关心它是什么服务，并且在系统中服务一般也只有一个实例。
- 而对对象则不同，我们会关心对象是谁，即哪一个；

3、分析每个场景参与者对象的基本状态特征（领域类图属性）。所谓的基本状态特征是指对象与生俱来的，对象从一开始被创建出来之后就具有的状态特征；最形象的例子就是人的身高体重，当一出生便具有了身高和体重这两个状态特征。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 169




6 通过用例图建立领域模型实践

再比如一篇博文，它从被写好之时起就具有了内容这个状态特征，但是我们可以随便修改博文的内容；但是有些状态特征是不能修改的，比如博文的创建时间一旦博文被创建之后便不能再被修改；需要注意的是我们不要把对象关联的一些关联信息也认为是对象的基本状态特征。比如某人有一本英语六级考试证书，那么人和证书之间的关系是拥有的关系，证书不是人固有的与生俱来的基本状态特征，而是人参与了某次考试这个场景后得来的；

4、分析每个场景参与者对象分别扮演什么角色参与场景，整个场景的完整交互过程是怎样的，对象在参与场景的过程中执行了哪些交互行为。

- 一个什么样的人或组织或物品以某种角色在某个时刻或某段时间内参与某个活动。
- 这个交互行为是“谁通知谁做什么事情？”，“行为的驱动者是谁？”、“行为的执行者是谁？”、一般行为的驱动者就是通知方，行为的执行者就是被通知方，被通知方拥有通知方要求做的事情执行行为。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 170




6 通过用例图建立领域模型实践

5、分析交互过程结束后分别会对每个场景参与者对象产生哪些基本状态特征的改变。当一个对象参与了某个交互活动后，一些基本状态特征会发生改变。——寻找领域模型中的方法

- 比如一个人参加一次100米跑步比赛后，心跳速度会加快；心跳速度就是人的基本特征。
- 比如一次应聘活动会产生一些与该活动相关的信息，如是否录取，笔试成绩，面试成绩等；
- 比如一次考试会产生考试成绩信息；一次借书会产生一个借阅信息（包含：借书人、被借的书、借书时间，还书时间）；在很多情况下，这些交互信息会在后续的其他交互场景中再次被更新。
- 比如，一次应聘一开始状态可能是“新投递”，表示应聘者刚刚投了简历并选了某个职位，后来她去参加笔试或面试了，那么这次应聘的状态就变为了“已笔试”或“已面试”；
- 在比如一个学生参加一次考试，刚开始还没有成绩，但后来老师批卷子后便有了考试成绩。
- 再比如一次借书后，如果这本书还没被还，则还书时间为空，而一旦还书了之后，便有了还书时间；

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 171




6 通过用例图建立领域模型实践

6、建立交互相关对象：

- 从上述基本状态特征的改变的这些规律中可以发现，交互其实是一个过程，并且该过程一旦开始后就会产生一些相关的信息，如应聘的状态，考试的成绩，借阅信息的归还时间等等。
- 通常我们会把交互过程本身所涉及的一切信息以及交互过程所产生的所有附加信息作为一个整体来进行考虑。通过设计一个对象，用来表示某一次交互的结果，这个结果包含交互过程本身所涉及的一切信息以及交互过程所产生的所有附加信息。
- 如“应聘”这个单词，你有时候会认为它是一个动词，以后会认为它是一个名词。在认为是动词时，我们关注的是交互本身，活动本身，强调行为；而在认为是名词时，我们关注的是应聘行为所产生的一切信息。
- 交互对象一方面体现了交互活动的参与者（交互时间，交互地点等）信息，另一方面体现了交互活动所产生的附加的信息，如成绩，应聘状态，借书还书时间等等。
- 因为这些信息不是历史，即不是不可改变的，相反，这些信息会在后续的其他交互活动中被更新。


18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 172



7 小结

- 面向对象是一种的程序设计方法，或者说它是一种程序设计范型，其基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。
- 面向对象的思想已经涉及到软件开发的各个方面，包括面向对象的分析、设计、编程和测试等。
- 面向对象建模技术所建立的四中模型，即用例模型、逻辑模型、交互模型和部署模型，分别从四个不同侧面描述了所要开发的系统。
- UML是一种基于面向对象的可视化建模语言。其提供了五种模型视图，包括用例模型视图、结构模型、行为模型视图、实现模型视图和部署模型视图，也称为UML的4+1模型视图。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 173



7 小结

- 分析建模的目标是创建各种表现形式，以描绘软件信息、功能和行为需求。
- 面向对象分析就是检查一组用例的问题域，尽量提取定义问题的类及其类之间的关系、并使用UML图建模和编写用例场景，以及开发活动图和泳道图来加以刻画。
- 基于类的建模使用从基于场景的描述中提取分析类，可以使用语法分析从文本叙述中提取候选类、属性和操作。
- CRC卡可以用于定义类之间的关系和获取类的职责以及协作类，并用逐步分析类聚合和继承关系及依赖。
- UML包图可用于描述系统的逻辑架构，使用层的方式划分系统，定义包之间的关联和依赖，便于开发人员分工和并发工作。
- 类建模和包图的描述为分析建模提供了软件的静态视图，而行为建模描述了动态行为。
- 行为模型使用SSD、操作契约、顺序图和状态图来分析系统的动态行为。

18:01:01 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 174



7 小结

- 面向过程开发模式：处理过程的抽象
- 面向对象开发模式：对象的行为，对象是数据抽象与过程抽象的综合。
- 面向对象软件系统=对象+类+继承+通信
- 对象一类（消息、实例、方法、属性、封装、继承、多态性、重载、复用）
- OMT的思想：建立描述系统数据结构的对象模型；建立描述系统控制结构的动态模型；建立描述系统功能的功能模型。
- UML具有一个完整的概念模型，提供了一套完整、全面的表达方法
 - 用户模型：实例图
 - 结构模型：类图、对象图、包图
 - 行为模型：状态图、顺序图、活动图、协同图
 - 实现模型：构建图
 - 环境模型：配置图