

第一章 软件工程概述

1. 软件危机

(1) 定义：在计算机软件的开发和维护过程中，所遇到的一系列严重问题。包含两方面问题：如何开发软件，以满足对软件日益增长的需求；如何维护数量不断膨胀的已有软件。

(2) 典型表现

- ① 对软件开发成本和进度的估计很不准确
- ② 用户对已完成的软件不满意的现象经常发生
- ③ 软件的质量可靠性差
- ④ 软件的可维护性差
- ⑤ 软件没有适当的文档
- ⑥ 软件成本所占比例逐年上升
- ⑦ 软件开发生产率提高的速度，远远跟不上计算机就要用普及深入的趋势

(3) 产生原因

- ① 软件本身的特点
 - 1) 软件本身规模庞大
 - 2) 软件开发管理困难
- ② 软件开发与维护的方法不正确
 - 1) 软件开发技术、开发工具、生产方式落后
 - 2) 对用户要求没有完整准确的认识就匆忙着手编写程序

(4) 解决方法

- ① 推广使用在实践中总结出来的成功技术和方法，并研究探索更好的方法
- ② 开发和使用更好的软件工具
- ③ 采取良好的组织管理措施

2. 软件工程

(1) 定义：指导软件开发和维护的一门工程学科，即把工程应用于软件中

(2) 本质特性：

- ① 关注大型程序的构造
- ② 中心课题是控制复杂性
- ③ 软件经常变化
- ④ 开发软件的效率非常重要
- ⑤ 和谐地合作是开发软件的关键
- ⑥ 软件必须有效地支持它的用户
- ⑦ 在软件工程领域中，通常由具有一种文化背景的人，替具有另一种文化背景的人创造产品

(3) 基本原理

- ① 用分阶段的生命周期计划严格管理
- ② 坚持进行阶段评审
- ③ 实行严格的产品控制

- ④ 采用现代程序设计技术
- ⑤ 结果应能清楚地审查
- ⑥ 开发小组的人员应该少而精
- ⑦ 承认不断改进软件工程实践的必要性

3. 软件生命周期

(1) 软件定义

- ① 问题定义：确定总目标
- ② 可行性研究：确定可行性
- ③ 需求分析：确定软件必须具备的功能

(2) 软件开发

- ① 总体设计：设计程序的体系结构和实现方案，确定程序由哪些模块组成，以及模块间的关系
- ② 详细设计：确定实现每个模块具体的算法和数据结构
- ③ 编码和单元测试：把设计的结果翻译成代码程序
- ④ 综合测试：通过测试使软件达到预定要求

(3) 运行维护：使系统持久地满足用户需要

4. 软件过程模型

- (1) 瀑布模型：规范的、文档驱动的方法，阶段有顺序性、依赖性
- (2) 快速原型模型：快速建立原形供用户试用，以获取真正需求
- (3) 增量模型：分解成一系列增量构件，分批逐步交付，能在较短

时间内提交有部分功能的产品，方便用户逐步学习

- (4) 螺旋模型：在每个阶段开始前加入风险分析，降低风险
- (5) 喷泉模型：支持面向对象开发过程，体现过程迭代和无缝特性
- (6) Rational 统一过程 RUP
 - ① 最佳实践：迭代式开发、管理需求、使用基于构件的体系结构、可视化建模、验证软件质量、控制软件变更
 - ② 重复一系列生命周期循环，每次循环结束都交付一个版本
 - ③ 工作阶段：初始、精化、构建、移交
 - ④ 核心工作流：业务建模、需求、分析与设计、实现、测试；部署、配置与变更管理、项目管理、环境。
 - ⑤ 不同阶段对不同工作流的侧重点不同
- (7) 以 XP 极限编程为代表的敏捷过程：适用于小型项目、有限资源、有限时间，对变化和不确定性快速反应
- (8) 微软模型：每一个生命周期发布一个版本，持续迭代循环，规划、设计、开发、稳定、发布

第二章 可行性研究

1. 可行性

- (1) 分析几种主要的可能解法的利弊，确定问题是否值得去解决
- (2) 技术可行性
- (3) 经济可行性

- (4) 操作可行性
- (5) 社会可行性（法律）

2. 可行性研究过程步骤

- (1) 复查系统规模和目标
- (2) 研究目前正在使用的系统
- (3) 导出新系统的高层逻辑模型
- (4) 进一步定义问题
- (5) 导出和评价供选择的解法
- (6) 推荐行动方针
- (7) 草拟开发计划
- (8) 书写文档提交审查

第八章 软件维护

1. 软件维护

- (1) 定义：在软件已经交付使用之后，为了改正错误或满足新的需要而修改软件的过程。

2. 软件维护分类

- (1) 改正性维护：诊断和改正正在使用过程中发现的软件错误
- (2) 适应性维护：修改软件以适应环境的变化
- (3) 完善性维护：根据用户的要求改进或扩充软件使它更完善
- (4) 预防性维护：为将来的维护活动预先做准备

3. 软件维护特点

- (1) 结构化维护与非结构化维护差别巨大
- (2) 维护的代价高昂
- (3) 维护的问题很多

第九章 面向对象方法学引论

1. 传统方法学

- (1) 采用结构化技术完成软件开发，把软件生命周期的全过程依次划分为若干个阶段，然后顺序地完成每个阶段的任务。

2. 面向对象方法学要点

- (1) 认为客观世界是由各种对象组成的，任何事物都是对象，复杂的对象可以由简单的对象以某种方式组合而成。
- (2) 把所有对象都分成类，每个类都定义了一组数据和一组方法。
- (3) 按照父类和子类的关系，把若干个相关类组成一个层次结构的系统（类等级）。
- (4) 对象彼此间仅能通过发送消息相互联系。

3. 对象

- (1) 对象是由描述该对象属性的数据，以及可以对这些数据施加的所有操作，封装在一起构成的统一体。

4. 面向对象方法学优点

- (1) 与人类习惯的思维方法一致

- (2) 稳定性好
- (3) 可重用性好
- (4) 较易开发大型软件产品
- (5) 可维护性好

5. 类间关系

- (1) 关联
- (2) 泛化（继承）：继承是指能够直接获得已有的性质和特征，而不必重复定义它们。继承是子类自动地共享父类中定义的数据和方法的机制。
- (3) 依赖
- (4) 细化

第十三章 软件项目管理

- 1. 软件规模估算、工作量估算
 - (1) 代码行技术、功能点技术
 - (2) COCOMO2 模型
- 2. 进度计划
 - (1) 甘特图、工程网络、关键路径
 - (2) 估算工程进度、机动时间
- 3. 人员组织
 - (1) 民主制程序员组、主程序员组、现代程序员组

4. 假设自己被指定为项目负责人，任务是开发一个应用系统，该系统类似于自己的小组以前做过的那些系统，但是规模更大且更复杂一些。客户已经写出了完整的需求文档。应选用哪种项目组结构？为什么？打算采用哪种（些）软件过程模型？为什么？

答：(1) 由于待开发的应用系统类似于以前做过的系统，开发人员已经积累了较丰富的经验，没有多少技术难题需要攻克。为了减少通信开销，充分发挥技术骨干的作用，统一意志统一行动，提高生产率，加快开发进度，项目组的组织结构以基于程序员组的形式为宜。

(2) 针对待开发的系统，客户已经写出了完整的需求文档，项目组又有开发类似系统的经验，因此，可以采用广大软件工程师熟悉的瀑布模型来开发本系统。

5. 假设自己被指派为一个软件公司的项目负责人，任务是开发一个技术上具有创新性的产品，该产品把虚拟现实硬件和最先进的软件结合在一起。由于家庭娱乐市场的竞争非常激烈，这项工作的压力很大。应选择哪种项目组结构？为什么？打算采用哪种（些）软件过程模型？为什么？

答：(1) 由于是技术上具有创新性的产品，所以需要采用民主制程序员组，大家可以集思广益，共同攻关技术难题。

(2) 要求把虚拟现实硬件和最先进的软件结合在一起，所以需要采用一种完整而且完美的模型进行开发，所以 RUP 最为合适。

6. 假设自己被指派作为一个大型软件产品公司的项目负责人，工作是管理该公司已被广泛应用的字处理软件的新版本开发。由于市场竞争激烈，公司规定了严格的完成期限并且对外公布了。应选择哪种项目组结构？为什么？打算采用哪种（些）软件过程模型？为什么？

答：(1) 应该选择现代程序员组，因为小组成员都能对发现程序错误持积极、主动的态度。能更好的适应竞争。

(2) 大型软件应采用演化模型中的螺旋模型。

10. Rational 统一过程主要适用于何种项目？

答：Rational 统一过程（RUP）强调采用迭代和渐增的方式来开发软件，整个项目开发过程由多个迭代过程组成。在每次迭代中只考虑系统的一部分需求，针对这部分需求进行分析、设计、实现、测试和部署等工作，每次迭代都是在系统已完成部分的基础上进行的，每次给系统增加些 N 的功能，如此循环往复地进行下去，直至完成最终项目。它适用于大型的需求不断变化的复杂软件系统项目。

11. 说明敏捷过程的适用范围。

答：敏捷过程具有对变化和不确定性的更快速、更敏捷的反应特性，而且在快速的同时仍然能够保持可持续的开发速度。较适用于开发可用资源及开发时间都有较苛刻约束的小型项目。

12. 说明微软过程的适用范围。

答：微软过程的每一个生命周期发布一个递进的软件版本，各个生命周期持续、快速地迭代循环。它综合了 Rational 统一过程和敏捷过程的许多优点，是对众多成功项目的开发经验的正确总结；适用于商业环境下具有有限资源和有限开发时间约束的项目。

7. 什么是软件生命周期模型?试比较瀑布模型、快速原型模型、增量模型和螺旋模型的优缺点,说明每种模型的适用范围。

答: (1) 瀑布模型

①优点

- a. 可强迫开发人员采用规范的方法;
- b. 严格地规定了每个阶段必须提交的文档;
- c. 要求每个阶段交出的所有产品都必须经过质量保证小组的仔细验证;
- d. 对文档的约束,使软件维护变得容易一些,且能降低软件预算。

②缺点

- a. 开发过程一般不能逆转,否则代价太大;
- b. 实际的项目开发很难严格按该模型进行;
- c. 客户往往很难清楚地给出所有的需求;
- d. 软件的实际情况必须到项目开发的后期客户才能看到。

③适用范围

- a. 用户的需求非常清楚全面,且在开发过程中没有或很少变化;
- b. 开发人员对软件的应用领域很熟悉;
- c. 用户的使用环境非常稳定;
- d. 开发工作对用户参与的要求很低。

(2) 快速原型模型

①优点

- a. 可以得到比较良好的需求定义,容易适应需求的变化;
- b. 有利于开发与培训的同步;
- c. 开发费用低、开发周期短且对用户更友好。

②缺点

- a. 客户与开发者对原型理解不同;
- b. 准确的原型设计比较困难;
- c. 不利于开发人员的创新。

③适用范围

- a. 对所开发的领域比较熟悉而且有快速的原型开发工具;
- b. 项目招投标时,可以以原型模型作为软件的开发模型;
- c. 进行产品移植或升级时,或对已有产品原型进行客户化工作时。

(3) 增量模型

①优点

- a. 能在较短时间内向用户提交可完成部分工作的产品;
- b. 逐步增加功能,减少了全新的软件可能给客户组织带来的冲击。

②缺点

- a. 并行开发构件有可能遇到不能集成的风险,软件必须具备开放式的体系结构;
- b. 增量模型的灵活性很容易退化为边做边改模型,从而是软件过程的控制失去整体性。

③适用范围

- a. 进行已有产品升级或新版本开发;
- b. 对完成期限严格要求的产品;
- c. 对所开发的领域比较熟悉而且已有原型系统。

(4) 螺旋模型

①优点

- a. 对可选方案和约束条件的强调有利于已有软件的重用;
- b. 减少了过多测试或测试不足所带来的风险;
- c. 在螺旋模型中维护只是模型的另一个周期,在维护和开发之间并没有本质区别。

②缺点

- a. 需要丰富的风险评估经验和专门知识,如果未能够及时标识风险,会造成重大损失;
- b. 过多的迭代次数会增加开发成本,延迟提交时间。

③适用范围

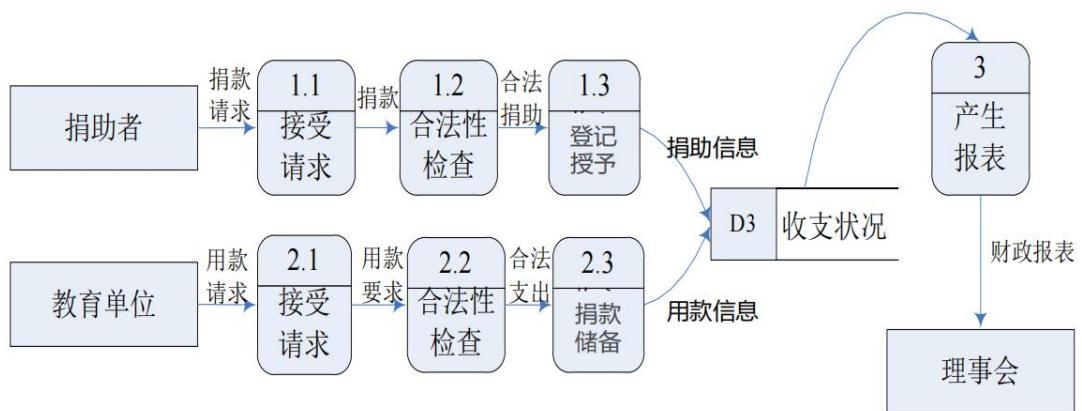
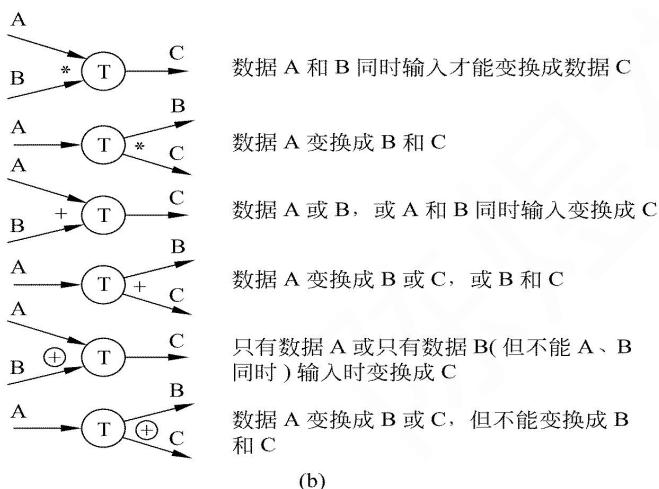
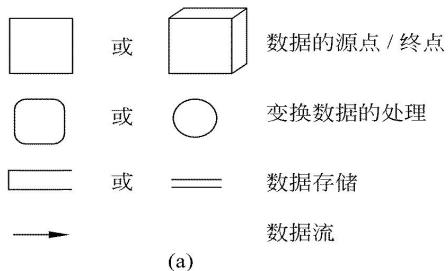
- 适用于内部开发的大规模软件项目。

开发模型	特点	适用场合
瀑布模型	线性模型，每一阶段必须完成规定的文档	需求明确的中、小型软件开发
快速原型模型	用户介入早，通过迭代完善用户需求，应用快速开发工具	需求模糊的小型软件开发
增量模型	每次迭代完成一个增量，可用于OO开发	容易分块的大型软件开发
螺旋模型	典型迭代模型，重视风险分析，可用于OO开发	具有不确定性的大型软件开发
喷泉模型	典型的OO过程模型，体现迭代和无缝的特性	需求明确的中小型软件开发

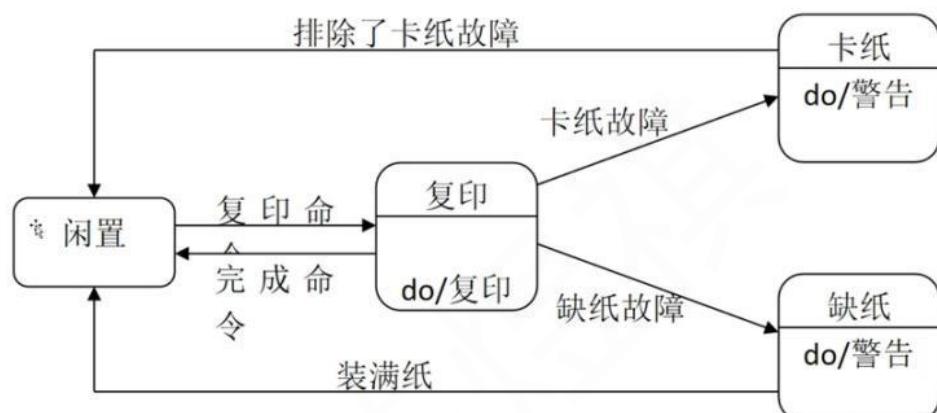
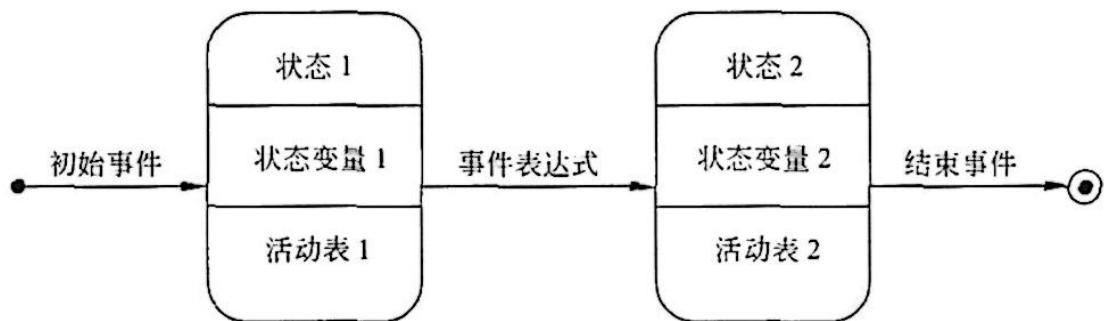
应用题

重要

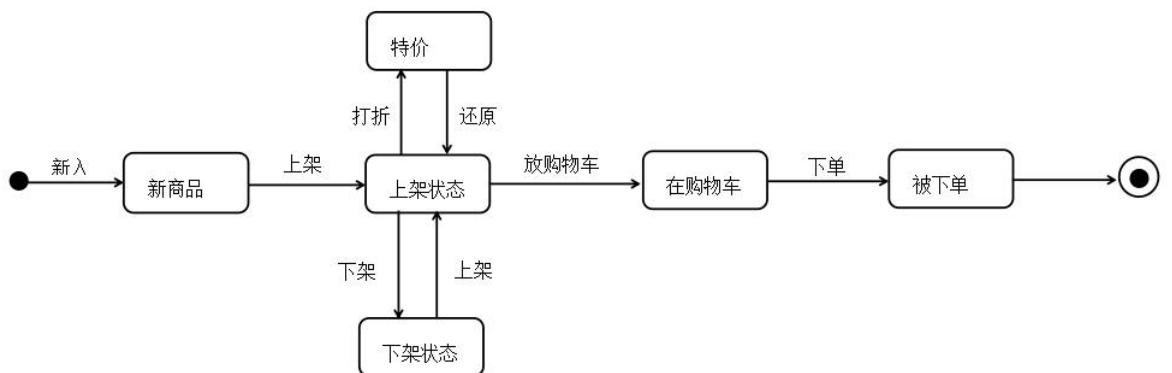
1. 数据流图



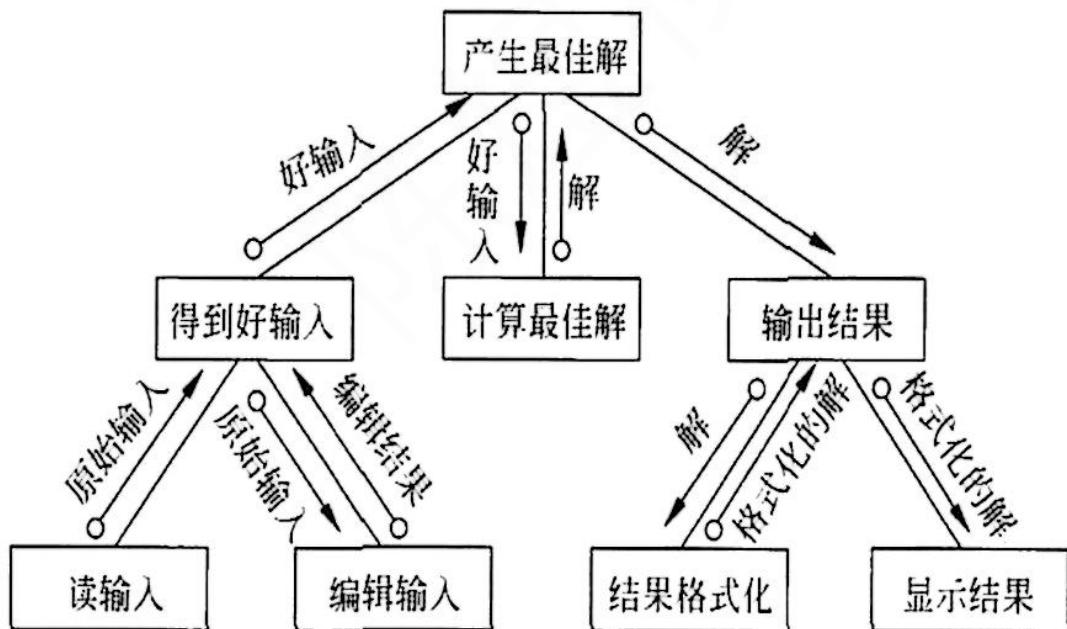
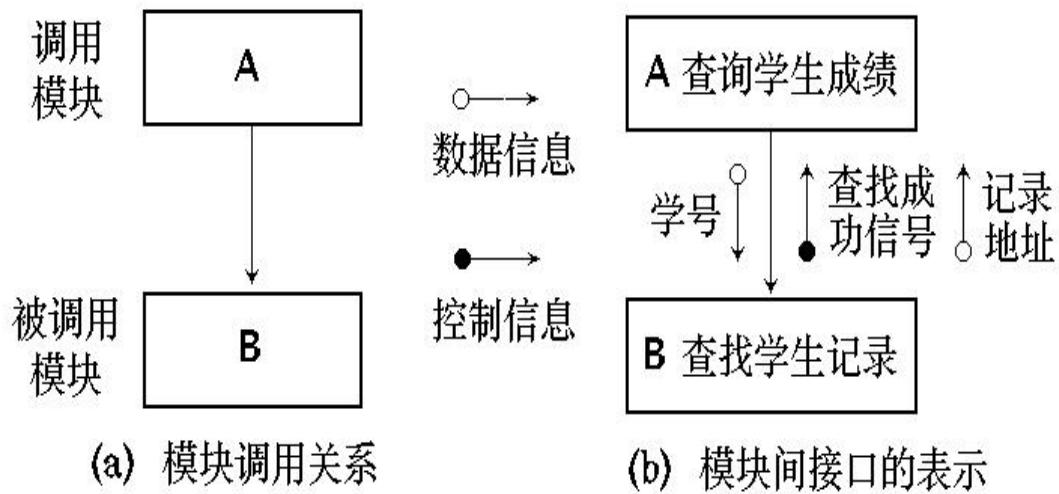
2. 状态转换图



复印机的状态转换图

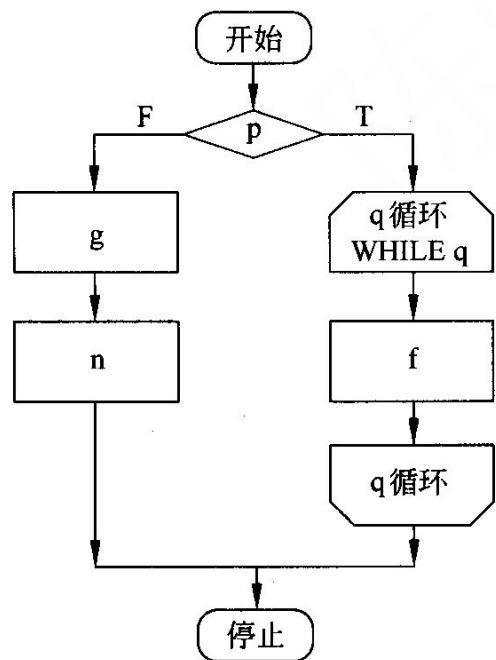
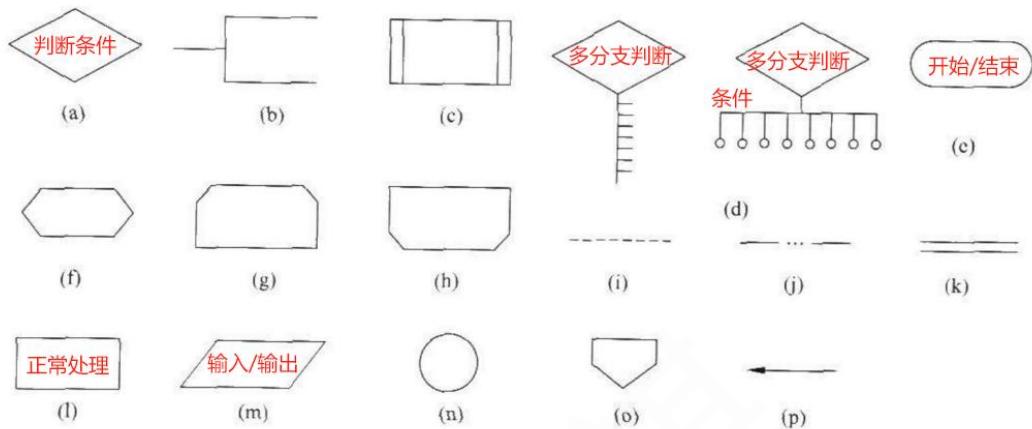


3. 软件结构图

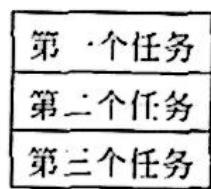


4. 程序流程图/盒图/PAD 图

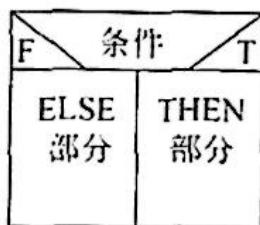
4.1 程序流程图



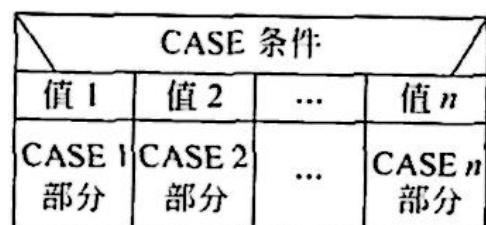
4.2 盒图



(a)



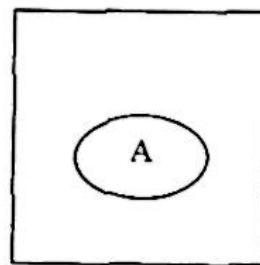
(b)



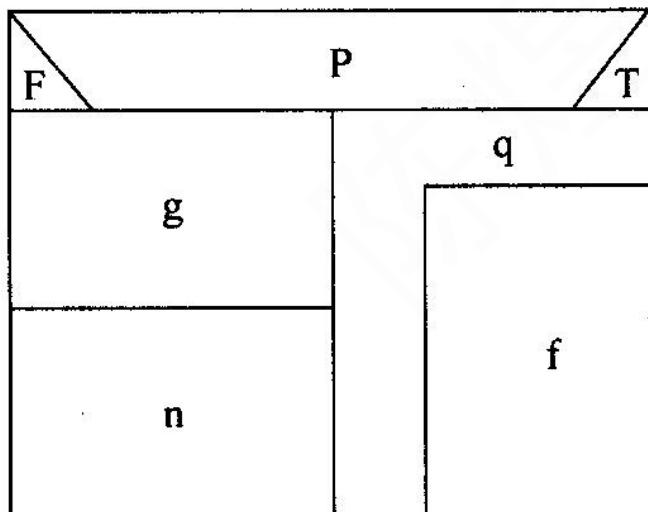
(c)



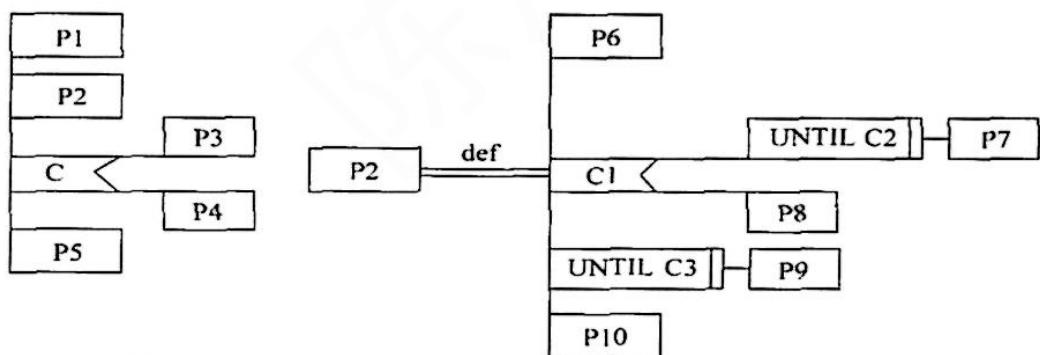
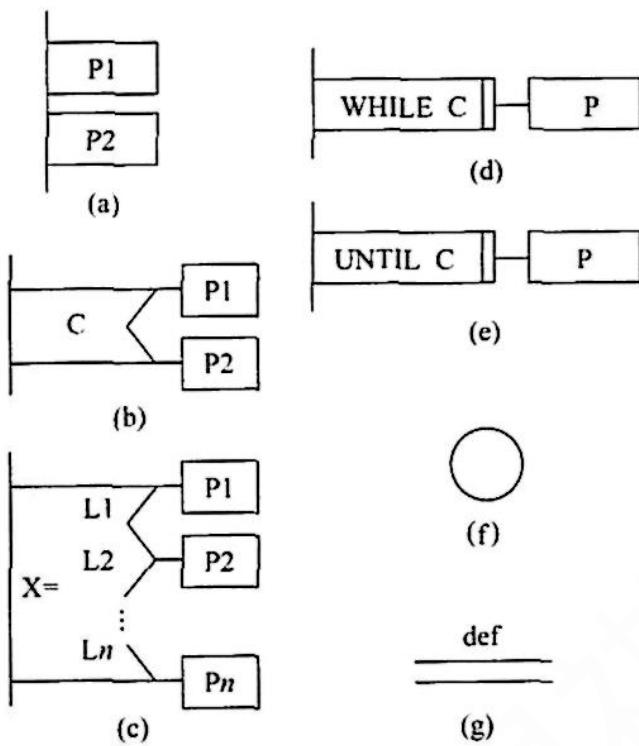
(d)



(e)



4.3 PAD 图



5. 黑盒/白盒测试

5.1 白盒测试

- (1) 语句覆盖：每个语句执行一次
- (2) 判定覆盖：每个语句执行一次，每个分支执行一次
- (3) 条件覆盖：每个语句执行一次，判定表达式中每个子条件两种结果
- (4) 判定/条件覆盖：(2) AND (3)
- (5) 条件组合覆盖：每个判定表达式内，所有子条件组合都执行一次

5.1 黑盒测试

(1) 等价类划分

1、输入值范围：一个有效类，两个无效类

2、输入值个数：一个有效类，两个无效类

3、输入值条件：针对具体情况，若干个有效类，若干个无效类

输入条件	有效等价类	无效等价类
地区码	1.空白; 2.三位数;	1.有非数字字符； 2.少于三位数字； 3.多于三位数字。
前 缀	3. 从 200 到 999 之间的三位	4.有非数字字符； 5.起始位为 ‘0’ ； 6.起始位为 ‘1’ ； 7.少于三位数字； 8.多于三位数字。
后 缀	4. 四位数字。	9. 有非数字字符； 10.少于四位数字； 11.多于四位数字。

1、尽可能多地覆盖有效类

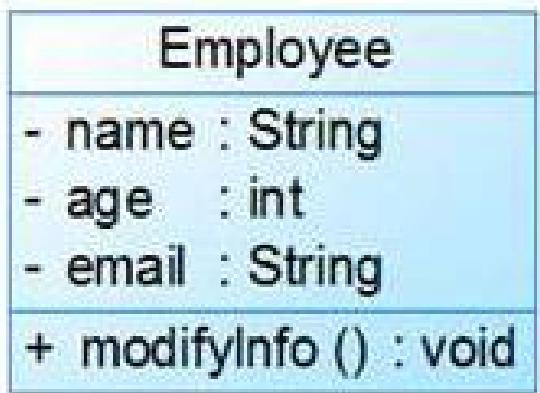
2、只覆盖一个无效类

方案	内容（覆盖的等价类）			输入	预期输出
	地区码	前缀	后缀		
1	空白	200~999 之间的三位数字	四位数字	()276-2345	有效
2	三位数字		四位数字	(635)805-9321	有效
3	有非数字字符			(0)723-4567	无效
4	少于三位数字			(33)234-5678	无效
5	多于三位数字			(5555)345-6789	无效
6		有非数字字符		(345)2-3456	无效
7		起始位为 ‘0’		(345)012-3456	无效
8		起始位为 ‘1’		(345)132-3456	无效
9		少于三位数字		(345) 92-3456	无效
10		多于三位数字		(345)4562-3456	无效
11			有非数字字符	(345)34256	无效
12			少于四位数字	(345)342-356	无效
13			多于四位数字	(345)562-34567	无效

(2) 边界值分析

刚好等于、刚刚小于、刚刚大于边界值

6. 类图



类名、属性、服务

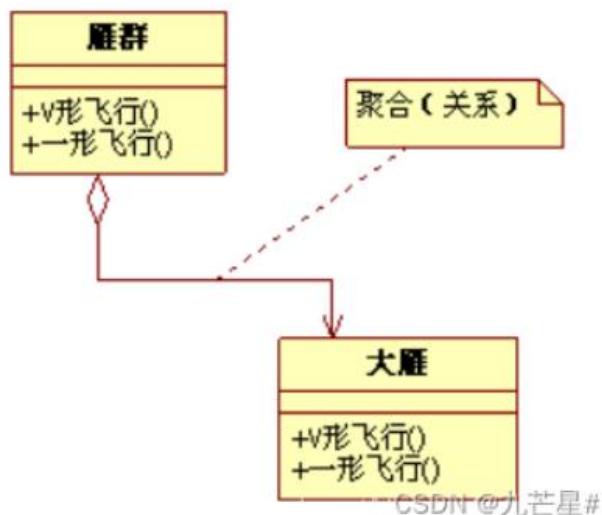
可见性 属性名:类型名 = 初值 {只读}

+公有, -私有, #保护

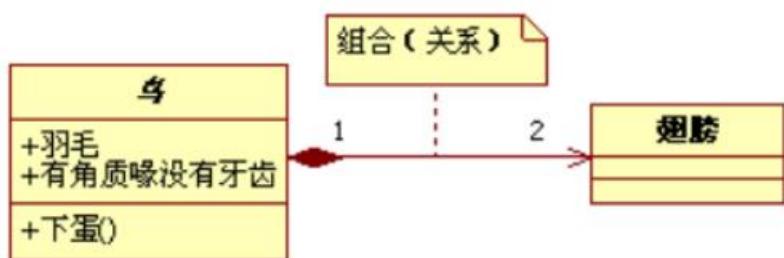
一般关联：关联名、黑三角、重数（最常见）



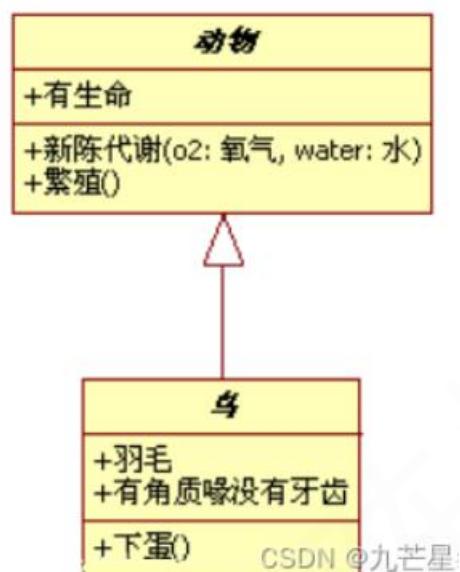
一般聚集：白色菱形



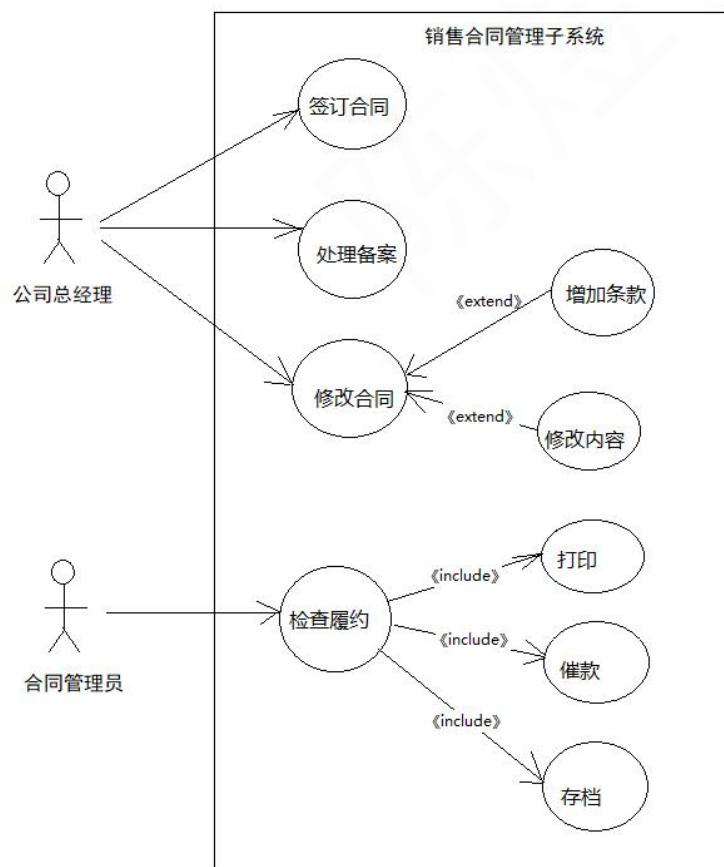
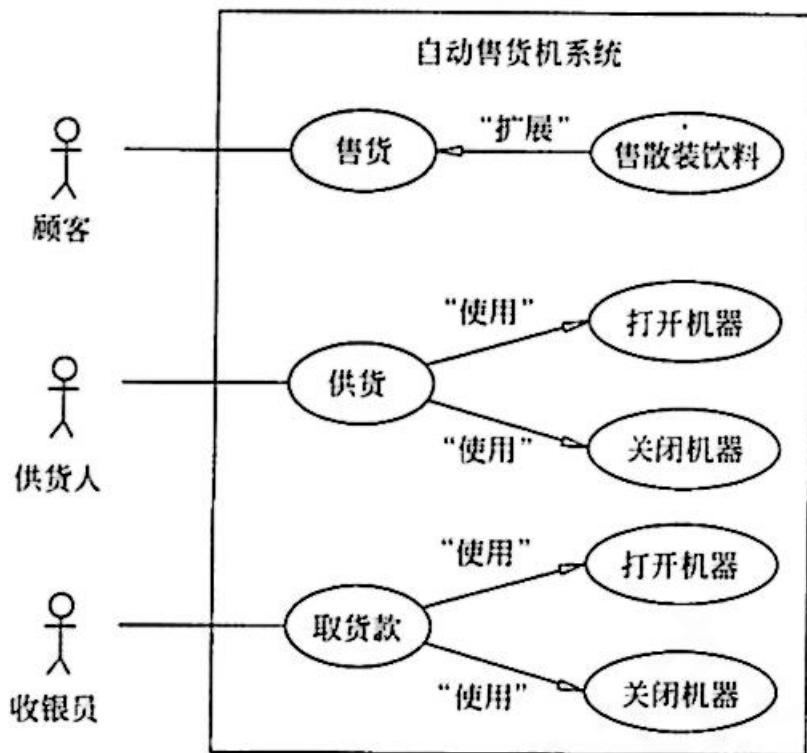
组合聚集：黑色菱形



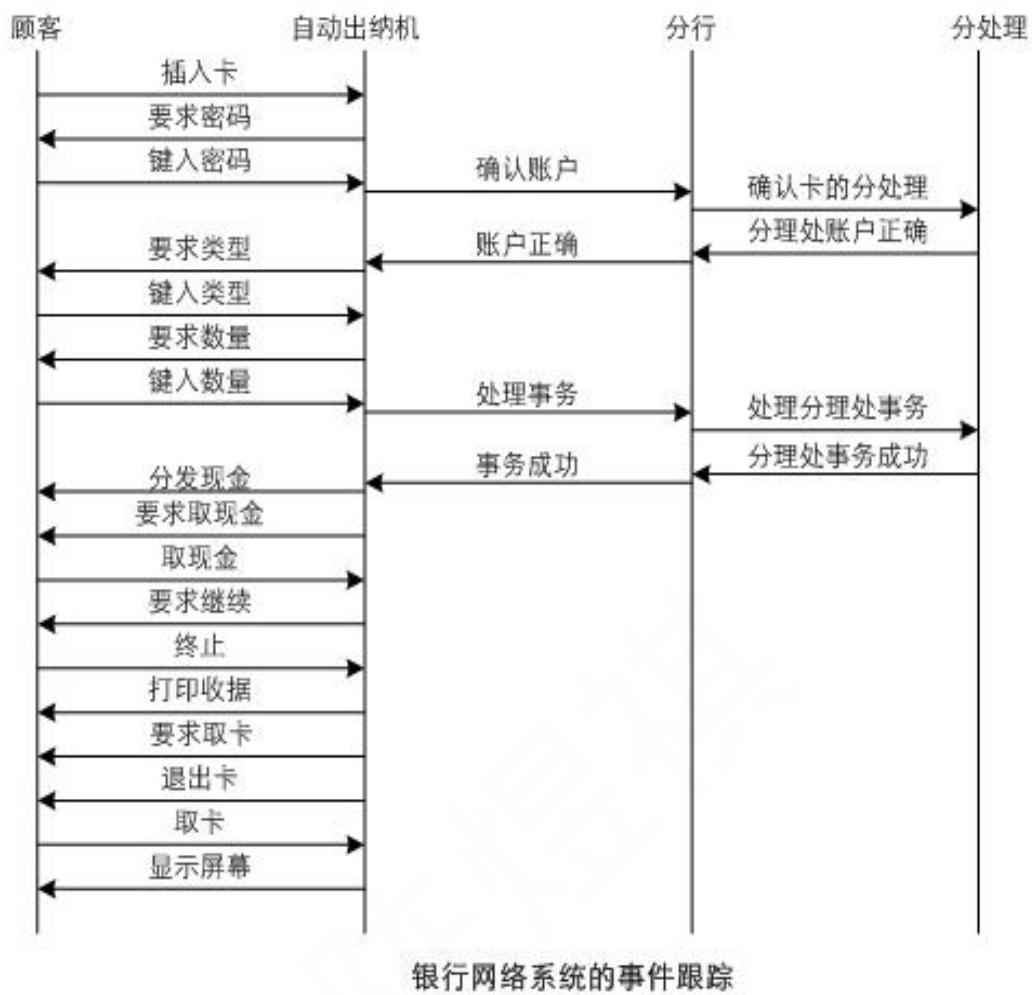
继承：白色三角



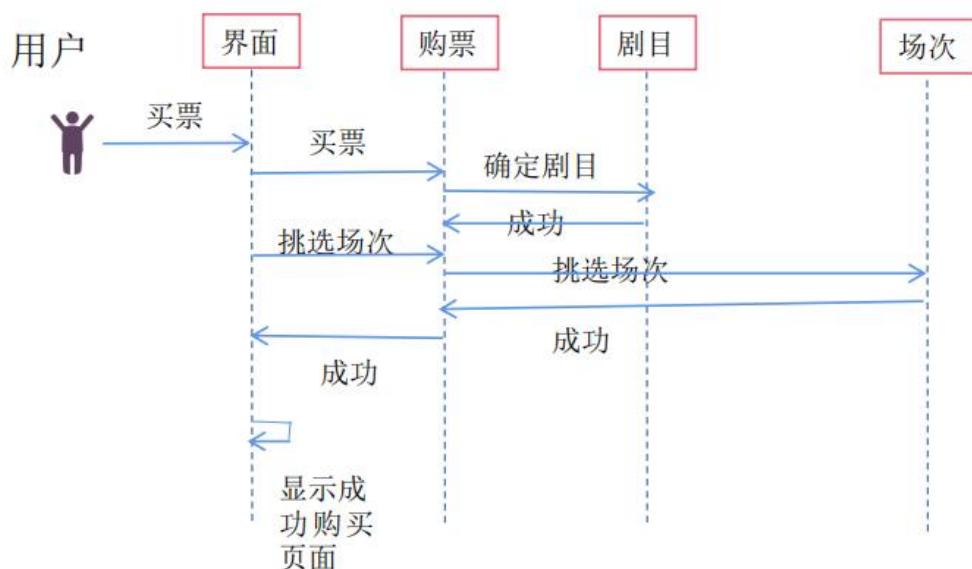
7. 用例图



8. 事件跟踪图

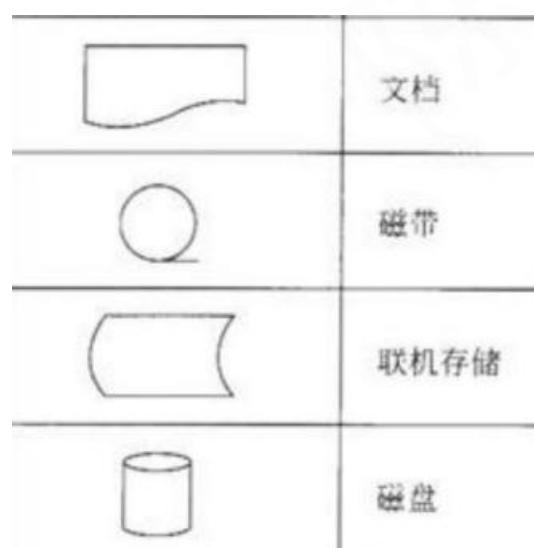
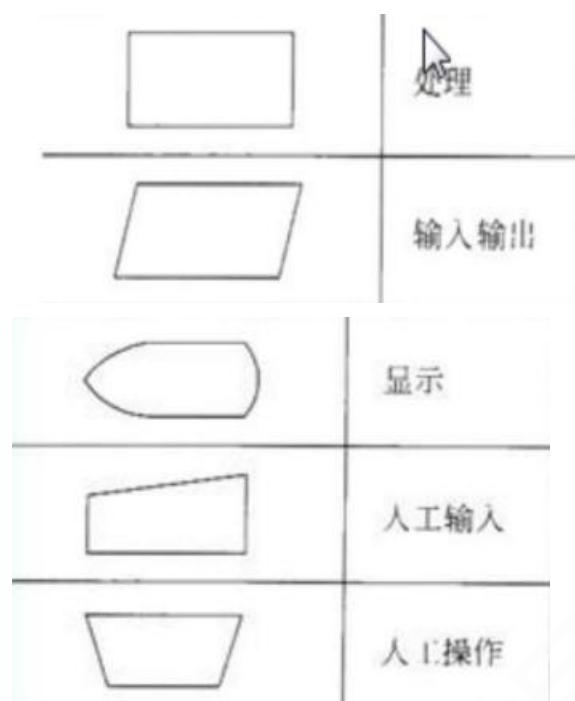


银行网络系统的事件跟踪



次要

1. 系统流程图



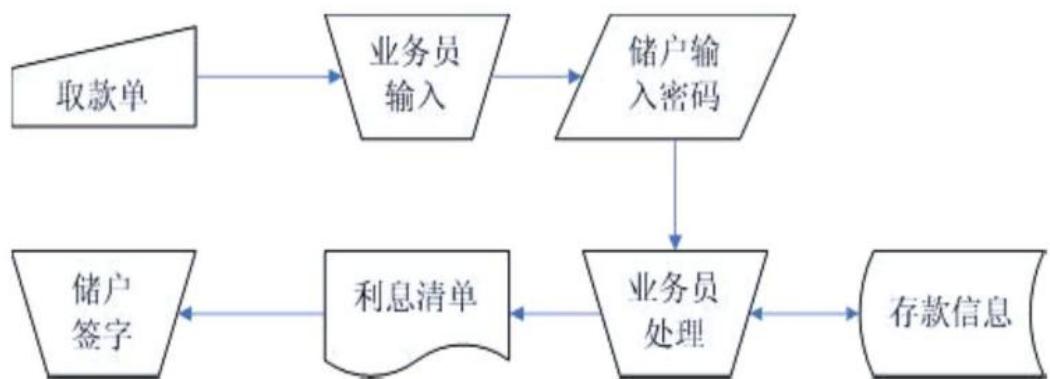
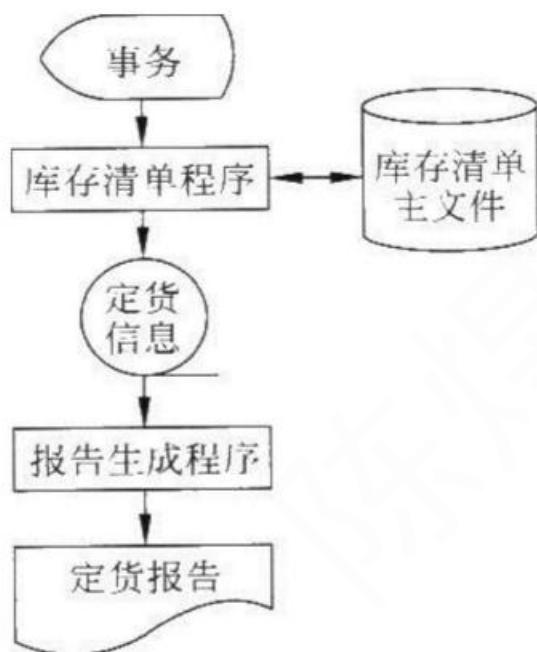


图 2-12 取款系统流程图



2. 数据字典

=定义， +连接， [a|b|c]， 1{字符}8， (可选)， “A00” .. “A99”

名字， 别名， 描述， 定义， 位置

名字:订货报表
别名:订货信息
描述:每天一次送给采购员的需要订货的零件表
定义:订货报表 = 零件编号 + 零件名称 + 订货数量 + 目前价格 + 主要供应商 + 次要供应商
位置:输出到打印机

名字:订货数量
别名:
描述:某个零件一次订货的数量
定义:订货数量 = 1{数字}5
位置:订货报表
订货信息

数据项名: 货物编号

别名: G-No, G-num

简述: 本公司所有的货物的编号

类型: 字符串

长度: 10

取值范围及含义:

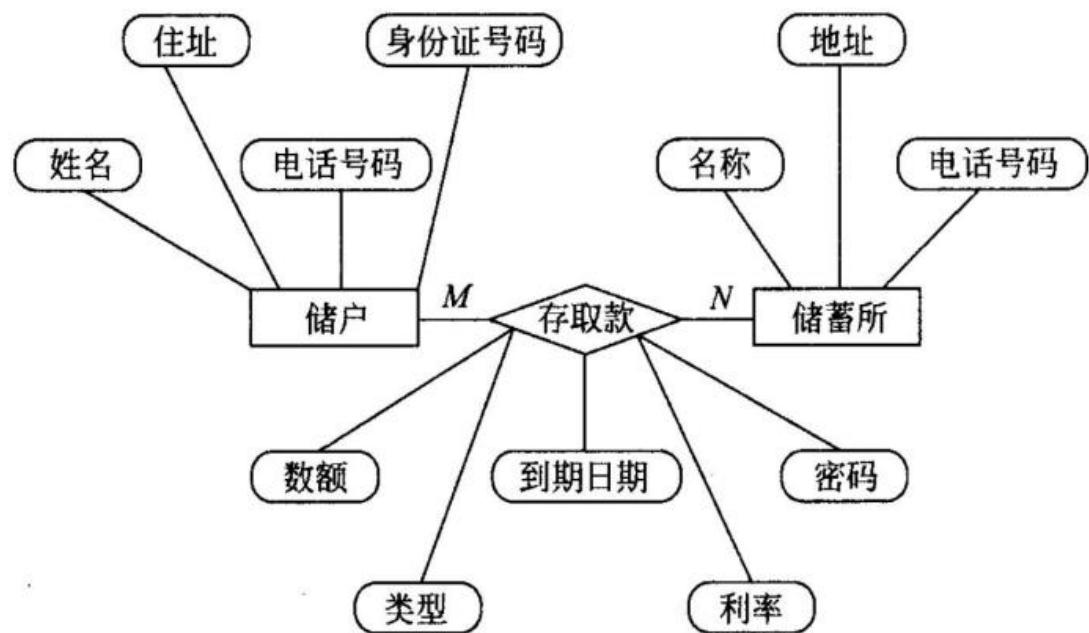
第1位: [J | G] (进口 / 国产)

第2 - 4位: L01.. L29 (类别)

第5 - 7位: “A00”.. “A99” (规格)

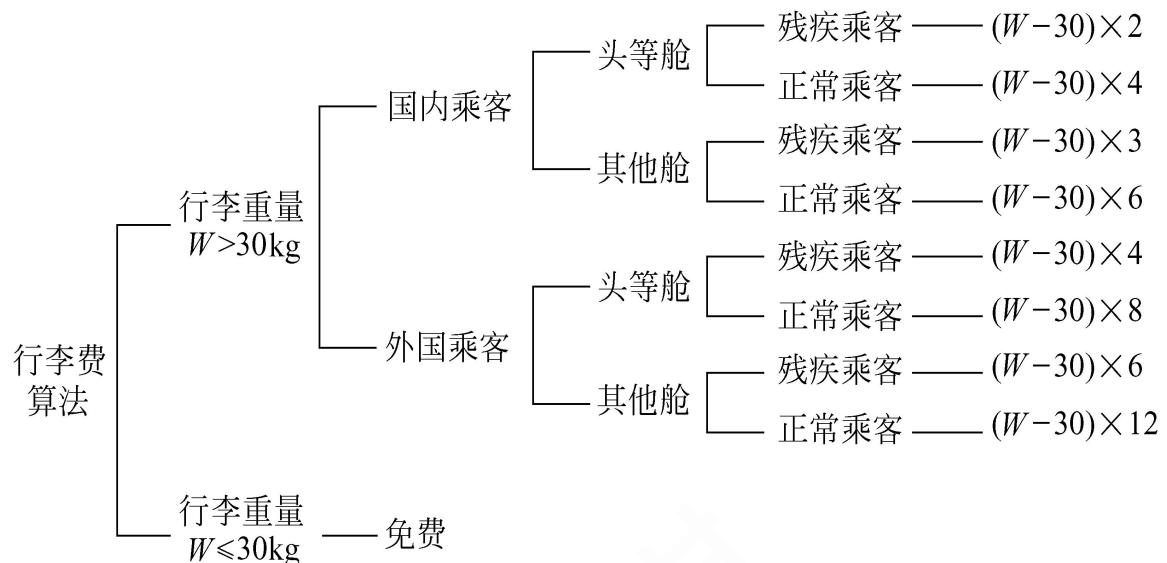
第8 - 10位: “001”.. “999” (品名编号)

3. E-R 图



4. 变换分析

5. 判定表/树



	1	2	3	4	5	6	7	8	9
国内乘客	T	T	T	T	F	F	F	F	F
头等舱	T	F	T	F	T	F	T	F	
残疾乘客	F	F	T	T	F	F	T	T	
行李重量 $W \leq 30\text{kg}$	T	F	F	F	F	F	F	F	
免费	×								
$(W-30) \times 2$				×					
$(W-30) \times 3$					×				
$(W-30) \times 4$		×						×	
$(W-30) \times 6$			×						×
$(W-30) \times 8$						×			
$(W-30) \times 12$							×		