# 第1章　软件工程学概述

## 1.1　复习笔记

四、软件过程

1. 定义

软件过程是为了获得高质量软件所需要完成的一系列任务的框架，它规定了完成各项任务的工作步骤。

1. 表示

使用生命周期模型简洁地描述软件过程。

（1）瀑布模型

②特点

第一，阶段间具有顺序性和依赖性；第二，推迟实现的观点；第三，质量保证的观点。

（2）快速原型模型

【注意】快速原型模型弥补了瀑布模型不适用于需求动态变更的缺点，其本质是“快速”。

（3）增量模型

④技术难点 a．要求软件体系结构必须是开放的。

b．增量模型本身是自相矛盾的。 c．增量模型的灵活性很容易退化为边做边改模型，从而使软件过程的控制失去整体性。

⑤风险更大的增量模型

图1-4是一种风险更大的增量模型。

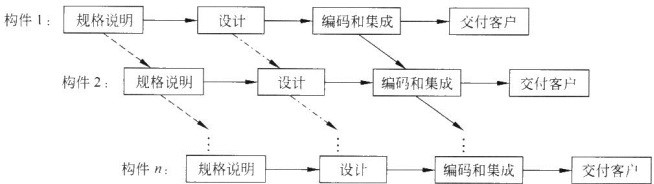


图1-4　风险更大的增量模型

【注意】每个增量构件的开发采用瀑布模型。

（5）喷泉模型

①开发流程图1-7所示的喷泉模型，是典型的面向对象的软件过程模型之一。

②特点

a．喷泉模型是一种以用户需求为动力，以对象为驱动的模型，主要用于描述面向对象的软件开发过程。

b．体现了面向对象软件开发过程迭代和无缝的特性。

（6）Rational统一过程（RUP）

②核心工作流

如上图纵轴所示，由上至下有九个核心工作流，前六个为核心过程工作流程，后三个为核心支持工作流程。

③工作阶段

a．初始阶段。 b．精化阶段。 c．构建阶段。 d．移交阶段。

④特点 a．采用迭代和渐增的方式开发软件。

b．具有多功能性和广泛适用性。

## 1.2　课后习题详解

1什么是软件危机？它有哪些典型表现？为什么会出现软件危机？

答：（1）软件危机是指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。

软件危机的典型表现：

①对软件开发成本和进度的估计不准确；

②用户对“已完成的”软件系统不满意的现象经常发生；

③软件产品的质量往往靠不住；

④软件常常是不可维护的；

⑤软件通常没有适当的文档资料；

⑥软件成本在计算机系统总成本中所占的比例逐年上升；

⑦软件开发跟不上计算机应用迅速普及深入的趋势。

软件危机出现的原因：

①客观原因

a．软件是计算机系统中的逻辑部件，缺乏“可见性”，因此管理和控制软件开发过程相当困难。

b．软件维护通常意味着改正或修改原来的设计，因此软件较难维护。 c．软件规模庞大，而程序复杂性将随着程序规模的增加而呈指数上升。

②主观原因 a．存在与软件开发和维护有关的许多错误认识和做法。

b．对用户要求没有完整准确的认识就匆忙着手编写程序。

c．开发人员只重视程序而忽视软件配置的其余成分（文档和数据等）。

d．软件开发人员轻视维护。

2假设自己是一家软件公司的总工程师，当把图1-13给手下的软件工程师们观看，告诉他们及早发现并改正错误的重要性时，有人不同意这个观点，认为要求在错误进入软件之前就清除它们是不现实的，并举例说：“如果一个故障是编码错误造成的，那么，一个人怎么能在设计阶段清除它呢？”应该怎么反驳他？

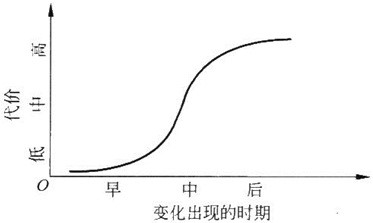


图1-13　引入同一变动付出的代价随时间变化的趋势

答：可以从以下两方面来反驳他：

1. 在软件开发的不同阶段进行修改需要付出的代价是很不相同的。在软件“已经完成”时再引入变动，当然需要付出更高的代价。在后期引入一个变动比在早期引入相同变动所需付出的代价高2～3个数量级。
2. 发生在编码时期的错误极有可能是在需求分析阶段由于操作不当产生的，所以必须及时消除错误，否则，到了后期软件运行和维护阶段再回过头来修改，将会付出更大的代价。

3什么是软件工程？它有哪些本质特性？怎样用软件工程消除软件危机？

答：（1）软件工程定义：

采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护它，这就是软件工程。

1. 软件工程本质特征：

①软件工程关注于大型程序的构造；

②软件工程的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。

1. 消除软件危机的方法：

①应该对计算机软件有一个正确的认识；

②应该充分认识到软件开发是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目；

③应该推广使用在实践中总结出来的开发软件的成功的技术和方法，并且研究探索更好更有效的技术和方法；

④应该开发和使用更好的软件工具。

4简述结构化范型和面向对象范型的要点，并分析它们的优缺点。

答：（1）结构化范型

①要点

结构化范型属于传统方法学。传统的软件开发方法大部分采用瀑布模型。这种模型要求每一阶段都以前一阶段形成的文档为基础完成工作。

每一阶段将要完成时，都要求开发人员进行验证或确认。

②优点 a．把软件生命周期划分成若干相对独立、简单的阶段，便于不同人员分工协作，降低开发的困难程度。

b．开发过程中采用科学的管理技术和良好的技术方法，且在每个阶段结束之前都进行严格的审查，保证了软件的质量，提高了软件的可维护性； c．大大提高了软件开发的成功率和生产率。

③缺点

a．不适用于软件规模庞大、或者对软件的需求是模糊的或会随时间变化而变化的情况。

b．结构化范型技术把数据和操作人为地分离成两个独立的部分，增加了软件开发与维护的难度。

（2）面向对象范型

①要点 a．把对象作为融合了数据及在数据上的操作行为的统一的软件构件。

b．把所有对象划分成各种对象类。

c．继承性。

d．对象彼此之间仅能通过传递消息互相联系。

②优点 a．降低了软件产品的复杂性； b．提高了软件的可理解性； c．简化了软件的开发和维护工作； d．提高了软件的可重用性。

③缺点面向对象性能比面向过程低。

6什么是软件过程？它与软件工程方法学有何关系？

答：（1）软件过程是为了开发出高质量的软件产品所需完成的一系列任务的框架，它规定了完成各项任务的工作步骤。软件过程定义了运用技术方法的顺序、应该交付的文档资料、为保证软件质量和协调软件变化必须采取的管理措施，以及标志完成了相应开发活动的里程碑。

（2）软件过程是软件工程方法学的三个重要组成部分之一。软件工程方法学的三个要素为：方法、工具和过程。

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：软件生命周期模型是软件开发的全部过程、活动和任务的结构框架。软件生命周期模型能清晰、直观地表达软件开发全过程，明确规定了要完成的主要活动和任务，用来作为项目开发的基础。

（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．瀑布模型是由文档驱动的；

b．缺乏对于需求变更和项目变化的适应性；

c．用户只有在生命周期的后期才能看到结果；

③适用范围

适用于用户需求明确、完整、无重大变化的软件项目开发。

（2）快速原型模型

①优点 a．克服了瀑布模型的缺点，对于需求变更具有更好的适应性； b．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．本质是快速，没有考虑软件的总体质量和可维护性，最终产品质量较差； 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．逐步增加产品功能，可以使用户有较充裕的时间学习和适应新产品。

②缺点

a．要求软件体系结构必须是开放的。

b．增量模型的灵活性很容易退化为边做边改模型，从而使软件过程的控制失去整体性。

c．增量模型本身是自相矛盾的。它一方面要求开发人员把软件看作一个整体，另一方面又要求开发人员把软件看作构件序列，每个构件本质上都独立于另一个构件。

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．有利于已有软件的重用，也有助于把软件质量作为软件开发的一个重要目标； 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．软件开发人员需要具备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。

8为什么说喷泉模型较好地体现了面向对象软件开发过程无缝和迭代的特性？

答：“喷泉”这个词体现了面向对象软件开发过程迭代和无缝的特性。喷泉模型的开发流程图中代表不同阶段的圆圈相互重叠，表示两个活动之间存在交迭；而面向对象方法在概念和表示方法上的一致性，保证了在各项开发活动之间的无缝过渡。在一个阶段内的向下箭头代表该阶段内的迭代（或求精）。

9试讨论Rational统一过程的优缺点。

答：（1）优点

提高了团队生产力，在迭代的开发过程、需求管理、基于组件的体系结构、可视化软件建模、验证软件质量以及控制软件变更等方面、针对所有关键的开发活动为每个开发成员提供了必要的准则、模板和工具指导，并确保全体成员共享相同的知识基础、它具有简洁和清晰的过程结构，为开发过程提供了较大的通用性。

（2）缺点 RUP只是一个开发过程，并没有涵盖软件过程的全部内容，例如它缺少关于软件运行和支持等方面的内容，此外，它没有支持多项目的开发结构，在一定程度上降低了在开发组织内大范围实现重用的可能性。

1. Rational统一过程主要适用于何种项目？

答：适用于大型的需求不断变化的复杂软件系统项目。

1. 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。因此适用于商业竞争环境下对小型项目提出的有限资源和有限开发时间的约束。

1. 说明微软过程的适用范围。

答：适用于商业环境下具有有限资源和有限开发时间约束的小型项目。

## 2.2　课后习题详解

1在软件开发的早期阶段为什么要进行可行性研究？应该从哪些方面研究目标系统的可行性？

答：（1）进行可行性研究的目的

可行性研究的目的是，用最小的代价在尽可能短的时间内研究并确定客户提出的问题能否解决。

（2）从下述三个方面研究每种解决方案的可行性

①技术可行性：使用现有的技术能否实现这个系统。

②经济可行性：这个系统的经济效益能否超过它的开发成本。

③操作可行性：这个系统的操作方式在该客户组织内是否行得通。

2为方便储户，某银行拟开发计算机储蓄系统。储户填写的存款单或取款单由业务员输入系统，如果是存款，系统记录存款人姓名、住址、存款类型、存款日期、利率等信息，并印出存款单给储户；如果是取款，系统计算利息并印出利息清单给储户。

写出问题定义并分析此系统的可行性。

答：（1）问题定义

①如果是存款，储户填写存款单，然后交给业务员键入系统，同时系统还要记录存款人姓名、住址（或电话号码）、身份证号码、存款类型、存款日期、利率等信息，完成后由系统打印存款单给储户。

②如果是取款，储户填写取款单，然后交给业务员，业务员把取款金额输入系统并要求储户输入密码以确认身份，核对密码正确无误后系统计算利息并印出利息清单给储户。

③为了满足储户的需求，该系统需要迅速的对用户的要求做出反馈，要对用户输入的信息作出最快的处理，所以就需要很大的主存容量，以及强大的数据库支持。由于是所面向的用户是广泛的储蓄用户群，所以需要系统强大的安全性能支持。

1. 可行性研究方法条件、假定和限制

①建议开发软件运行的最短寿命：5年。

②进行系统方案选择比较的期限：2个月。

③经费来源和使用限制：定制银行。

④硬件、软件、运行环境和开发环境的条件和限制：银行中心拥有大型

机以及用来支持的数据库，各个银行网点都有安好的PC机，安装有

Windows2000及以上的操作系统。

⑤建议开发软件投入使用的最迟时间：开发完成后试运行1个月。

1. 可行性研究方法

通过与银行熟练业务员进行深入讨论，制定详细用户调查问卷，真正了解用户以及银行业务员的实际需求，根据业务员提供的信息以及问题定义再综合调查问卷中用户提出的意见进行改进。最终确定项目需要解决的问题，并确定问题能不能被解决。决定可行性的主要因素：项目开发成本、所需设备置办成本、技术是否能满足需求、操作人员的熟练程度、资源有效性。

1. 处理流程和数据流程

①系统流程图

a．存款（如图2-5）

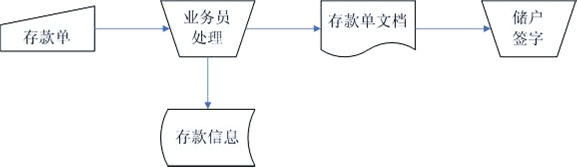


图2-5　存款系统流程图 b．取款（如图2-6）

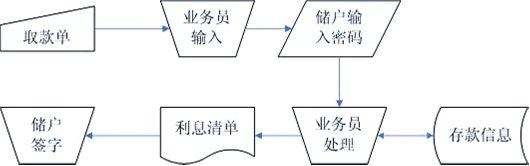


图2-6　取款系统流程图

②数据流图（如图2-7）

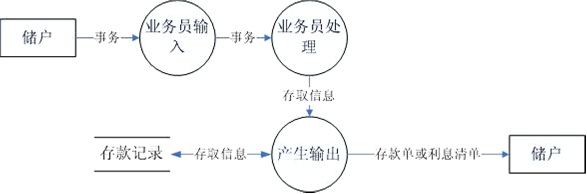


图2-7　银行业务数据流图

1. 分析影响因素

①工作负荷

当前大多数银行所使用的银行储蓄系统在办理业务时手续繁琐，人工业务操作过多，办理一个客户的业务就需耗费较长的时间，给银行工作人员增加了非常大的负担和额外的工作负荷。

②费用支出包括银行业务员及其他工作人员的工资，系统维护所需资金。

③人员需要大量的业务员、客服人员，系统维护人员以及其他工作人员。

④设备包括打印机、PC机、笔记本电脑。

⑤局限性工作效率低下，不能符合大量群众的及时需求，给群众的生活带来不便。具体问题如下：

a．当前银行使用的银行储蓄系统在办理业务时仅仅靠手工操作，人工业务占据了全部。对银行业务员的业务素质和数量都提出了很高的要求。

b．用纸张记录保存用户存款记录查找繁琐，耗时久，不方便，且容易丢失。且人工记录易发生差错。

c．存款记录保密性差，任何一个业务员都可以随意更改查阅用户数据，用户的资料容易泄漏。

d．业务员服务的时间有限，不能24小时办理业务，没有应急处理。

e．简单的业务经改进可由机器完成，这样节约了人力成本，也提高效率。

f．对现有系统的改进维护只能依靠增多业务员数量，增多银行分行数目，提高业务员业务素质来实现。增多业务员数量会导致银行人员支出大大增多，同时需要多选址建设分行，或者扩大分行规模，费用代价巨大。业务员的业务素质需要培养，不能短期见成效，这又浪费资金和人力资源。对现用系统的改进型维护已经不能解决储户越来越多、储户时间越来越长的问题。

3为方便旅客，某航空公司拟开发一个机票预订系统。旅行社把预订机票的旅客信息（姓名、性别、工作单位、身份证号码、旅行时间、旅行目的地等）输入进该系统，系统为旅客安排航班，印出取票通知和账单，旅客在飞机起飞的前一天凭取票通知和账单交款取票，系统校对无误即印出机票给

旅客。

写出问题定义并分析此系统的可行性。

答：（1）问题定义

①目标：在一个月内建立一个高效率，无差错的航空公司机票预定系统。

②存在的主要问题：人工不易管理，手续繁琐。

（2）可行性分析

①经济可行性

a．成本估算

打印机一台（2000元）＋开发费（3500元）＝5500元 b．效益估算

该系统有很好的社会效益，提高了公司售票效率，方便旅客，售票方便、科学。

②技术可行性经过调查分析，得到目前航空公司机票预定系统流程图如图2-8所示。

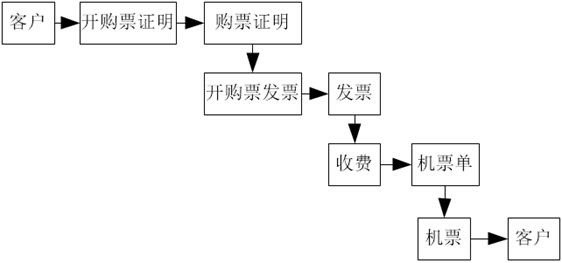


图2-8　订票系统流程图

③操作的可行性

比较以上两图看出，与人工系统相比，计算机保留了原有的主要工作流程，可以看出计算机系统是人工系统的优化，操作也不复杂，工作人员在短时间经过培训就可熟练掌握。

④结论

由于经济、技术、操作三方面的可行性分析都通过，因此开发航空公司机票预定系统是可行的。

4目前住院病人主要由护士护理，这样做不仅需要大量护士，而且由于不能随时观察危重病人的病情变化，还可能会延误抢救时机。某医院打算开发一个以计算机为中心的患者监护系统，试写出问题定义，并且分析开发这个系统的可行性。

医院对患者监护系统的基本要求是随时接收每个病人的生理信号（脉搏、体温、血压、心电图等），定时记录病人情况以形成患者日志，当某个病人的生理信号超出医生规定的安全范围时向值班护士发出警告信息，此外，护士在需要时还可以要求系统印出某个指定病人的病情报告。

答：（1）问题定义

①本系统的数据源点是“病人”和“护士”，他们分别提供生理信号和要求的病情报告相关信息。从系统应该“定时记录病人情况以形成患者日志”这项要求可以想到，还应该有一个提供日期和时间信息的“时钟”作为数据源点。

②本系统的数据终点是接收警告信息和病情报告的护士。系统对病人生理信号的处理功能主要是“接收信号”“分析信号”和“产生警告信息”。此外，系统还应该具有“定时取样生理信号”“更新日志”和“产生病情报告”的功能。

（2）分析可行性

为了分析病人生理信号是否超出了医生规定的安全范围，应该存储“患者安全范围”信息。此外，定时记录病人生理信号所形成的“患者日志”显然也是一个数据存储。本系统的基本系统模型如图2-9所示，图2-10是本系统的功能级数据流图。

# 第3章　需求分析

## 3.2　课后习题详解

1为什么要进行需求分析？通常对软件系统有哪些需求？

答：（1）需求分析的原因

为了开发真正满足用户需求的成功的软件产品，必须进行需求分析。

（2）对软件系统的需求

功能需求、性能需求、可靠性和可用性需求、出错处理、接口需求、约束、逆向需求、将来可能提出的要求。

2怎样与用户有效地沟通以获取用户的真实需求？

答：可采用以下四种方法获取真实需求：

1. 访谈；
2. 面向数据流自顶向下求精；
3. 简易的应用规格说明技术；
4. 快速建立软件原型。

4分析习题2第3题所述的机票预订系统。试用实体-联系图描绘本系统中的数据对象并用数据流图描绘本系统的功能。

答：（1）数据流图，如图3-5所示。



图3-5　购票系统数据流图

（2）E-R图，如图3-6所示。

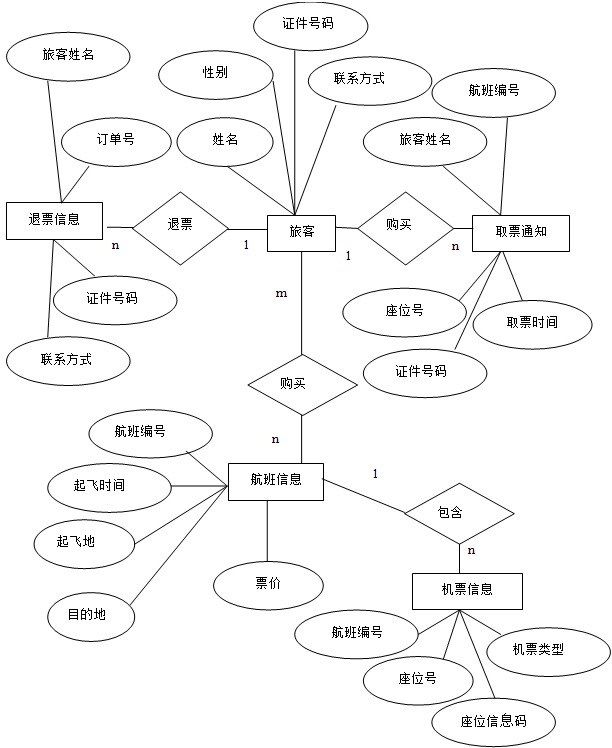


图3-6　购票系统E-R图

5分析习题2第4题所述的患者监护系统。试用实体联系图描绘本系统中的数据对象，画出本系统的顶层IPO图。

答：E-R图，如图3-7所示。IPO图如图3-8所示。

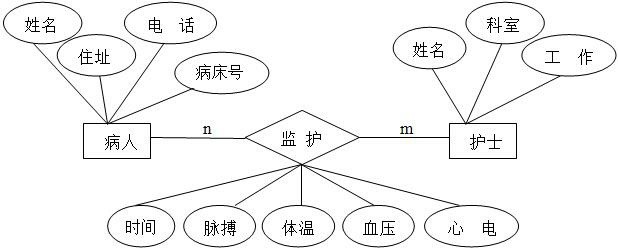


图3-7　患者监护系统E-R图

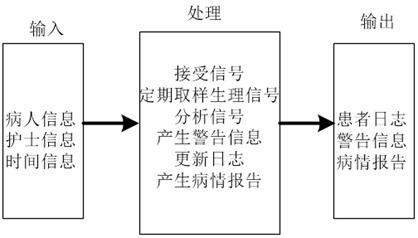


图3-8　患者监护系统IPO图

# 第5章　总体设计

四、描绘软件结构的图形工具

1层次图

1. 作用

层次图用来描绘软件的层次结构。

1. 符号

①矩形框代表一个模块。

②方框间的连线表示调用关系。

2HIPO图

HIPO图是“层次图加输入/处理/输出图”的缩写。在层次图里除了最顶层的方框外，每个方框都加了编号。

3结构图结构图的符号表示分为两类：

（1）基本符号

①方框代表一个模块，框内注明模块的名字或主要功能；

②方框之间的箭头（或直线）表示模块的调用关系；

③尾部是空心圆的箭头表示传递的是数据，实心圆的箭头表示传递的是控制信息。

（2）特殊符号

①表示当模块M中某个判定为真时调用模块A，为假时调用模块B，如图5-1所示；

②表示模块M循环调用模块A、B和C，如图5-2所示。

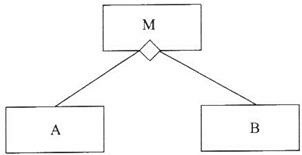


图5-1　判定为真时调用A，为假时调用B

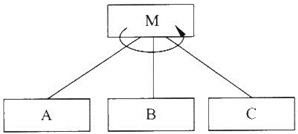


图5-2　模块M循环调用模块A、B、C

## 5.2　课后习题详解

1为每种类型的模块耦合举一个具体例子。

答：

1.非直接耦合：例如在实数的运算中，有两个模块分别完成n个实数的求和、求乘积功能，这两个模块之间没有直接联系，只是在主模块调用中有执行的前后顺序而已。

2.数据耦合：如有两个模块，其中，模块A完成两个数的比较，返回其中的较大值；模块B完成四个数的比较，返回其中的最大值。因此，为了完成四个数的比较，B模块调用A模块即可，在这种调用关系中，将模块B的四个参数两两传给模块A，模块A的运算结果作为参数参加B 的运算。这两个模块之间发生数据耦合。

3.标记耦合：若一组模块通过参数传递记录信息，这就是标记耦合。

4.控制耦合：遥控器与电器。

5.外部耦合：C语言中都访问被说明为extern类型的外部变量的各个模块之间发生外部耦合。

6.公共耦合：几个模块对同一个数据库进行查询就发生公共耦合。

7.内容耦合：两个模块之间有一部分代码重复（只可能发生在汇编程序中）或一个模块访问另一个模块的内部数据时，模块之间发生内容耦合。

2为每种类型的模块内聚举一个具体例子。

答：

1.逻辑内聚

一个子程序将打印季度开支报告、月份开支报告和日开支报告。具体打印哪一个，将由传入的控制标志决定，这个子程序具有逻辑内聚性，因为它的内部逻辑是由输入进去的外部控制标志决定的。

2.时间内聚

若一个模块包含了需要在同一时间段中执行的多个任务，则称该模块的内聚为时间内聚。例如，将多个变量的初始化放在同一个模块中实现，或将需要同时使用的多个库文件的打开操作放在同一个模块中，都会产生时间内聚的模块。

3.过程内聚

一个子程序，它产生读取雇员的名字，然后是地址，最后是它的电话号码。这种顺序之所以重要，仅仅是因为它符合用户的要求，用户希望按这种顺序进行屏幕输入。另外一个子程序将读取关于雇员的其他信息。

这个子程序是过程内聚性，因为是由一个特定顺序而不是其他任何原因，把这些操作组合在一起的。

4.顺序内聚

一个按给出的生日计算雇员年龄、退休时间的子程序，如果它是利用所计算的年龄来确定雇员将要退休的时间，那么它就具有顺序内聚性。而如果它是分别计算年龄和退休时间的，但使用相同生日数据，那它就只具有通信内聚性。

5.功能内聚

计算雇员年龄并给出生日的子程序就是功能内聚性的，因为它只完成一项工作，而且完成得很好。

6.偶然内聚

一个函数内调用读取文件函数实现对文件的读取，同时实现打印当前时间的功能，这两个功能之间除了前后执行之外没有什么联系，这就属于偶然内聚。

7.通信内聚

模块A实现将传入的Date类型数据转换成String类型，以及将Date类型数据插入数据库，这两个操作都是对“Date类型数据”而言的。模块A中就是通信内聚。

3用面向数据流的方法设计下列系统的软件结构。

1. 储蓄系统（参见习题2第2题）。
2. 机票预订系统（参见习题2第3题）。
3. 患者监护系统（参见习题2第4题）。

答：（1）储蓄系统

①数据流图，如图5-5和5-6所示

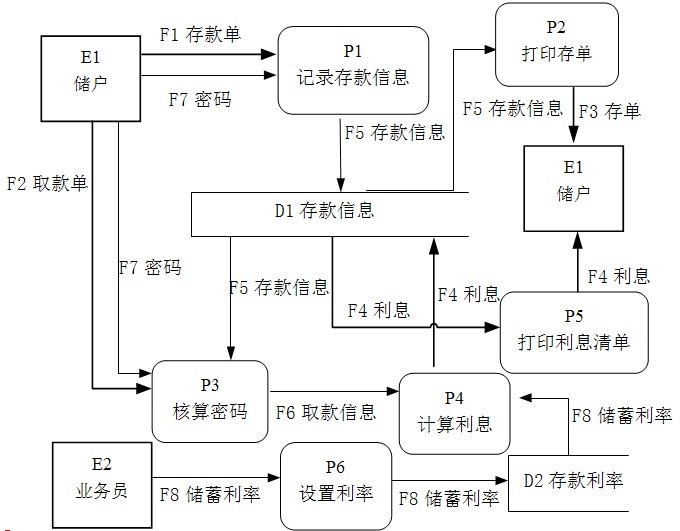


图5-5　储蓄系统数据流图（a）

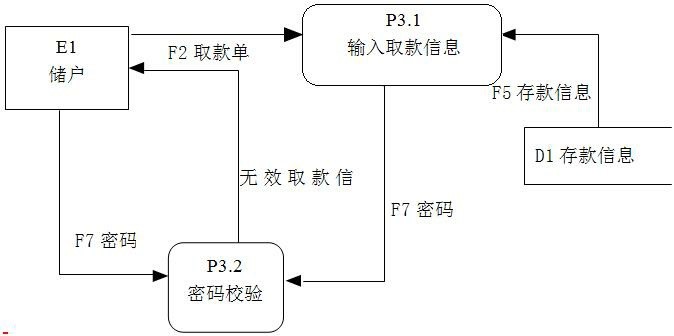


图5-6　储蓄系统数据流图（b）

②软件结构图，如图5-7所示。

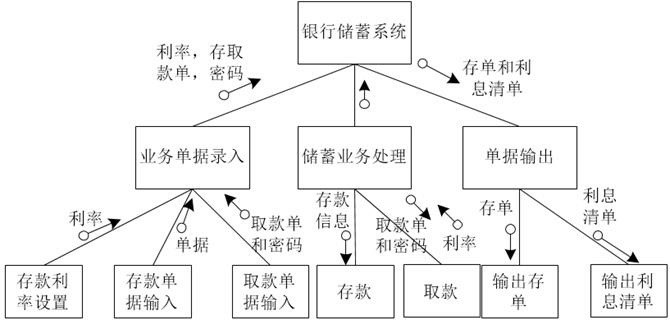


图5-7　储蓄系统软件结构图

1. 机票预定系统①数据流图，如图5-8、5-9、5-10、5-11、5-12所示。

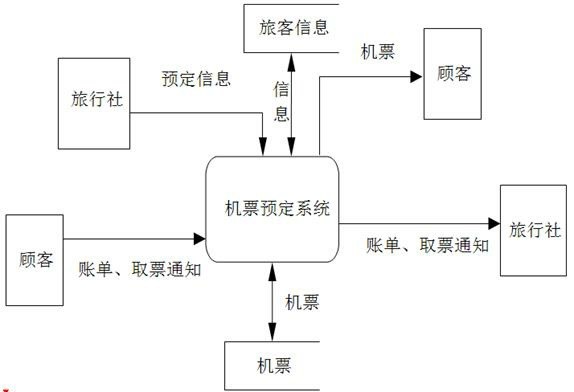


图5-8　机票预定系统数据流图（a）

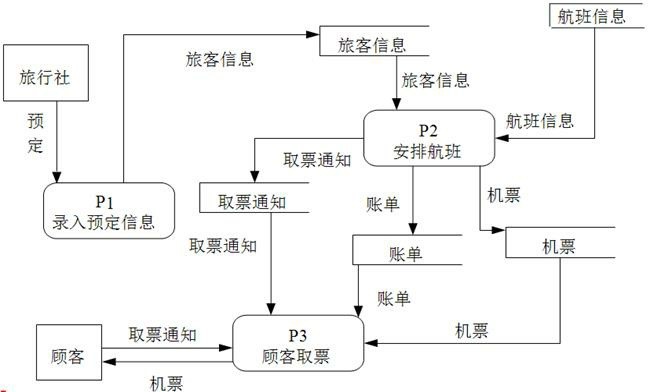


图5-9　机票预定系统数据流图（b）

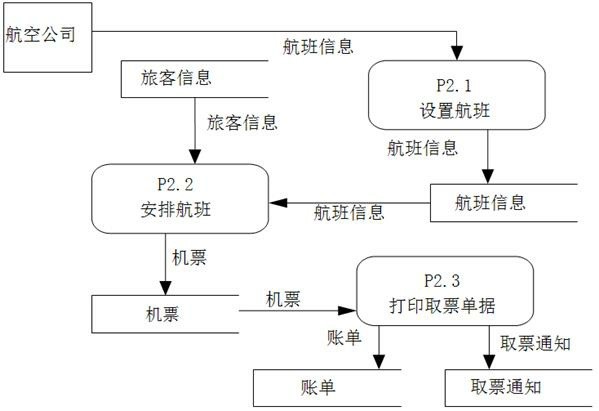


图5-10　机票预定系统数据流图（c）

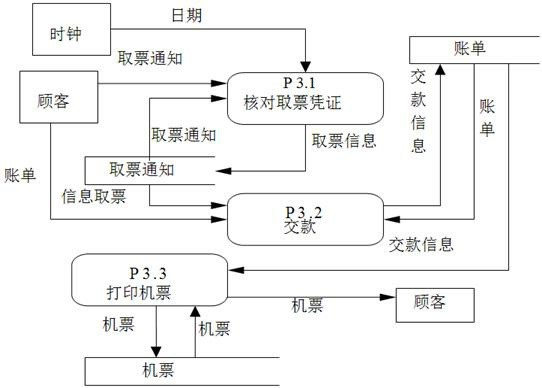


图5-11　机票预定系统数据流图（d）

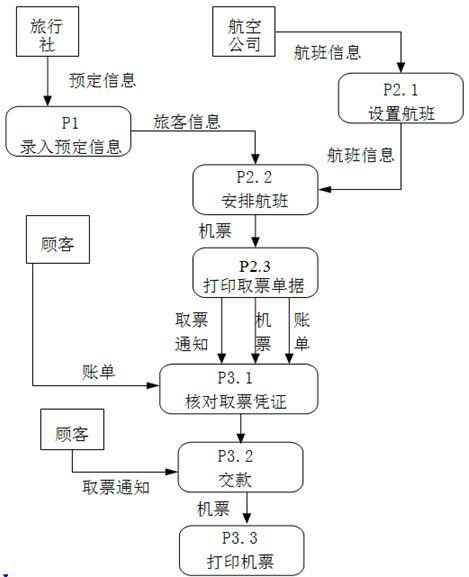


图5-12　机票预定系统数据流图（e）

②软件结构图，如图5-13所示。

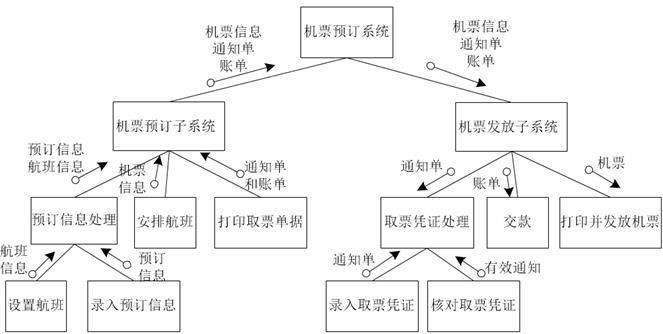


图5-13　机票预定系统软件结构图

1. 患者监护系统

①数据流图，如图5-14所示。

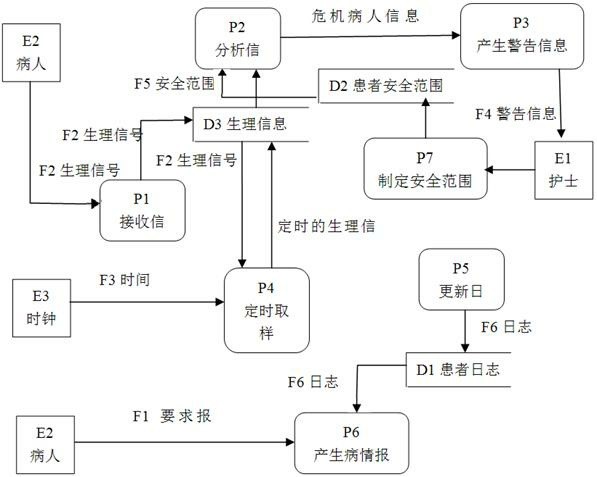


图5-14　患者监护系统数据流图

②软件结构图，如图5-15所示。

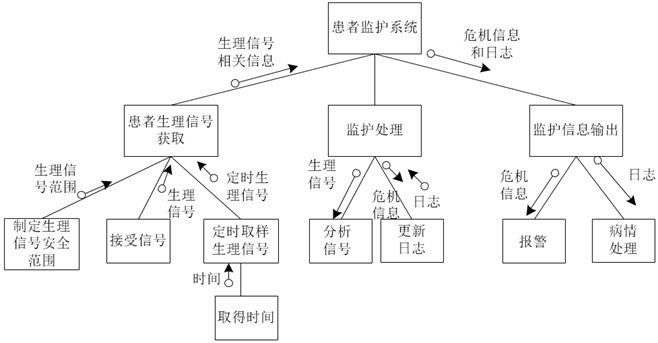


图5-15　患者监护系统软件结构图

4美国某大学共有200名教师，校方与教师工会刚刚签订一项协议。按照协议，所有年工资超过$26000（含$26000）的教师工资将保持不变，年工资少于$26000的教师将增加工资，所增加的工资数按下述方法计算：给每个由此教师所赡养的人（包括教师本人）每年补助$100，此外，教师有一年工龄每年再多补助$50，但是，增加后的年工资总额不能多于$26000。

教师的工资档案储存在行政办公室的光盘上，档案中有目前的年工资、赡养的人数、雇用日期等信息。需要写一个程序计算并印出每名教师的原有工资和调整后的新工资。要求：

1. 搜索工资档案数据，找出年工资少于$26000的人，计算新工资，校核是否超过$26000，储存新工资，印出新旧工资对照表；
2. 把工资档案数据按工资从最低到最高的次序排序，当工资数额超过$26000时即停止排序，计算新工资，校核是否超过限额，储存新工资，印出结果。
3. 画出此系统的数据流图。
4. 写出需求说明。
5. 设计上述的工资调整程序（要求用HIPO图描绘设计结果），设计时分别采用下述两种算法，并比较这两种算法的优缺点：

（4）所画出的数据流图适用于哪种算法？

答：（1）数据流程图，如图5-16所示。

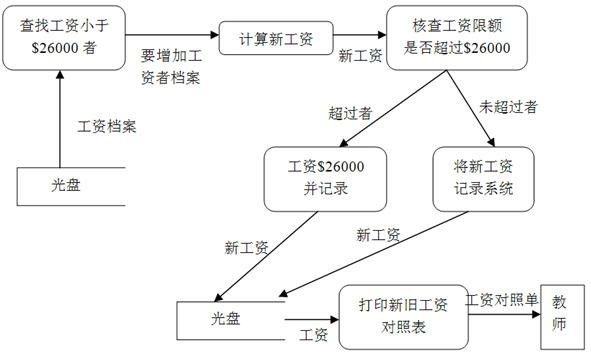


图5-16　工资系统数据流图

（2）需求说明

①任务概述

对于年工资未超过$26000的教师涨工资，即每赡养一人补助$100，每有一年工龄补助$50，涨工资的上限为$26000。

②数据描述

采用工资表、VF。

③功能需求 a．档案管理包括获得教师当前工资、获取赡养人数、获取工龄、更新教师信息。

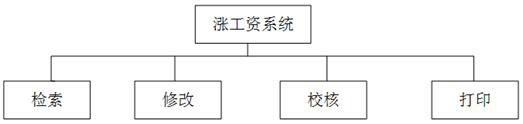
b．数据计算包括计算工资、根据限额确定工资。

c．查询管理包括排序、查询。

d．打印报表

打印工资对照单。

(3)HIPO图，如图5-17



所画的流程图适合（a）种算法。

5下面将给出两个人玩的扑克牌游戏的一种玩法，试设计一个模拟程序，它的基本功能是：

1. 发两手牌（利用随机数产生器）。
2. 确定赢者和赢牌的类型。
3. 模拟N次游戏，计算每种类型牌赢或平局的概率。

要求用HIPO图描绘设计结果并且画出高层控制流程图。

扑克牌游戏规则如下：

（1）有两个人玩，分别称为A和B。

（2）一副扑克牌有52张牌，4种花色（方块、梅花、红桃和黑桃），每种花色的牌的点数按升序排列有2，3，4，…，10，J，Q，K，A等13 种。

1. 给每个人发3张牌，牌面向上（即，亮牌），赢者立即可以确定。
2. 最高等级的一手牌称为同花，即3张牌均为同一种花色，最大的同花牌是同一种花色的Q、K、A。
3. 第二等级的牌称为顺子，即点数连续的3张牌，最大的顺子是花色不同的Q、K、A。
4. 第三等级的牌是同点，即点数相同的3张牌，最大的同点是A、A、

A。

1. 第四等级的牌是对子，即3张牌中有两张点数相同，最大的对子是

A、A、K。

1. 第五等级的牌是杂牌，即除去上列4等之外的任何一手牌，最大的杂牌是不同花色的A、K、J。
2. 若两人的牌类型不同，则等级高者胜；若等级相同，则点数高者胜；若点数也相同，则为平局。

答：（1）程序的HIPO图如图5-19所示。

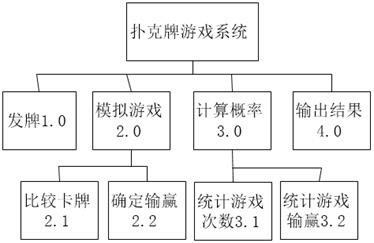


图5-19　系统HIPO图

（2）高层控制流程图，如图5-20所示。

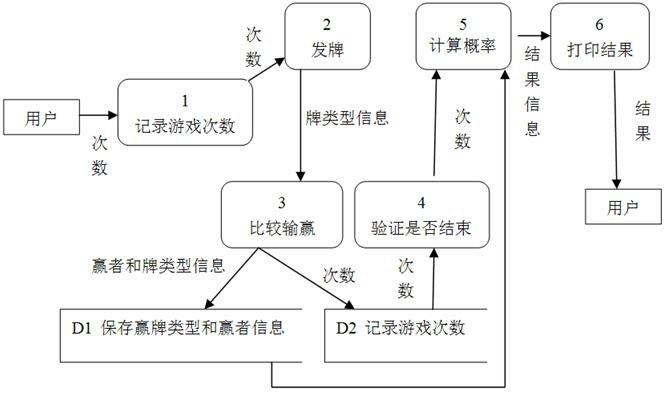


图5-20　高层控制流程图

# 第9章　面向对象方法学引论

## 9.1　复习笔记

一、面向对象方法学概述

1要点

1. 定义

面向对象方法是一种以数据或信息为主线，把数据和处理相结合的方法。

1. 要点

①对象；

②类；

③继承性；

④封装性。

【注意】考生需理解四个要点各自的含义。

2面向对象方法学的优点

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。

二、面向对象的概念

1对象

1. 对象的定义

①面向对象程序设计的角度：对象是具有相同状态的一组操作的集合。

②结构化的角度：对象是封装了数据结构及可以施加在这些数据结构上的操作的封装体。

1. 对象的特点

①以数据为中心；

②对象是主动的；

③实现了数据封装；

④具有并行性；

⑤模块独立性好。

2其他概念

1. 类

类是对具有相同数据和相同操作的一组相似对象的定义。

1. 实例

实例就是由某个特定的类所描述的一个具体的对象。

【注意】类是对象的抽象，对象是类的实例。

1. 消息

①定义：消息就是要求某个对象执行在定义它的那个类中所定义的某个操作的规格说明。

②组成：一个消息由接收消息的对象、消息选择符、零个或多个变元组成。

1. 方法

方法就是对象所能执行的操作，也就是类中所定义的服务。

1. 属性

属性就是类中所定义的数据，它是对客观世界实体所具有的性质的抽象。

1. 封装

封装是把数据和实现操作的代码集中起来放在对象内部。

1. 继承

①定义继承是子类自动地共享基类中定义的数据和方法的机制。

②特点 a．继承具有传递性； b．低层的性质将屏蔽高层的同名性质。

③分类 a．单继承：一个类只允许有一个父类，即类等级为树形结构。

b．多重继承：允许一个类有多个父类。

1. 多态性

①定义

多态性指在类等级的不同层次中可以共享一个方法的名字，不同层次中的每个类各自按自己的需要来实现这个行为。

【注意】在C++中，多态性是通过虚函数来实现的。

②动态联编

在运行时刻根据接收消息的对象所属于的类，决定执行哪个特定版本的函数，这称为动态联编（滞后联编）。

③优点 a．增加了面向对象软件系统的灵活性，进一步减少了信息冗余。

b．显著提高了软件的可重用性和可扩充性。

1. 重载

①分类

a．函数重载

在同一作用域内的若干个参数特征不同的函数可以使用相同的函数名字。

【注意】常考点：重载函数的形式参数（指参数的个数或者类型或者顺序）必须不同，而与函数返回值类型无关。

b．运算符重载同一个运算符可以施加于不同类型的操作数上面。

②实现 a．函数重载是通过静态联编（先前联编）实现的。

b．运算符重载是在编译时根据被操作数的类型，决定使用该算符的哪种语义。

③优点进一步提高了面向对象系统的灵活性和可读性。

三、面向对象建模

1模型

1. 定义

模型是为了理解事物而对事物作出的一种抽象，是对事物的一种无歧义的书面描述。

1. 作用

①是一种思考工具，利用这种工具可以把知识规范地表示出来。

②可以帮助人们思考问题、定义术语、在选择术语时作出适当的假设，并且有助于保持定义和假设的一致性。

2模型分类

1. 对象模型：描述系统数据结构。
2. 动态模型：描述系统控制结构。
3. 功能模型：描述系统功能。

四、对象模型

1概念

1. 定义

它是对模拟客观世界实体的对象以及对象彼此间的关系的映射，描述了系统的静态结构。【注意】对象模型表示静态的、结构化的系统的数据性质。

1. 工具

使用UML提供的类图来建立对象模型。

2类图的基本符号

类图描述类及类与类之间的静态关系。类图是一种静态模型，它是创建其他UML图的基础。

1. 类

UML中类的图形符号为长方形，用两条横线把长方形分为上、中、下3 个区域，3个区域分别放类的名字、属性和服务，如图9-1所示。



图9-1　表示类的图形符号

【注意】类名应该是富于描述性的、简洁的而且无二义性的。

1. 属性

① UML描述属性的语法格式为：可见性　属性名：类型名＝初值{性质串}

②属性的可见性分为公有的（＋）、私有的（－）和保护的（#），没有默认的可见性。

（3）服务

①服务也就是操作，UML描述操作的语法格式为：可见性　操作名（参数表）：返回值类型{性质串}

②操作的可见性分为公有的（＋）、私有的（－）和保护的（#），没有默认的可见性。

③参数表中描述一个参数的语法为：参数名：类型名＝默认值

3表示关系的符号

类与类之间通常有关联、泛化（继承）、依赖和细化4种关系。

（1）关联

①定义关联表示两个类的对象之间存在某种语义上的联系。

②普通关联 a．定义

只要在类与类之间存在连接关系就可以用普通关联表示。

b．表示第一，普通关联的图示符号是连接两个类之间的直线。

第二，关联是双向的，可为关联起一个名字。在名字前面（或后面）加一个表示关联方向的黑三角。

第三，在表示关联的直线两端可以写上重数。

③限定关联 a．定义

限定关联用在一对多或多对多的关联关系中，可以把重数从一对多变成一对一，或从多对多简化成多对一。

b．表示在类图中把限定词放在关联关系末端的一个小方框内。

④关联类 a．定义

为了说明关联的性质，可能需要一些附加信息。关联类可以用来记录相关信息。

b．表示

关联中的每个连接与关联类的一个对象相联系。关联类通过一条虚线与关联连接。

（2）聚集

聚集（聚合）是关联的特例。表示类与类之间的关系是整体与部分的关系。

①共享聚集

a．定义

如果在聚集关系中处于部分方的对象可同时参与多个处于整体方对象的构成，则该聚集称为共享聚集。

b．表示在表示关联关系的直线末端紧挨着整体类的地方画一个空心菱形。

②组合聚集 a．定义

如果部分类完全隶属于整体类，部分与整体共存，整体不存在了部分也会随之消失，则该聚集称为组合聚集。

b．表示

在表示关联关系的直线末端紧挨着整体类的地方画一个实心菱形。

（3）泛化

①定义

UML中的泛化关系就是继承关系，它是通用元素和具体元素之间的一种分类关系。

②表示

在UML中，用一端为空心三角形的连线表示泛化关系，三角形的顶角紧挨着通用元素。

③分类泛化关系包括普通泛化和受限泛化。

④约束预定义的约束有4种：多重、不相交、完全和不完全。

⑤继承分类

分为多重继承、不相交继承、完全继承、不完全继承四类。

（4）依赖和细化

①依赖关系 a．定义

依赖关系描述两个模型元素之间的语义连接关系：其中一个模型元素是独立的，另一个模型元素不是独立的，它依赖于独立的模型元素，如果独立的模型元素改变了，将影响依赖于它的模型元素。

b．表示

在UML类图中用带箭头的虚线连接有依赖关系的两个类，箭头指向独立的类。

②细化关系 a．定义对同一个事物在不同抽象层次上描述时，这些描述之间具有细化关系。 b．表示细化的图示符号为由元素B指向元素A的一端为空心三角形的虚线。

1. 动态模型

1概念

动态模型表示瞬时的、行为化的系统的控制性质，它规定了对象模型中的对象的合法变化序列。

2建模

每个类的动态行为用一张状态图来描绘，各个类的状态图通过共享事件合并起来，从而构成系统的动态模型。

1. 功能模型

1概念

1. 定义

功能模型表示变化的系统的功能性质，它指明了系统应该做什么。

1. 组成

功能模型由一组数据流图组成。

2用例图

在UML中把用用例图建立起来的系统模型称为用例模型。

1. 定义

用例模型描述的是外部行为者所理解的系统功能。用例模型的建立是系统开发者和用户反复讨论的结果，它描述了开发者和用户对需求规格所达成的共识。

1. 组成和表示

①系统，用方框表示。

②用例，用椭圆表示。

③行为者，用线条人表示。

④关系，用连线表示。

1. 关系的分类

UML用例之间主要有扩展和使用两种关系，它们是泛化关系的两种不同形式。

①扩展关系

向一个用例中添加一些动作后构成了另一个用例，这两个用例之间的关系就是扩展关系，后者继承前者的一些行为，通常把后者称为扩展用例。

②使用关系一个用例使用另一个用例时，这两个用例之间就构成了使用关系。七、3种模型之间的关系

1针对每个类建立的动态模型，描述了类实例的生命周期或运行周期。

2状态转换驱使行为发生，这些行为在数据流图中被映射成处理，在用例图中被映射成用例，它们同时与类图中的服务相对应。

3功能模型中的处理对应于对象模型中的类所提供的服务。

4数据流图中的数据存储，以及数据的源点/终点，通常是对象模型中的对象。

5数据流图中的数据流，往往是对象模型中对象的属性值，也可能是整个对象。

6用例图中的行为者，可能是对象模型中的对象。

7功能模型中的处理可能产生动态模型中的事件。

8对象模型描述了数据流图中的数据流、数据存储以及数据源点/终点的结构。

## 9.2　课后习题详解

1什么是面向对象方法学？它有哪些优点？

答：（1）面向对象方法学是尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类解决问题的方法与过程，使描述问题的问题空间与实现解法的解空间在结构上尽可能一致的方法学科。面向对象方法学是同时使用对象、类、继承和消息的方法。

（2）面向对象方法学优点：

①与人类习惯的思维方法一致。

②面向对象软件稳定性好。

③面向对象软件可重用性好。

④较易开发大型软件产品。

⑤可维护性好。

2什么是“对象”？它与传统的数据有何异同？

答：（1）对象是封装了数据结构及可以施加在这些数据结构上的操作的封装体。

（2）传统数据结构忽略了数据和操作之间的内在联系，其解空间与问题空间并不一致；而对象是由描述内部状态表示静态属性的数据，以及可以对这些数据施加的操作（表示对象的动态行为）封装在一起所构成的统一体。对象之间通过传递消息互相联系，以模拟现实世界中不同事物彼此之间的联系。

3什么是“类”？

答：类是对具有相同数据结构和相同操作的一组相似对象的定义，即类是对具有相同属性和行为的一个或多个对象的描述。

4什么是“继承”？

答：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。在面向对象的软件技术中，继承是子类自动地共享基类中定义的数据和方法的机制。

5什么是模型？开发软件为何要建模？

答：（1）模型是为了理解事物而对事物作出的一种抽象，是对事物的一种无歧义的书面描述。模型由一组图示符号和组织这些符号的规则组成，利用它们来定义和描述问题域中的术语和概念。模型是一种思考工具，利用这种工具可以把知识规范地表示出来。

（2）由于建模忽略了事物的非本质属性，因此模型比原始事物更容易操作。对于那些因过分复杂而不能直接理解的系统，特别需要建立模型，模型通过把系统的重要部分分解成人的头脑一次能处理的若干个子部分，从而减少了系统的复杂程度。

6什么是对象模型？建立对象模型时主要使用哪些图形符号？这些符号的含义是什么？

答：（1）对象模型表示静态的、结构化的系统的数据性质。它是对模拟客观世界实体的对象以及对象彼此间的关系的映射，描述了系统的静态结构。

1. 通常使用UML提供的类图来建立对象模型。
2. 在UML中“类”的实际含义是一个类以及属于该类的对象。

7什么是动态模型？建立动态模型时主要使用哪些图形符号？这些符号的含义是什么？

答：（1）动态模型表示瞬时的、行为化的系统的控制性质，它规定了对象模型中对象的合法变化序列。

1. 在UML中，使用状态图和事件追踪图来建立动态模型。
2. 其符号的含义为：

①状态图表示需要考察的对象的动态行为。

②事件追踪图表示其运行规律和行为规则。

8什么是功能模型？建立功能模型时主要使用哪些图形符号？答：（1）功能模型表示软件系统的功能性质，它指明了系统应该“做什么”，因此更直接地反映了用户对目标系统的需求。

（2）在UML中主要使用用例图来建立功能模型，一幅用例图包括的模型元素有系统、行为者、用例以及用例之间的关系。

9试用面向对象观点分析、研究第2章中给出的订货系统的例子。在这个例子中有哪些类？试建立订货系统的对象模型。

答：订货系统中的类如图9-2所示（类之间的关系省略）：

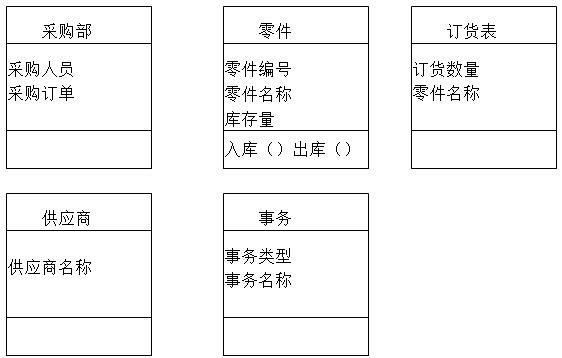


图9-2　订货系统存在的类

10 建立订货系统的用例模型。

答：仓库管理员通过放在仓库中的终端把零件入库或出库事务报告给订货系统，系统接收到事务信息之后应该处理事务；采购员需要使用订货系统提供的产生报表功能，以获取订货报表。用例图如图9-3所示。

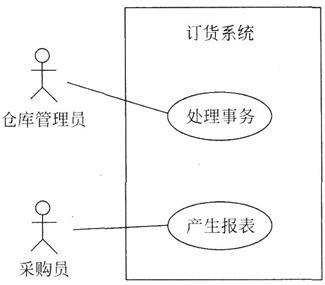


图9-3　订货系统用例图

# 第10章　面向对象分析

## 10.1　复习笔记

一、面向对象分析的基本过程

1概述

1. 定义

面向对象分析，就是抽取和整理用户需求并建立问题域精确模型的过程。

1. 过程

①分析陈述用户需求的文件；

②深入理解用户需求，抽象出目标系统的本质属性；

③用模型准确地表示。

1. 目的

全面深入地理解问题域和用户的真实需求，建立问题域的精确模型。

23个子模型与5个层次

1. 3个模型

面向对象建模得到的模型包含系统的三个要素，即

①静态结构（对象模型）；

②交互次序（动态模型）；

③数据变换（功能模型）。

【注意】3种模型中，对象模型是最基本、最重要、最核心的。

1. 5个层次

复杂问题的对象模型由5个层次组成，如图10-1所示。



图10-1　复杂问题的对象模型的5个层次

二、需求陈述

1内容

1. 阐明“做什么”而不是“怎样做”。
2. 描述用户的需求而不是提出解决问题的方法。
3. 指出哪些是系统必要的性质，哪些是任选的性质。
4. 避免对设计策略施加过多的约束，不描述系统的内部结构。2

书写要点

1. 做到语法正确，慎重选用名词、动词、形容词和同义词。
2. 必须把需求与实现策略区分开。
3. 需求陈述可简可繁。
4. 避免出现具有二义性的、不完整的、不一致的内容。

三、建立对象模型

1先建立对象模型的原因

1. 静态数据结构对应用细节依赖较少，比较容易确定。
2. 当用户的需求变化时，静态数据结构相对来说比较稳定。

2信息来源

需求陈述、应用领域的专业知识、客观世界的常识是建立对象模型时的主要信息来源。

3建立对象模型的步骤

1. 确定类与对象。

①找出候选的类与对象；

②筛选出正确的类与对象。

1. 确定关联。

【注意】在分析确定关联的过程中，不必花过多的精力去区分关联和聚集，聚集是一种特殊的关联。

1. 划分主题。
2. 确定属性。
3. 识别继承关系。
4. 反复修改。

四、建立动态模型

1概念

1. 适用性

①对于仅存储静态数据的系统来说，动态模型并没有什么意义。

②在开发交互式系统时，动态模型起着很重要的作用。

③应用系统的主要工作是收集输入信息时，建立正确的动态模型是至关重要的。

1. 步骤

①编写典型交互行为的脚本。 ②从脚本中提取出事件，确定触发每个事件的动作对象以及接受事件的目标对象。

③排列事件发生的次序，确定每个对象的状态及状态间的转换关系，用状态图描绘。

④比较各个对象的状态图，确保事件之间的匹配。

2编写脚本

1. 定义

脚本是指系统在某一执行期间内出现的一系列事件。

1. 目的

编写脚本的目的是保证不遗漏重要的交互步骤，有助于确保交互过程的正确性、清晰性。

3设想用户界面

1. 重要性

用户界面的好坏往往对用户是否喜欢、是否接受一个系统起很重要的作用。

1. 目的

确保能够完成全部必要的信息交换，而不会丢失重要的信息。【注意】当前阶段重要的是在界面下的信息交换方式，而不是用户界面的细节。

4画事件跟踪图

1. 一条竖线代表一个对象；
2. 每个事件用一条水平的箭头线表示；
3. 箭头方向从事件的发送对象指向接受对象；
4. 时间从上向下递增；
5. 用箭头线在垂直方向上的相对位置表示事件发生的先后，不表示事件间的时间差。

## 10.2　课后习题详解

1用面向对象方法分析研究本书习题2第2题中描述的储蓄系统，试建立它的对象模型、动态模型和功能模型。

答：（1）对象模型储蓄系统的对象模型，其类图如图10-2所示。

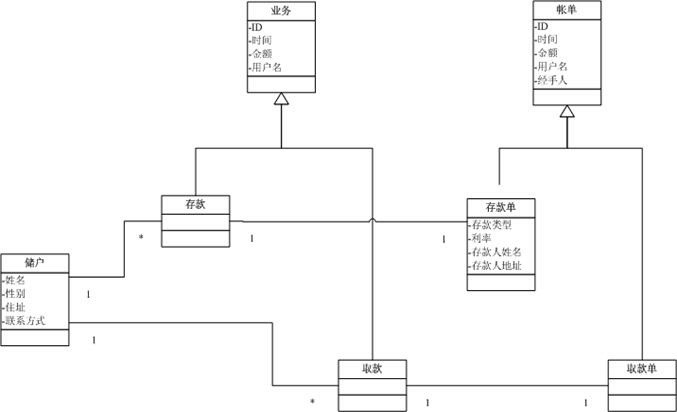


图10-2　储蓄系统的对象模型

（2）动态模型

①编写脚本

a．正常情况脚本

第一，储户有存款要求，填写存款单，包含储户个人信息，存款金额和存款类型；第二，业务员查收存款，审核存款与存款单存款金额吻合；第三，存款单生效；

第四，储户有取款要求，填写取款单，包含个人账号、密码（待定）和存款金额；

第五，业务员审核存款，验证储户身份，确定储户存款金额＞＝取款金额；第六，审核通过，取款单生效；第七，系统打印利息清单，业务员把本金和利息返回储户。

b．异常情况脚本

第一，储户有存款要求，填写存款单，包含储户个人信息，存款金额和存款类型；第二，业务员查收存款，审核存款与存款单存款金额不符；第三，存款单作废；

第四，储户有取款要求，填写取款单，包含个人账号、密码（待定）和存款金额；第五，业务员审核存款，验证储户身份，发现储户存款金额＜取款金额；第六，取款单作废。

②事件追踪图

储蓄系统的事件追踪图，如图10-3所示。状态图略。

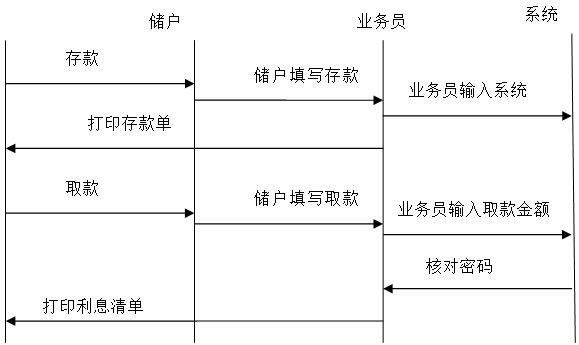


图10-3　储蓄系统的事件追踪图

（3）功能模型

储蓄系统的数据流图，如图10-4所示。

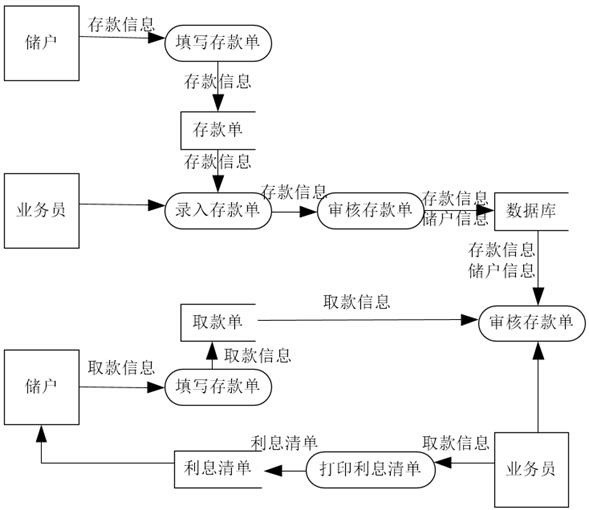


图10-4　储蓄系统的数据流图

2用面向对象方法分析研究本书习题2第3题中描述的机票预订系统，试建立它的对象模型、动态模型和功能模型。

答：（1）对象模型

该系统中有以下对象：旅客、旅行社、系统、机票，其对象模型中类图如图10-5所示。

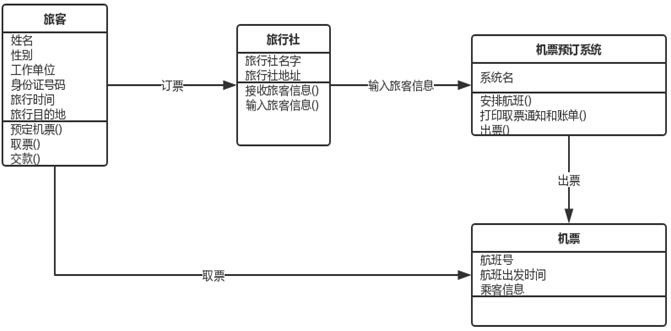


图10-5　机票预订系统类图

1. 动态模型

动态模型中事件跟踪图如图10-6所示。

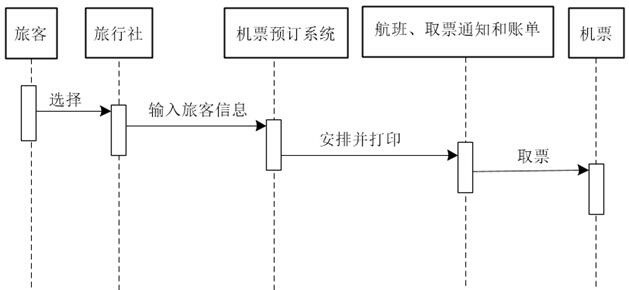


图10-6　机票预订系统事件跟踪图

动态模型中状态转换图如图10-7所示。



图10-7　机票预订系统的状态转换图

1. 功能模型

机票预订系统的数据流图如图10-8所示。

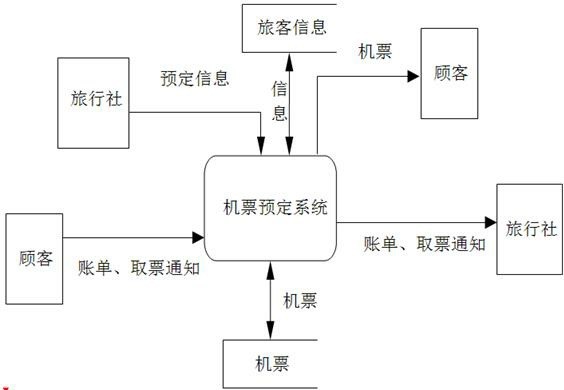


图10-8　机票预订系统的数据流图

3用面向对象方法分析研究本书习题2第4题中描述的患者监护系统，试建立它的对象模型、动态模型和功能模型。

答：（1）对象模型

该系统中有以下对象：病人、护士、系统，对象模型中类图如图10-9所示。

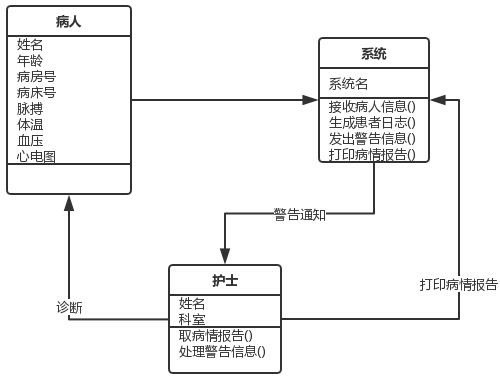


图10-9　患者监护系统类图

1. 动态模型

患者监护系统的动态模型中事件跟踪图如图10-10所示。

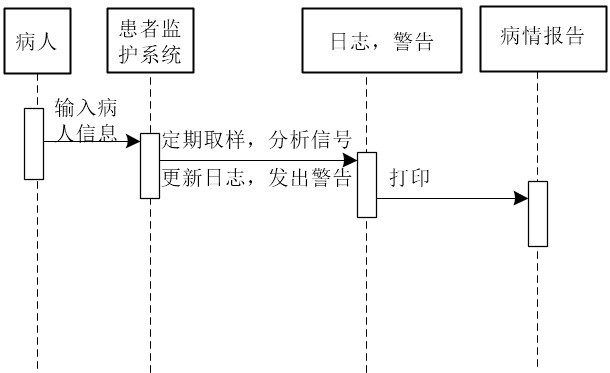


图10-10　患者监护系统事件跟踪图动态模型中状态转换图如图10-11所示。



图10-11　患者监护系统状态转换图

1. 功能模型

患者监护系统中，系统的功能级数据流图如图10-12所示。

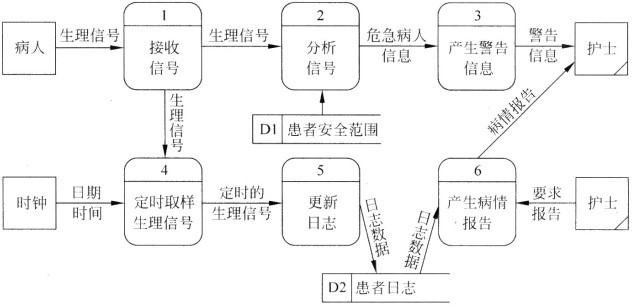


图10-12　患者监护系统功能级数据流图

4下面是自动售货机系统的需求陈述，试建立它的对象模型、动态模型和功能模型：

自动售货机系统是一种无人售货系统。售货时，顾客把硬币投入机器的投币口中，机器检查硬币的大小、重量、厚度及边缘类型。有效的硬币是一元币、五角币、一角币、五分币、二分币和一分币。其他货币都被认为是假币。机器拒绝接收假币，并将其从退币孔退出。当机器接收了有效的硬币之后，就把硬币送入硬币储藏器中。顾客支付的货币根据硬币的面值进行累加。

自动售货机装有货物分配器。每个货物分配器中包含零个或多个价格相同的货物。顾客通过选择货物分配器来选择货物。如果货物分配器中有货物，而且顾客支付的货币值不小于该货物的价格，货物将被分配到货物传送孔送给顾客，并将适当的零钱返回到退币孔。如果分配器是空的，则和顾客支付的货币值相等的硬币将被送回到退币孔。如果顾客支付的货币值少于所选择的分配器中货物的价格，机器将等待顾客投进更多的货币。如果顾客决定不买所选择的货物，他投放进的货币将从退币孔中退出。

答：（1）对象模型

①建立流程 a．候选类

自动售货机、顾客、硬币、投币口、假币、退币孔、硬币储藏器、货物分配器、货物、货物传送孔、零钱。

b．筛选优化类

第一，硬币、假币、零钱是货币的属性。属于系统内部“检验”功能处理的内容，它们与“货物”都属于物流，应该去掉。

第二，顾客决定买不买所选货物，需要一个按钮通知硬币储藏器，所以应增加一个“选择按钮”。

第三，进行投币检验需要一个检验器。

c．类集合

自动售货机、顾客、投币口、退币口、硬币储藏器、货物分配器、货物传送孔、硬币检查器。

②对象模型

自动售货机系统的对象模型，如图10-13所示。

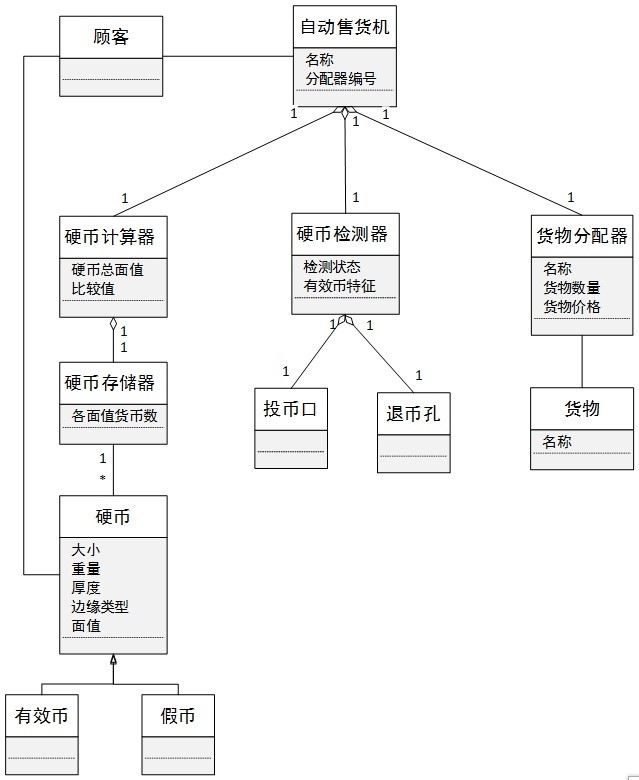


图10-13　自动售货机系统的对象模型

（2）动态模型

①编写脚本 a．正常情况脚本第一，顾客投入了5元真硬币。

第二，投币机检验，钱币合格，送往硬币储藏器，储藏器进行累加。

第三，顾客按“选择按钮”，指定一种3.5元的货物。

第四，按钮将信息送往货物分配器。

第五，货物分配器检查货物是否还有。还有，则将该货物的价格及所购数量计算后通知硬币储藏器需款总数。

第六，硬币储藏器将找零的硬币送往退币孔。

第七，货物分配器将应交付的货物送往货物传送口。

b．异常情况脚本1（投币不足）第一，顾客投入了5元真硬币。

第二，投币机检验，钱币合格，送往硬币储藏器，储藏器进行累加。

第三，顾客按“选择按钮”，指定一种6元的货物。

第四，按钮将信息送往货物分配器。

第五，货物分配器检查货物是否还有。还有，则将该货物的价格及所购数量计算后通知硬币储藏器需款总数。第六，硬币储藏器将对投款数与需款数进行对比后不发出任何信息，机器处于等待状态。

c．异常情况脚本2（投入假币）第一，顾客投入了5元硬币。

第二，投币机检验，其中有假币，投币机即刻将所投币送往退币孔。

d．异常情况脚本3（无货）第一，顾客投入了5元真硬币。

第二，投币机检验，钱币合格，送往硬币储藏器，储藏器进行累加。

第三，顾客按“选择按钮”，指定一种3.5元的货物。

第四，按钮将信息送往货物分配器。

第五，货物分配器检查货物是否还有。没有货物，则将无货信息送往硬币储藏器。

第六，硬币储藏器将5元硬币送往退币孔。

e．异常情况脚本4（顾客决定不买）第一，顾客投入了5元真硬币。

第二，投币机检验，钱币合格，送往硬币储藏器，储藏器进行累加。

第三，顾客按“选择按钮”，决定不买了。

第四，按钮将信息送往货物分配器。第五，硬币储藏器将5元硬币送往退币孔。

②建模

自动售货机系统的动态模型图如图10-14所示。事件跟踪图和状态图略。

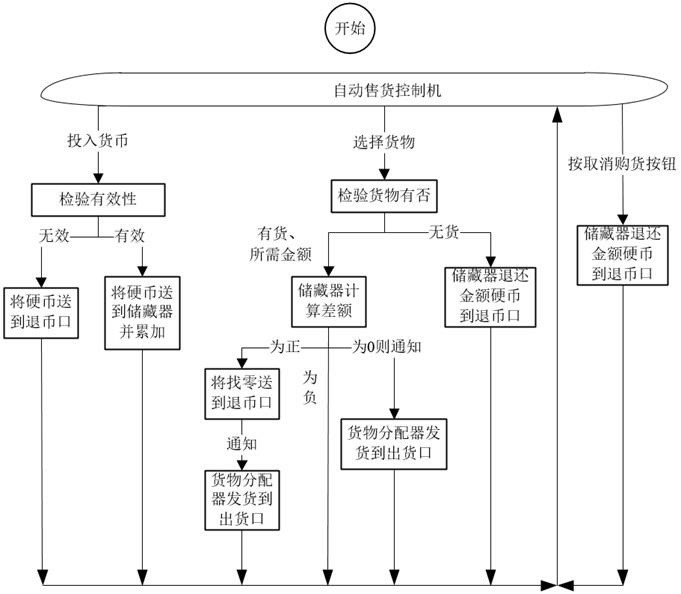


图10-14　自动售货机系统的动态模型图

（3）功能模型

自动售货机系统的程序流程图，如图10-15所示。功能级数据流图略。

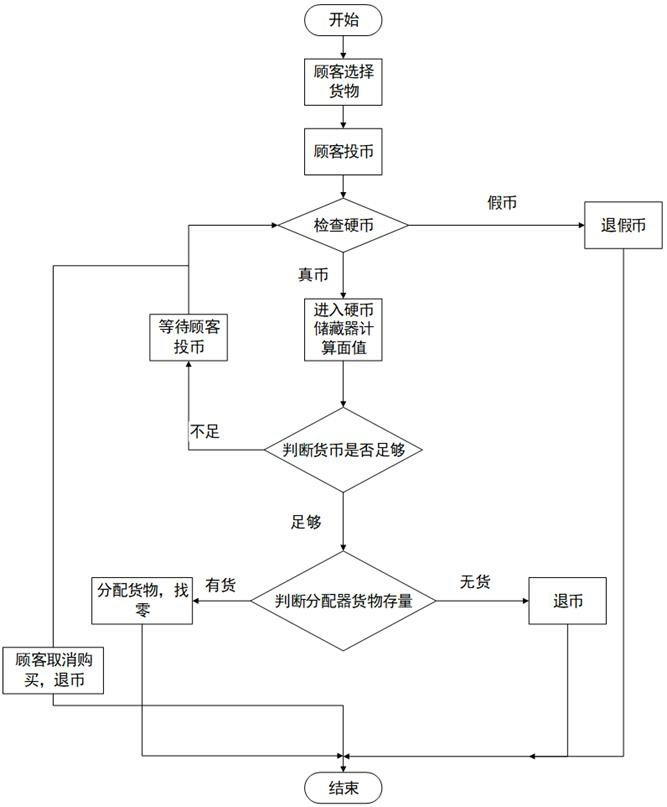


图10-15　自动售货机系统的程序流程图