第一章

一、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？

软件危机是指在计算机软件开发、使用与维护过程中遇到的一系列严重问题和难题。它包括两方面：如何开发软件，已满足对软件日益增长的需求；如何维护数量不断增长的已有软件。

软件危机的典型表现：

(1) 对软件开发成本和进度的估计常常很不准确。常常出现实际成本比估算成本高出一个数量级、实际进度比计划进度拖延几个月甚至几年的现象。而为了赶进度和节约成本所采取的一些权宜之计又往往损害了软件产品的质量。这些都降低了开发商的信誉，引起用户不满。

(2) 用户对已完成的软件不满意的现象时有发生。

(3) 软件产品的质量往往是靠不住的。　(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。文档资料不全或不合格，必将给软件开发和维护工作带来许多难以想象的困难和难以解决的问题。

(6) 软件成本、软件维护费在计算机系统总成本中所占比例逐年上升。

(7) 开发生产率提高的速度远跟不上计算机应用普及的需求。

软件危机出现的原因：

(1) 来自软件自身的特点：是逻辑部件，缺乏可见性；规模庞大、复杂，修改、维护困难。

(2) 软件开发与维护的方法不当：忽视需求分析；认为软件开发等于程序编写；轻视软件维护。

(3) 供求矛盾将是一个永恒的主题：面对日益增长的软件需求，人们显得力不从心。

二、假设自己是一家软件公司的总工程师，当把图 1.1 给手下的软件工程师们观看，告诉他们及时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楚它们是不现实的，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？

答：在软件开发的不同阶段进行修改付出的代价是很不相同的，在早期引入变动，涉及的面较少，因而代价也比较低；在开发的中期，软件配置的许多成分已经完成，引入一个变动要对所有已完成的配置成分都做相应的修改，

不仅工作量大，而且逻辑上也更复杂，因此付出的代价剧增；在软件“已经完成”是在引入变动，当然付出的代价更高。一个故障是代码错误造成的，有时这种错误是不可避免的，但要修改的成本是很小的，因为这不是

整体构架的错误。

三、什么是软件工程？它有哪些本质特征？怎样用软件工程消除软件危机？

1993 年 IEEE 的定义：软件工程是：① 把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程，也就是把工程应用于软件；② 研究①中提到的途径。

软件工程的本质特征：

(1) 软件工程关注于大型程序(软件系统)的构造　(2) 软件工程的中心课题是分解问题，控制复杂性　(3) 软件是经常变化的，开发过程中必须考虑软件将来可能的变化

(4) 开发软件的效率非常重要，因此，软件工程的一个重要课题就是，寻求开发与维护软件的更好更有效的方法和工具

(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) 在软件工程领域中是由具有一种文化背景的人替具有另一种文化背景的人(完成一些工作)消除软件危机的途径：

(1) 对计算机软件有一个正确的认识(软件≠程序)

(2) 必须充分认识到软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目

(3) 推广使用在实践中总结出来的开发软件的成功技术和方法

(4) 开发和使用更好的软件工具

四、简述结构化范型和面向对象范型的要点，并分析他们的优缺点。

1. 传统方法学：也称为生命周期方法学或结构化范型。 优点：把软件生命周期划分成基干个阶段，每个阶段的任务相对独立，而且比较简单，便于不同人员分工协作， 从而降低了整个软件开发过程的困难程度。缺点：当软件规模庞大时，或者对软件的需求是模糊的或会承受时 间而变化的时候，开发出的软件往往不成功；而且维护起来仍然很困难。

2. 面向对象方法学：优点：降低了软件产品的复杂性；提高了软件的可理解性；简化了软件的开发和维护工作； 促进了软件重用。

五、1-5 根据历史数据可以做出如下的假设 ：

对计算机存储容量的需求大致按下面公式描述的趋势逐年增加：M=4080e0.28(Y-1960)

存储器的价格按下面公式描述的趋势逐年下降：P1=0.3×0.72Y-1974(美分/位)

如果计算机字长为 16 位，则存储器价格下降的趋势为：P2=0.048×0.72Y-1974(美元/字)

在上列公式中 Y 代表年份，M 是存储容量(字数)，P1 和 P2 代表价格。

基于上述假设可以比较计算机硬件和软件成本的变化趋势。要求计算：
(1) 在 1985 年对计算机存储容量的需求估计是多少?如果字长为 16 位，这个存储器的价格是多少?
存储容量需求 M=4080e0.28（1985-1960）=4474263（字）
存储器价格 P=0.048*0.72（1985-1974）*4474263=5789 美元

(2) 假设在 1985 年一名程序员每天可开发出 10 条指令，程序员的平均工资是每月 4000 美元。如果一条指令为一个字长，计算使存储器装满程序所需用的成本。
需要工作量 4474263/200=22371（人/月）
指令成本 22371*4000=89484000 美元

(3) 假设在 1995 年存储器字长为 32 位，一名程序员每天可开发出 30 条指令，程序员的月平均工资为 6000 美元，重复(1)、(2)题。
需求估计 M=4080e0.28（1995-1960）=73577679 字
存储器价格 0.003*32*0.72（1995-1974）*73577679=7127 美元
工作量 73577679/600=122629(人/月）
成本 122629*6000=735776790 美元

六、什么是软件过程?它与软件工程方法学有何关系?
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。
软件工程方法学：通常把在软件生命周期全过程中使用的一整套技术方法的集合称为方法学，也称范型。
软件过程是软件工程方法学的 3 个重要组成部分之一。

七、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，说明每种模型的使用范围。
软件生命周期模型是跨越整个生存期的系统开发、运作和维护所实施的全部过程、活动和任务的结构框架。
瀑布模型　优点：它提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。虽然有不少缺陷但比在软件开发中随意的状态要好得多。
缺点：(1) 实际的项目大部分情况难以按照该模型给出的顺序进行，而且这种模型的迭代是间接的，这很容易由微小的变化而造成大的混乱。
(2) 经常情况下客户难以表达真正的需求，而这种模型却要求如此，这种模型是不欢迎具有二义性问题存在的。
(3) 客户要等到开发周期的晚期才能看到程序运行的测试版本，而在这时发现大的错误时，可能引起客户的惊慌，而后果也可能是灾难性的。
快速原型模型
优点：使用户能够感受到实际的系统，使开发者能够快速地构造出系统的框架。
缺点：产品的先天性不足，因为开发者常常需要做实现上的折中，可能采用不合适的操作系统或程序设计语言，以使原型能够尽快工作。
增量模型
优点：(1) 人员分配灵活，刚开始不用投入大量人力资源，当核心产品很受欢迎时，可增加人力实现下一个增量。
(2) 当配备的人员不能在设定的期限内完成产品时，它提供了一种先推出核心产品的途径，这样就可以先发布部分功能给客户，对客户起到镇静剂的作用。
缺点：(1) 至始至终开发者和客户纠缠在一起，直到完全版本出来。
(2) 适合于软件需求不明确、设计方案有一定风险的软件项目。
该模型具有一定的市场。
螺旋模型
优点：对于大型系统及软件的开发，这种模型是一个很好的方法。开发者和客户能够较好地对待和理解每一个演化级别上的风险。
缺点：(1) 需要相当的风险分析评估的专门技术，且成功依赖于这种技术。
(2) 很明显一个大的没有被发现的风险问题，将会导致问题的发生，可能导致演化　　的方法失去控制。
(3) 这种模型相对比较新，应用不广泛，其功效需要进一步的验证。
该模型适合于大型软件的开发

八、为什么说喷泉模型较好的体现了面向对象软件开发过程无缝和迭代的特性？
因为使用面向对象方法学开发软件时，各个阶段都使用统一的概念和表示符号，因此，整个开发过程都是吻合一致的，或者说是无缝连接的，这自然就很容易实现各个开发步骤的反复多次迭代，达到认识的逐步深化，而喷泉模型则很好的体现了面向对象软件开发过

程迭代和无缝的特性。

九、试讨论 Rational 统一过程的优缺点。

优点:提高了团队生产力，在迭代的开发过程、需求管理、基于组建的体系结构、可视化软件建模、验证软件质量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、模版和工具指导，并确保全体成员共享相同的知识基础。它建立了简洁和清晰的过程结构，为开发过程提供较大的通用性。

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，他没有支持多项目的开发结构，这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。

十．Rational 统一过程主要适用于何种项目？

大型的需求不断变化的复杂软件系统项目

十一. 说明敏捷过程的适用范围

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

十二. 说明微软过程的适用范围

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的软件过程模式

第二章

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）开发一个软件时，需要判断原定的系统模型和目标是否现实，系统完成后所能带来的效益是否大到值得投资开发这个系统的程度，如果做不到这些，那么花费在这些工程上的任何时间、人力、软硬件资源和经费，都是无谓的浪费。可行性研究的实质是要进行一次大大压缩简化了的系统分析和设计过程，就是在较高层次上以较抽象的方式进行的系统分析和设计的过程。可行性研究的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决。

（2）一般说来，至少应该从以下三个方面研究每种解法的可行性：

a.技术可行性。对要开发项目的功能、 性能和限制条件进行分析， 确定在现有的资源条件下，技术风险有多大，项目是否能实现，这些即为技术可行性研究的内容。这里的资源包括已有的或可以搞到的硬件、软件资源，现有技术人员的技术水平和已有的工作基础。

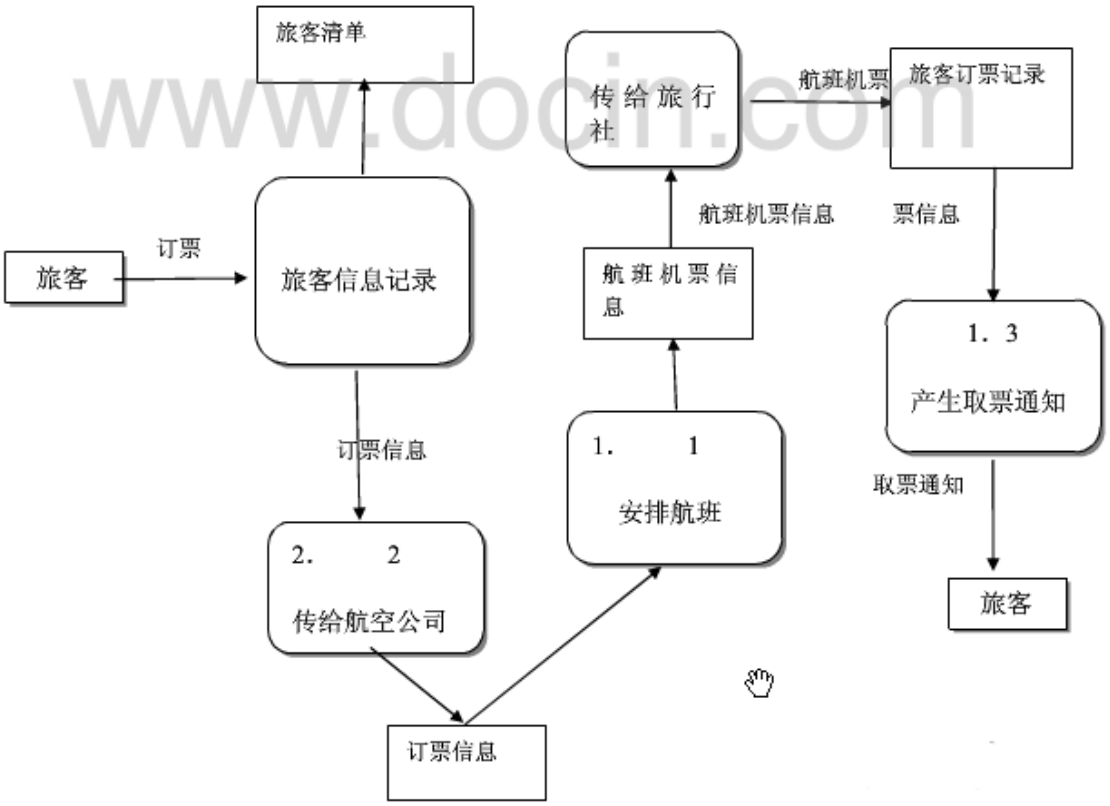
b.经济可行性。进行开发成本的估算以及了解取得效益的评估， 确定要开发的项目是否值得投资开发，这些即为经济可行性研究的内容对于大多数系统，一般衡量经济上是否合算，应考虑一个“底线”，经济可行性研究范围较广，包括成本—效益分析， 长期公司经营策略， 开发所需的成本和资源，潜在的市场前景。

c.操作可行性。有时还要研究社会可行性问题，研究要开发的项目是否存在任何侵犯、妨碍等责任问题。社会可行性所涉及的范围也比较广，它包括合同、责任、 侵权和其他一些技术人员常常不了解的陷阱等。

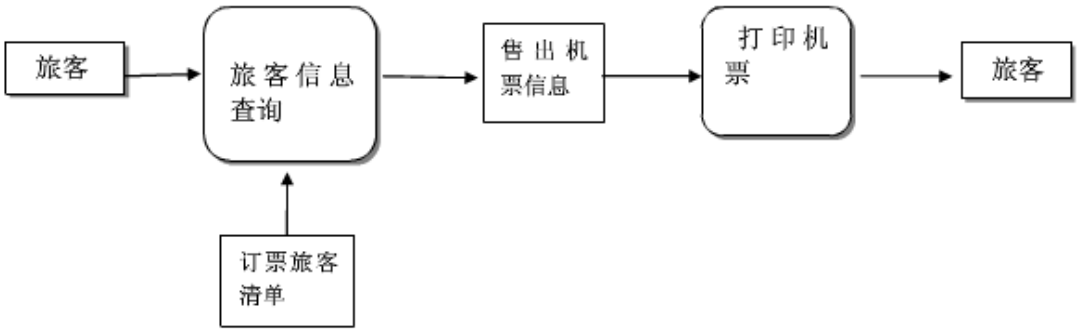
必要时还应该从法律、社会效益等更广泛的方面研究每种解法的可行性。

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键入系统，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。 请写出问题定义并分析此系统的可行性。

数据流程图:



取票图：



航空订票系统技术在目前是一个技术上成熟的系统，并且在航空公司内部准备采取有力措施保证资金和人员配置等。因此，分阶段开发“航空订票系统”的构想是可行的。为了使航空公司适应现代化市场竞争的需求，促进机票预订管理信息化，不断满足旅客预订机票的要求，争取更好的经济效益，可立即着手系统的开发与完善。

3、为方便旅客，某航空公司拟开发一个机票预定系统。旅行社把预定机票的旅客信息（姓名、性别、工作单位、身份证号码、旅行时间、旅行目的地等）输入进入该系统，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系统的可行性。

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

3> 建立新系统

① 经济可行性 ——→成本效益分析

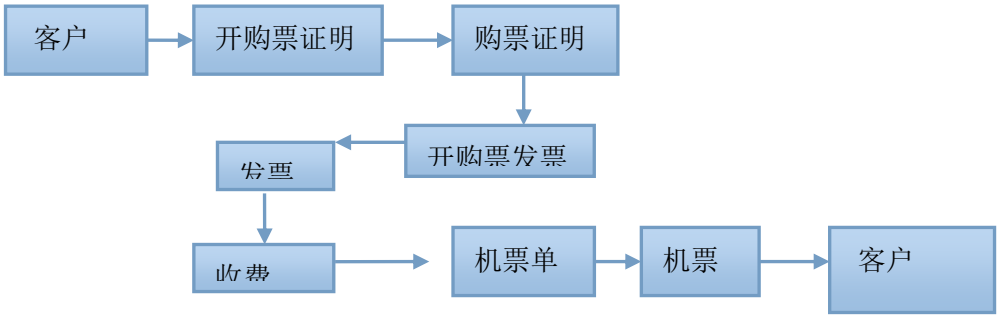
成本估算：打印机一台（2000 元）+开发费（3500 元）=5500 元

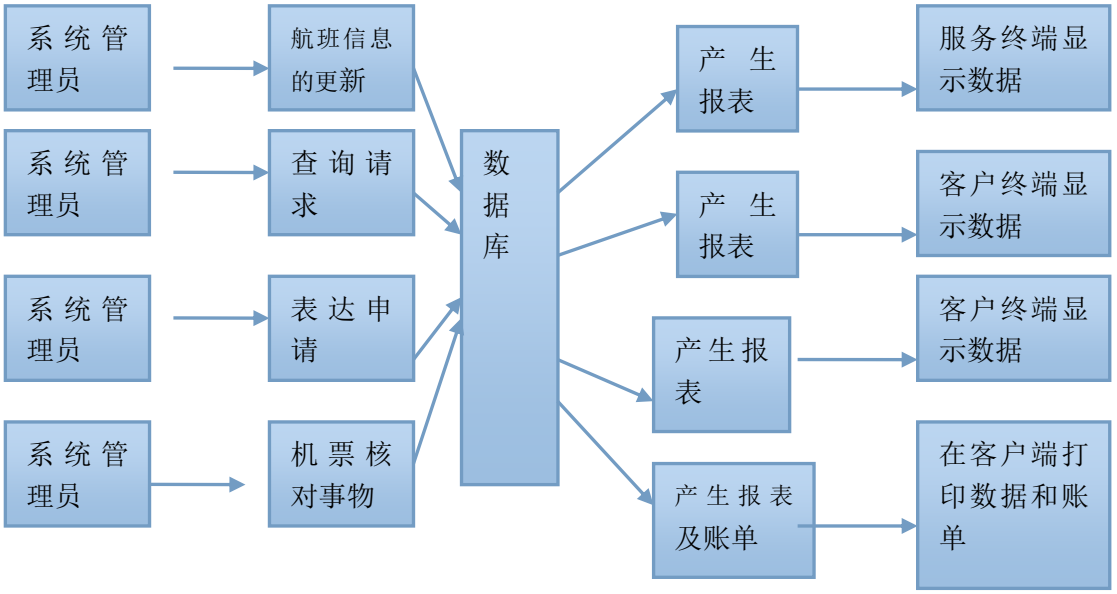
可承担

效益估算:该系统有很好的社会效益，提高了航空公司售票效率，方便了旅客，售票方便化，科学化

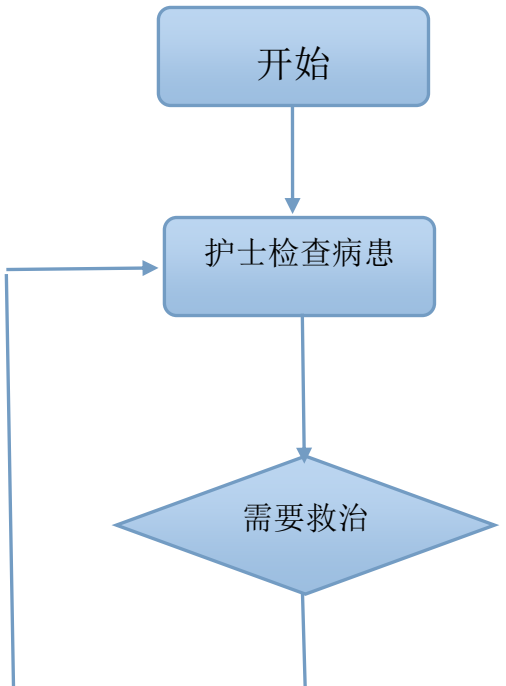
② 技术可行性

经过调查分析，得到目前航空公司机票预定系统流程图如下：

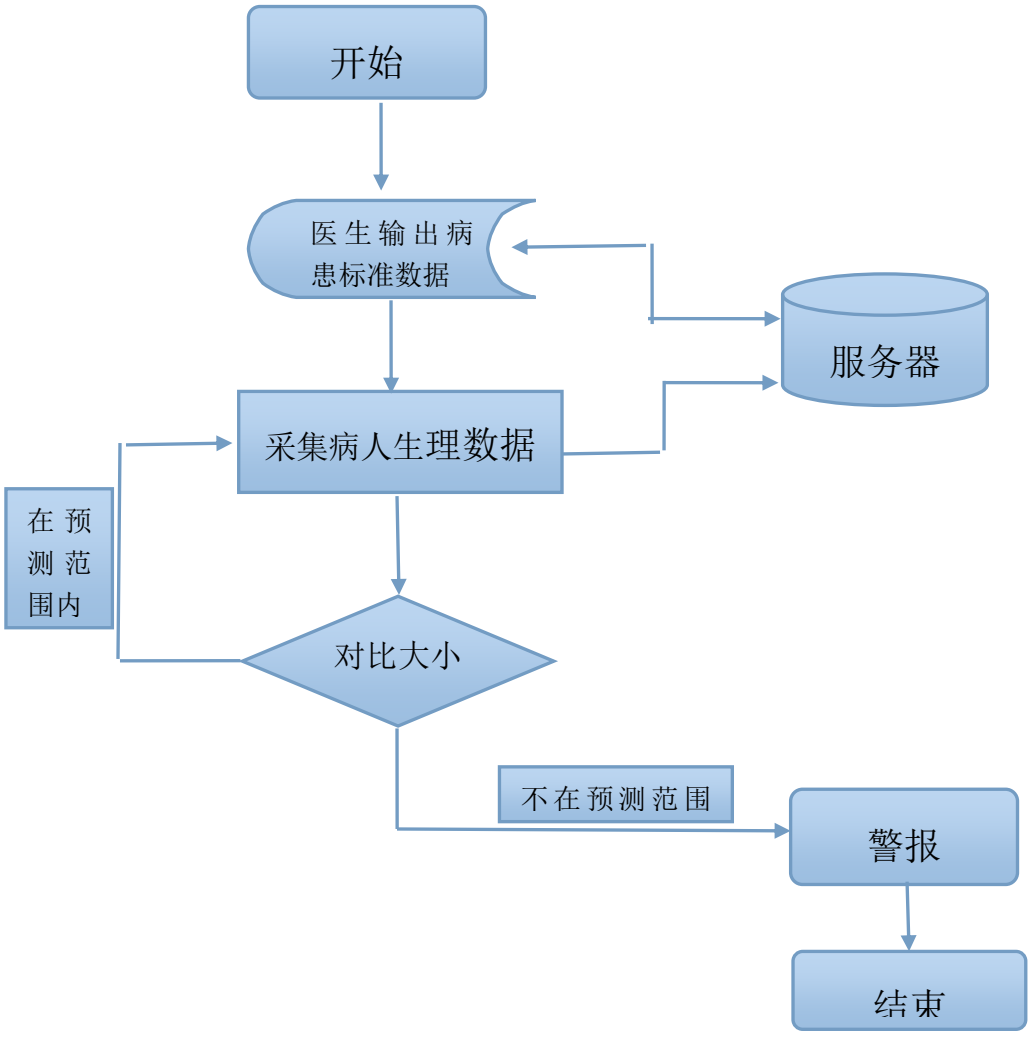




4.目前住院病人主要由护士护理，这样做不仅需要大量护士，而且由于不能随时观察危重病人的病情变化，还可能会延误抢救时机。某医院打算开发一个以计算机为中心的患者监护系统，试写出问题定义，并且分析开发这个系统的可行性可行性分析--原系统分析：



可行性分析-逻辑图：


开始
医生输出病患标准数据
服务器
采集病人生理数据
在预测范围内
对比大小
不在预测范围
警报
结束


医生
D1：标准数据

技术可行性;

虽然生理数据的采集需要涉及大量的专业精密仪器，软件工程师并不精通，但是可以在专业人士的帮助下完成。

经济可行性：

支出方面由医院方面承担，是否可行取决于医院方面是否能支付所需的费用。

操作可行性：

医生并不具有软件维护的能力，在数据库的维护上需要专业人士来进行，因为病患数目并不会太多，所以只需一个人或者几个人定期对数据库进行管理和维护就行了。

5 北京某高校可用的电话号码有以下几类：校内电话号码由 4 位数字组成，第一位数字不是 0.校外电话又分为本市电话和外地电话两类。拨校外电话需要先拨 0，若是本市电话则接着拨 8 位数字（第一位不是 0），若是外地电话则拨 3 位区码后再拨 8 位电话号码（第一位不是 0）。

答：

电话号码＝ ［校内电话号码｜校外电话号码］

校内电话号码＝ 非零数字＋ 3 位数字

校外电话号码＝ ［本市号码｜外地号码］

本市号码＝ 数字零＋ 8 位数字

外地号码＝ 数字零＋ 3 位数字＋ 8 位数字

非零数字＝ ［1 ｜2 ｜3 ｜4 ｜5 ｜6 ｜7 ｜8 ｜9］

数字零＝ 0

3 位数字＝ 3 {数字} 3

8 位数字＝ 非零数字＋ 7 位数字

7 位数字＝ 7 {数字} 7

数字＝ ［0 ｜1 ｜2 ｜3 ｜4 ｜5 ｜6 ｜7 ｜8 ｜9］

其中，[ ]意思是或，从校内电话号码或校外电话号码中选择一个；{ }表示重复，两边的数字表示重复次数的下限和上限；=意思是定义为；+意思是和，连接两个分量。

第三章

（1）.为什么要进行需求分析？通常对软件系统有哪些要求？

答： 1) 为了开发出真正满足用户需求的软件产品，首先必须知道用户的需求。对软件需求的深入理解是软件开发工作获得成功的前提条件，不论我们把设计和编码工作做得如何出色，不能真正满足用户需求的程序只会令用户失望，给开发者带来烦恼。

2)确定对系统的综合要求：1、功能需求；2、性能需求；3、可靠性和可用性需求；4、出错处理需求；5、接口需求；6、约束；7、逆向需求；8、将来可以提出的要求，分析系统的数据要求。

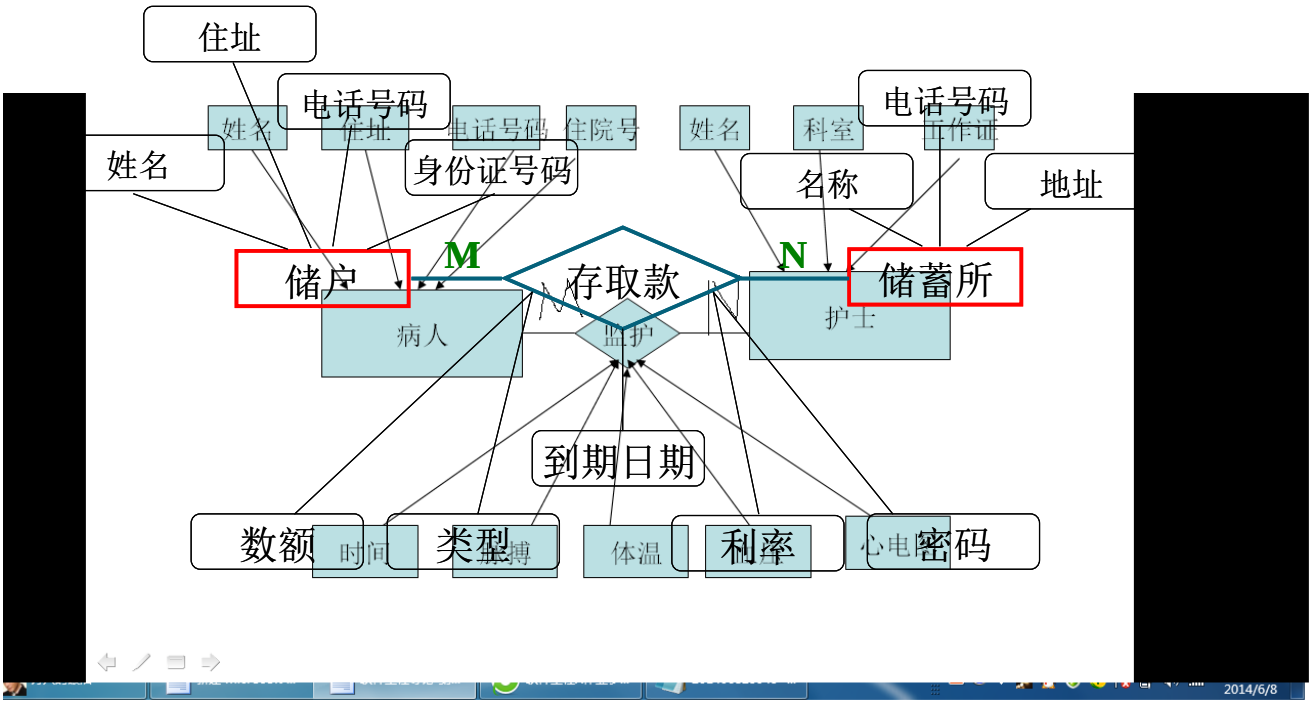
（2）怎样与用户有效地沟通以获取用户的真实需求？

答案 ：

与用户沟通获取需求的方法：访谈；面向数据流自顶向下求精；简易的应用规格说明技术；快速建立软件原型

3 银行计算机储蓄系统的工作过程大致如下：储户填写的存款单或取款单由业务员键入系统，如果是存款则系统记录存款人姓名、住址、身份证号码等存款信息，并打印出存款存单给储户；如果是取款且存款时留有密码，则系统首先核对储户密码，若密码正确或存款时未留密码，则系统计算利息并打印出利息清单给储户。

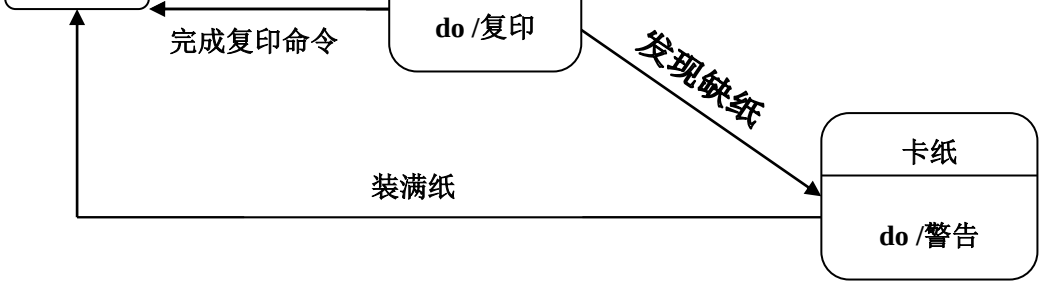
答案： 用 ER 图描绘系统中的数据对象。



（6 ）复印机的工作过程大致如下：未接到复印命令时处于闲置状态，一旦接到复印命令则进入复印状态，完成一个复印命令规定的工作后又回到闲置状态，等待下一个复印命令；如果执行复印命令时发现没纸，则进入缺纸状态，发出警告，等待装纸，装满纸后进入闲置状态，准备接收复印命令；如果复印时发生卡纸故障，则进入卡纸状态，发出警告等待维修人员来排除故障，故障排除后回到闲置状态。

请用状态转换图描绘复印机的行为。

答案： 从问题陈述可知，复印机的状态主要有“闲置”、“复印”、“缺纸”和“卡纸”。引起状态转换的事件主要是“复印命令”、“完成复印命令”、“发现缺纸”、“装满纸”、“发生卡纸故障”和“排除了卡纸故障”。



第四章

1，举例说明形式化说明技术和欠形式化方法的优缺点。

答：

	优点	缺点
形式化说明	1，简洁准确的描述物理现象，对象获动作的结果 2，可以在不同软件工程活动之间平滑的过度。 3，它提供了高层确认的手段	大多形式化的规格说明主要关注系统的功能和数据，而时序的问题，控制和行为等方面的需求却更难于表示
非形式化说明	难度低	可能存在矛盾，二义性，含糊性，不完整性级抽象层次混乱等问题

以一个简单的俄罗斯方块游戏系统规格说明为例，用自然语言描述如下：

游戏的每个状态对应一个游戏界面，开始状态下，但变量 cd_start=1 是进入正常游戏的状态，cd_start=2 时 进入读取游戏状态，cd_start=3 是进入得分榜界面查看。在得分榜界面按任意键返回开始界面，在读取游戏界 面，当游戏数据读取完成后进入正常游戏状态，正常游戏状态下，同时按下左键和右键进入储存游戏界面，数 据储存结束后返回正常游戏状态，在正常游戏状态下，如果变量 game_res=0，则游戏结束，进入游戏结束画面。 可见，用自然语言书写的系统规格说明书，罗嗦繁杂，并且可能存在矛盾，二义性，含糊性，不完整性及抽象 层次混乱等问题。

用有穷状态的描述如下：

状态机 J：{开始，正常游戏（游戏进行中），得分榜，读取游戏，储存游戏，游戏结束}

输入集 K：{按键 UP，按键 DOWN，按键 LEFT，按键 RIGHT，寄存器变量 cd_start ，寄存器变量 game_res}

转换函数 T：如图 4.1 所示

初始态 S：{开始}

终态集 F：{游戏结束}

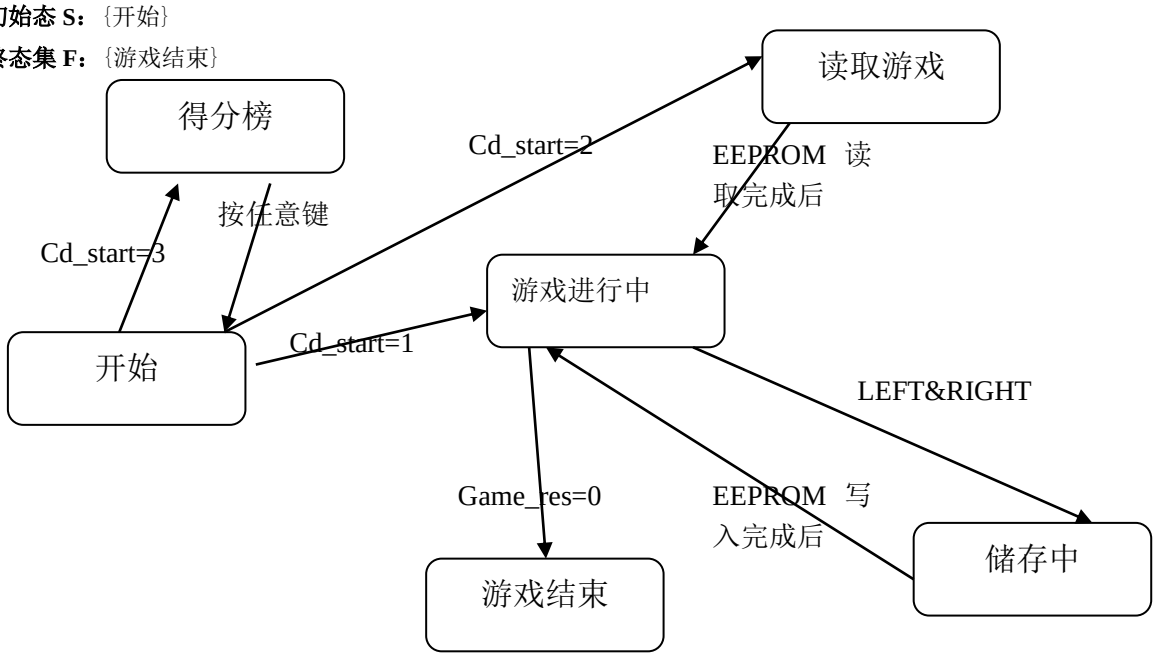


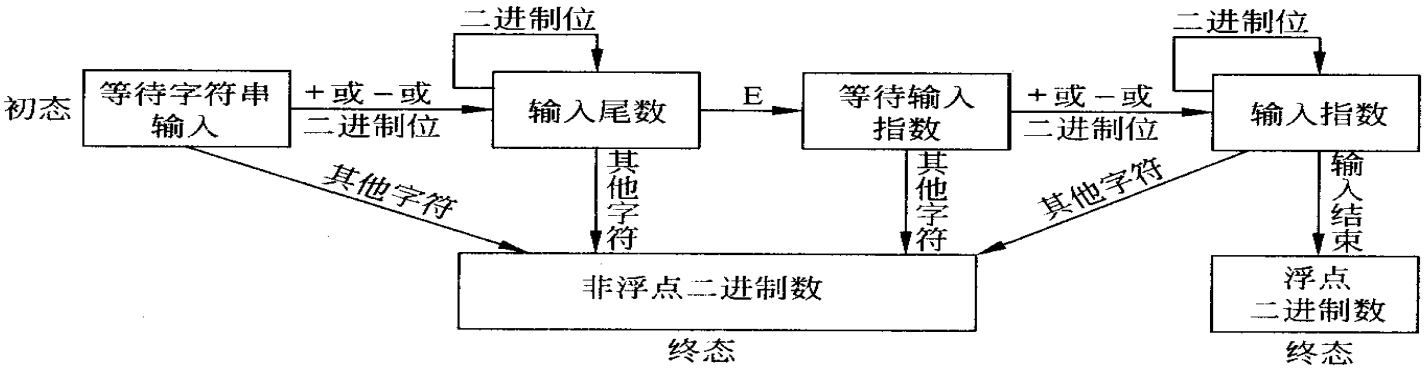
图 4.1 游戏状态转换图

可见用形式化语言描述，更加简洁，准确，无歧义。

2，在什么情况下应该使用形式化说明技术？使用形式化说明技术应遵守哪些规则？

答：

在用非形式化的方式描述时，存在矛盾，二义性，含糊性，不完整性级抽象层次混乱等问题时用形式化说明技术。



I，应该测试，测试再测试；

J，应该重用。

3，一个浮点二进制数的构成是：一个可选的符号(+或-)，后跟一个或多个二进制位，再跟上一个字符 E，再加上另一个可选符号(+或-)及一个或多个二进制位。例如，下列的字符串都是浮点二进制数：

110101E-101

-100111E11101

+1E0

更形式化地，浮点二进制数定义如下：

〈floating　point binary〉 ::=［〈sign〉］〈bitstring〉 E［〈sign〉］〈bitstring〉

〈sign〉 ::=+｜-

〈bitstring〉 ::=〈bit〉［〈bitstring〉］

〈bit〉 ::=0｜1

其中，

符号::=表示定义为；

符号［...］表示可选项；

符号 a｜b 表示 a 或 b。

假设有这样一个有穷状态机：以一串字符为输入，判断字符串中是否含有合法的浮点二进制数。试对这个有穷状态机进行规格说明。

4，考虑下述的自动化图书馆流通系统：每本书都有一个条形码，每个人都有一个带条形码的卡片。但一个借阅人想借一本书时，图书管理员扫描书上的条形码和借阅人卡片的条形码，然后在计算机终端上输入 C；当归还一本书时，图书管理员将再次扫描，并输入 R。图书管理员可以把一些书加到（+）图书集合中，也可以删除（—）它们。借阅人可以再终端上查找到某个作者所有的书（输入“A=”和作者名字），或具有指定标题的所有书籍（输入“T=”和标题），或属于特定主题范围内的所有图书（输入“S=”加主题范围）。最后，如果借阅人想借的书已被别人借走，图书管理员将给这本书设置一个预约，以便书归还时把书留给预约的借阅人（输入“H=”加书号）。

试用有穷状态机说明上述的图书流通系统

：图书馆流通系统的有穷状态机描述如下：

（一）图书状态的有穷状态机描述

态机 J：{书在图书馆 S1，书被借出 S2，书被预约 S3}

入集 K：{书上条形码，借阅卡条形码，终端输入各种命令}

换函数 T：如图 4.4.1 所示

始态 S：{书在图书馆 S1，书被借出 S2}

态集 F：{书被借出 S2，书被预约 S3}

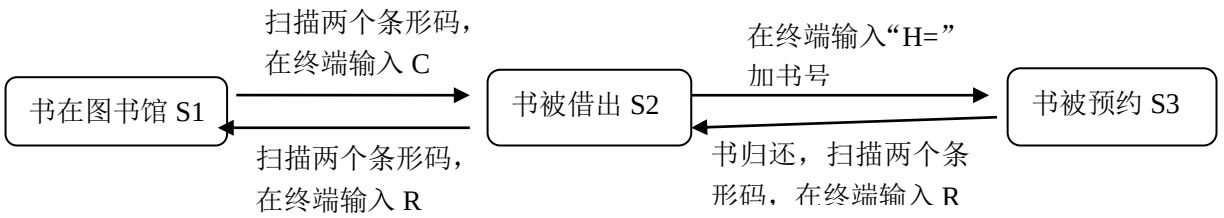


图 4.1.1

二)图书馆终端管理员模式的有穷状态机描述

态机 J：{管理员设置状态，书入库，书出库（删除），预约}

入集 K：{终端输入管理员命令，书的各种状态（S1，S2，S3}

换函数 T：如图 4.4.2 所示

始态 S：{管理员设置状态}

态集 F：{书入库，书出库（删除），预约，}

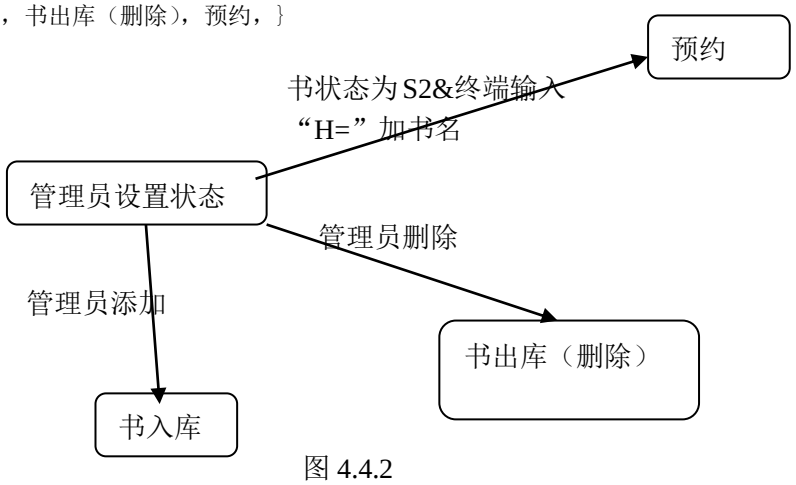


图 4.4.2

二)图书馆终端用户模式的有穷状态机描述

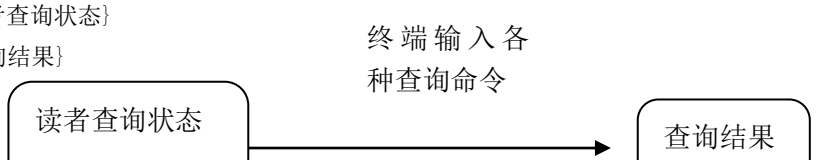
态机 J：{读者查询状态，查询结果}

入集 K：{终端输入用户查询命令，书的各种状态（S1，S2，S3）}

换函数 T：如图 4.4.3 所示

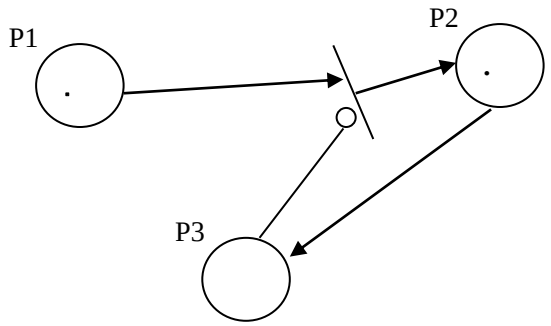
始态 S：{读者查询状态}

态集 F：{查询结果}



5，试用 Petri 网说明第四题所述图书馆中一本书的循环过程，在规格说明中应该包括操作 H、C 及 R。

答：其中 P1 表示书在图书馆 P2 表示书在读者手上，P3 书被预约



6，试用 Z 语言对第四题所描述图书馆图书流通系统做一个完整的规格说明。

答：（这题真的不会！）

BOOK_STATE

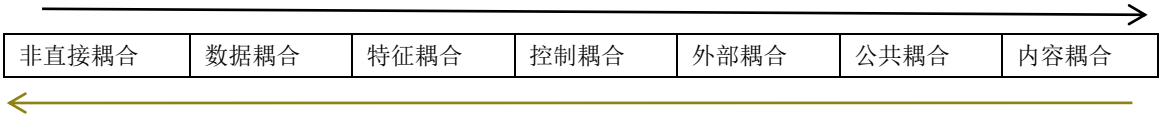
Book_in,Book_out,Book_reserve
Book_in∩Book_out∩Book_reserve=Φ Book_in∪Book_out∪Book_reserve=BOOK_STATE

第五章

5.1 为每种类型的模块耦合举一个具体的例子。

答：耦合式对一个软件结构内不同模块之间互联程度的度量。耦合强弱取决于接口的复杂度，进入或访问某一模块的点，以及通过接口的数据。一般模块之间的可能的连接方式有七种，构成耦合的七种类型，它们的关系为：

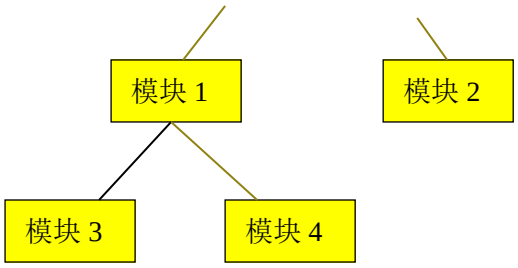
低　　耦合性　　高



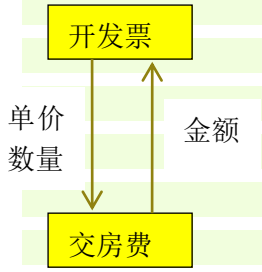
强　　独立性　　弱

下面举例说明以上耦合：

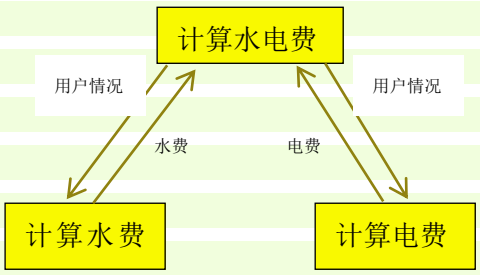
A． 非直接耦合：两个模块没有直接的关系（模块 1 和模块 2），独立性最强



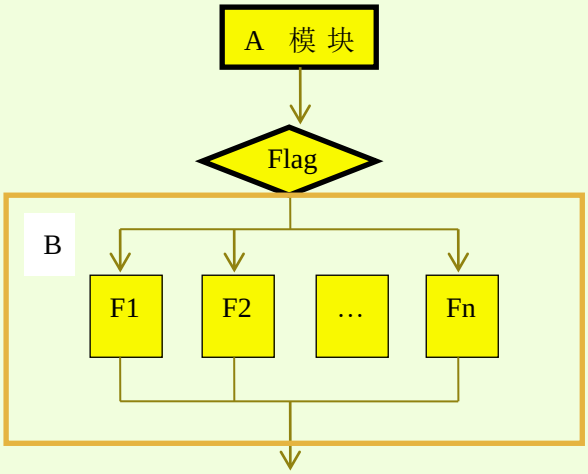
B.数据耦合：即一个模块访问另一个模块的时候，彼此之间是通过数据参数来交换输入、输出信息的，这种耦合为数据耦合。这种耦合较为松散，模块间独立性较强。



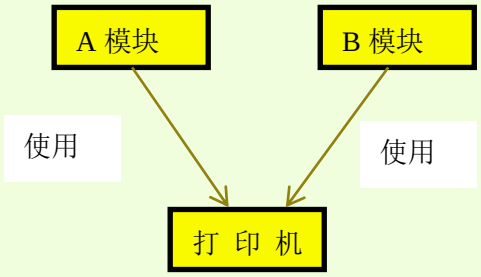
C.特征耦合：即一组模块通过参数传递记录信息，用户情况是个数据结构，图中模块都与此有关，“计算水费”和“计算电费”本没有关系，由于引用了此数据结构产生了依赖关系



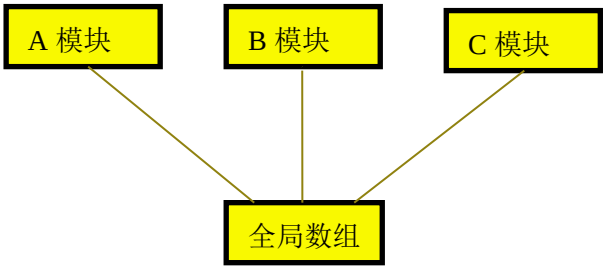
D.控制耦合：即如果一个模块通过传送开关、标志、名字等控制信息，明显地控制选择另一模块的功能，就是控制耦合



E.外部耦合：一组模块都访问同一全局简单变量而不是同一全局数据结构，而且不是通过参数表传递该全局变量的信息，则称之为外部耦合。



F.公共耦合：若一组模块都访问同一个公共数据环境，则它们之间的耦合就称为公共耦合。



G.内容耦合：如果出现以下情况之一，两个模块就发生了内容耦合

① 一个模块访问另一个模块的内部数据。

② 一个模块不通过正常入口儿转到另一个模块的内部

③ 两个模块有一部分程序代码重叠（只可能发生在汇编程序中）

④ 一个模块有多个入口（这意味着一个模块有几种功能）

```
Sub     AA(….)
…
…
Goto L
…
End sub
Sub    BB(..)
…
…
L: …
…
End    sub
```

5.2 为每种类型的模块内聚举一个例子

答：内聚标志着一个模块内各个元素之间彼此结合的紧密程度，它是信息隐藏和局部化概念的自然扩展。

低内聚：

A． 偶然内聚：如果一个模块完成一组任务，这些任务彼此间即使有关系，关系也是很松散的。这就叫做偶然内聚

偶然内聚的例子：在模块 T 中有 A,B,C 三条语句，至少从表面上看来这三条语句没什么联系，只是因为 D,E,F,G 中都有这三条语句，为了节省空间才把这三条语句作为一个模板放在一起。

B． 逻辑内聚：如果一个模块完成的任务在逻辑上属于相同或相似的一类(例如一个模块产生各种类型的全部输出)，称为逻辑内聚

逻辑内聚的例子：某一个模块将打印，年，月，日，具体打印什么，将由传入的控制标志所决定。

C． 时间内聚：一个模块包含的任务必须在同一段时间内执行（例如，模块完成各种初始化工作），称为时间内聚

时间内聚的例子：将多个变量的初始化放在同一个模块中实现。

中内聚：

A． 过程内聚：如果一个模块内的处理元素是相关的，而且必须以特定次序执行，称为过程内聚

过程内聚的例子：一个子程序，将开始读取学生的学号，然后是姓名，最后将读取分数，是由于特定的顺序而将这些操作组合在一起的

B． 通讯内聚：如果模块中所有的元素都使用同一个输入数据和（或）产生同一个输出数据，则称为通讯内聚

通讯内聚的例子：有一个子程序，它将打印实验报告，并且在完成后重新初始化传进来的实验数据。这个程序具有通讯内聚性。因为这两个操作由于使用同一个数据源联系在了一起。

高内聚：

A． 顺序内聚：如果一个模块内的处理元素和同一个功能密切相关，而且这些处理必须顺序执行（通常一个处理元素的输出数据作为下一个处理元素的输入数据），则称为顺序内聚。

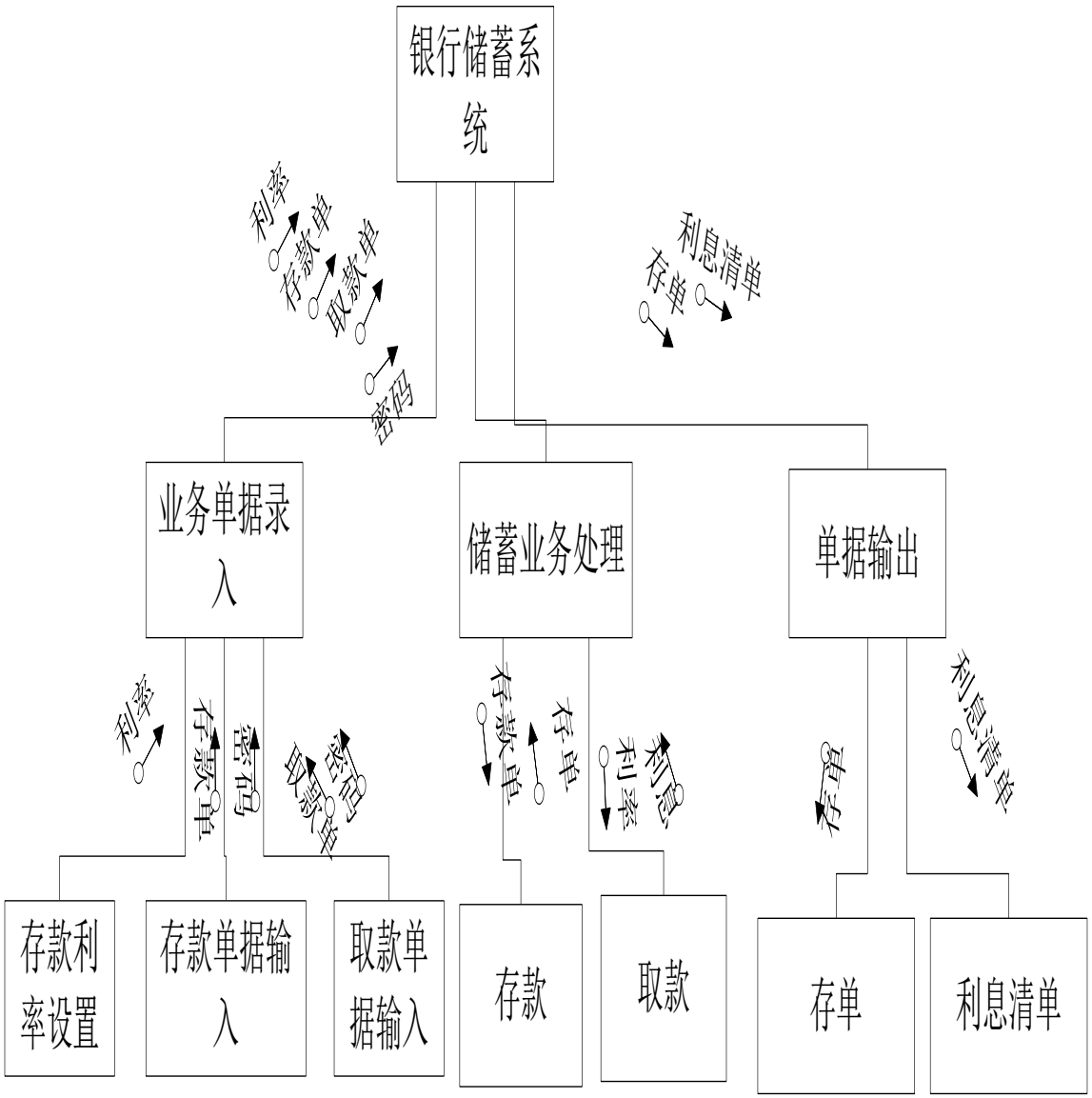
顺序内聚的例子：有一个子程序，通过给出的生日，先计算出年龄。再根据年龄算出退休的时间，则这个程序具有顺序内聚性。

B． 功能内聚：如果模块内所有的元素属于一个整体完成一个单一的功能，则成为功能内聚。

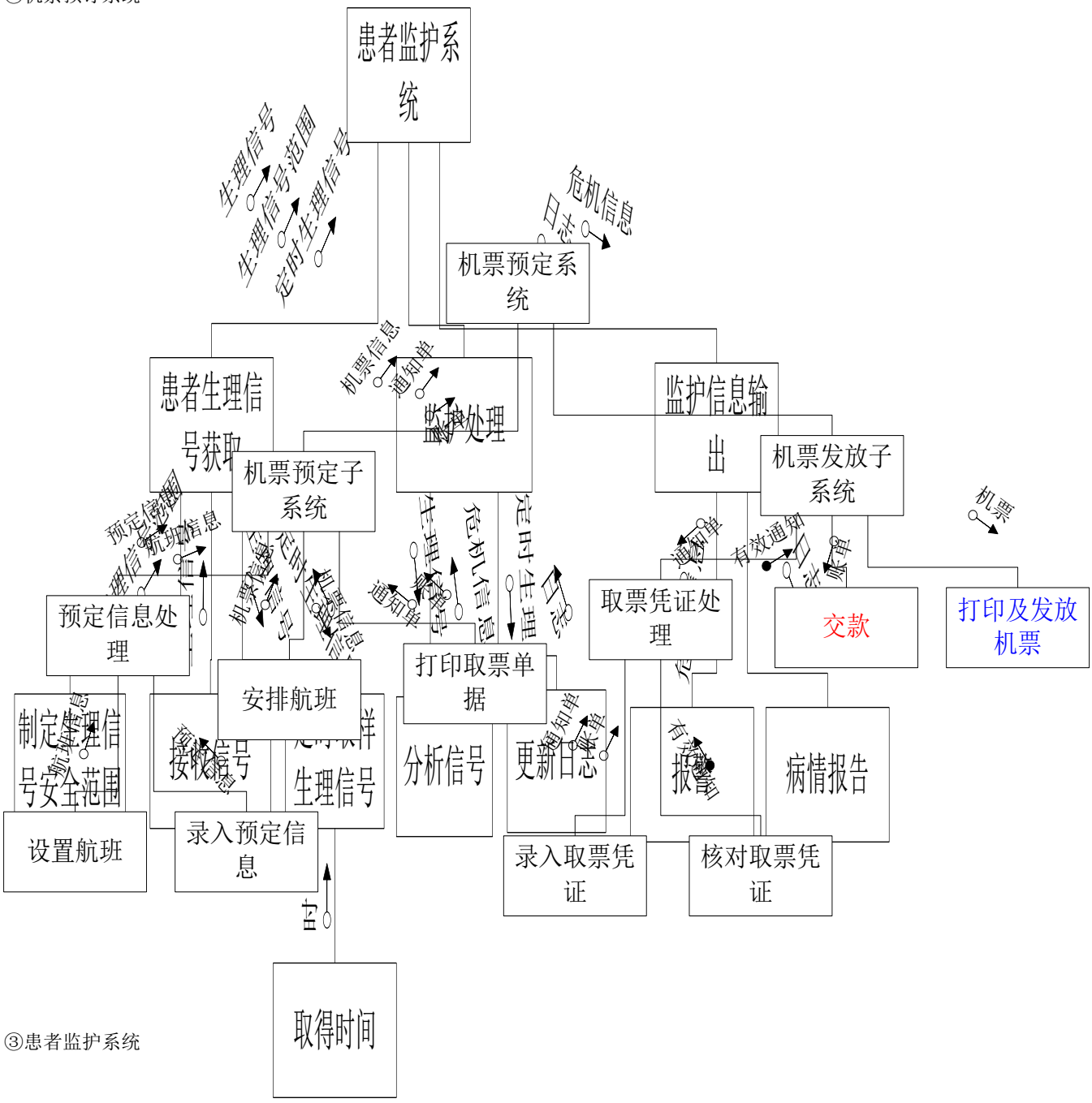
功能内聚的例子：一个程序中所有的操作都是为了算出一个人的年龄

5.3 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


银行储蓄系统
利率
存款单
取款单
密码
存单
利息清单
业务单据录入
储蓄业务处理
单据输出
利率
存款单
密码
取款单
存款
存单
利率
存单
利息清单
存款利率设置
存款单据输入
取款单据输入
存款
取款
存单
利息清单

②机票预订系统



③患者监护系统

5.4 美国某大学有 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，所增加工资数额按下述方法计算：给每位教师所赡养的人（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。

教师工资档案存储在行政办公室的磁带上，档案中有目前的年工资、赡养的人数、雇佣日期等信息。需要写一个程序计算并印出每名教师的原工资和调整后的新工资。

要求：（1）画出此系统的数据流图；（2）写出需求说明；

（3）设计上述的工资调整程序(要求用 HIPO 图描绘设计结果)，设计时分别采用两种算法，并比较两种算法的优缺点：

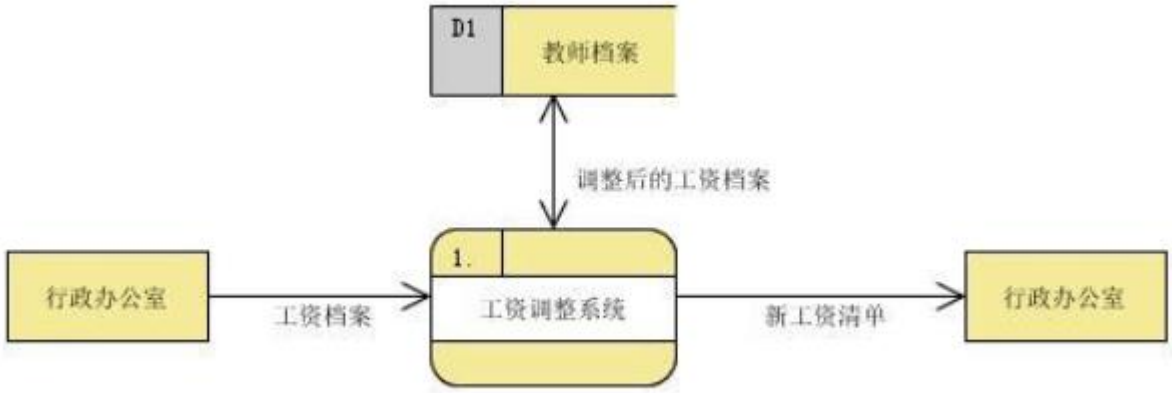
（a）搜索工资档案数据，找出年工资少于$26000 的人，计算新工资，校核是否超过$26000，存储新工资，印出新旧工资对照表；

（b）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

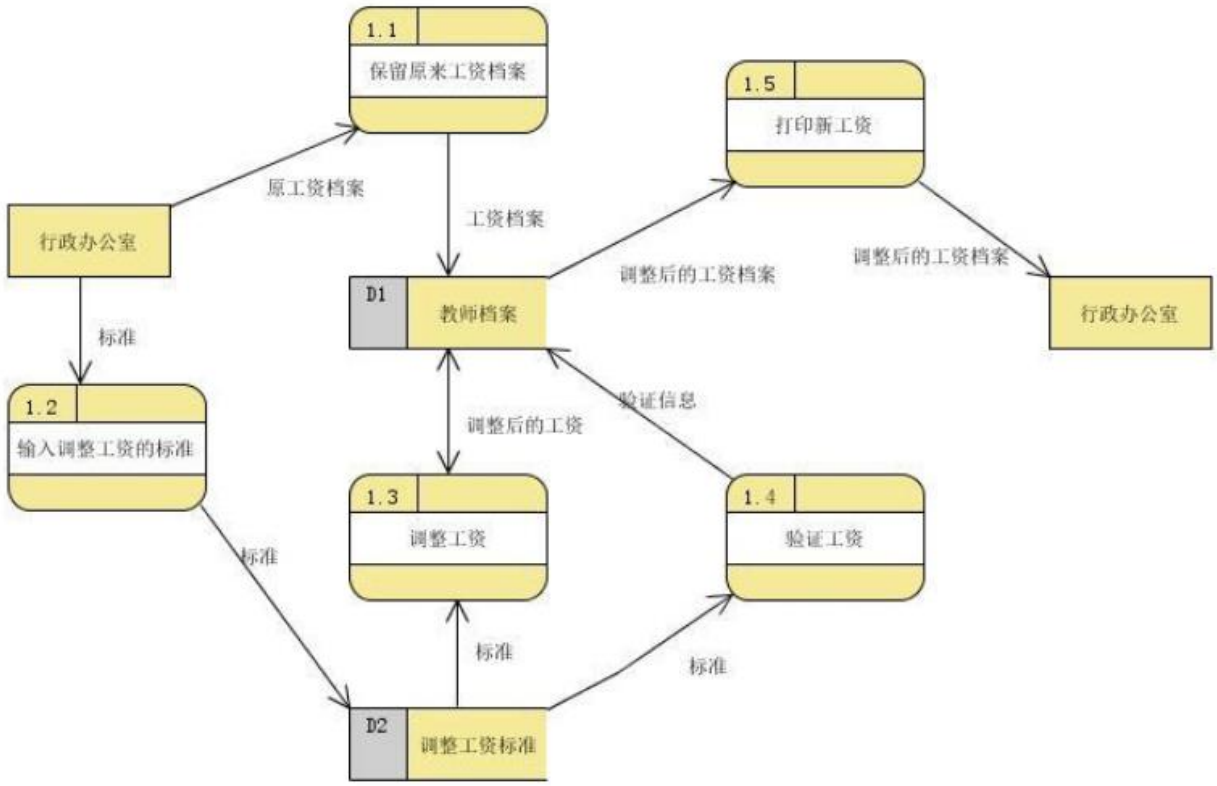
时即停止排序，计算新工资，校核是否超过限额，存储新工资，印出结果。

（4）你所画出的数据流图适应用那种算法？

解：（1）数据流图：



分解后：



（2）需求说明：

1. 功能需求：可以输入调资的标准，输入教师档案，经调资给出新的教师档案，

需要存储档案

2. 性能需求：软件的响应时间应小于0.5s，更新处理要快

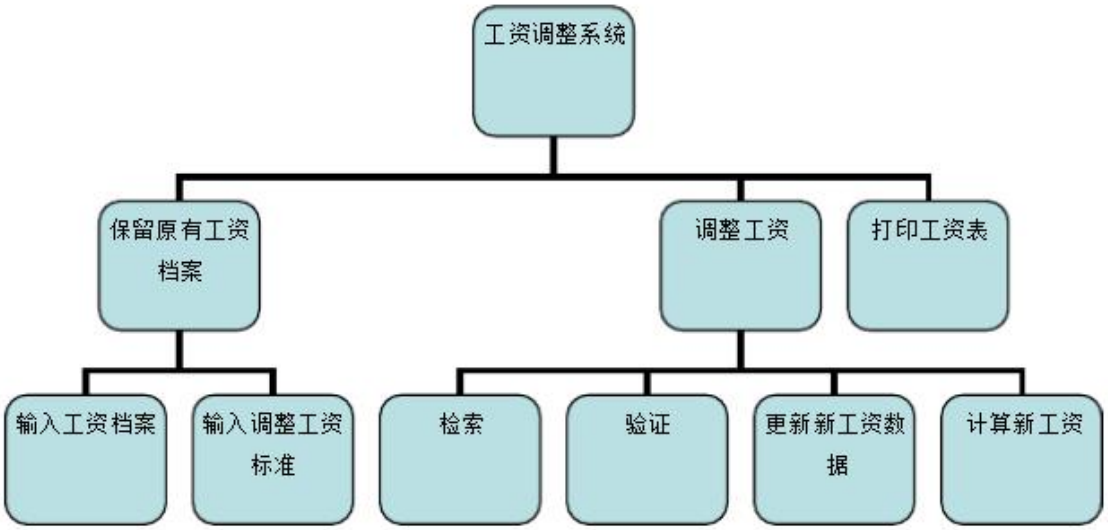
3. 灵活需求：当需求发生某些变化时，该软件应该能够适应这些变化

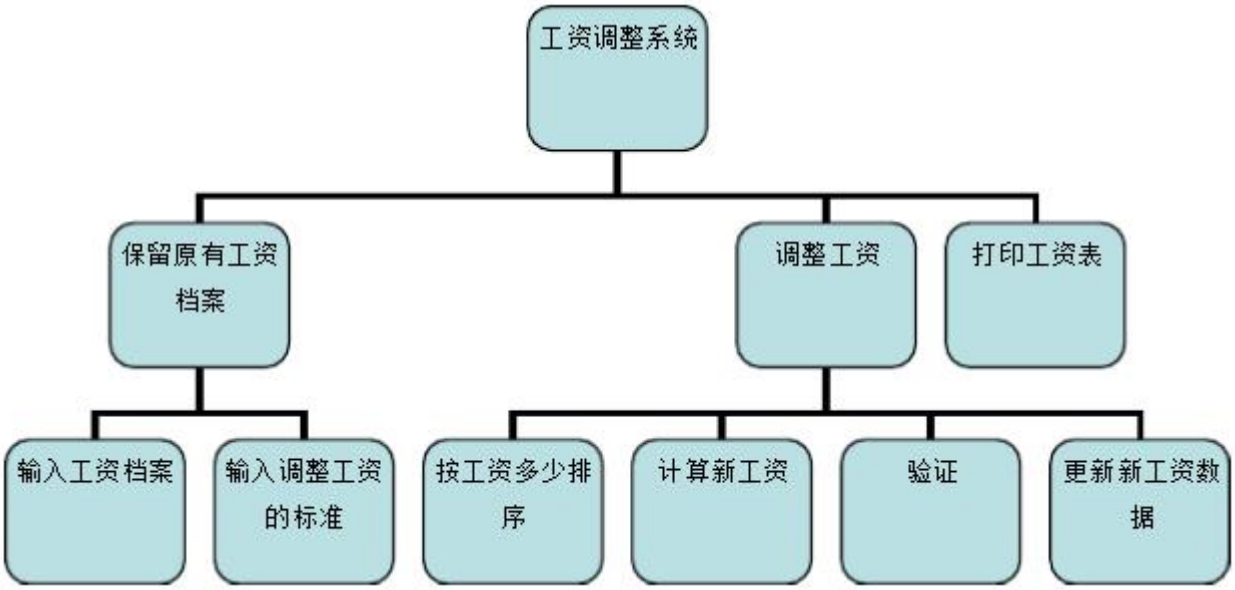
4. 故障处理要求：出现错误时，应给予警告或提示

（3）

A.

B.





（a）比较耗时，因为它要检索所有的档案，（b）从速度上看比较快，但是（b）

需要排序算法，比较复杂，（a）对于设计来讲比较简单 .

（4）画出的数据流图比较适合（A）的算法

5.5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）有两个人玩分别为 A 和 B。

（2）一副扑克牌有52张牌，4种花色（黑桃、红桃、梅花、方块），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。

（3）给每个人发三张牌，牌面向上，赢者立即可以确定。

（4）最高等级的一手牌成为同花，即3张牌均为同一种花色，最大的同花是同一种花色的 Q,K,A。

（6）第三等级的牌是同点，即点数相同的三张牌，最大的同点是 AAA。

（7）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。

（8）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。

（9）若两个人的牌类型不同，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。

程序：#include "stdio.h"

```
int rabl(int a,int b,int *r)
{
        int l,k,m,i,p;
        k=b-a+1;
        l=2;
        while(i<=l)
        {
                k=k+k+k+k+k;
                k=k%m;
                l=k/4+a;
                if(l<=b)  {p=l;i=i+1;}
        }
        *r=k;
```

```
          return(p);
}
int max(int T[10][10])
{
          int t=0;
          if(T[0][0]>T[1][0])
                    t=T[0][0];
          else t=T[1][0];
          if(t<T[2][0])
                    t=T[2][0];
          return t;
}
int E1(int T[10][10])
{
          if(T[0][1]==T[1][1]&&T[1][1]==T[2][1])
                    return 1;
          else  return 0;
}
int E2(int T[10][10])
{
          int q=0;
          if(((max(T[10][10])-1)==T[0][0]||(max(T[10][10])-1)==T[1][0]||(max(T[10][10])
                    -1)==T[2][0])&&((max(T[10][10])-2)==T[0][0]||(max(T[10][10])-2)==T[1][0]
||
                     (max(T[10][10])-2)==T[2][0])) // if(q=max(T[][10]))
                     return 1;
          else
                     return 0;
```

```
}
int E3(int T[10][10])
{
          if(T[0][0]==T[1][0]==T[2][0])
                    return 1;
          else return 0;
}
int E4(int T[10][10])
{
          if(T[0][0]==T[1][0]&&T[0][0]!=T[2][0])
                    return 1;
          else if(T[0][0]==T[2][0]&&T[0][0]!=T[1][0])
                    return 1;
          else if(T[1][0]==T[2][0]&&T[1][0]!=T[0][0])
                    return 1;
          else return 0;
}
void main()
{
          int times=0,e1=0,e2=0,e3=0,e4=0,e5=0;
          int A[10][10],B[10][10];
          int r1=2,r2=3;
          printf("请输入游戏的次数\n");
          scanf("%d",&times);
          for(int j=0;j<times;j++)
          {
                    for(int i=0;i<3;i++)
```

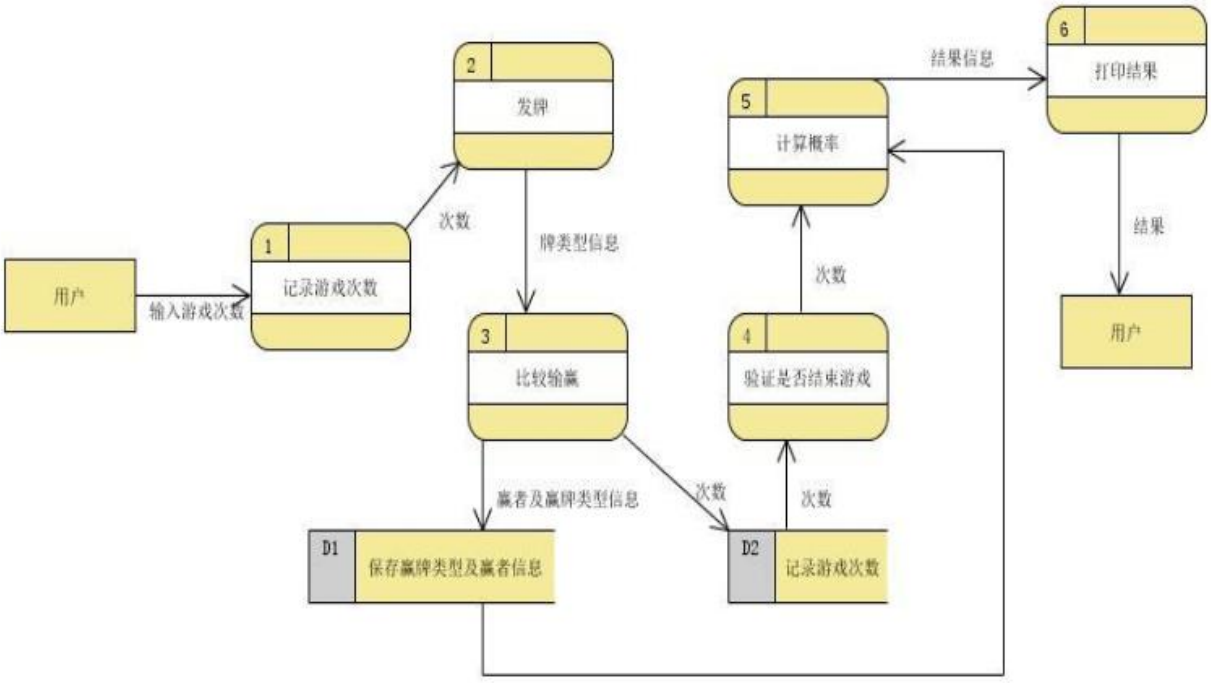
```
{
        A[i][0]=rabl(1,13,&r1);
        A[i][1]=rabl(14,17,&r2);
        B[i][0]=rabl(1,13,&r1);
        B[i][1]=rabl(14,17,&r2);
}
if(E1(A[][10])>E1(B[][10])){
        e1++;
        printf("A 赢，同花顺\n");
}
else if(E1(A[][10])<E1(B[][10]))
{
        e1++;
        printf("B 赢，同花顺\n");
}
else if(E1(A[][10])==E1(B[][10])&&E1(B[][10])==1)
{
        e1++;
        if(max(A[][10])>max(B[][10]))
                printf("A 赢，同花顺\n");
        else
                printf("B 赢，同花顺\n");
}
else if(E2(A[][10])>E2(B[][10]))
{
        e2++;
        printf("A 赢，顺子\n");
```

```
}
else if(E2(A[][10])<E2(B[][10]))
{
        e2++;
        printf("B 赢，顺子\n");
}
else if(E2(A[][10])==E2(B[][10])&&E2(B[][10])==1)
{
        e2++;
        if(max(A[][10])>max(B[][10]))
                printf("A 赢，顺子\n");
        else
                printf("B 赢，顺子\n");
}
else if(E3(A[][10])>E3(B[][10]))
{
        e3++;
        printf("A 赢，同点\n");
}
else if(E3(A[][10])<E3(B[][10]))
{
        e3++;
        printf("B 赢，同点\n");
}
else if(E3(A[][10])==E3(B[][10])&&E3(B[][10])==1)
{
        e3++;
```

```
            if(max(A[][10])>max(B[][10]))
                        printf("A 赢，同点\n");
            else
                        printf("B 赢，同点\n");
}
else if(E4(A[][10])>E4(B[][10])) {
            e4++;
            printf("A 赢，对子\n");
}
else if(E4(A[][10])>E4(B[][10]))
{
            e4++;
            printf("B 赢，对子\n");
}
else if(E4(A[][10])==E4(B[][10])&&E4(B[][10])==1) ;
{
            e4++;
            if(max(A[][10])>max(B[][10]))
                        printf("A 赢，对子\n");
            else
                        printf("B 赢，对子\n");
}
else
{
            if(max(A[][10])>max(B[][10]))
                        printf("A 赢，杂牌\n");
            else
```

```
                printf("B 赢，杂牌\n");
            }
        }
        printf("同花顺赢牌概率为%d，顺子赢牌概率为%d，同点赢牌概率为%d，对子赢牌概率为%d，杂牌赢牌概率为%d",e1/times,e2/times,e3/times,e4/times,e5/times);
}
```

控制流程图：



。

第六章

1、假设只有 SEQUENCE 和 DO_WHILE 两种控制结构，怎么利用它们完成 IF_THEN_ELSE 操作？

解：转化如下：

```
K = 1
DO WHILE （条件 .AND. K.EQ.1)
```

```
        程序块 1
        K=K+1
    END DO
    DO WHILE （ (.NOT. 条件) .AND. K.EQ.1)
        程序块 2
        K=K+1
    END DO
```

2、假设只有 SEQUENCE 和 IF_THEN_ELSE 两种控制结构，怎么利用它们完成 DO_WHILE 操作？

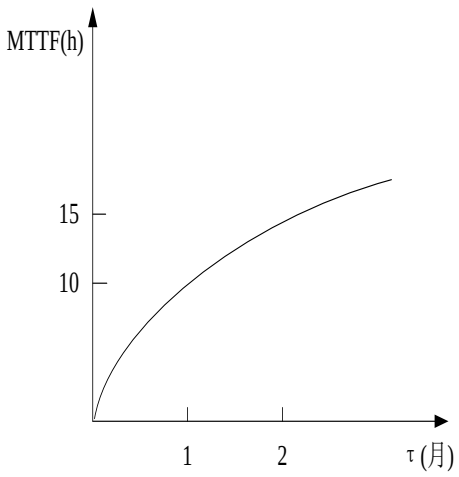
解：转化如下；

```
label:  IF (条件) THEN
            程序块
            GOTO label
        ELSE
            程序块
        END IF
```

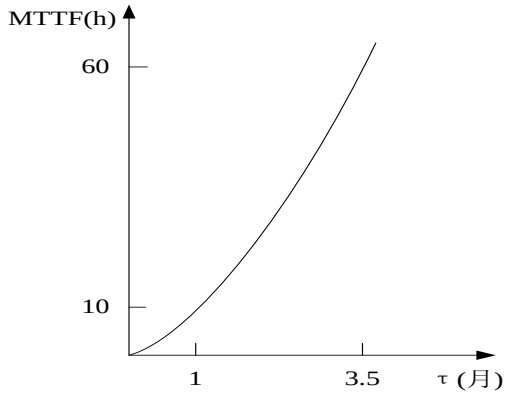
3、画出下列伪代码程序的流程图和盒图：

```
START
IF p THEN
    WHILE q DO
        f
    END DO
ELSE
    BLOCK
        g
        n
    END BLOCK
END IF
STOP
```

解：流程图：



盒图：



4、图 6.18 给出的程序流程图代表一个非结构化的程序，问：

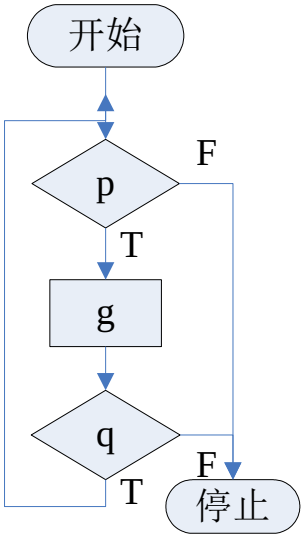
（1）为什么说它是非结构化的？

（2）设计一个等价的结构化程序。

（3）在（2）题的设计中使用附加的标识变量 flag 了吗？

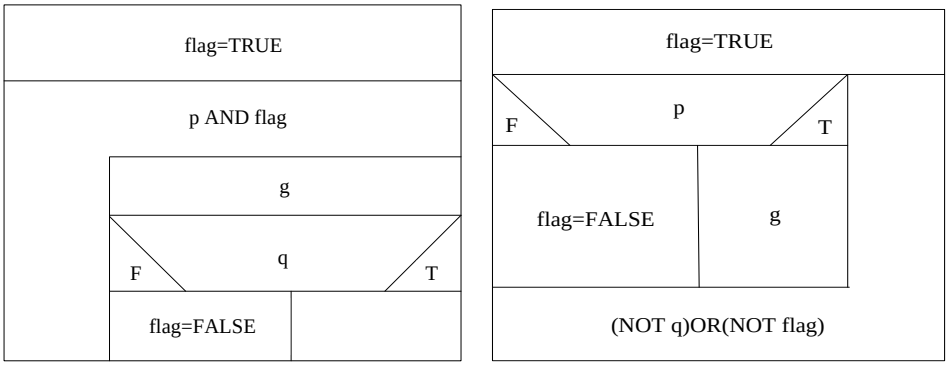
若没用，在设计一个使用 flag 的程序；

若用了，在设计一个不用 flag 的程序。

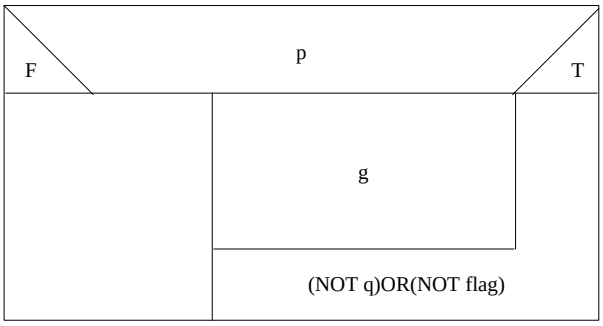


解：（1）通常所说的结构化程序，是按照狭义的结构程序的定义衡量，符合定义规定的程序，每个代码块只有一个入口和一个出口。图示的程序的循环控制结构有两个出口，显然不符合狭义的结构程序的定义，因此是非结构化的程序。

（2）使用附加的标志变量 flag，至少有两种方法可以把该程序改造为等价的结构化程序，图示盒图描绘了等价的结构化程序。



（3）不使用 flag 把该程序改造为等价的结构化程序的方法如图所示。



5、研究下面的伪码程序：

```
LOOP:Set I to(START+FINISH)/2
        IF TABLE(I)=ITME goto FOUND
        IF TABLE(I)<ITME Set START to (I+1)
        IF TABLE(I)>TME Set FINSH to (I-1)
        IF (FINSH-START)>1 goto LOOP
        IF TABLE(START)=ITEM goto FOUND
        IF TABLE(FINSH)=ITEM goto FOUND
        Set FLAG to 0
        Goto DONE
FOUND：Set FLAG to 1
DONE：EXIT
```

要求：

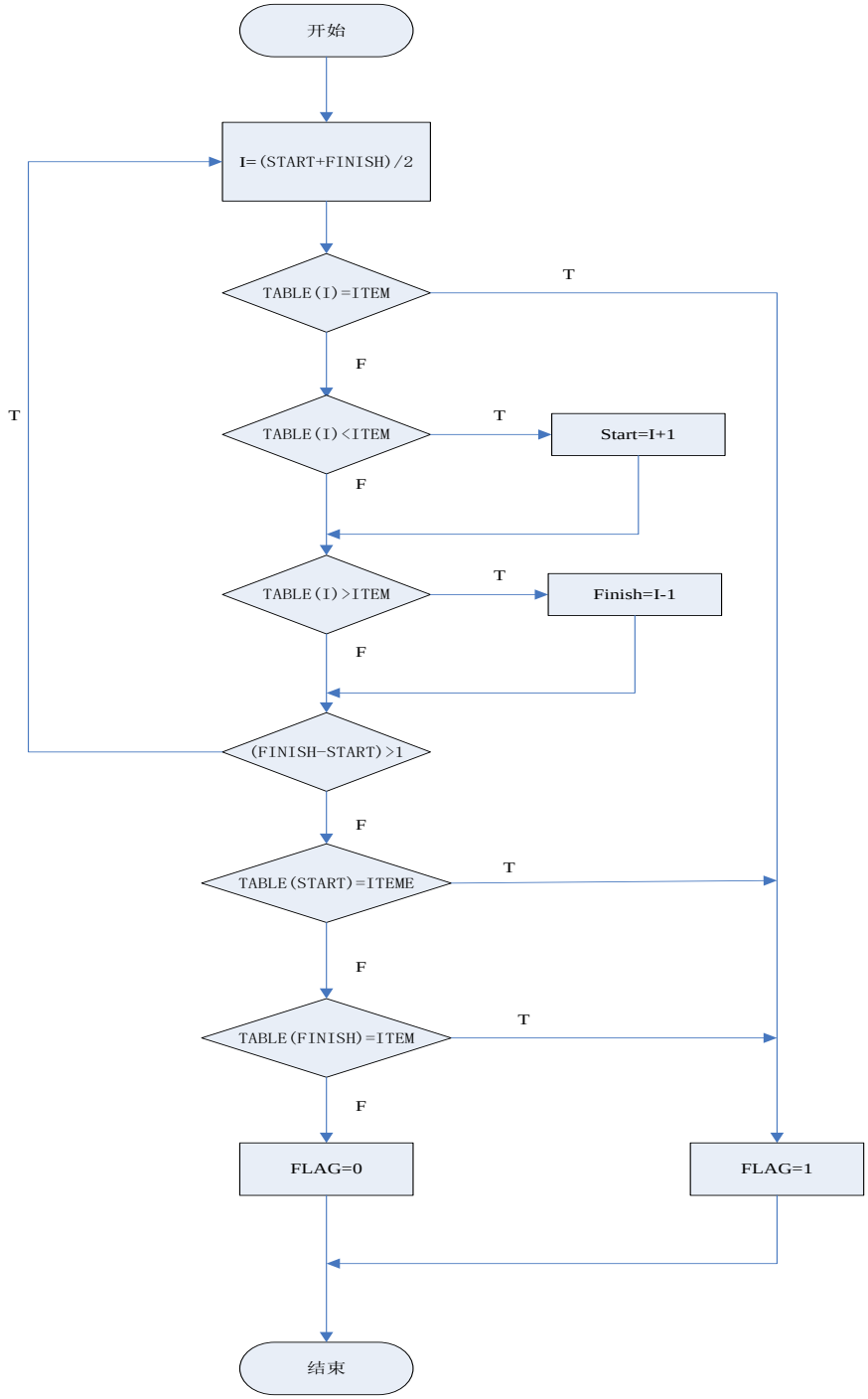
（1）画出流程图。

（2）程序是结构化的吗？说明理由。

（3）若此程序是非结构化，设计一个等价的结构化程序并画出流程图。

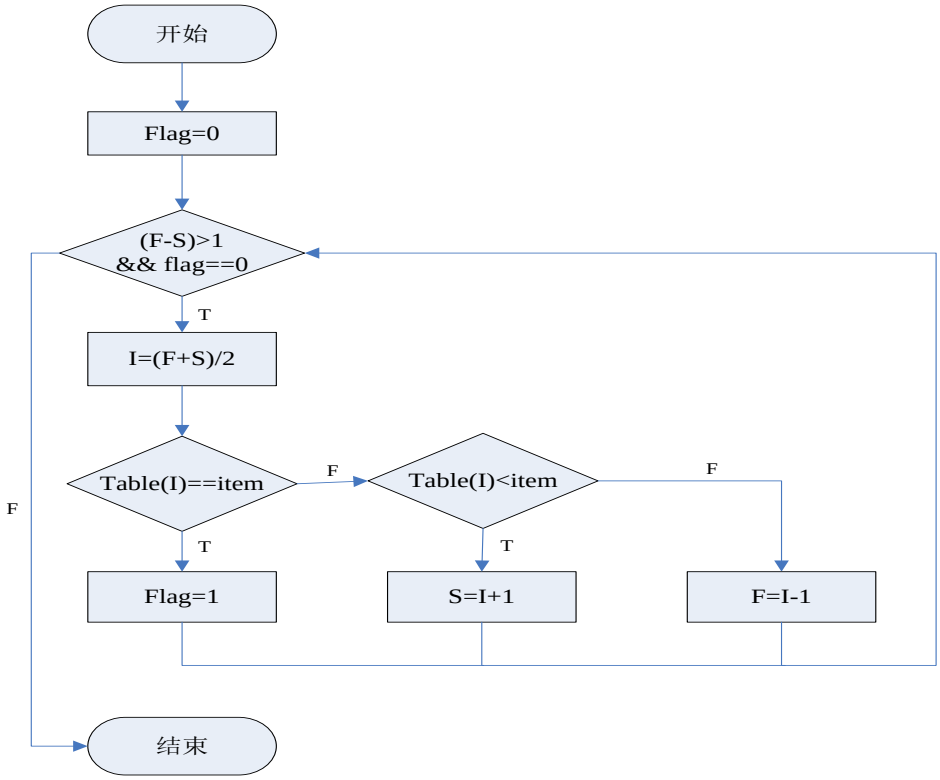
（4）此程序的功能是什么？它完成预定功能有什么隐含的前提条件吗？

解：（1）该程序流程图如下：



（2）该程序不是结构化的，结构化的程序只有一个入口和一个出口，而该程序的流程途中有两个出口。

（3）等价的结构化程序的流程图如下：

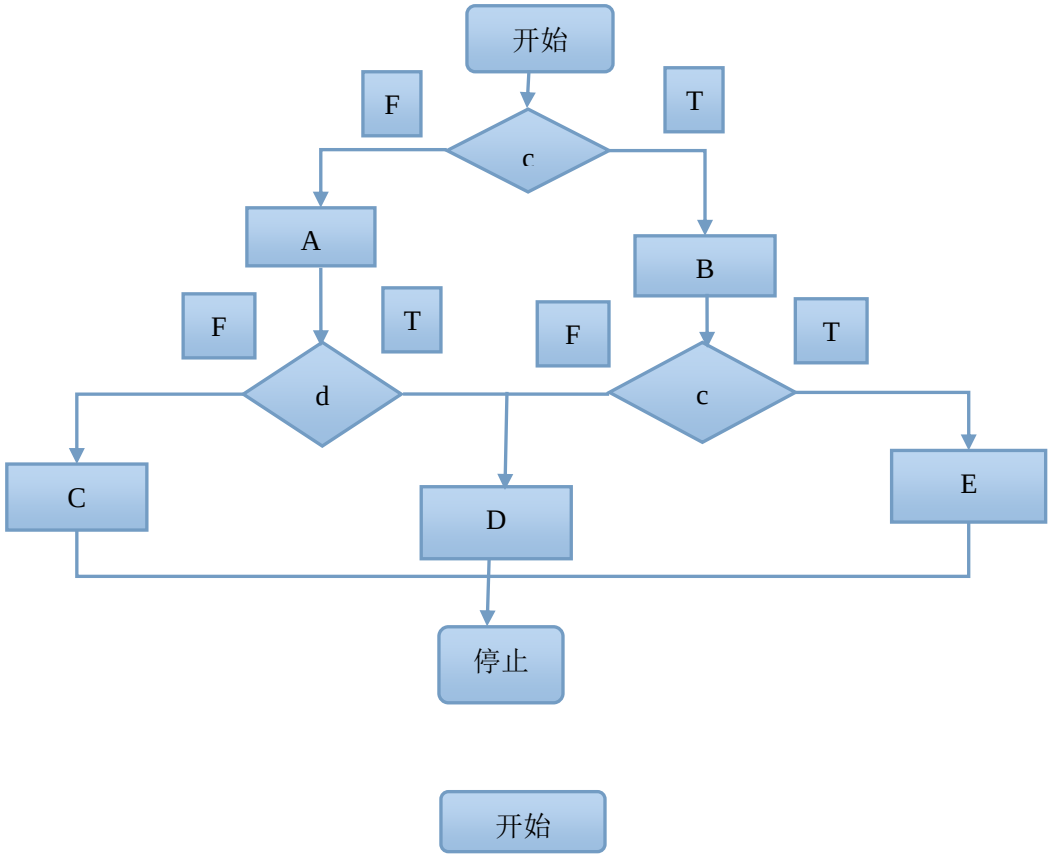


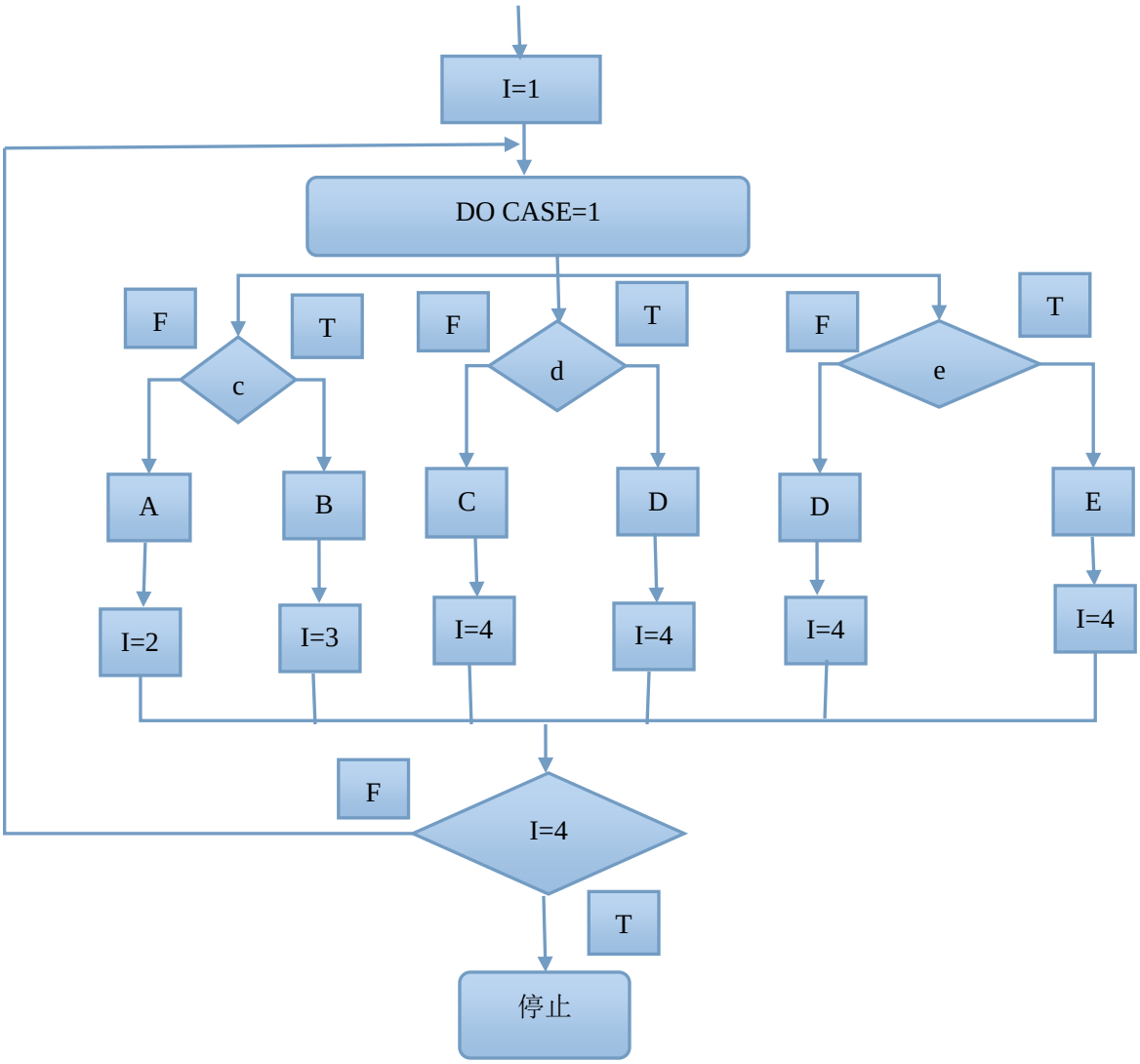
（4）此程序有二分查找的功能，它完成预定功能的隐含前提条件是现有序列为从小到大顺序排好的有序序列。

6．用 Ashcroft_Manna 技术可以将非结构化的程序转化为结构化程序，图 6.19 是一个转换的例子。

(1)能否从这个例子总结出 Ashcroft_Manna 技术的一些基本方法?

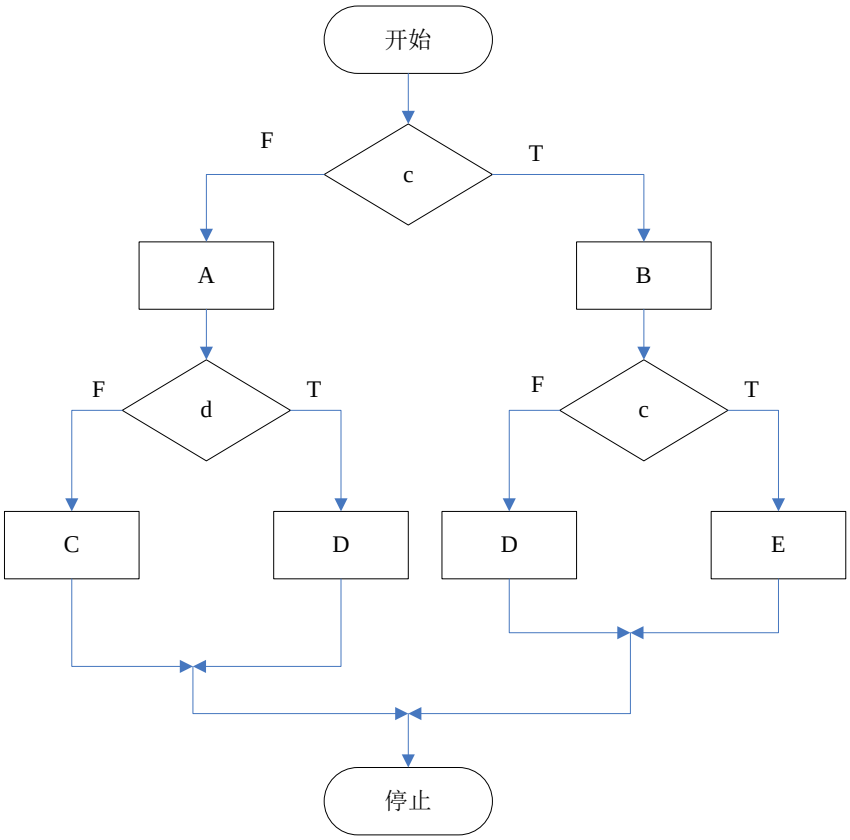
(2)进一步简化 6.19(b)给出的结构化设计。





解：（1）从这个例子中看出，Ashcroft_Manna 技术的基本方法是，当待改造的序含有嵌套的非结构化的IF语句时，改造后的程序中增加DO-CASE语句和DO-UNTIL语句，并增加一个辅助变量I，I的初始值为 1 。最外层的IF语句在I= 1 时执行，执行完这个IF语句后把I赋值为随后应该执行的内层IF语句所对应的CASE标号值。DO-CASE语句的最大分支数（可执行的最大标号值）等于IF语句的个数。当执行完最内层的IF语句之后，把I赋值为可执行的最大标号值加 1 ，而DO-UNTIL 循环的结束条件就是I等于这个值。

（2）进一步简化后的结构化程序的流程图如下所示。



7、某交易所规定给经纪人的手续费计算方法如下：总手续费等于基本手续费加上与交易中的每股价格和股数有关的附加手续费。如果交易金额少于 1000 元，则基本手续费为交易金额的 8.4%；如果交易总金额在 1000 元~10000 元之间，则基本手续费为交易金额的 5%，再加 34 元；如果金额超过 10000 元，则基本手续费为交易金额的 4%加上 134 元。当每股售价低于 14 元时，附加手续费为基本手续费的 5%，除非买进、卖出的股数不是 100 的倍数，在这种情况下附加手续费的 9%。当每股售价在 14 元到 25 元之间时，附加手续费为基本手续费的 2%，除非交易的股数不是 100 的倍数，在这种情况下附加手续费的 6%。当每股售价超过 25 元时，如果交易的股数（即不是 100 的倍数），则附加手续费为基本手续费的 4%，否则附加手续费为基本手续费的 1%。

要求：

（1）用判定表表示手续费的计算方法。

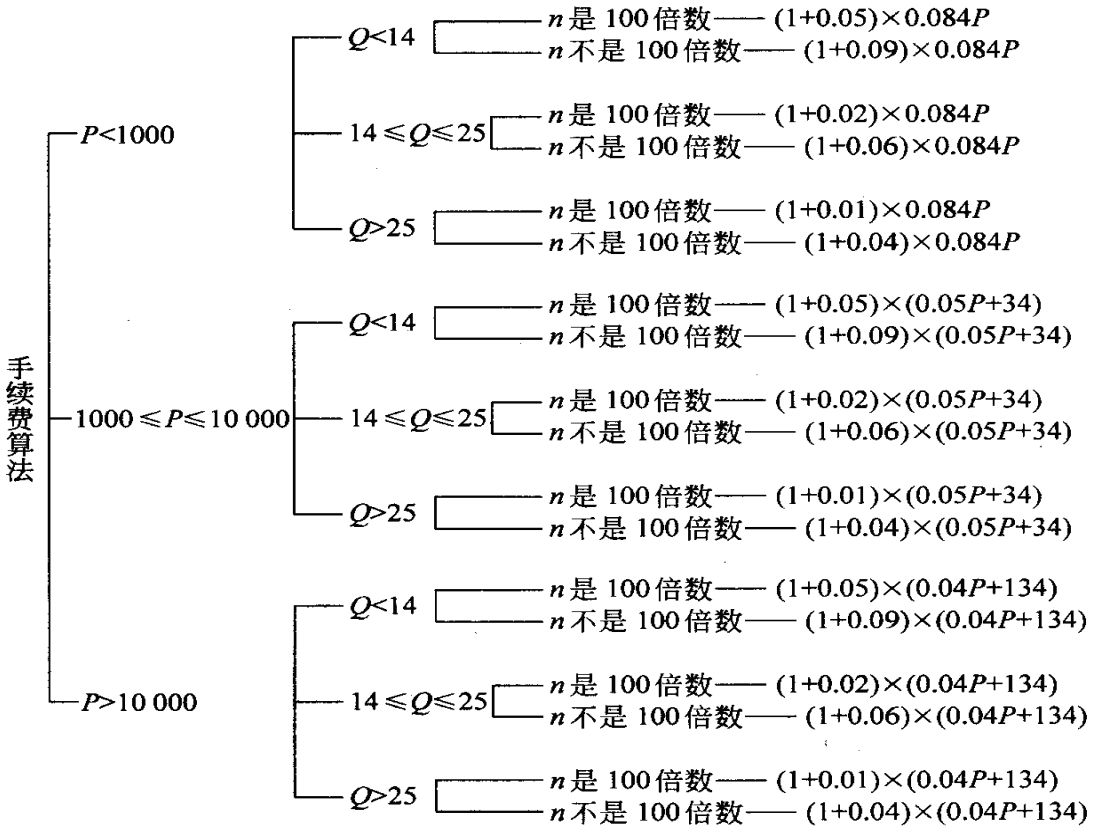
（2）用判定数表示手续费的计算方法。

解：令 P 代表交易的总金额，Q 代表每股的售价，n 代表交易的股数。

（1）表示手续费计算方法的判定表如图所示

	规则																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$P<1000$	T	T	T	T	T	T	F	F	F	F	F	F	F	F	F	F	F	F
$1000\leqslant P\leqslant 10000$	F	F	F	F	F	F	T	T	T	T	T	T	F	F	F	F	F	F
$P>10000$	F	F	F	F	F	F	F	F	F	F	F	F	T	T	T	T	T	T
$Q<14$	T	T	F	F	F	F	T	T	F	F	F	F	T	T	F	F	F	F
$14\leqslant Q\leqslant 25$	F	F	T	T	F	F	F	F	T	T	F	F	F	F	T	T	F	F
$Q>25$	F	F	F	F	T	T	F	F	F	F	T	T	F	F	F	F	T	T
n 是 100 的倍数	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
$(1+0.05)\times 0.084P$	×																	
$(1+0.09)\times 0.084P$		×																
$(1+0.02)\times 0.084P$			×															
$(1+0.06)\times 0.084P$				×														
$(1+0.01)\times 0.084P$					×													
$(1+0.04)\times 0.084P$						×												
$(1+0.05)\times(0.05P+34)$							×											
$(1+0.09)\times(0.05P+34)$								×										
$(1+0.02)\times(0.05P+34)$									×									
$(1+0.06)\times(0.05P+34)$										×								
$(1+0.01)\times(0.05P+34)$											×							
$(1+0.04)\times(0.05P+34)$												×						
$(1+0.05)\times(0.04P+134)$													×					
$(1+0.09)\times(0.04P+134)$														×				
$(1+0.02)\times(0.04P+134)$															×			
$(1+0.06)\times(0.04P+134)$																×		
$(1+0.01)\times(0.04P+134)$																	×	
$(1+0.04)\times(0.04P+134)$																		×

（2）表示手续费计算方法的判定树

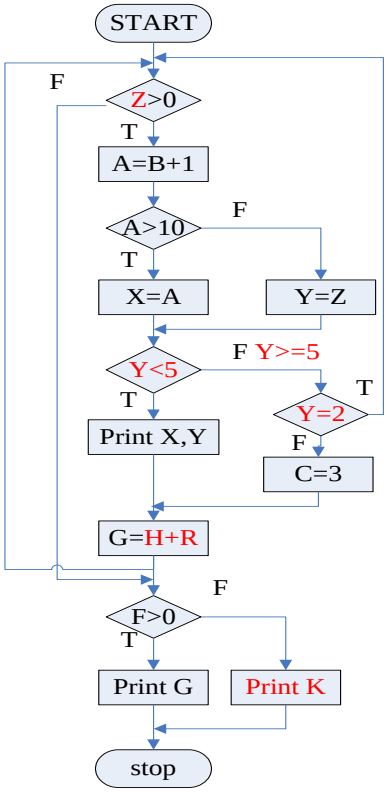


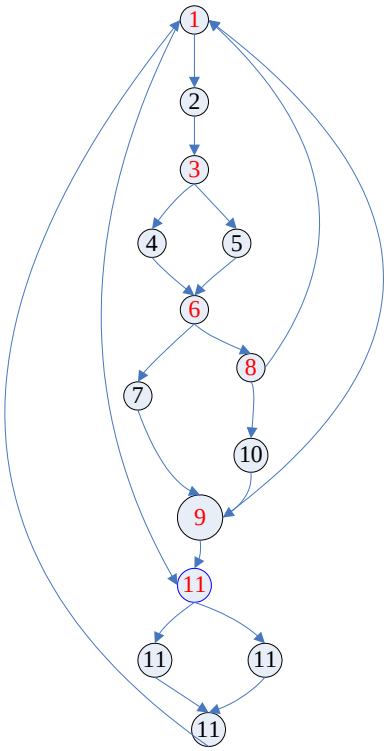
8、画出下列伪码程序的流图，计算它的环形复杂度。你觉得这个程序的逻辑有什么问题吗？

C　EXAMPLE

```
LOOP:DO WHILE  X>0
            A=B+1
       IF A>10
            THEN X=A
       ELSE Y=Z
           END IF
IF Y<5
    THEN PRINT X,Y
     ELSE IF Y=2
             THEN GOTO LOOP
     ELSE C=3
     END IF
END IF
G=H+R
END DO
IF F>0
    THEN PRINT G
    ELSE PRINT K
END IF
STOP
```

解：流程图：





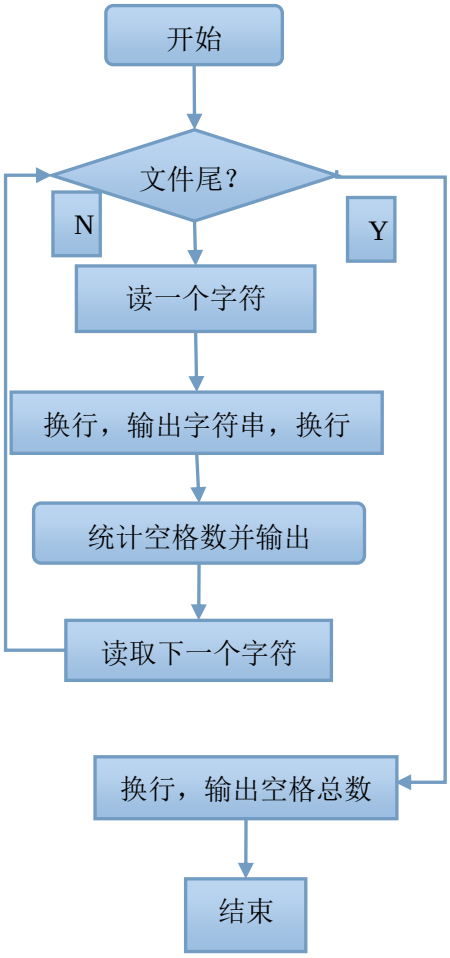
环形复杂度：

V(G)＝结点 *E*－弧数 *n*＋1=17-11+1=7=判断结点 ＋*1=6* ＋*1*＝*7*=封闭的区域数

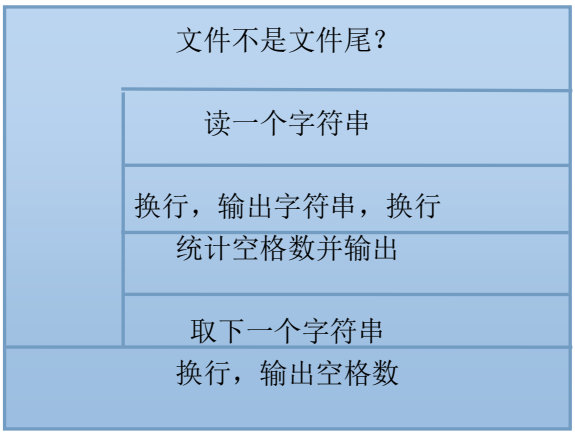
逻辑有问题，当 Z>0 时，容易形成死循环；条件 Y<5 包含条件 Y=2。

9、把统计空格程序的 Jackson 图(图 6.13）该画为等价的程序流程图和盒图。

解：流程图：

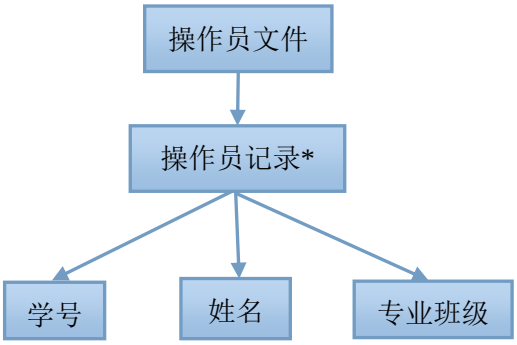


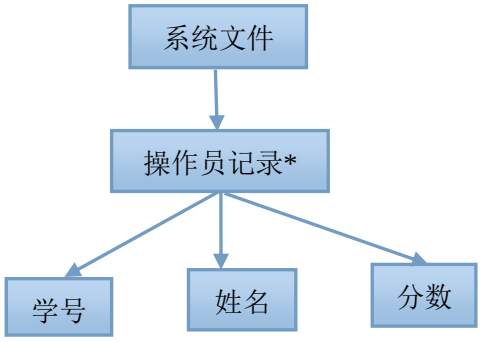
盒图：

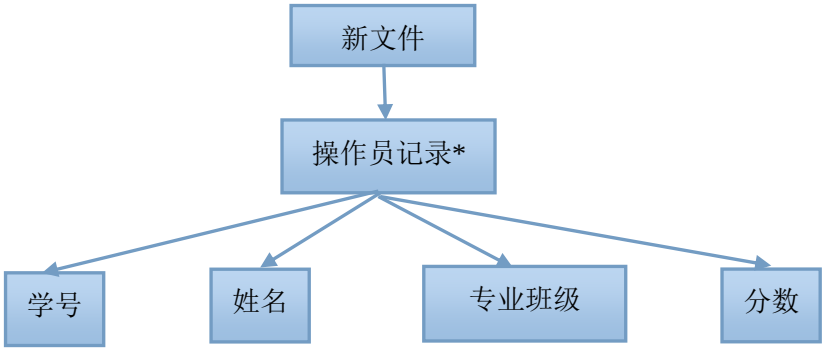


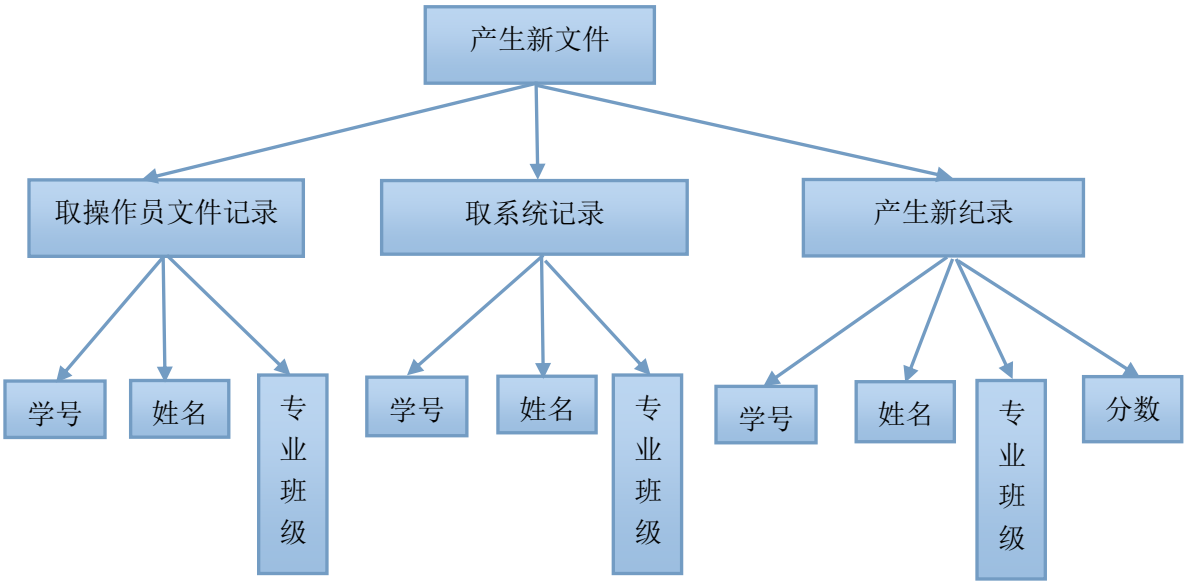
10、人机对话由操作员信息和系统信息交替组成。假设一段对话总是由操作员信息开始以系统信息结束，用 Jackson 图描绘这样的人机对话过程。

解：









第七章

第一题

1、下面给出的伪码中有一个错误。请仔细阅读这段伪码，说明该伪码的语法特点，找出并改正伪码中的错误。字频统计程序的伪码如下：

INITIALIZE the Program
READ the first text record
DO WHILE there are more words in the text record
DO WHILE there are more words in the text record
EXTRACT the next text word
SEARCH the word-table for the extracted word
IF the extracted word is found
INCREMENT the word's occurrence count
ELSE
INSERT the extracted word into the table
END IF
INCREMENT the words-processed count
END DO at the end of the text record
READ the next text record
END DO when all text records have heen read
PRINT the table and summary information
TERMINATE the program

答：INSERT the extracted word into the table 在这个后面，有没有给这个 word 的 occurrence/count 赋值为1

第二题

2、研究下面给出的伪码程序，要求：
(1) 画出它的程序流程图。
(2) 它是结构化的还是非结构化的?说明理由。
(3) 若是非结构化的，则
(a) 把它改造成仅用 3 种控制结构的结构化程序；
(b) 写出这个结构化设计的伪码；
(c) 用盒图表示这个结构化程序。
(4) 找出并改正程序逻辑中的错误。
COMMENT:PROGRAM SEARCHES FOR FIRST N REFERENCES
TO A TOPIC IN AN INFORMATION RETRIEVAL
SYSTEM WITH T TOTAL ENTRIES
INPUT N
INPUT KEYWORD(S)FOR TOPIC
I=O
MATCH=0
DO WHILE I≤T
I=I+1
IF WORD=KEYWORD
THEN MATCH=MATCH+1
STORE IN BUFFER

END

IF MATCH=N

THEN GOTO OUTPUT

END

END

IF N=0

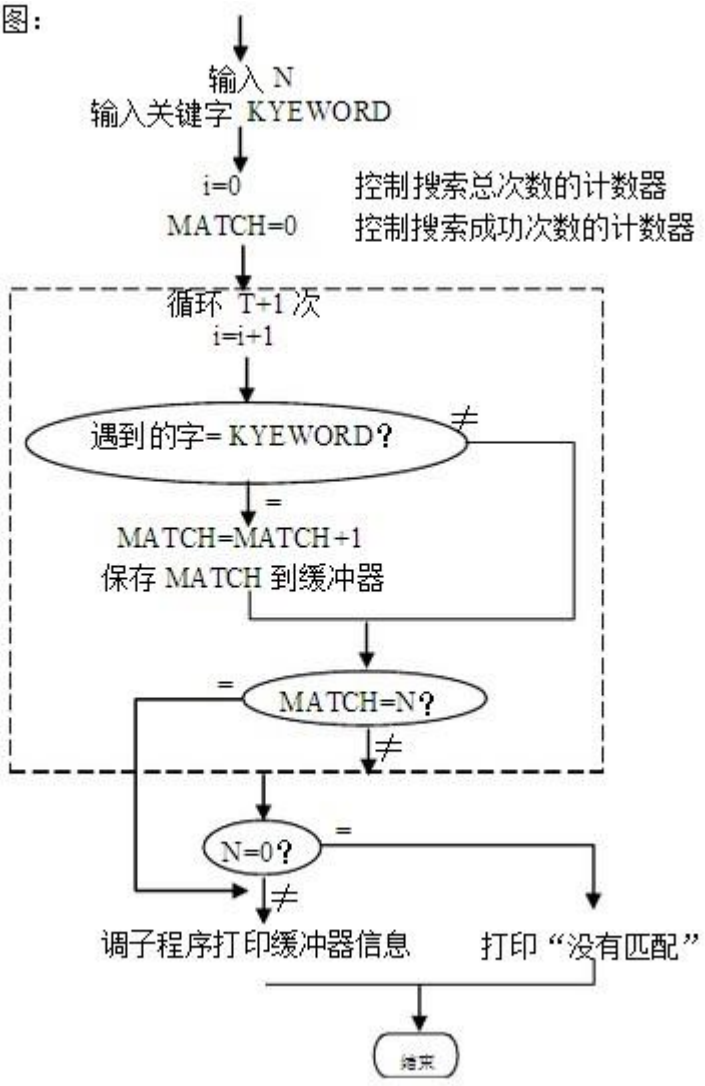
THEN PRINT ″NO MATCH″

OUTPUT:ELSE CALL SUBROUTINE TO PRINT BUFFER

INFORMATION

END

解：(1) 程序流程图：



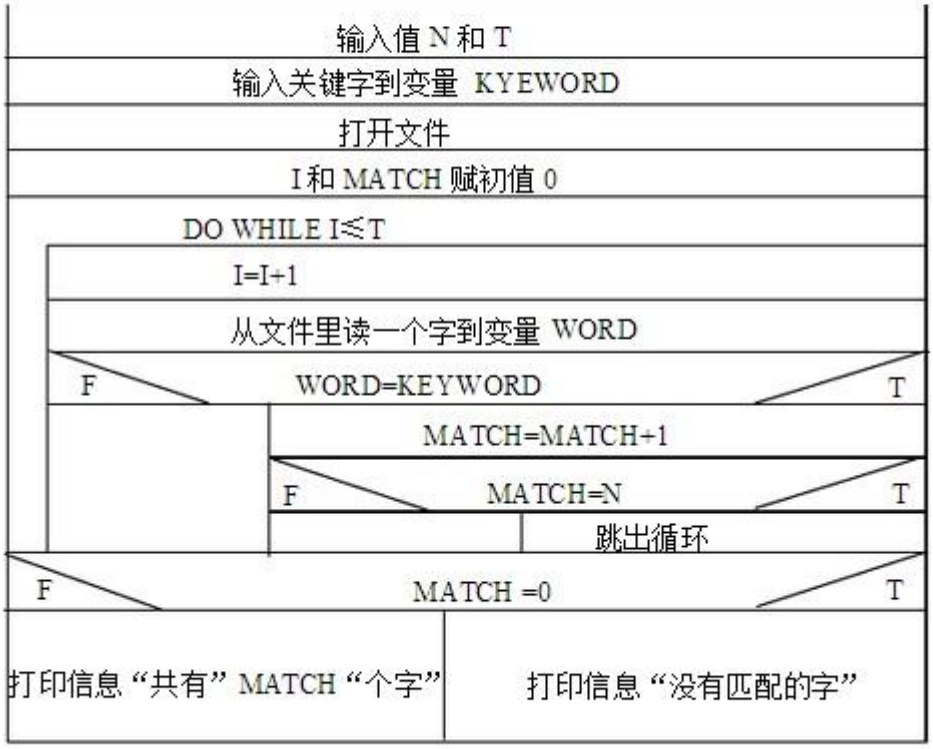
⑵ 此程序是非结构化的，它有一个 GOTO 语句，并且是从一个循环体内转到循环体外的一个条件语句内部。

⑶ 修改后的伪码如下：

```
INPUT N，T                              输入 N
INPUT KEYWORD(S) FOR TOPIC              输入有关话题的关键字
OPEN FILE                               打开文件
I=0
MATCH=0
DO WHILE I≤T                            循环—最多可做 T 次
   I=I+1
   READ A WORD OF FILE TO WORD          从文件里读一个字到变量 WORD
   IF WORD=KEYWORD
      THEN MATCH=MATCH+1
       IF MATCH=N   THEN EXIT           搜索到了 N 个关键字，就跳出循环
       END IF
   END IF
END DO
```

IF MATCH =0
　THEN PRINT "NO MATCH"　　若 MATCH =0 就打印“没有相匹配”
ELSE　PRINT “共搜索到”；MATCH；“个匹配的关键字”　　否则打印信息
END IF

修改后的程序框图（盒图）



⑷ 程序中的错误：语句“IF WORD=KEYWORD”里的变量“WORD”没有预先赋值。
程序中没有预先输入 T 的值
“N=0”应该是“MATCH=0”

第三题

3、在第 2 题的设计中若输入的 N 值或 KEYWORD 不合理，会发生问题。

(1) 给出这些变量的不合理值的例子。

(2) 将这些不合理值输入程序会有什么后果?

(3) 怎样在程序中加入防错措施，以防止出现这些问题?

（1）答：N=1，KEYWOED=#。

（2）答：无法打印出任何信息。

（3）答：加入循环

第四题

4、（1）什么是模块测试和集成测试？它们各有什么特点？

答：模块测试是对每个单独的模块，分别用黑盒和白盒测试技术，测试它的功能是否正确，

检查模块控制结构中的特定路径并发现最大数量的错误。

其特点是：主要应用白盒测试的技术，对多个模块的测试可以并发的进行。

集成测试是把模块装配在一起形成完整的软件包，在装配的同时进行测试。

特点是：可能发生接口问题。

（2）假设有一个由 1000 行 FORTRAN 语句构成的程序，估计在对它进行测试期间将发现多少个错误？为什么？

答：月 25 至 100 个错误，美国的一些统计数字告诉我们通常这个比值在 0.005~0.02 之间，也就是说，测试之前每 1000 条指令中大约有 5~20 个错误。假设测试之前每 1000 条指令中有 10 个错误，则估计对它进行测试期间将发现的错误数为：5000*10/1000=50。

（3）设计下列伪码程序的语句覆盖和路径覆盖测试用例：

```
START
INPUT(A,B,C)
IF  A＞5
THEN X=10
ELSE X=1
END IF
IF B>10
THEN X=20
ELSE X=2
END IF
IF C>15
THEN X=30
ELSE X=3
END IF
PRINT (X,Y,Z)
STOP
```

答：此程序的语句覆盖用例：①A=5，B=10，C=15；②A=6，B=11，C=16 条件覆盖用例为：①A=5，B=10，C=15；②A=6，B=11，C=16。

语句覆盖测试用例

序号	判定			输入			预期的输		
	1	2	3	A	B	C	X	Y	Z
1	F	F	F	1	1	1	1	2	3
2	T	T	T	20	40	60	10	20	30

语句覆盖的含义是，选择足够多的测试数据，使被测试程序中的每个语句至少执行一次。

序号	判定			输入			预
	1	2	3	A	B	C	X
1	F	F	F	1	1	1	1
2	F	F	T	1	1	60	1
3	F	T	F	1	40	1	1
4	F	T	T	1	40	60	1
5	T	F	F	20	1	1	10
6	T	F	T	20	1	60	10
7	T	T	F	20	40	1	10
8	T	T	T	20	40	60	10

第五题

5、某图书馆有一个使用 CRT 终端的信息检索系统，该系统有下列 4 个基本检索命令

要求：

(1) 设计测试数据以全面测试系统的正常操作；

(2) 设计测试数据以测试系统的非正常操作

名 称	语 法	操 作
BROWSE （浏览）	b(关键字)	系统搜索给出的关键字，找出字母排列与此关键字最相近的字。然后在屏幕上显示约 20 个加了行号的字，与给出的关键字完全相同的字约在中央
SELECT （选取）	s(屏幕上的行号)	系统创建一个文件保存含有由行号指定的关键字的全部图书的索引，这些索引都有编号(第一个索引的编号为 1，第二个为 2……依此类推)
DISPLAY （显示）	d(索引号)	系统在屏幕上显示与给定的索引号有关的信息，这些信息与通常在图书馆的目录卡片上给出的信息相同。这条命令接在 BROWSE/SELECT 或 FIND 命令后面用，以显示文件中的索引信息
FIND （查找）	f(作者姓名)	系统搜索指定的作者姓名，并在屏幕上显示该作者的著作的索引号，同时把这些索引存入文件

解：⑴测试系统正常操作的测试数据

①顺序执行下列 3 个命令：

b（KEYWORD）

s（L）

d（N）

其中，KEYWORD 是正确的关键字；L 是执行命令 b 后在屏幕上显示的约 20 个行号中的一个（至少应该使 L 分别为第一个、最后一个和中央一个行号）；N 是执行命令 s 后列出的索引号中的一个（至少应该使 N 分别为第一个、最后一个和中央一个索引号）。

针对若干个不同的 KEYWORD 重复执行上述命令序列。

②顺序执行下列 2 个命令：

f（NAME）

d（N）

其中，NAME 是已知的作者姓名；N 是执行命令 f 后列出的索引号中的一个（至少应该使 N 分别为第一个、最后一个和中央一个索引号）。

针对若干个不同的 NAME 重复执行上述命令序列。

⑵测试系统非正常操作的测试数据

①用过长的关键字作为命令 b 的参数：例如，b（reliability software and hardware combined）

预期的输出：系统截短过长的关键字，例如，上列命令中的关键字可能性被截短为 reliability software

②用不正确的关键字作为命令 b 的参数：例如，b（AARDVARK）

预期的输出：显示出最接近的匹配结果，例如，执行上列命令后可能显示

1．AARON，JULES（book）

③用比执行命令 b 后列出的最大行号大 1 的数作为命令 s 的参数

预期的输出：“命令 s 的参数不在行号列表中”

④用数字和标点符号作为命令 b 和命令 f 的参数
预期的输出：“参数类型错”
⑤用字母字符作为命令 s 和命令 d 的参数
预期的输出：“参数类型错”
⑥用 0 和负数作为命令 s 和命令 d 的参数
预期的输出：“参数数值错”
⑦命令顺序错：例如，没执行命令 b 就执行命令 s，或没执行命令 s 就执行命令 d
预期的输出：“命令顺序错”
⑧命令语法错：例如，遗漏命令名 b、s、d 或 f；或命令参数没用圆括号括起来
预期的输出：“命令语法错”
⑨命令参数空：例如，b（）、s（）、d 或 f（）
预期的输出：系统提供默认参数或给出出错信息
⑩使用拼错了的作者姓名作为 f 的参数
预期的输出：“找不到这们作者的著作”

第六题

6、航空公司 A 向软件公司 B 订购了一个规划飞行路线的程序。假设你是软件公司 C 的软件工程师，A 公司已雇用你所在的公司对上述程序进行验收测试。任务是，根据下述事实设计验收测试的输入数据，解释你选取这些数据的理由。
领航员向程序输入出发点和目的地，以及根据天气和飞机型号而初步确定的飞行高度。程序读入途中的风向风力等数据，并且制定出 3 套飞行计划(高度，速度，方向及途中的 5 个位置校核点)。所制定的飞行计划应做到燃料消耗和飞行时间都最少。

用正常的输入数据作为测试数据

① 向程序输入常规的出发点，目的地，飞机型号，5 个位置校核点，高度和速度。
② 输入 3~5 组出发点和目的地，重复执行步骤 1
③ 输入固定的出发点、目的地、飞机型号、5 个位置校核点和高度，分别输入 3~5 个不同的速度，重复执行步骤 1
④ 输入固定的出发点、目的地、飞机型号、5 个位置校核点和速度，分别输入 3~5 个不同的高度，重复执行步骤 1
⑤ 输入固定的出发点、目的地、飞机型号、速度和高度，分别输入 3~5 组不同的位置校核点，重复执行步骤 1
⑥ 输入固定的出发点、目的地、5 个位置校核点和高度，分别输入 3~5 个不同的飞机型号，重复执行步骤 1
⑦ 输入固定的目的地、5 个位置校核点和高度，分别输入 3~5 个不同的飞机型号，重复执行步骤 1
⑧ 输入固定的出发点、5 个位置校核点和高度，分别输入 3~5 个不同的目的地，重复执行步骤 1

（2） 用特殊的数据值作为测试数据

① 分别输入非常高和非常低的数据组合测试
② 用负数测试
③ 输入数字 0 进行测试
④ 分别输入相距非常远和非常近的出发点和目的地测试

ay 和 size，size 小于数组的大小，并给出需要查找的值，该值在 somearray 中；
预期的输出：返回-1；
⑤ 首先给出某个数组 somearray 和 size，size 大于数组的大小，并给出需要查找的值，该值不在 somearray 中；
预期的输出：返回-1；
⑥ 首先给出某个数组 somearray 和 size，size 大于数组的大小，并给出需要查找的值，该值在 somearray 中；
预期的输出：返回-1；

第七题

7、严格说来，有两种不同的路径覆盖测试，分别为程序路径覆盖和程序图路径覆盖。这两种测试可分别称为程序的自然执行和强, , 迫执行。所谓自然执行是指测试者(人或计算机)读入程序中的条件表达式，根据程序变量的当前值计算该条件表达式的值(真或假)，并相应地分支。强迫执行是在用程序图作为程序的抽象模型时产生的一个人为的概念，它可以简化测试问题。强迫执行的含义是，一旦遇到条件表达式，测试者就强迫程序分两种情况(条件表达式的值为真和为假)执行。显然，强迫执行将遍历程序图的所有路径，然而由于各个条件表达式之间存在相互依赖的关系，这些路径中的某一些在自然执行时可能永远也不会进入。 为了使强迫执行的概念在实际工作中有用，它简化测试工具的好处应该超过它使用额外的不可能达到的测试用例所带来的坏处。在绝大多数情况下，强迫执行的测试数并不比自然执行的测试数大很多，此外，对强迫执行的定义实际上包含了一种技术，能够缩短在测试含有循环的程序时所需要的运行时间。 程序的大部分执行时间通常用于重复执行程序中的 DO 循环，特别是嵌套的循环。因此必须发明一种技术，使得每个 DO 循环只执行一遍。这样做并不会降低测试的功效，因为经验表明第一次或最后一次执行循环时最容易出错。
Laemmel 教授提出的自动测试每条路径的技术如下： 当编写程序时每个 DO 循环应该写成一种包含测试变量 T 和模式变量 M 的特殊形式，因此
DO I= 1 TO 38
应变成
DO I=1 TO M*38+(1-M)*T
可见，当 M=0 时处于测试模式，而 M=1 时处于正常运行模式。当处于测试模式时，令 T=0 则该循环一次也不执行，令 T=1 则该循环只执行一次。 类似地应该使用模式变量和测试变量改写 IF 语句，例如
IF X+Y>0 THEN Z=X
ELSE Z=Y 应变成
IF M*(X+Y)+T>0 THEN Z=X
ELSE
Z=Y 正常运行时令 M=1 和 T=0，测试期间令 M=0，为测试 THEN 部分需令 T=+1，测试 ELSE 部分则令 T=-1。
要求：
(1) 选取一个包含循环和 IF 语句的程序，用 Laemmel 技术修改这个程序，上机实际测试这个程序并解释所得到的结果。
(2) 设计一个程序按照 Laemmel 技术自动修改待测试的程序。利用这个测试工具修改上一问中人工修改的程序，两次修改得到的结果一致吗?
(3) 怎样把 Laemmel 技术推广到包含 WHILE DO 和 REPEAT UNTIL 语句的程序?
(4) 试分析 Laemmel 技术的优缺点并提出改进意见。

（1）答：if (x+y)>2

A=x

Else

A=y

改为：if M*(x+y)+T>2

A=x

Else

A=y

结果一致。

(2)答：两次结果是一致的

(3)答：WHILE DO 和 IF ELSE 修改技术类似。

(4)答：优点是可以使得结果更为精确。缺点是并不适用所有的程序，有时会有程序运行变得复杂。

第八题

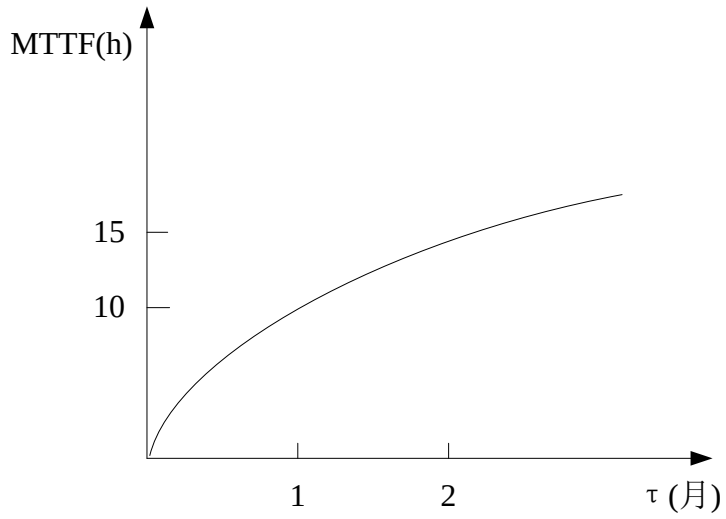
8、对一个包含 10000 条机器指令的程序进行一个月集成测试后，总共改正了 15 个错误，此时 MTTF=10h；经过两个月测试后，总共改正了 25 个错误（第二个月改正了 10 个错误），MTTF=15h。

要求：

（1）根据上述数据确定 MTTF 与测试时间之间的函数关系，画出 MTTF 与测试时间 τ 的关系曲线。在化这条曲线是做了什么假设？

（2）为做到 MTTF=100h，必须进行多长时间的集成测试？当集成测试结束时总共改正了多少个错误，还有多少个错误潜伏在程序中？

答：（1）MTTF 与测试时间 τ 的关系曲线如下：



假设调试过程中没有引入新的错误。

（2）根据估算平均无故障时间的公式可得：

1/K(Et/10000-100/10000)=10

1/K (Et/10000-300/10000)=15

计算可得：K＝333，Et=45

当 MTTF=100h 时，有

1/333(45/10000-Ec/10000)=100

计算可得：Ec=42.按前两个月测试改错的进度估算，需进行 3 个月的集成测试。

当测试结束时，共改正了 42 个错误，还有 3 个错误潜伏在程序中。

第九题

9．如对一个长度为 100000 条指令的程序进行集成测试期间记录下下面的数据：

（a）7 月 1 日：集成测试开始，没有发现错误。

（b）8 月 2 日：总共改正 100 个错误，此时 MTTF=0.4h

（c）9 月 1 日：总共改正 300 个错误，此时 MTTF=2h

根据上列数据完成下列各题。

估计程序中的错误总数。

为使 MTTF 达到 10h，必须测试和调试这个程序多长时间？

画出 MTTF 和测试时间 τ 之间的函数关系曲线。

答：（1）根据估算平均无故障时间的公式可得：

1/K(Et/100000-100/100000)=0.4

1/K (Et/100000-300/100000)=2

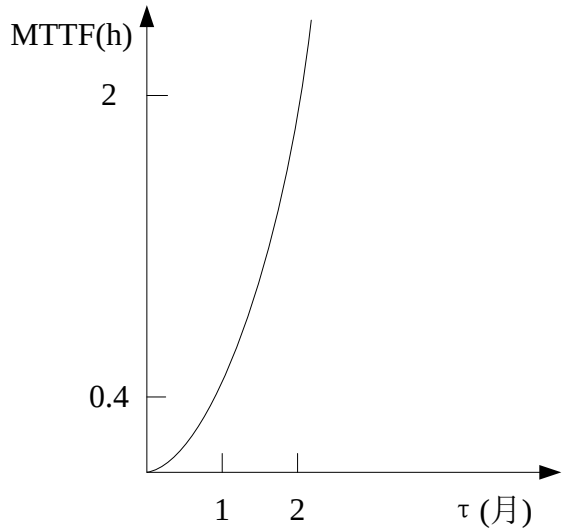
计算可得：K＝1000，Et=350 即程序中的错误总数为 350。

（2）当 MTTF＝10h 时，有

1/K (350/100000- Ec/100000)=10

计算可得：Ec=340.按前两个月测试改错的进度估算，还进行 2 个月的集成测试。

（3）MTTF 和测试时间 τ 之间的函数关系曲线如下：



第十题

10、在测试一个长度为 24000 条指令的程序时，第一个月由甲、乙两名测试员各自独立测试这个程序。经一个月测试后，甲发现并改正 20 个错误，使 MTTF 达到 10h。与此同时，乙发现 24 个错误，其中 6 个甲也发现了。以后由甲一个人继续测试这个程序。问：

（1）刚开始测试时程序中总共有多少个潜在的错误？

（2）为使 MTTF 达到 60h，必须再改正多少个错误？还需多长测试时间？

（3）画出 MTTF 与集成测试时间 τ 之间的函数关系曲线。

答：（1）根据公式：B0=B2B1/bc，可得：

B0＝20*24/6=80，即刚开始测试时程序中总共有 80 个错误。

（2）根据估算平均无故障时间的公式可得：

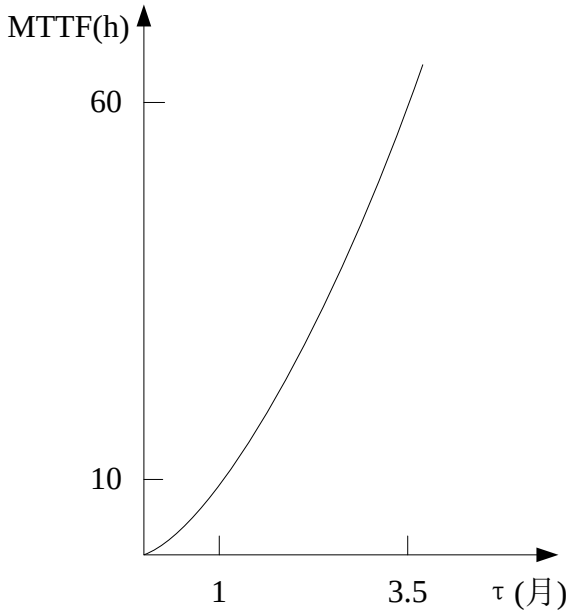
1/K(80/24000-20/24000)=10

1/K(80/24000- Ec/24000)=60

计算可得：K=40，Ec=70

即还需要改正 50 个错误。根据甲的改正进度，估计还需要 3 个月。

（3）MTTF 与集成测试时间 τ 之间的函数关系曲线如下：



第八章

一.答：软件的可维护性与哪些因素有关？在软件开发过程中应该采取哪些措施来提高软件产品的可维护性？

答：（1）、可理解性

（2）、可预测性

（3）、可修改性

（4）、可移植性

（5）、可重用性

在每个阶段结束前的技术审查和管理复查中，应该着重对可维护性进行复审过程中，应该对将来要改进的部分和可能要改的部分加以注意指明，应该讨论软件的可移植性问题，考虑可能影响软件维护的系统界面。在设计和编码过程中应该尽量使用可重用的软件构件，每个测试步骤都可以暗示在软件正式交付使用之前，程序中可能需要做预防性维护的部分。在完成每项维护工作之后，都应该对软件维护本身仔细认真地复审。

二：答：应该选取 a 和 c 因为文档是影响软件可维护性的决定因素。因此，文档甚至比可执行的程序代码更重要，文档必须和程序代码同时维护，只有和程序代码完全一致才能是真正有价值的代码。

文档修改：针对系统中当前正在修改的那些部分文档建立完整的文档。
代码重构：首先用重构工具分析源代码，标注出和结构化程序设计概念相违背的部分。然后重构有问题的代码（此项可以工作可自动化进行）。最后，复审和测试生成的重构代码（以保证没有引入异常）并更新代码文档。

三:　答：对储蓄系统：一般大型软件的维护成本远远高于开发成本若干倍，所以在设计时就应该考虑到软件维护成本，而且在开发过程中应该使用标准的程序设计语言和标准的操作系统接口，可以大大提高软件的可维护性，也可以减少软件存在的错误；

对机票预订系统：往往人一多系统瘫痪几率就大，在开发人员不在场的情况下，很容易出现系统错误，维护软件也是很困难的，也会给很多人带来不便，所以，在设计过程中应该严格科学的管理规划还有合理设计模块，是各个模块的独立性越高，这样对软件的改进越方便，也便于快速纠错；

对患者监护系统：应该要考虑它的完善性和预防性，要能满足用户在使用过程中的增加和修改工作，还要为了改善未来的可维护性或可靠性而修改软件。更要考虑系统数据的隐秘及安全，随时备份。

第13章

1. 研究本书2.4.2小杰所述的订货系统，要求：
（1）用代码行技术估算本系统的规模；
（2）用功能点技术估算本系统的规模；
（3）用静态单变量模型估算开发本系统所需的工作量；
（4）假设由一个人开发本系统，试制定进度计划；
（5）假设由两个人开发本系统，试制定进度计划；（不会做）
2. 研究本书习题2第2题中描述的储蓄系统，要求：
（1）用代码行技术估算本系统的规模；
（2）用功能点技术估算本系统的规模；
（3）用静态单变量模型估算开发本系统所需的工作量；
（4）假设由一个人开发本系统，试制定进度计划；
（5）假设由两个人开发本系统，试制定进度计划；（不会做）

3.下面叙述对一个计算机辅助设计（CAD）软件的需求：该 CAD 软件接受由工程师提供的二维或三维几何图形数据。工程师通过用户界面与 CAD 系统交互并控制它，该用户界面应该表现出良好的人机界面设计特征。几何图形数据及其他支持信息都保存在一个 CAD 数据库中。开发必要的分析、设计模块，以产生所需要的设计结果，这些输出将显示在各种不同的图形设备上。应该适当地设计软件，以便与外部设备交互并控制它们。所用的外部设备包括鼠标、数字化扫描仪和激光打印机。要求：
（1）进一步精化上述要求，把 CAD 软件的功能分解成若干个子功能；
（2）用代码行技术估算每个子功能的规模；
（3）用功能点技术估算每个子功能的规模；
（4）从历史数据得知，开发这类系统的平均生产率是 620LOC/PM，如果软件工程师的平均月工资是 8000 元，请估算开发本系统的工作量和成本。
（5）如果从历史数据得知，开发这类系统的平均生产率是 6.5FP/PM，请估算开发本系统的工作量和成本

答：（1）习题中仅对需求做出了粗略描述，每项都都应该进一步扩展，以提供细节需求和定量约束。例如，在开始估算软件规模之前，需要确定“良好的人机界面设计特征”的具体含义，以及对“CAD 数据库”的规模和复杂度的具体需求。

经过对需求的进一步精化，分解出软件的下述 7 个主要的子功能：• 用户界面及控制机制；• 二维几何图形分析；• 三维几何图形分析；• 数据库管理；

• 计算机图形显示机制；• 外部设备控制；• 设计分析模块。

（2）为了用代码行技术估算软件规模，应该针对每个子功能都分别估计出下述 3 个值：乐观值（即最小规模 a），悲观值（即最大规模 b）和可能值（即最可能规模 m）。分别算出这 3 种规模的平均值，然后用下式的加权平均法计算每个子功能规模，结果示于表 10.4

表 10.4 代码行技术的估算表

功能	乐观值	可能值	悲观值	估计值
用户界面及控制机制	1500	2200	3500	2300
二维几何图形分析	3800	5400	6400	5300
三维几何图形分析	4600	6900	8600	6800
数据库管理	1850	3200	5450	3350
计算机图形显示机制	3100	4900	7000	4950
外部设备控制	1400	2150	2600	2100
设计分析模块	6200	8500	10200	8400
估算出的总代码行数				33200

（3）使用功能点技术估算软件规模时,对软件的分解是基于信息域特性而不是基于软件功能。表 10.5 给出了对 5 个信息域特征的估计值。为了计算未调整的功能点数，假设每个信息域都是平均级的。

接下来估计 14 个技术复杂性因素的值，并且计算 DI 的值，表 10.6 列出了得到的结果。

表 10.5 估算调整的功能点数

功能	乐观值	可能值	悲观值	估计值	特性系数	UFP 数
输入数	20	24	30	24	4	96
输出数	12	15	22	16	5	80
查询数	16	22	28	2	4	88
文件数	4	4	5	4	10	10
外部接口数	2	2	3	2	7	14
总计数值						38

表 10.6 估算复杂性因素

因 素	估计值	因 素	估计值
数据通信	2	复杂的计算	5
分布式数据处理	0	可重用性	4
性能标准	4	安装方便	3
高负荷硬件	2	操作方便	4
高处理率	4	可移植性	5
联机数据输入	4	可维护性	5
终端用户效率	4	DI	49
联机更新	3		

然后用下式计算技术复杂性因子：

TCF＝0.65＋0.01×DI

＝1.14

最后计算功能点数

FP＝UFP×TCP

＝31×1.14

＝363

(4) 用代码行估算，开发本系统的工件量为

E＝33200/620

≈54(人月)

开发本系统的成本为

8000×54＝432000（元）

(5) 用功能点技术估算，开发本系统的工作量为

E＝363/6.5

≈56(人月)

开发本系统的成本为

8000×56＝448000(元)

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。应选用哪些项目组结构？为什么？打算采用哪种（些）软件过程模型？为什么？

答：根据上述，应该主程序员组的项目组结构。因为项目小组已经开发过类似的系统，开发人员已经具备了一定的经验。这个时候开发过程遇到的难题不会很多，所以应该减少通信开销，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。　应该采用“已定义级”的软件过程模型。因为客户已经写出了完整的需求文档，而且项

3 / 3　　目小组已经有过类似的开发经验。软件过程已经文档化和标准化。这种过程模型是基于在软件机构中对已定义的过程模型的活动、人员和职责都有共同的理解。

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穿心性的产品，该产品把虚拟现实硬件和嘴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。应该选择哪种项目组结构?为什么?打算采用哪种软件过程模型?为什么?

答：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，项目组织结构以基于主程序员组的形式为宜。

针对待开发的系统，客户已经挟持了完整的需求文本，项目组又有开发类似系统的经验，因此，可采用广大软件工程师熟悉的瀑布型来开发本系统

6.假设自己被指派作为一个大型软件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。公司严格规定了严格的完成期限并且对外公布了,应该选择哪种项目组结构?为什么?打算采用哪种软件过程模型?为什么?

答：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。大型软件应该采用演化模型中的螺旋模型，

7.什么是软件质量？试叙述它与软件可靠性的关系。

答：软件质量是软件与明确地叙述的功能和性能需求、文档中明确描述的开发标准以及任何专业开发的软件产品都应该具有的隐含特征一致的程度。

8.一个程序能既正确又不可靠吗？解释一下自己的答案。

答：能。所谓软件可靠性,是程序在给定的时间间隔内按照规格说明书的规定成功地运行的概率.通常认为,软件可靠性既包含正确性又包含健壮性,也就是说,不仅在预定环境下程序应该能正确地完成预期功能,而且在硬件发生故障,输入的数据无效或用户操作错误等意外环境下,程序也应该能做出适当的响应.如果一个程序在预定环境下能够正确地完成预期的功能,但是在意外环境下不能做出适当的响应,则该程序就是既正确又不可靠

9. 仅当每个与会者都在事先作了准备时，正式的技术复审才能缺德预期的效果。如果自己是复审小组的组长，怎样发现事先没做准备的与会者？打算采取什么措施来促使大家事先做准备？

答： 软件复审包括了对需求文档、详细设计、数据库设计、功能设计、编码功能实现及质量、错误跟踪等的审查，以避免使用过程中出现更多的差错。反复审查是为了确保质量，保证不出现更多的错误和异常，软件复审就是以对质量保证为目的的。

对每个与会者提些软件配置复审因素问题：变更指令中指令的变更是否完成？每个附加变更是否已经纳入到系统中？是否进行了正式技术审核？是否遵循软件工程标准？变更的软件配置项是否作了特殊标记而得到强调？是否注明变更日期和变更执行人员？软件配置项属性是否反映了变更？是否遵循与变更有关的注释，记录及报告的软件配置管理规程？相关的软件配置项是否都得到了同步更新？等问题。

经常督促大家事先做准备，平时严格要求每次在准备开会前几天再次提醒每个与会者，开会时记录下每个与会者回答问题的准确性程度，根据此内容对与会者进行适当的赏罚。

10.什么是基线？为什么要建立基线？

答：是已经通过了正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变它。建立基线的三大原因是：重现性、可追踪性和报告。

重现性是指及时返回并重新生成软件系统给定发布版的能力，或者是在项目中的早些时候重新生成开发环境的能力。可追踪性建立项目工件之间的前后继承关系。其目的在于确保设计满足要求、代码实施设计以及用正确代码编译可执行文件。报告来源于一个基线内容同另一个基线内容的比较。基线比较有助于调试并生成发布说明。

建立基线后，需要标注所有组成构件和基线，以便能够对其进行识别和重新建立。

11.配置审计和技术复审有何不同？可否把它们的功能放在一次复审终完成？

答：正式的技术复审关注被修改后的配置对象的技术正确性。复审者审查该对象以确定它与其他软件配置项的一致性，并检查是否有遗漏或副作用，软件配置审计通过评估配置对象的那些通常不在复审过程中考虑的特征，而成为对正式技术复审的补充。

12.CMM 的基本思想是什么？为什么要把能力成熟度划分为 5 个等级？

答：CMM 的基本思想是，由于问题是由我们管理软件过程的方法不当引起的，所以软件技术的运用并不会自动提高软件的生产率和质量。把能力成熟度划分成 5 个等级的原因是：对软件的改进不可能一撅而就