# UML （Unified Modeling Language 统一建模语言）

# 1 面向对象分析与设计（OOA/D）

**建模：**建立人们能够理解的模型，比如地图，各种各样，帮助你到达目的地；

**分析（analysis）**对问题和需求的调查研究

**设计（design）**满足需求的概念上的解决方案；

**面向对象分析（Object-Oriented analysis）**在问题域内发现和描述对象

**面向对象设计（Object-Oriented design）**定义软件对象及它们之间的关系，完成任务。

**UML**不是OOA/D，也不是方法，它仅仅只是一种图形**表示法**。

建模三大要素：表示法、过程、工具。

**用例：**使用案例（使用过程），做什么。

**领域模型（OOA）：**识别问题中的概念，使用之中的使用者，行为等；确定哪些对象

**定义设计类图（静态建模）：**从领域模型及交互如中获得启示，定义软件类，包括属性、方法等等

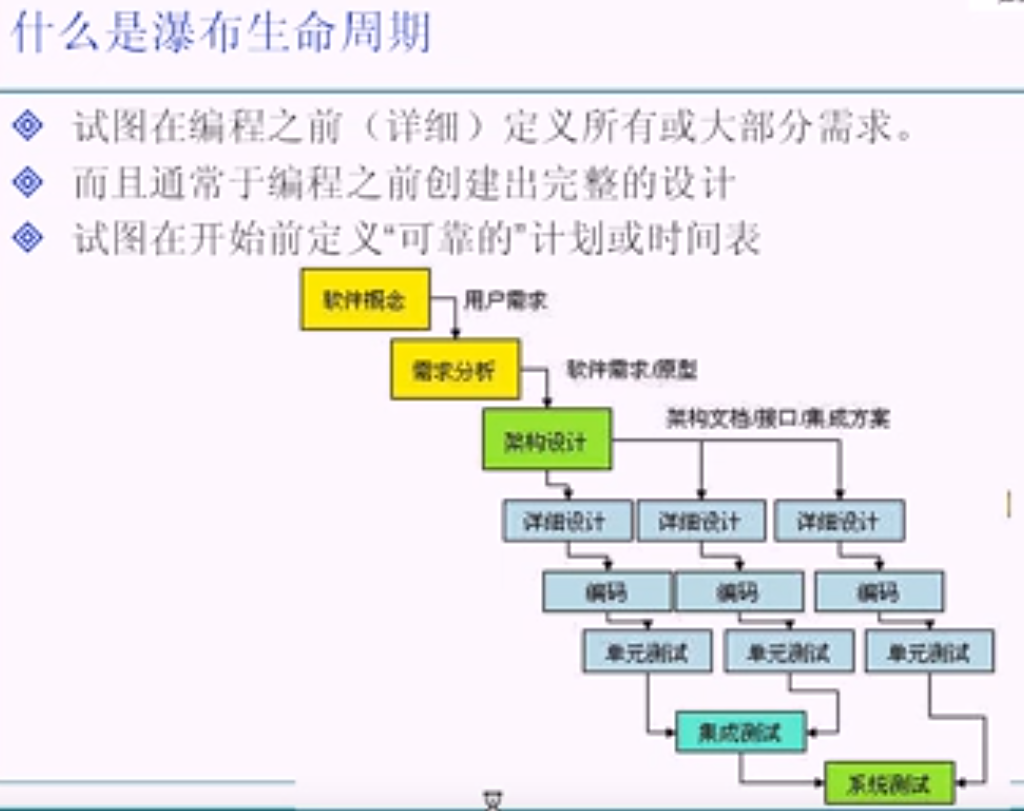
# 2 UP 过程与“瀑布”模型

软件开发过程（software development process）描述了构造、部署以及维护软件的方式。

统一软件开发过程（The Unified Software Development Process）是一种流行的构造面向对象系统的迭代软件开发过程。特别是，Rational统一过程（Rational Unified Process，RUP）是对统一过程的详细精化，并且已经被广泛采纳。

UP也可以引进其他方法中有用的实战，Extreme Programming，XP极限编程，其中的测试驱动开发（test-driven development）、重构（refactoring）和持续集成（continuous integration）等。

过程：先做什么，再做什么；



很难做到，唯一不变的就是变化。

# 3 UML概述

是描述、构造和文档系统制品的可视化语言。是一个庞大的图形化表示方法体系。

应用三种方式

草图

蓝图

编程语言

三要素

表示法-图形

过程-（UML与过程无关，但最好用于RUP）

工具（比如：Ratinal Rose）

指导性的意义，编程的辅助，不能主导编程。

**UML包括：**

**事务：**

结构：类、接口、构建（模块，jar包）、节点（服务器、数据库、打印机）等

行为：交互（消息）、状态等

分组：包、子系统等

注释：注释

**关系：**

依赖、关联（聚合、组合）、泛化generalization（java里叫继承）、实现realize（java的实现）

**图：**

用例图、交互图（顺序图、协作图）、类图(静态建模，其他都是动态)、活动图、状态图等

**扩展机制：**Stereotype、Tagged Value、Constraint

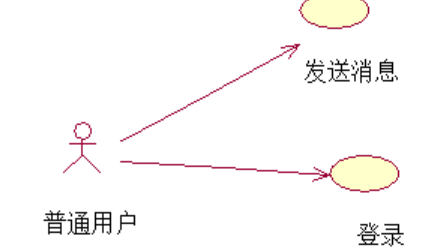
Rational Rose：一种建模工具

# 4 需求分析与用例（用例图use case view）

用例：使用的过程，强调用户的目标和观点

参与者(actor)

场景（scenano）



# 5 类图

类图允许我们去标记静态内容及类之间的关系

**类**的基本表示法

名称

属性（类型，可见性public、protected、private）

方法（参数，返回值）

**接口**的基本表示法

图形表示法

构造性表示法

**包**

**关系**

依赖（一个事务的变化影响另外一个事务）

关联（关联名、导航，角色，多重性、聚合（整体与部分）、组合（强聚合））

泛化（generalization）extends

实现（realize）implements

delete只删除图不删除模型；Ctrl+d是都删除

**正向工程：**从UML图形生成Java代码；

**逆向工程：**从Java代码生成UML图形；

【注意】 画图的目的不是为了生成代码；写代码的目的不是为了生成图形，只是辅助手段

MDA：模型驱动架构

什么时候用类图？

任何时候；类图是UML中最重要的图；不要尝试使用类图描述所有的细节，保持类图的简单

对概念建模（领域模型）

分析（分析类图）

实体类（领域模型）

控制类

边界类

# 6 领域模型（分析模型）

是OO分析中最重要的和经典的模型

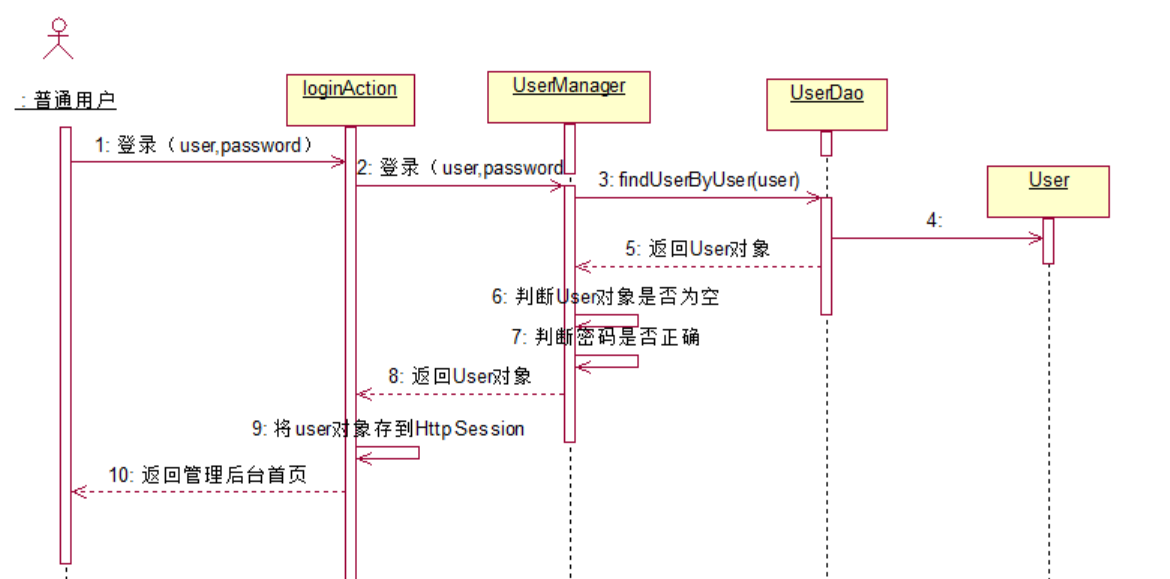
领域模型（domain model），也成为了概念模型，领域对象模型、分析对象模型

领域模型包括：概念、关联、属性

# 7 交互图（顺序图和协作图）

顺序图：对动态方面建模

协作图：一般用的少，这些对象大概是怎么交互的（不要画太多，不然很难看），一般用在设计的时候



没有表达不正确的，主要表达在正常的情况，如果还要表达不正确的应该用活动图。

# 8 UML活动图

activity diagram,描述事物或对象的活动变化流程，类似流程图。

简单活动：不能再分解的活动

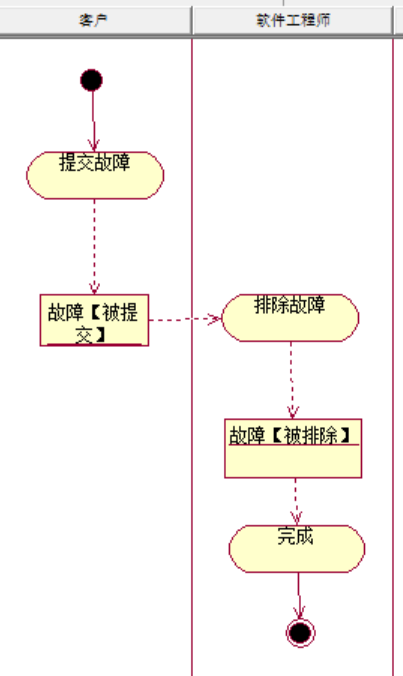
复合活动：可以再分解的

分支：

分劈：

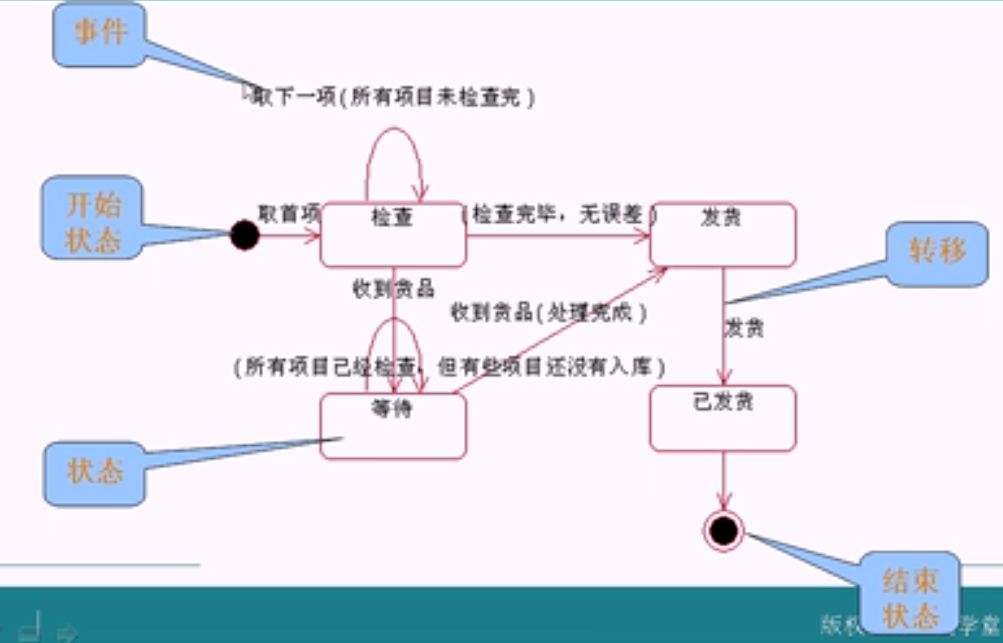
汇合：

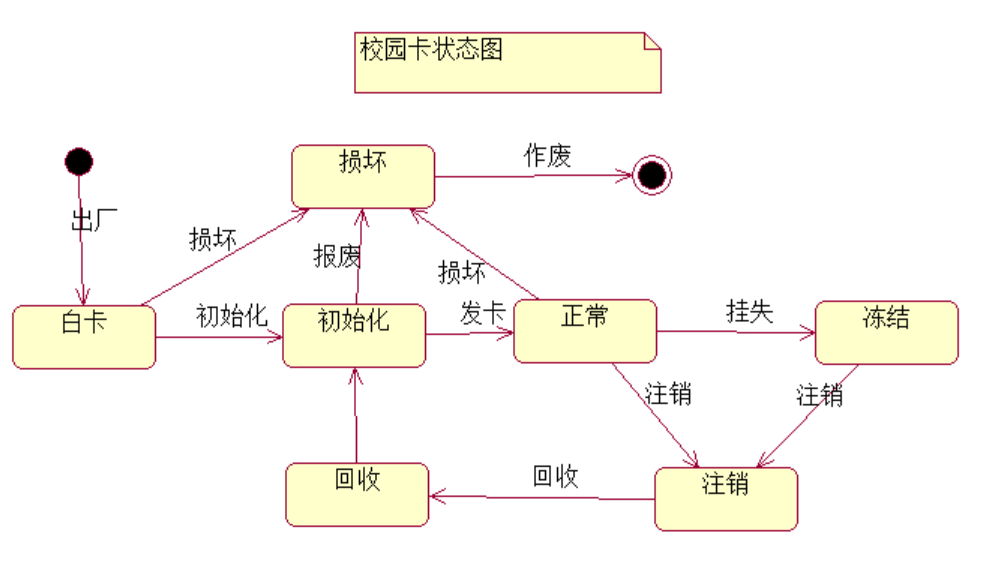
泳道：活动图中的区域划分，每一个泳道代表一个责任区，一个泳道中包含一组相关活动图



# 9 UML状态图

statechart diagram





# 10 GRASP与GOF（设计模型）

单独看23种设计模式；越早看越好。是一种想法

# 11 部署图

deployment diagram用来描述系统中计算机节点的拓扑结构和通信路径与节点运行的软件等；由于Visio更丰富好看，一般部署图用Visio画