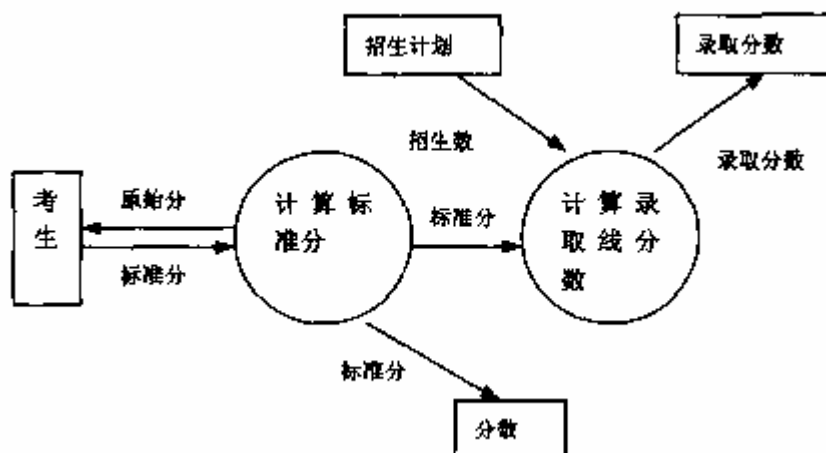


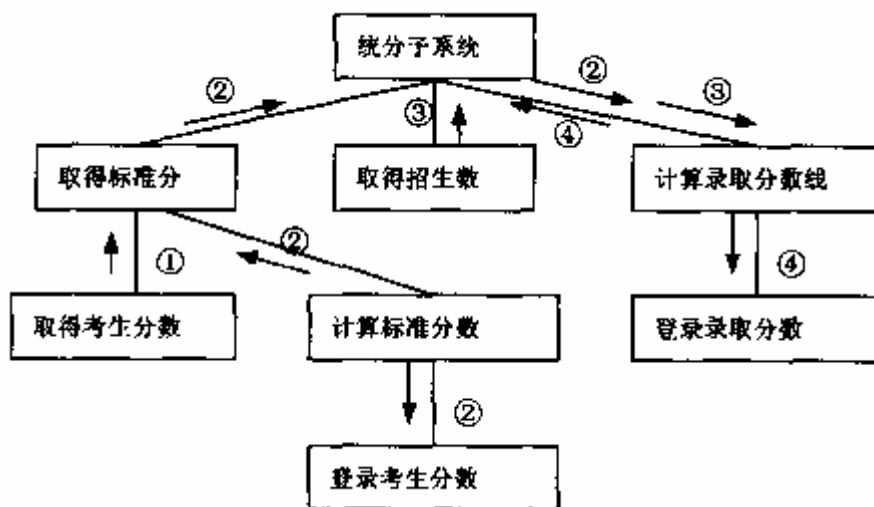
高校录取统分子系统有如下功能：

- (1)计算标准分：根据考生原始分计算，得到标准分，存入考生分数文件。
- (2)计算录取线分：根据标准分、招生计划文件中的招生人数，计算录取线，存入录取线文件。试根据要求画出该系统的数据流程图，并将其转换为软件结构图。

(1) 数据流图如图



(2) 软件结构图如图所示



①原始分。②标准分。③招生计划。④录取线分。

如果要开发“图书馆管理信息系统”，你认为采用什么开发模型比较合适？请说明理由。

答：开发“图书馆管理信息系统”采用瀑布模型比较合适。因为瀑布模型在开发过程中各阶段非常清晰，便于评审、跟踪、管理和控制，并且“图书馆管理信息系统”也基本符合瀑布模型的条件：

- (1) “图书馆管理信息系统”在开发过程中需求变化很少。
- (2) 分析和设计人员对“图书馆管理信息系统”比较熟悉。
- (3) “图书馆管理信息系统”是低风险项目。
- (4) “图书馆管理信息系统”的用户使用环境比较稳定。
- (5) “图书馆管理信息系统”的用户除提出需求以外，很少参与开发工作。

简述结构化程序设计方法的基本要点。

- (1)采用自顶向下，逐步求精的程序设计方法。
- (2)使用三种基本控制结构构造程序，分别是顺序，选择和循环
- (3)采用主程序员组的组织形式。

根据下列条件使用等价类划分法设计测试用例。

某一 8 位微机，其八进制常数定义为：以零开头的数是八进制整数，其值的范围是-177~177，如 05，0127，-065。

用等价划分法：

- (1) 划分等价类并编号，如表 1

表 1 八进制常量输入条件的等价类

输入条件	合理等价类	不合理等价类
八进制整数	1. 2—4 位以 0 打头的数字串 2. 以-0 打头的 3—5 位数字串	3. 以非 0 非-打头的串 4. 0 打头含有非数字字符的串 5. 以-0 打头含有非数字字符的串 6. 多于 5 个字符 7. -后非 0 的多位串 8. -后有非数字字符 9. 后多于 4 个数字
八进制数范围	10. 在-177—177 之间	11. 小于-177 12. 大于 177

- (2)为合理等价类设计测试用例。表 2 中有两个合理等价类，设计两个例子。

测试结果	期望结果	覆盖范围
026	显示有效输入	1, 10
-0145	显示有效输入	2, 10

- (3)为不合理等价类设计测试用例。表 3 中有两个合理等价类，设计相应例子。

测试结果	期望结果	覆盖范围
135	显示无效输入	3
0B45	显示无效输入	4
.....

为提高软件的可理解性，可采取哪些措施？

可理解性是指理解软件的结构、接口、功能和内部过程的难易程度。

提高软件可理解性的措施有：

采用模块化的程序结构；

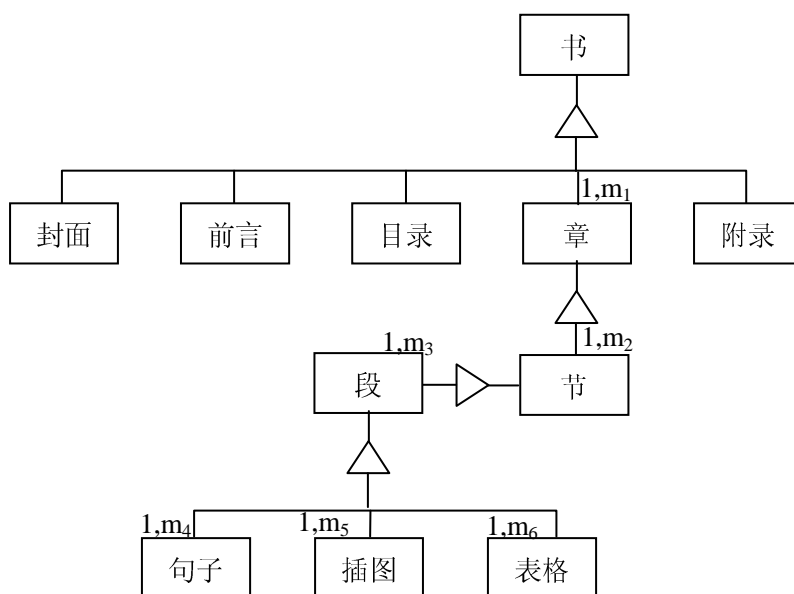
书写详细正确的文档；

采用结构化程序设计；

书写源程序的内部文档；
使用良好的编程语言；
具有良好的程序设计风格。

一本书的组成有一个封面，一个目录，一个前言，若干章，每章有若干节，每节有若干段，每段有若干句子，每节有 0 个或多个插图，还有 0 个或多个表格，最后有一个附录。
试建立该书的对象模型。

书的对象模型：



衡量模块独立的两个标准是什么？它们各表示什么含义？

衡量模块的独立性的标准是两个定性的度量标准：耦合性和内聚性。

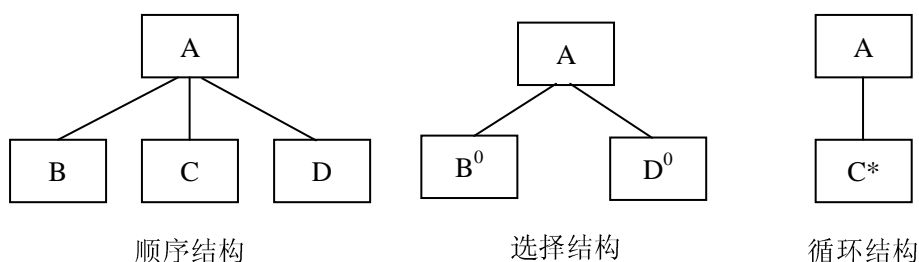
(1)耦合性。也称块间联系。指软件系统结构中各模块间相互联系紧密程度的一种度量。模块之间联系越紧密，其耦合性就越强，模块的独立性则越差。模块间耦合高低取决于模块间接口的复杂性、调用的方式及传递的信息。

(2)内聚性。又称块内联系。指模块的功能强度的度量，即一个模块内部各个元素彼此结合的紧密程度的度量。若一个模块内各元素(语句之间、程序段之间)联系得越紧密，则它的内聚性就越高。

耦合性与内聚性是模块独立性的两个定性标准，将软件系统划分模块时，尽量做到高内聚低耦合，提高模块的独立性，为设计高质量的软件结构奠定基础。

试画出 Jackson 方法所用的结构图的三种基本结构。

三种基本结构如图所示：



简述三种面向对象模型的主要功能。

对象模型：表示了静态的结构化的系统数据性质，描绘了系统的静态结构，从客观世界的对象关系角度来描绘对象。

动态模型：该模型描述了系统的控制结构，它表示了瞬间的、行为化的系统控制性质，它关心的是系统的控制及操作的执行顺序，它从对象的事件和状态的角度出发，表现了对对象的相互行为。

功能模型：描述了系统的所有计算。指出发生的时间、事件和客体。

等价分类法的基本思想是什么？

根据程序的输入特性，将程序的定义域划分为有限个等价区段——“等价类”，从等价类中选择出的用例具有“代表性”，即测试某个等价类的代表值就等价于对这一类其他值的测试。如果某个等价类的一个输入数据(代表值)测试中查出了错误，说明该类中其他测试用例也会有错误。

中级结构性成本模型 COCOMO 是一个()。

- A. 静态单变量模型
- B. 动态单变量模型
- C. 静态多变量模型
- D. 动态多变量模型

软件工程是()。

A、是结构化程序设计的指导方法
B、是软件开发技术和软件工程管理学为内容的学科

C、是指导计算机软件开发和维护的工程学科
D、是指导软件开发的工程方法。

在 McCall 软件质量度量模型中，()属于面向软件产品操作的质量因素。

- A. 可用性
- B. 可维护性
- C. 适应性
- D. 互操作性

原型方法是用户和设计者之间的一种交互过程，选用于()系统。

A、需求不确定性较高的
B、需求确定的
C、管理信息
D、决策支持

面向对象的开发方法中，()将是面向对象技术领域内占主导地位的标准建模语

- A. Booch
- B. Coad 方法
- C. UML 语言
- D. OMT 方法

瀑布模型本质上是一种(

- A. 线性顺序模型
- B. 顺序迭代模型
- C. 线性迭代模型
- D. 及早见产品模型

Jackson 结构化程序设计方法是英国人 M.A.Jackson 提出的，它是一种面向()的设计方法。

A、对象
B、数据流
C、数据结构
D、控制结构

在快速原型模型的开发过程中，仅作为需求说明的补充形式的原型是()

- A. 探索型模型
- B. 演化型模型
- C. 实验型模型
- D. 提交型模型

要将一个复杂的系统分析清楚，常用的方法是结构化分析方法，结构化分析方法就是()。

A、面向数据流自顶向下逐步求精的方法
B、由内向外进行分析的方法

C、先局部后整体的分析方法

D、使用 IPO 图形工具分析的方法

软件可维护性的特性中相互矛盾的是()。

A. 可修改性和可理解性

B. 可测试性和可理解性

C. 效率和可修改性

D. 可理解性和可读性

可行性研究是从()两个方面研究的。

A、经济可行性和单位领导态度

B、经济可行性和技术可行性

C、经济可行性和开发商

D、单位领导态度和开发商

软件维护产生的副作用,是指()。

A. 开发时的错误

B. 隐含的错误

C. 因修改软件而造成的错误

D. 运行时误操作

概要设计过程是()。

A、先确定系统的实现方案,然后在结构设计阶段中确定软件的模块结构

B、确定软件的模块结构,再设计出系统的所有程序和数据文件

C、设计出系统的 HIPO 图并对所有模块进行描述

D、规划出系统的后期设计总体结构

用黑盒技术设计测试用例的方法之一为()。

A. 因果图

B. 逻辑覆盖

C. 循环覆盖

D. 基本路径测试

设函数 $C(X)$ 定义问题 X 的复杂度,函数 $E(X)$ 确定解决问题需要的工作量,对于问题 $P1$ 和 $P2$,若 $C(P1) > C(P2)$ 则 $E(P1) > E(P2)$, $C(P1+P2) > C(P1) + C(P2)$ 从而 $E(P1+P2) > E(P1) + E(P2)$,是说明了()。

A、问题 $P1$ 、 $P2$ 越复杂所需的工作量越大

B、独立解决问题 $P1$ 、 $P2$ 所需用工作量比合并 $P1$ 、 $P2$ 解决 所需用工作量少

C、问题的复杂程度和解决问题时所需的处理工作量有函数关系

D、通过数学推导找出解决问题的最小工作量表达式

软件测试方法中的()属于静态测试方法。

A. 黑盒法

B. 路径覆盖

C. 错误推测

D. 人工检测

Ada 语言是在 Pascal 语言基础上开发出来的,适用于()项目的应用领域。

A. 科学工程计算

B. 数据库应用

C. 实时处理

D. 人工智能

程序的三种基本结构是()。

A、过程,子程序,分程序

B、顺序,条件,循环

C、递归,堆栈,队列

D、调用,返回,转移

程序设计语言的工程特性之一为()。

A. 软件的可重用性

B. 数据结构的描述性

C. 抽象类型的描述性

D. 数据库的易操作性

结构化程序设计的一种基本方法是()。

A、筛选法

B、递归法

C、归纳法

D、逐步求精法

软件详细设计的主要任务是确定每个模块的()。

A. 算法和使用的数据结构

B. 外部接口

C. 功能 D. 编程

使用 Jackson 方法可以实现 ()。

A、从数据结构导出程序结构 B、从数据流图导出初始结构图

C、从模块结构导出数据结构 D、从模块结构导出程序结构

PAD 图为()提供了有力的上具。

A. 系统分析 B. 软件的自动化生成

C. 自动分析数据 D. 测试软件

软件测试的目的是 ()。

A、证明程序中没有错误 B、发现程序中的错误

C、测量程序的动态特性 D、检查程序中的语法错误

软件结构图的形态特征能反映程序重用率的是()

A. 深度 B. 宽度

C. 扇入 D. 扇出

使用黑盒测试时, 测试用例应根据 ()。

A、程序的内部逻辑 B、程序的复杂程度 C、使用说明书 D、程序的功能

在大型数据处理系统的功能分析与设计中, 数据库的概念设计对应于系统开发的 ()。

A. 需求分析 B. 概要设计

C. 详细设计 D. 程序设计

测试的分析方法是通过分析程序 () 来设计测试用例的方法。

A、应用范围 B、内部逻辑 C、功能 D、输入数据

为了提高模块的独立性, 模块内部最好是 ()。

A. 逻辑内聚 B. 时间内聚

C. 功能内聚 D. 通信内聚

白盒法测试程序时常按照给定的覆盖条件选取测试用例, () 即是判定覆盖, 又是条件把覆盖, 它并不保证各种条件都能取得所有可能的值。

A、语句 B、判定 C、条件 D、判定/条件

软件结构图中, 模块框之间若有直线连接, 表示它们之间存在 ()。

A. 调用关系 B. 组成关系

C. 链接关系 D. 顺序执行关系

软件可移植性是用来衡量软件的 () 的重要尺度之一。

A、通用性 B、效率 C、质量 D、人机关系

需求分析最终结果是产生()

A. 项目开发计划 B. 需求规格说明书

C. 设计说明书 D. 可行性分析报告

可行性研究要进行一次()

A. 深入的 B. 详尽的

C. 彻底的 D. 简化的、压缩了的

尽管有不合法的输入, 软件仍能继续正常工作的能力称 ()。

A、可理解性 B、健壮性 C、可用性 D、兼容性

软件是一种()

- A. 程序
- B. 数据
- C. 逻辑产品
- D. 逻辑产品

信息模型是描述（ ）。

- A、对象及其相互之间的关系
- B、数据结构
- C、属性和服务
- D、对象的状态

软件生存周期模型不包括()

- A. 瀑布模型
- B. 对象模型
- C. 增量模型
- D. 喷泉模型

面向对象的分析就是用（ ）描述对象及其相互关系。

- A、信息模型、状态模型
- B、状态模型、处理模型
- C、信息模型、状态模型、处理模型
- D、信息模型、状态模型、数据模型

软件可行性研究实质上是要进行一次()需求分析

- A. 简化、压缩的
- B. 详细的
- C. 彻底的
- D. 深入的

面向对象分析的独到之处是（ ）。

- A、将数据和处理结合在一起而不是分离
- B、数据和处理分别独立考虑
- C、仅考虑数据结构
- D、仅考虑处理过程

需求分析阶段最重要的技术文档之一是()。

- A. 项目开发计划
- B. 设计说明书
- C. 需求规格说明书
- D. 可行性分析报告

通常所说的结构化设计即 SD 是属于面向（ ）的设计方法。

- A、数据结构
- B、数据流
- C、对象
- D、以上均可

在以下模块耦合性类型中，模块之间独立性最差的类型是()

- A. 无直接耦合
- B. 标记耦合
- C. 控制耦合
- D. 内容耦合

重用率高的模块在软件结构图中的特征是()。

- A. 扇出数大
- B. 扇入数大
- C. 内聚性高
- D. 扇出数小

软件详细设计的主要任务是对()进行的详细设计

- A. 算法和数据结构
- B. 外部接口
- C. 软件编码
- D. 功能

Jackson 方法是一种面向()的方法。

- A. 数据流
- B. 对象
- C. 数据结构
- D. 控制流

第一个体现结构化编程思想的程序设计语言

- A. Fortran
- B. C
- C. Pascal
- D. Cobol

一个成功的测试可以()

- A. 表明程序没有错误
- B. 发现所有错误
- C. 提高软件可靠性
- D. 发现尚未发现的错误

黑盒技术设计测试用例的方法之一为()。

- A. 因果图
- B. 逻辑覆盖
- C. 循环覆盖
- D. 基本路径测试

在软件维护的内容中, 占维护活动工作量比例最高的是()

- A. 改正性维护
- B. 适应性维护
- C. 完善性维护
- D. 预防性维护

面向对象程序设计语言不同于其他语言的最主要特点是()。

- A. 模块性
- B. 抽象性
- C. 共享性
- D. 继承性

软件维护的副作用

- A. 开发时的错误
- B. 隐含的错误
- C. 因修改软件而造成的错误
- D. 运行时误操作

增量模型本质上是一种()

- A. 线性顺序模型
- B. 整体开发模型
- C. 非整体开发模型
- D. 快速原型模型

在快速原型的开发过程中, 用于及早向用户提交原型系统的原形模型是

- A. 探索型原型
- B. 实验型原型
- C. 演化型原型
- D. 增量构造原型

关联是建立()之间关系的一种手段。

- A. 类
- B. 对象
- C. 角色
- D. 属性

面向对象技术特别强调的是()

- A. 对象的数据结构
- B. 对象的使用方式
- C. 程序结构
- D. 结构化编程

对象模型由一个或若干()组成。

- A. 对象
- B. 模板
- C. 属性
- D. 数据流图

在 Mccall 质量度量模型中, 不属于面向软件产品修改的特性是()

- A. 可维护性
- B. 可测试性
- C. 可靠性
- D. 适应性

Putnam 成本估算模型是一种()

- A. 静态单变量模型
- B. 静态多变量模型
- C. 动态单变量模型
- D. 动态多变量模型

开发软件所需高成本和产品的低质量之间有着尖锐的矛盾, 这种现象称作()。

- A. 软件工程
- B. 软件周期
- C. 软件危机
- D. 软件产生

研究开发所需要的成本和资源是属于可行性研究中的()。

- A. 技术可行性
- B. 经济可行性
- C. 社会可行性
- D. 法律可行性

IDEFO 图反映不出系统()。

- A. 系统做什么
- B. 系统功能如何实现
- C. 系统由谁来做
- D. 系统实现的约束条件

模块的内聚性最高的是()。

- A. 逻辑内聚
- B. 时间内聚
- C. 偶然内聚
- D. 功能内聚

在 SD 方法中全面指导模块划分的最重要的原则是()。

- A. 程序模块化
- B. 模块高内聚
- C. 模块低耦合
- D. 模块独立性

软件详细设计主要采用的方法是()。

- A. 模块设计
- B. 结构化设计
- C. PDL 语言
- D. 结构化程序设计

下列关于 JSP 方法不正确的说法是()。

- A. JSP 方法主要用于规模不大的数据处理系统
- B. JSP 方法不明确的划分软件概要设计和详细设计的两个阶段
- C. JSP 方法适用于输入数据和输出数据之间有对应关系的问题求解
- D. JSP 方法根据输入、输出的数据结构，按一定的规则映射成软件的体系结构，因此它只适用于详细设计阶段。

不适合作为科学与工程计算的语言是()。

- A. Pascal
- B. C
- C. Fortran
- D. Prolog

黑盒测试在设计测试用例时，主要需要研究()。

- A. 需求规格说明与概要设计说明
- B. 详细设计说明
- C. 项目开发计划
- D. 概要设计说明与详细设计说明

若有一个计算类型的程序，它的输入量只有一个 x ，其范围是 $[-1.0, 1.0]$ ，输入的角度考虑一组测试用例：-1.001，-1.0，1.0，1.001。设计这组测试用例的方法是()。

- A. 条件覆盖法
- B. 等价分类法
- C. 边界值分析法
- D. 错误推测法

下列属于维护阶段的文档是()。

- A. 软件规格说明
- B. 用户操作手册
- C. 软件问题报告
- D. 软件测试分析报告

快速原型模型的主要特点之一是()。

- A. 开发完毕才见到产品
- B. 及早提供全部完整的软件产品
- C. 开发完毕后才见到工作软件
- D. 及早提供工作软件

因计算机硬件和软件环境的变化而作出的修改软件的过程称()。

- A. 校正性维护
- B. 适应性维护
- C. 完善性维护
- D. 预防性维护

类库这种机制是()级别的信息共享。

- A. 同一类
- B. 不同类
- C. 同一应用
- D. 不同应用

只有单重继承的类层次结构是()层次结构。

- A. 网状型
- B. 星型
- C. 树型
- D. 环型

在屏蔽硬件错误的冗错技术中, 冗余附加技术有()。

- A. 三模冗余
- B. 信息冗余
- C. 冗余备份程序的存储及调用
- D. 关键程序和数据冗余存储和调用

软件按照设计的要求, 在规定时间内和条件下达到不出故障, 持续运行的要求的质量特性称为()

- A. 可用性
- B. 可靠性
- C. 正确性
- D. 完整性

下列文档与维护人员有关的有()。

- A. 软件需求说明书
- B. 项目开发计划
- C. 概要设计说明书
- D. 操作手册

采用 Gann 图表示软件项目进度安排, 下列说法中正确的 i

- A. 能够反映多个任务之间的复杂关系
- B. 能够直观表示任务之间相互依赖制约关系
- C. 能够表示哪些任务是关键任务
- D. 能够表示子任务之间的并行和串行关系

下列模型属于成本估算方法的有()。

- A. COCOMO 模型
- B. McCall 模型
- C. McCabe 度量法
- D. 时间估算法

语句覆盖与条件覆盖能否做到 100%? 为什么?

语句覆盖是指选择足够的测试用例, 使得运行这些测试用例时, 被测程序的每个可执行语句都至少执行一次。

条件覆盖是指选择足够的测试用例, 使得运行这些测试用例时, 被测程序的每个判定中的每个条件的所有可能结果都至少出现一次。

什么是测试用例? 动态测试有哪些方法?

所谓“测试用例”是指为寻找程序中的错误而精心设计的一组测试数据, 每个测试用例一般是一个二元组(输入数据、预期结果)。动态测试主要有两种方法: 白盒法和黑盒法。白盒法是从被测试程序的内部逻辑入手来设计测试用例; 黑盒法着重测试被测试程序的功能, 而不关心内部如何实现其功能的结构, 是从用户观点出发的测试。但无论白盒法还是黑盒法都只能选择一些有代表性的测试用例进行有限的测试。

(1)白盒法测试: 有六种用于有判定存在的逻辑覆盖标准, 即程序内部路径的覆盖程度。对于有循环存在的覆盖用限制循环次数的办法来测试。但是对于一个实际问题的程序测试中, 其路径是一个庞大的数字, 基本路径测试告诉了最少路径计算办法, 在分析程序控制流程图环路复杂性基础上, 导出基本路径集合, 从而设计测试用例, 保证这些基本路径至少执行一次, 以最少的用例发现尽量多的错误。

(2)黑盒法测试: 具体方法有四种, 采用黑盒法的综合策略是先等值类划分法(包括边界值分析法, 即取边值上的数)设计出测试用例, 然后用错误推测法补充。如果被测试程序含有多个条件的逻辑组合, 则开始就用因果图法。

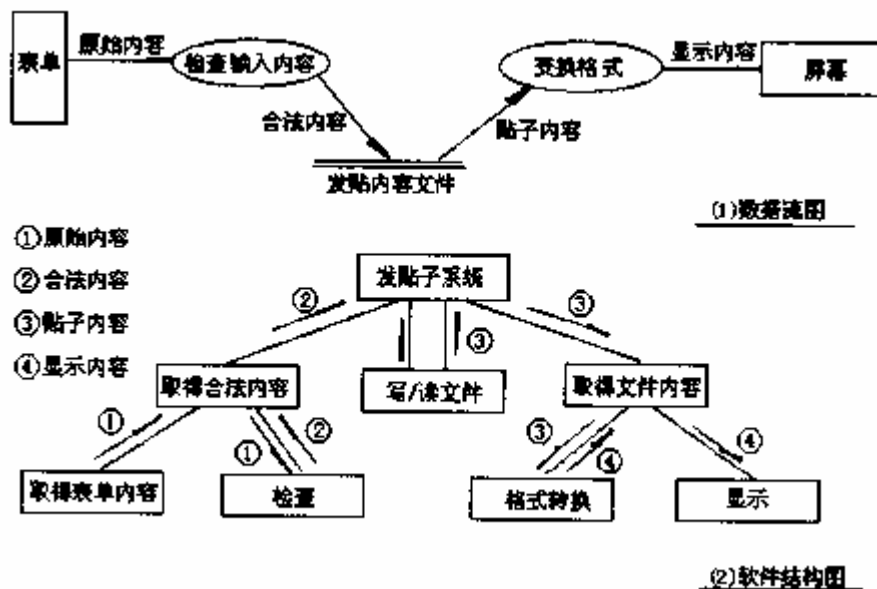
某 BBS 模板的发帖子系统有如下功能:

(1) 记录发帖内容：访客在表单中输入文字，系统进行检查。

(2)显示帖子内容：读出文件，按一定格式显示在屏幕上。

请根据要求画出该系统的数据流程图，并将其转换为软件结构图。

解答参考如图所示。



假设你被任命为一家软件公司的项目负责人，你的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已对外公布。你打算采用哪种软件生命周期模型？为什么？

这个项目是开发该公司已被广泛应用的字处理软件的新版本，从上述事实至少可以得出3点结论：第一，旧版本相当于一个原型，通过收集用户对旧版本的反映，较容易确定对新版本的需求，没必要再专门建立一个原型系统来分析用户的需求；第二，该公司的软件工程师对字处理软件很熟悉，有开发字处理软件的丰富经验，具有采用增量模型开发新版字处理软件所需要的技术水平；第三，该软件受到广大用户的喜爱，今后很可能还要开发更新的版本，因此，应该把该软件的体系结构设计成开放式的，以利于今后的改进和扩充。

综上所述，采用增量模型来完成这个项目比较恰当。

如何理解“规格说明必须容忍不完整和可扩充”。

我们把描述需求的文档叫做软件需求规格说明书，同时为确切表达用户对软件的输入输出要求，还需要制定数据要求说明书及编写初步的用户手册。

需求规格说明书是用户和开发者之间的一个协约。

实际的需求获取过程中，存在若干风险，包括用户参与不足，需求不断增加，需求模棱两可等等，用户由于不明白需求分析的重要性，有时只做一份很简略的规格说明，仅涉及到产品概念上的内容，而希望开发人员在开发过程中去完善，这种情况下的规格说明势必不够完整，需要开发过程中双方不断完善和补充。

另一方面，由于在需求阶段，很多需求细节客户自己未必清楚，因此这一阶段也无法达成一个一成不变的规格说明，必须允许客户在开发过程中不断明确自己的需求，这就要容忍规格说明不完整和可扩充。

用 PAD 图描述下列问题的控制结构：有一个表 A(1)、A(2)、…、A(N)按递增顺序排列。给定一个 key 值，在表中用折半法查找，若找到，将表位置 i 送入 x，否则将零送到 x，同时将 key 值插入表中。

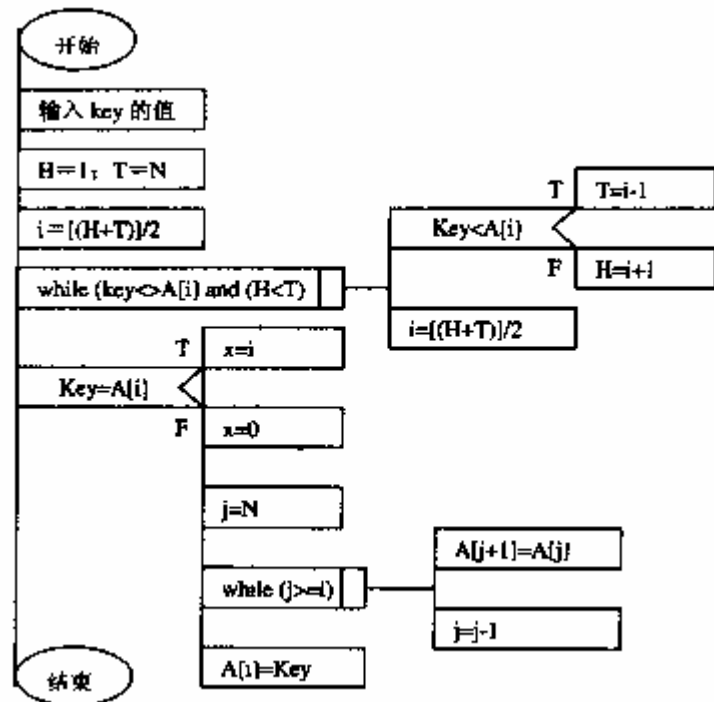
算法为：

(1)置初值 H=1(表头)，T=N(表尾)。

(2) 置 $i = [(H + T)/2]$ (取整)。

(3)若 $key = A(i)$ ，则找到，i 送到 x；若 $key > A(i)$ ，则 key 在表的后半部分，i + 1 送入 H；若 $key < A(i)$ ，则 key 在表的前半部分，i - 1 送入 T，重复第 2 步查找直到 $H > T$ 为止。

(4)查不到时，将 A(i)……A(N)移到 A(i+1)……A(N+1)，key 值送入 A(i)中 PAD 如图所示。



什么是模块的影响范围？什么是软件的控制范围？它们之间应建立什么关系？

模块的影响范围（作用域）：受该模块内一个判定影响的所有模块的集合

模块的控制范围（控制域）：这个模块本身以及所有直接或间接从属于它的模块的集合

一个模块的影响范围（作用域），应处在这个模块的模块的控制范围（控制域）内。

某旅行社根据旅游淡季、旺季及是否团体订票，确定旅游票的折扣率。

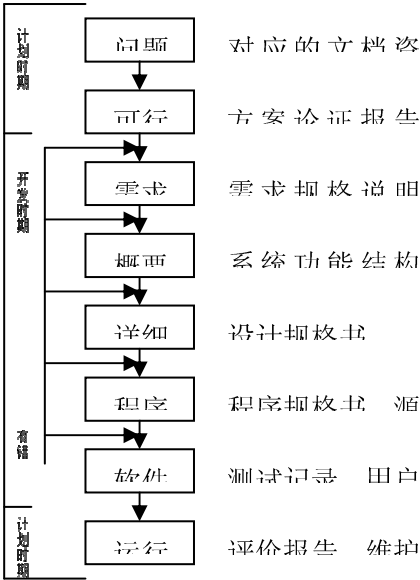
具体规定如下：人数在 20 人以上的属团体，20 人以下的是散客。每年的 4~5 月、7~8 月、10 月为旅游旺季，其余为旅游淡季。旅游旺季，团体票优惠 5%，散客不优惠。旅游淡季，团体票优惠 30%，散客优惠 20%。试用判定表表示旅游订票的优惠规定。

旅游旺季↙	T	T	F	F↘
团体↙	T	F	T	F↘
优惠 5%↙	X↘			
不优惠↙		X↘		
优惠 30%↙			X↘	
优惠 20%↙				X↘

某航空公司根据节假日及是否团体订票，确定机票的折扣率。具体规定如下：人数在 10 人以上的属团体，10 人以下的是散客。每年的五一、十一和春节期间旅客团体票上调 5%，散客上调 10%。其他时间团体票下调 30%，散客下调 20%。试用判定表表示旅游订票的优惠规定。

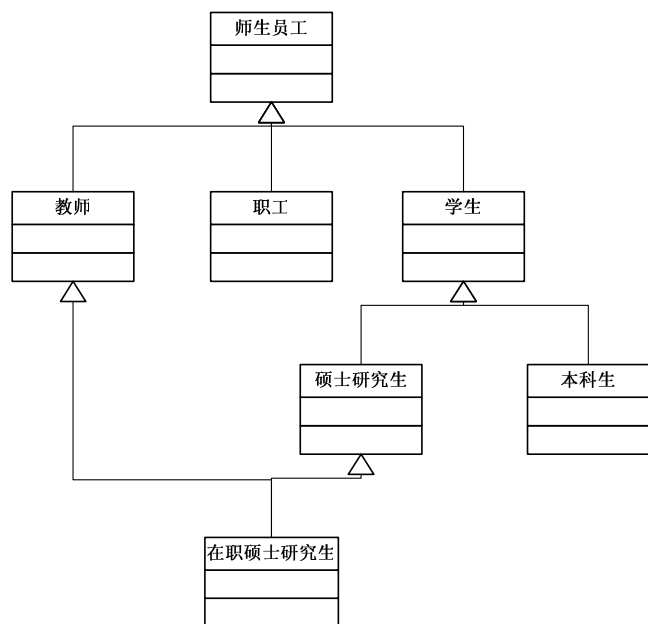
节假日	T	T	F	F
团体	T	F	T	F
上调 5%	X			
上调 10%		X		
下调 30%			X	
下调 20%				X

画出瀑布模型，并指出各阶段产生的文档

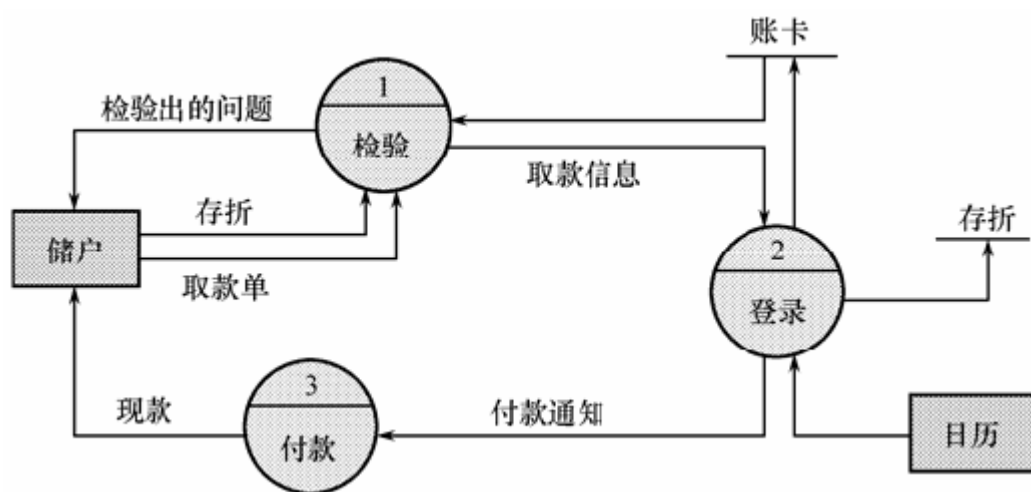


面向对象方法。

某教务处理系统涉及到以下几个类：师生员工，教师，职工，学生，本科生，硕士研究生，在职研究生（本校的某些教师可攻读硕士学位）。试用类图画出这些类之间的关系。



画出银行取款过程的 DFD 图。问题描述为：储户用存折取款，首先填写取款单，根据“账卡”中的信息检验取款单与存折，如有问题，将问题反馈给储户，否则，登录“储户存款数据库”，修改相应数据，并更新“账卡”，同时发出付款通知，出纳向储户付款。



请采用等价分类法（黑盒法）对程序 A 构造最少的测试用例。

[程序 A 的说明]

输入三角形的三条边（以整数表示），判断该三条边是否构成直角三角形，等边三角形，等腰三角形以及三角形。

假设输入的三条边长度是 a b c 。并且 $a \leq b \leq c$

输入条件	有效等价类	无效等价类
为三角形	$b+a>c$ and $b-a<c$	$b+a\leq c$ or $b-a\geq c$
为直角三角形	三角形前提下 $a^2+b^2=c^2$	三角形前提下 $a^2+b^2\neq c^2$
等腰三角形	三角形前提下 $a=b$ or $a=c$	三角形前提下 $a\neq b$ and

		$b < c$
等边三角形	三角形前提下 $a=b=c$	三角形前提下 $a \ b \ c$ 不全相等

测试用例：

$a=2 \ b=4 \ c=5$ 三角形

$a=1 \ b=4 \ c=5$ 非三角形

$a=3 \ b=4 \ c=5$ 直角三角形，非等腰，非等边

$a=3 \ b=4 \ c=4$ 等腰三角形，非直角，非等边

$a=3 \ b=3 \ c=3$ 等边三角形，等腰，等边

某旅馆的电话服务如下：

可以拨分机号和外线号码。分机号是从 7201 至 7299。外线号码先拨 9，然后是市话号码或长话号码。长话号码是以区号和市话号码组成。区号是从 100 到 300 中任意的数字串。市话号码是以局号和分局号组成。局号可以是 455，466，888，552 中任意一个号码。分局号是任意长度为 4 的数字串。

要求：写出在数据字典中，电话号码的数据条目的定义(即组成)。

电话号码=分机号 | 外线号码

分机号=7201...7299

外线号码=9+ [市话号码 | 长话号码]

长话号码=区号+市话号码

区号=100...300

市话号码=局号+分局号

局号= [455 | 466 | 888 | 552]

分局号=4 {数字} 4

逻辑内聚（教材第 4 章设计工程 4-7 页的 4.2.3.1 小节）

通过模块把几种相关的功能组合在一起，每次调用时，由传送给模块的控制型参数来确定该模块应执行哪一种功能。

逻辑内聚模块比巧合内聚模块的内聚程度要高，因为它表明了各部分之间在功能上的相关关系。

可移植性（PPT 中：SE13 编码部分 / SE15 维护部分）

可移植性是指程序从某一环境移植到另一环境的能力以及难易程度。

影响软件可移植性的因素有：信息隐蔽原则、模块独立、模块化、高内聚低耦合、良好的程序结构、不用标准文本以外的语句等。

面向对象（教材第 7 章面向对象的分析与设计首页及次页）

采用对象、分类、继承、通过消息的通信这四个基本概念开发的软件系统称为面向对象的。

成本估算模型中，Putnam 估算模型是一种_____模型。

功能模型描述子系统的所有计算，它由_____组成。

在软件开发阶段用来减少错误，提高软件可维护性的技术称为_____的技术

测试是一种挑剔性行为，_____是测试自己程序的障碍。

程序设计语言的心理特性所表现的简洁性是指程序员必须记住的_____的数量。

JSP 方法主要体现程序结构的设计，不明确地划分_____和详细设计的两个阶段。

按数据流的类型、结构化设计方法有两种设计策略，它们是变换分析设计和_____。

数据流图仅反映系统必须完成的逻辑功能，所以它是一种_____模型。

社会可行性所涉及的范围包括合同、责任、侵权、_____，其他一些技术人员常常不了解的陷阱等。

软件开发方法是一种使用早已定义好的技术集及_____来组织软件生产的过程。

软件工程研究的主要内容是软件开发技术和_____两个方面。

经济可行性研究范围包括_____、公司经营长期策略、开发所需的成本和资源、潜在的市场前景。

数据流图和数据字典共同构成了系统的_____模型，是需求规格说明书的主要组成部分。

结构化设计方法中，要把数据流图转换成软件结构，若某个加工将它的输入流分离成许多发散的数据流，形成许多加工路径，并根据输入的值选择其中一个路径来执行，这种特征的帅称为_____的数据流图。

人的记忆特性有两方面：联想方式和_____。

调试的目的是确定错误的原因和位置并改正错误，因此调试应该由_____进行。

面向对象设计中存在的三种内聚为操作内聚、_____和一般、具体内聚。

实现容错技术的主要手段是冗余，通常冗余技术包括结构冗余、_____、时间冗余和冗余附加技术。

软件工具通常由工具、_____和用户工具三部分组成。

程序设计工作台由支持程序开发过程的_____组成。

瀑布模型是以文档为驱动，适合于_____的软件项目的模型。

纯收入是指在整个生存周期之内的_____与投资之差。

数据项是指_____数据单元。

软件概要设计的主要任务就是_____。

结构化程序设计方法是使用_____构造程序。

软件开发是一个自顶向下逐步细化和求精过程，而软件测试是一个_____集成的过程。

软件维护的内容包括校正性维持，适应性维护、_____和预防性维护。

CASE 工具的平台集成指工具运行在相同的_____平台上。

在建立对象的功能模型时，使用的数据流图中包含有处理、数据流、和_____。

将开发项目与一个或多个已完成的类似项目进行比较。找出与某个相类似项目的若干不同之处，并估算每个不同之处对成本的影响，得出开发项目的总成本。这种成本估算方法称为_____。

某厂对部分职工重新分配工作的政策是：年龄在 20 岁以下者，初中文化程度脱产学习，高中文化程度当电工。20 岁至 40 岁之间，中学文化程度，男性当钳工，女性当车工，大学文化程度都当技术员。年龄在 40 岁以上者，中学文化程度当材料员，大学文化程度当技术

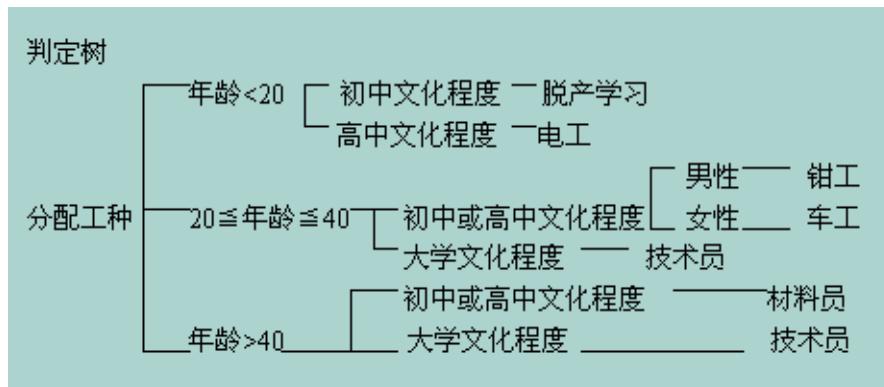
员。请用结构化语言、判定表或判定树描述上述问题的加工逻辑。

结构化语言：

CASE 年龄<20
CASE 文化程度 初中
分配工种 脱产学习
CASE 文化程度 高中
分配工种 电工
ENDCASE
CASE 年龄<=40.AND.>=20
CASE 文化程度 初中.OR.高中
CASE 性别 男
分配工种 钳工
CASE 性别 女
分配工种 车工
ENDCASE
ENDCASE
CASE 文化程度 大学
分配工种 技术员
ENDCASE
CASE 年龄>40
CASE 文化程度 初中.OR.高中
分配工种 材料员
CASE 文化程度 大学
分配工种 技术员
ENDCASE
ENDCASE

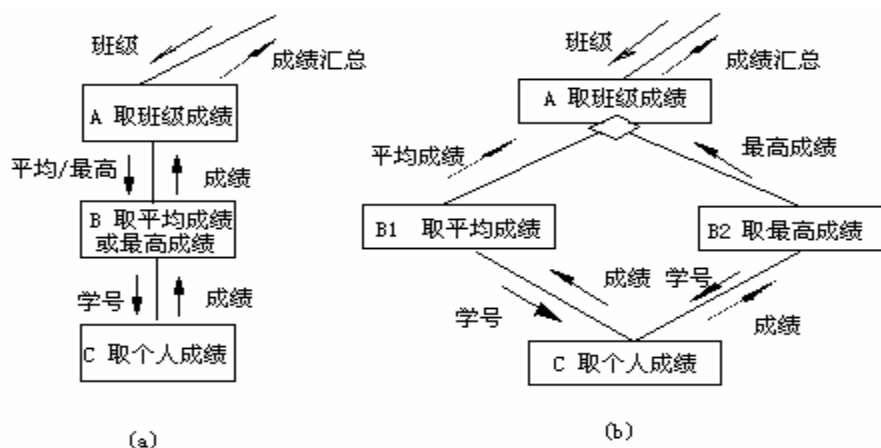
取值表			
条件名	取值	符号	取值数
年龄	年龄<20	C	M 1.=3
	20≤年龄≤40	Y	
	年龄>40	L	
性别	男	M	M 2.=2
	女	F	
文化程度	初中	J	M 3.=3
	高中	S	
	大学	U	

判定表																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
年龄	C	C	C	C	C	C	Y	Y	Y	Y	Y	Y	L	L	L	L	L	L
性别	M	M	M	F	F	F	M	M	M	F	F	F	M	M	M	F	F	F
文化程度	J	S	U	J	S	U	J	S	U	J	S	U	J	S	U	J	S	U
脱产学习	✓		/	✓		/												
电工		✓	/		✓	/												
钳工			/			/	✓	✓										
车工			/			/				✓	✓							
技术员			/			/			✓			✓			✓			✓
材料员			/			/							✓	✓		✓	✓	



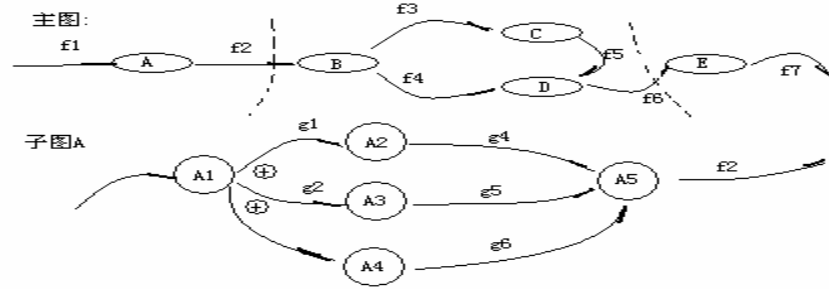
下图是某系学籍管理的一部分，(a),(b)分别是同一模块 A 的两个不同设计方案，你认为哪一个设计方案较好?请陈述理由。

答：我认为(b)方案较好。要评价一个软件结构设计，主要看模块的独立性，要从软件结构的耦合性和内聚性两个方面来衡量。对于(a)方案，模块 A 与模块 B 之间形成了控制耦合，因此独立性不高，而在 B 模块内部，有两个不同的功能被放在一起，形成的是逻辑内聚，内聚性不高。对于(b)方案，模块 A 与 B1 和 B2 之间是数据耦合，独立性较强，在 B1 和 B2 内部，只完成单一的功能，是功能内聚，因此内聚性高，可维护性好，因此相比而言 (b)方案比(a)方案更好些。

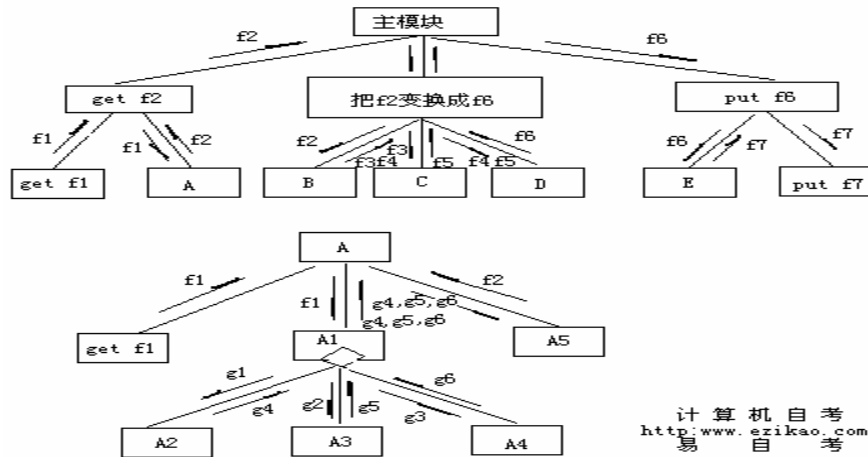


的 DFD 转换为软件结构图。

(注：图中⊕表示“或者”)



解：我们可以看出主图是变换型DFD，而子图是事务型DFD。
根据结构设计的方法，得到的结构图如下：



计算机自考
http://www.ezikao.com
易 考

某旅馆的电话服务如下：

可以拨分机号和外线号码。分机号是从7201至7299。外线号码先拨9，然后是市话号码或长话号码。长话号码是以区号和市话号码组成。区号是从100到300中任意的数字串。市话号码是以局号和分局号组成。局号可以是455，466，888，552中任意一个号码。分局号是任意长度为4的数字串。

要求：写出在数据字典中，电话号码的数据条目的定义(即组成)。

答：

电话号码=分机号 | 外线号码

分机号=7201...7299

外线号码=9+ [市话号码 | 长话号码]

长话号码=区号+市话号码

区号=100...300

市话号码=局号+分局号

局号= [455 | 466 | 888 | 552]

分局号=4{数字}4

附加题：

1、对下列子程序进行调试：

```
procedure example(y,z: real; var x: real)
```

```
begin
```

```
    if (y>1) and (z=0) then x:=x/y;
```

```
    if (y=2) or (x=1) then x:=x+1;
```

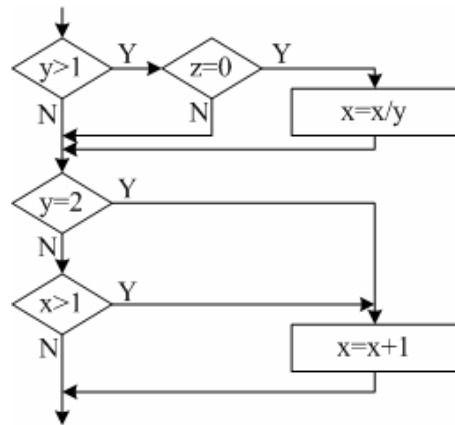
end.

该子程序接受x, y, z的值，并将计算结果x的值返回给调用程序。答题要求：

(1)画出流程图。

(2)用白盒法设计测试用例，并写出它所满足的是何种覆盖标准、预期结果以及路径。

答：（1）流程图



(2) 用白盒法中条件组合覆盖设计测试用例：

- ① y=2, z=0, x=4;
- ② y=2, z=1, x=1;
- ③ y=1, z=0, x=2;
- ④ y=1, z=1, x=1.

2、某报表处理系统要求用户输入处理报表的日期，日期限制在 2003 年 1 月至 2008 年 12 月，即系统只能对该段期间的报表进行处理，如日期不在此范围内，则显示输入错误信息。系统日期规定由年、月的 6 位数字字符组成，前四位代表年，后两位代表月。现要求用黑盒测试法中的边界值法写出测试用例。

答：

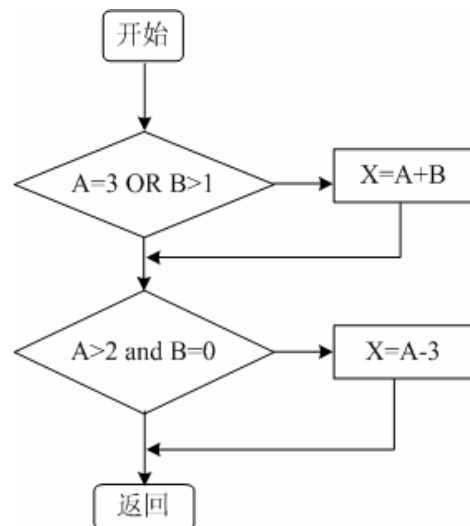
输入条件	测试用例说明	测试数据
报表日期的类型及长度	1个数字字符	5
	5个数字字符	20035
	7个数字字符	2003005
	有1个非数字字符	2003.5
	全部是非数字字符	MAY---
	6个数字字符	200305
日期范围	在有效范围	200301
	边界上选取	200812
	数据	200300
		200813
月份范围	月份为1月	200301
	月份为12月	200312
	月份<1	200300
	月份>12	200313

3、对以下程序进行测试：

```
PROCEDURE EX (A, B: REAL; VAR X: REAL);  
  BEGIN  
    IF (A=3) OR (B>1) THEN X:=A×B  
    IF (A>2) AND (B=0) THEN X:=A-3  
  END
```

先画出程序流程图。再按不同逻辑覆盖法设计一种（写出名称）测试数据。

答：



语句覆盖 A=3 B=0

4、程序功能说明书指出，某程序的输入条件为：每个学生可以选修1至3门课程，试用黑盒法设计测试用例。

(1)按等价分类法设计测试用例（要求列出设计过程）

(2)按边缘值分析法设计测试用例。

答：

1) 合理等价类

选修1门

选修2门

选修3门

不合理等价类

没选课

选3门以上

测试用例

选修=2

选修=0

选修>3

2) 边缘值法

0, 1, 3, 4

5、设被测试的程序段为：

```

begin
    s1 ;
    if (x=0) and (y>2)
        then s2;
    if (x<1) or (y=1)
        then s3 ;
    s4 ;
end

```

可供选择的测试数据组为：

	x	y
I	0	3
II	1	2
III	-1	2
IV	3	1

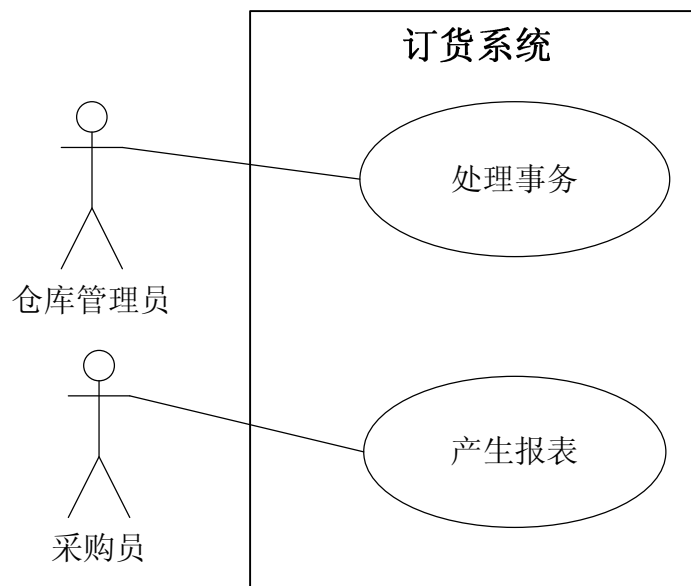
找出实现（1）语句覆盖；（2）条件覆盖；（3）判定覆盖至少要选择的数据组。

答：（1）I （2）I, IV （3）I, II

试建立下述订货系统的用例模型。

假设一家工厂的采购部每天需要一张订货报表，报表按零件编号排序，表中列出所有需要再次订货的零件。对于每个需要再次订货的零件应该列出下述数据：零件编号，零件名称，订货数量，目前价格，主要供应者，次要供应者。零件入库或出库称为事务，通过放在仓库中的终端把事务报告给订货系统。当某种零件的库存数量少于库存量临界值时就应该再次订货。

解析：从对这个订货系统的需求可以知道，仓库管理员通过放在仓库中的终端把零件入库/出库事务报告给订货系统，系统接收到事务之后应该处理事务；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。综上所述，可画出下图所示的用例图。



订货系统用例图

试用面向对象分析设计下述程序：

在显示器屏幕上圆心坐标为(100,100)的位置画一个半径为 40 的圆，在圆心坐标为(200,300)的位置画一个半径为 20 的圆，在圆心坐标为(400,150)的位置画一条弧，弧的起始角度为 30 度，结束角度为 120 度，半径为 50。

解析：面向对象方法模仿人类习惯的思维解题方法，用对象分解取代了功能分解，也就是把程序分解成一系列对象，每个对象都既有自己的数据，又有处理这些数据的方法。不同对象之间通过发送消息向对方提出服务要求，接受消息的对象主动完成指定功能提供所需要的服务。程序中所有对象分工协作，共同完成整个程序的功能。

从本题中给出的对这个简单图形程序的需求可以看出，这个程序中只涉及两类实体(即对象)，它们分别是圆和弧。

从需求陈述中不难看出，圆的基本属性是圆心坐标和半径，弧的基本属性是圆心坐标、半径、起始角度和结束角度。但是，通常不可能在需求陈述中找到所有属性，还必须借助于领域知识和常识，才能分析得出所需要的全部属性。众所周知，一个图形既可以在屏幕上显示出来，也可以不显示出来。也就是说，一个图形可以处于两种可能的状态之一(可见或不可见)。因此，本问题中的圆和弧都应该再增加一个属性——可见性。

分析需求得知，圆和弧都应该提供在屏幕上“画自己”的服务。所谓画自己，就是用当前的前景颜色在屏幕上显示自己的形状。这个程序很简单，在需求陈述中只提出了这一项最基本的功能需求。但是，根据常识一个图形既可以在屏幕上显示出来，也可以隐藏起来(实际上是用背景颜色显示)。既然已经设置了“可见性”这个属性来表明图形当前是否处于可见状态，自然也应该再提供“隐藏自己”这样一个服务。

此外，为了便于使用，通常对象的每个属性都是可以访问的。当然，可以访问并不是可以从对象外面随意读/写对象的属性，那样做将违反信息隐藏原理，也违背由对象主动提供服务而不是被动地接受处理的面向对象设计准则。所谓可以访问是指提供了读/写对象属性的服务。

综合上面的分析结果，可以用图 6-所示的类图形象地描绘程序中的两类对象。

圆	弧
-圆心坐标	-圆心坐标
-半径	-半径
-可见性	-起始角度
+读/写圆心坐标()	-结束角度
+读/写半径()	-可见性
+读/写可见性()	+读/写圆心坐标()
+显示()	+读/写半径()
+隐藏()	+读/写起始角度()
	+读/写结束角度()
	+读/写可见性()
	+显示()
	+隐藏()

图 6-圆类和弧类

从图 6-可以看出，圆和弧的许多属性和服务都是公共的。如果分别定义圆类和弧类，则这些公共的属性和服务需要在每个类中重复定义，这样做势必形成许多冗余信息。反之，如果让圆作为父类，弧作为从圆派生出来的子类，则在圆类中定义了圆心坐标、半径和可见性等属性之后，弧类就可以直接继承这些属性而无需再次重复定义它们，因此，在弧类中仅需定义本类特有的属性(起始角度和结束角度)。类似地，在圆类中定义了读/写圆心坐标、读/写半径和读/写可见性等服务之后，在弧类中只需定义读/写起始角度和读/写结束角度等弧类特有的服务。需要注意的是，虽然在图 6-中圆类和弧类都有名字相同的服务“显示”和

“隐藏”，但是它们的具体功能是不同的(显示或隐藏的图形形状不同)。因此，在把弧类作为圆类的子类之后，仍然需要在这两个类中分别定义“显示”和“隐藏”服务。

在这个简单程序中仅涉及圆和弧两类图形，当开发更复杂的图形程序时，将涉及更多的图形种类。但是，任何一种图形都有“坐标”和“可见性”等基本属性。当然，针对不同的图形，坐标的物理含义可能不同，例如，对圆来说指圆心坐标，对矩形来说指某个顶点的坐标。坐标和可见性实质上是屏幕上“点”的属性，如果把这两个基本属性抽取出来，放在点类中定义，并把点类作为各种图形类的公共父类，则不仅可进一步减少冗余信息，还能提高程序的可扩充性。相应地，读/写坐标和读/写可见性等服务也应该放在点类中定义。当然，点类中还应该定义它专用的显示和隐藏服务。

进一步分析“点”的属性可以看出，它们属于两类不同的基本信息：一类信息描述了点在哪里(坐标)，另一类信息描述了点的状态(可见性)。在上述两类信息中，坐标是更基本的信息。因此，可以定义一个更基本的基类“位置”，它仅仅拥有坐标信息，代表一个几何意义上的点。从位置类派生出屏幕上的点类，它继承了位置类中定义的每样东西(属性和服务)，并且加进了该类特有的新内容。

图 6-所示类图描绘了通过上述分析、设计过程得出的类等级。为简明起见，图中没有列出读/写属性值的常规服务。实际上，位置类提供读/写坐标服务，点类提供读/写可见性服务，圆类提供读/写半径服务，弧类提供读/写起始角度和读/写结束角度服务。

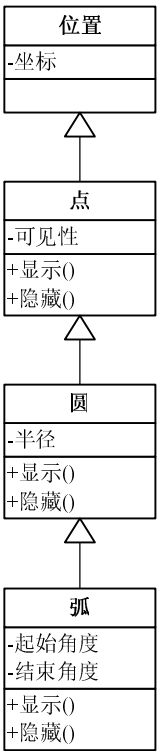


图 6-简单图形程序的类等级

P145-146 第 6-10 题

6. 建立下述牙科诊所管理系统的对象模型：

王大夫在小镇上开了一家牙科诊所。他有一个牙科助手、一个牙科保健员和一个接待员。王大夫需要一个软件系统来管理预约。

当病人打电话预约时，接待员将查阅预约登记表，如果病人申请的就诊时间与已定下的预约时间冲突，则接待员建议一个就诊时间以安排病人尽早得到诊治。如果病人同意建议的

就诊时间，接待员将输入约定时间和病人的名字。系统将核实病人的名字并提供记录的病人数据，数据包括病人的病历号等。在每次治疗或清洗后，助手或保健员将标记相应的预约诊治已经完成，如果必要的话会安排病人下一次再来。

系统能够按病人姓名和按日期进行查询，能够显示记录的病人数据和预约信息。接待员可以取消预约，可以打印出前两天预约尚未接诊的病人清单。系统可以从病人记录中获知病人的电话号码。接待员还可以打印出关于所有病人的每天和每周的工作安排。

7. 建立第 6 题所述牙科诊所管理系统的用例模型。
8. 用数据流图建立第 6 题所述牙科诊所管理系统的功能模型。
9. 写出第 6 题所述牙科诊所管理系统的脚本。
10. 画出第 6 题所述牙科诊所管理系统的状态图。

解析：

从对牙科诊所问题的陈述中，可以找出下列名词作为对象的修造者：

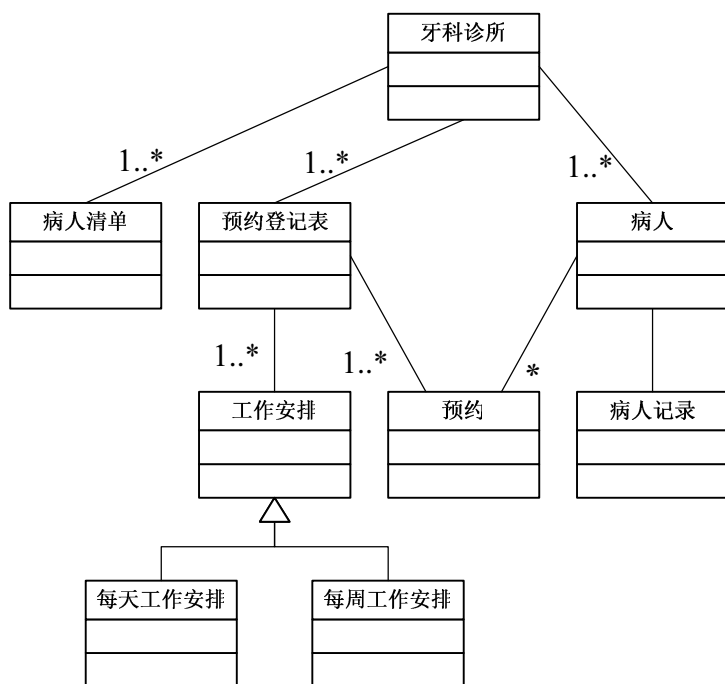
王大夫，小镇，牙科诊所，牙科助手，牙科保健员，接待员，软件系统，预约，病人，预约登记表，就诊时间，预约时间，约定时间，系统名字，记录的病人数据，病历号，姓名，日期，预约信息，病人清单，病人记录，电话号码，每天工作安排，每周工作安排。

通常，通过词法分析找到的修造对象中有许多并不是问题域中真正有意义的对象，因此，必须对这些候选对象进行严格的筛选，从中删去不正确的或不必要的，只保留确实应该记录其信息或需要其提供服务的那些对象。

具体说到牙科诊所问题，“王大夫”只不过是牙医的一个实例，实际上，本软件系统的主要功能是管理病人的预约，并不关心诊所内每名工作人员的分工，因此，牙医、牙科助手、牙科保健员和接待员都不是问题域中的对象；“小镇”是牙科诊所的地址属性，不是独立的对象；“软件系统”和“系统”是同义词，指的是将要开发的软件产品，不是问题域中的对象；“就诊时间”、“预约时间”和“约定时间”在本问题陈述中的含义相同，指的都是预约的就诊时间，实际上，预约的就诊时间既包括日期又包括时间，但是，它们是预约登记表包含的属性，不是问题域中独立的对象；“名字”和“姓名”是同义词，应该作为病人和预约登记表的属性；“记录的病人数据”实际上就是“病人记录”，可以统一使用“病人记录”作为对象名；“病历号”和“电话号码”是病人记录的属性，不是独立的对象；从问题陈述可知，“病人清单”是已预约但尚未就诊的病人名单，应该包含病人姓名、预约的就诊时间等内容，它和“预约信息”包含的内容基本相同，可以只保留“病人清单”作为问题域中的对象。

接下来分析确定问题域中对象彼此之间的关系。“每天工作安排”和“每周工作安排”有许多共同点，可以从它们泛化出一个父类“工作安排”。此外，问题域的对象之间还有下述关联关系：牙科诊所诊治多名病人；一位病人有一份病人记录；一位病人可能预约多次也可能一次也没预约；牙科诊所在一段时间内将打印出多份病人清单；牙科诊所开业以来已经建立了多份预约登记表；预约登记表中记录了多位病人的预约；根据预约登记表在不同时间可以制定出不同的工作安排。

综上所述，可以画出图 7-所示的牙科诊所管理系统的对象模型。

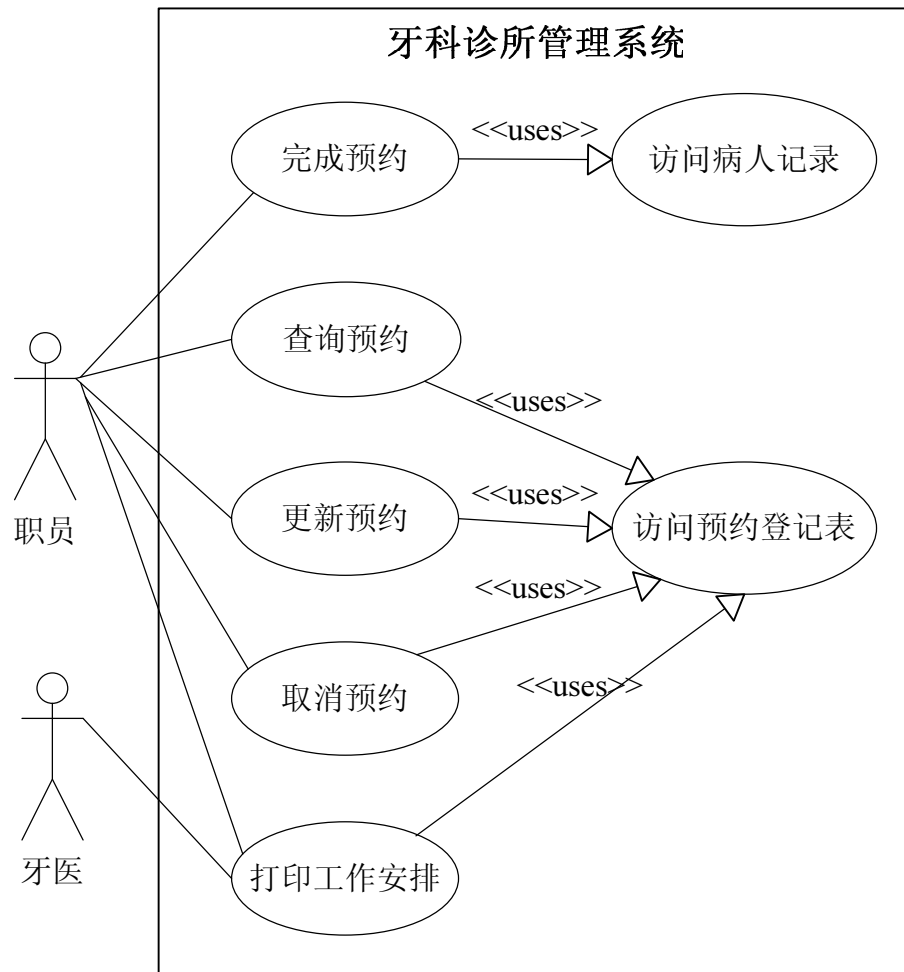


牙科诊所对象模型

7. 解析：用例图从用户角度描述系统的功能，它必须包含用户关心的所有关键功能。用户通常就是用例图中的行为者。为了画出系统的用例图，首先应该找出系统的用户，然后根据用户对系统功能的需求确定用例。

从对牙科诊所问题的陈述可知，接待员负责处理病人预约事务，为此他需要访问预约登记表和病人记录，接待员也可以取消预约，此外，接待员还可以根据预约登记表打印出关于所有病人的每天和每周的工作安排，牙医将按照工作安排诊治病人；在病人就诊后，助手或保健员将标记相应的预约诊治已完成，必要时还将安排病人下次再来，也就是说，他们将更新预约登记表的内容；系统能够按照病人姓名和日期查询预约信息，这项功能需求虽然没有指明行为者，但是这并不意味着没有行为者也可以有用例，事实上，一个用例至少必须与一个行为者相关联，可以认为“查询预约”这个用例的行为者是牙科诊所的职员。在牙科诊所问题中，没有必要区分接待员、助手和保健员在业务工作中扮演的不同角色，可以把他们统称为职员。

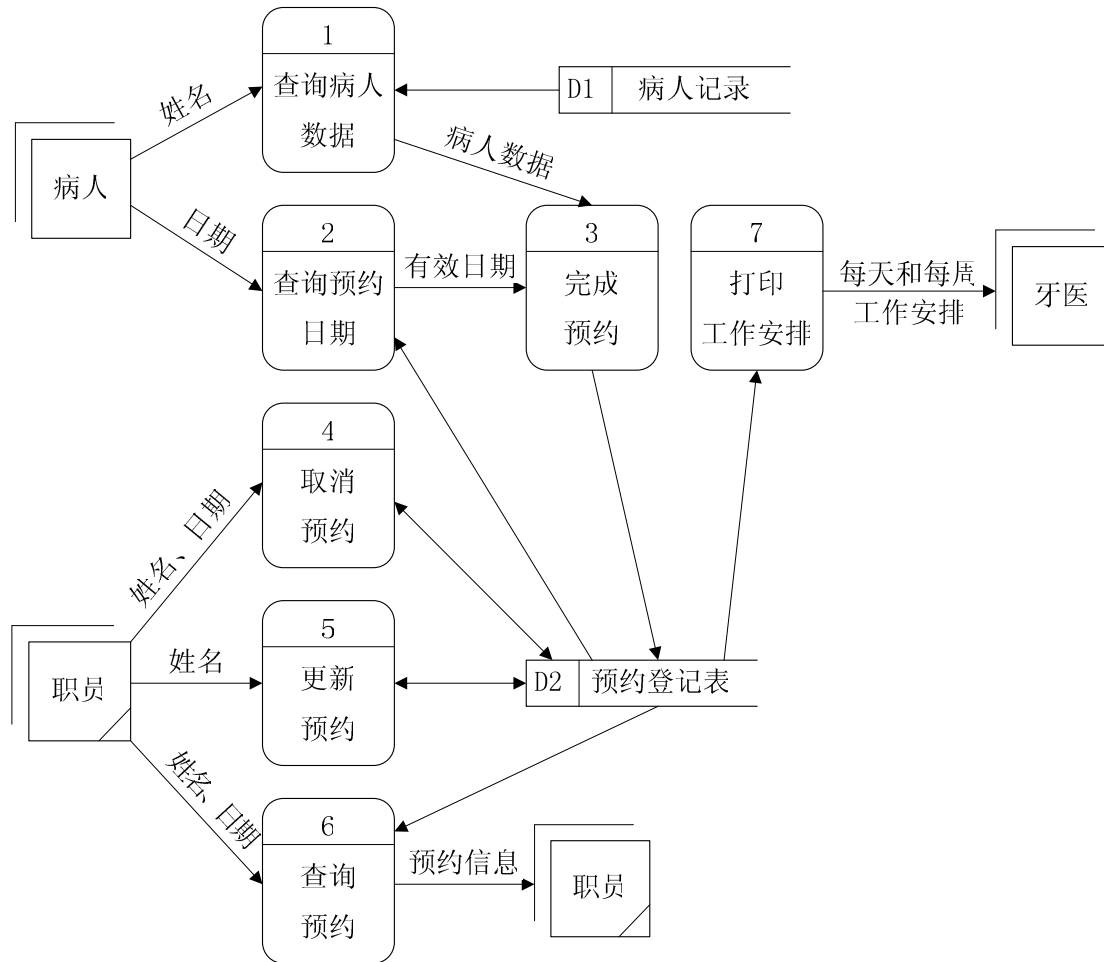
综上所述，可以画出图 7-所示的牙科诊所管理系统的用例图。



牙科诊所管理系统的用例图

8. 解析从牙科诊所管理系统的需求陈述得知，当进行预约时病人提供姓名、希望的就诊日期等数据，系统查询预约登记表，以确定一个有效的就诊日期，此外，系统还将查询病人记录以获得病历号等病人数据。在每次预约诊治完成之后，应该更新预约登记表，以标记相应的预约诊治已经完成，必要时将约定下次就诊日期。诊所职员可以按照病人姓名和日期查询预约信息，也可以取消预约。此外，系统可以打印出每天和每周的工作安排给牙医。

根据上述的系统功能，可以画出图 7-所示的牙科诊所管理系统的数据流图。



牙科诊所管理系统的数据流图

9. 解析脚本从用户角度描述系统典型的工作过程。根据对牙科诊所管理系统的需求，至少可以设想出下述 3 个脚本。

(1) 正常情况

病人甲请求预约。系统识别出病人的名字。系统建议一个就诊时间。病人同意该时间，接待员输入该预约。在预约的就诊日期到来之前两天，系统输出一份包含病人姓名和电话号码等信息的提醒清单。接待员打电话提醒病人。病人如约到来。治疗完之后，牙医助手安排该病人的下一次预约。

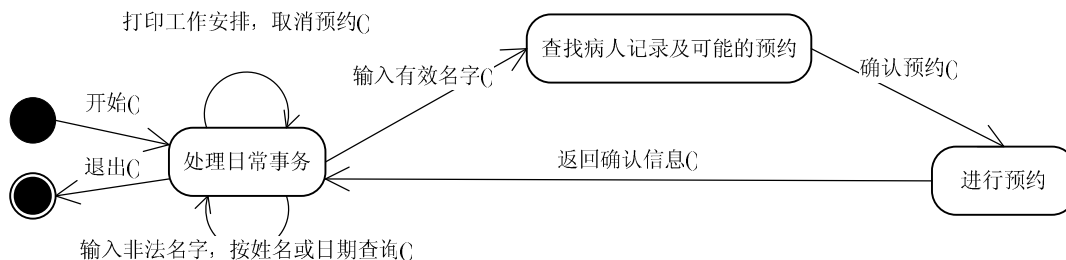
(2) 新病人

病人乙请求预约。系统不认识该病人的名字，必须把该病人的信息输入到病人记录系统中并为他建立一个记录。

(3) 多个预约

病人丙请求在未来两年内进行 16 次预约。接待员将其姓名输入到系统中，系统提出建议的预约就诊时间，病人同意后接待员输入病人认可的预约。

10. 解析：状态图描绘系统可能的状态转换。牙科诊所管理系统的主要功能是实现病人预约，根据需求陈述和在第 9 题解答中给出的脚本，可以画出图 7-所示的牙科诊所管理系统状态图。图中把除了完成病人预约之外的事务笼统地称为日常事务。



牙科诊所管理系统状态图

如果一个程序有两个输入数据, 每个输入都是一个 32 位的二进制整数, 那么这个程序有多少个可能的输入? 如果每微秒可进行一次测试, 那么对所有可能的输入进行测试需要多长时间?

答: 每个 32 位的二进制整数具有 2^{32} 个可能的值, 因此, 具有两个整数输入的程序应该具有 2^{64} 个可能的输入。

每微秒可进行一次测试, 即每秒可进行 106 个测试, 因此, 每天可进行的测试数为:

$$60 \times 60 \times 24 \times 10^6 = 8.64 \times 10^{10}$$

这等于每年大约可进行 3.139×10^{13} 个测试。

因为 $2^{10} = 1024 \approx 10^3$, 所以 $2^{64} = (2^{10})^6 \cdot 2^4 \approx 10^{19.2}$ 。

$$\frac{10^{19.2}}{3.139 \times 10^{13}} > 10^5, \text{ 所以做完全部测试将至少需要 } 10^5 \text{ 年 (即 10 万年)}。$$

4. 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 Y=20
    ELSE Y=2
END IF
IF C>15
    THEN Z=30
    ELSE Z=3
END IF
PRINT (X, Y, Z)
STOP
  
```

答: (1) 语句覆盖的测试用例

因为每个判定表达式为真或为假时均有赋值语句, 为了使每个语句都至少执行一次, 总共需要两组测试数据, 以便使得每个判定表达式取值为真或为假各一次。下面是实现语句覆

盖的典型测试用例：

① 使 3 个判定表达式之值全为假

输入：A=1, B=1, C=1

预期的输出：X=1, Y=2, Z=3

② 使 3 个判定表达式之值全为真

输入：A=20, B=40, C=60

预期的输出：X=10, Y=20, Z=30

(2) 路径覆盖的测试用例

本程序共有 8 条可能的执行通路，为做到路径覆盖总共需要 8 组测试数据。下面是实现路径覆盖的典型测试用例：

① 3 个判定表达式之值全为假

输入：A=1, B=1, C=1

预期的输出：X=1, Y=2, Z=3

② 3 个判定表达式依次为假、假、真

输入：A=1, B=1, C=60

预期的输出：X=1, Y=2, Z=30

③ 3 个判定表达式依次为假、真、假

输入：A=1, B=40, C=1

预期的输出：X=1, Y=20, Z=3

④ 3 个判定表达式依次为假、真、真

输入：A=1, B=40, C=60

预期的输出：X=1, Y=20, Z=30

⑤ 3 个判定表达式依次为真、假、假

输入：A=20, B=1, C=1

预期的输出：X=10, Y=2, Z=3

⑥ 3 个判定表达式依次为真、假、真

输入：A=20, B=1, C=60

预期的输出：X=10, Y=2, Z=30

⑦ 3 个判定表达式依次为真、真、假

输入：A=20, B=40, C=1

预期的输出：X=10, Y=20, Z=3

⑧ 3 个判定表达式全为真

输入：A=20, B=40, C=60

预期的输出：X=10, Y=20, Z=30

6. 使用基本路径测试方法, 设计测试下面列出的伪码程序的测试用例：

```
1:      START
          INPUT (A, B, C, D)
2:      IF (A>0)
3:          AND (B>0)
4:      THEN X=A+B
5:      ELSE X=A-B
6:      END
7:      IF (C>A)
```

```

8:      OR (D>B)
9:      THEN Y=C-D
10:     ELSE Y=C+D
11:     END
12:     PRINT (X, Y)
      STOP

```

答：用基本路径测试方法设计测试用例的过程，有下述 4 个步骤：

(1) 根据过程设计的结果画出流图

与本题给出的伪码程序相对应的流图如图 4.1 所示。

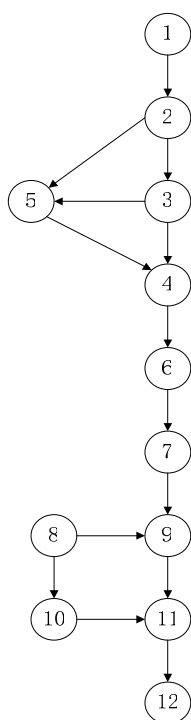


图 4.1 与第 6 题伪码程序对应的流图

(2) 计算流图的环形复杂度

使用下述 3 种方法中的任一种都可以算出图 4.1 所示流图的环形复杂度为 5。

① 该流图共有 15 条边，12 个结点，所以环形复杂度为

$$15 - 12 + 2 = 5$$

② 该流图共有 5 个区域，因此环形复杂度为 5。

③ 该流图中共有 4 个判定结点，因此环形复杂度为

$$4 + 1 = 5$$

(3) 确定线性独立的路径的基本集合

所谓线性独立路径是指至少引入程序的一个新语句集合或一个新条件的路径，用流图术语来描述，独立路径至少包含一条在定义该路径之前不曾用过的边。

使用基本路径测试法设计测试用例时，程序的环形复杂度决定了程序中独立路径的数量，而且这个数值是确保程序中所有语句至少被执行一次所需的测试数量的上界。

对于图 4.1 所示流图来说，由于它的环形复杂度为 5，因此共有 5 长独立路径。下面列出了 5 条独立路径：

路径 1: 1—2—3—4—6—7—9—11—12

路径 2: 1—2—5—6—7—9—11—12
 路径 3: 1—2—3—5—6—7—9—11—12
 路径 4: 1—2—3—4—6—7—8—9—11—12
 路径 5: 1—2—3—4—6—7—8—10—11—12

(4) 设计可强制执行基本路径的测试用例

① 执行路径 1 (两个判定表达式全为真)

输入: A=1, B=1, C=2, D=2 (任意)

预期的输出: X=2, Y=0

② 执行路径 2 (第一个判定表达式为假, 第二个判定表达式为真)

输入: A=0, B=1, C=2, D=0 (任意)

预期的输出: X=-1, Y=2

③ 执行路径 3 (第一个判定表达式为假, 第二个判定表达式为真)

输入: A=1, B=0, C=2, D=2 (任意)

预期的输出: X=1, Y=2

④ 执行路径 4 (两个判定表达式全为真)

输入: A=1, B=1, C=0, D=-1

预期的输出: X=2, Y=1

⑤ 执行路径 5 (第一个判定表达式为假, 第二个判定表达式为真)

输入: A=1, B=1, C=0, D=2 (任意)

预期的输出: X=2, Y=2

2. 分析图 3.1 所示的层次图, 确定每个模块的内聚类型。

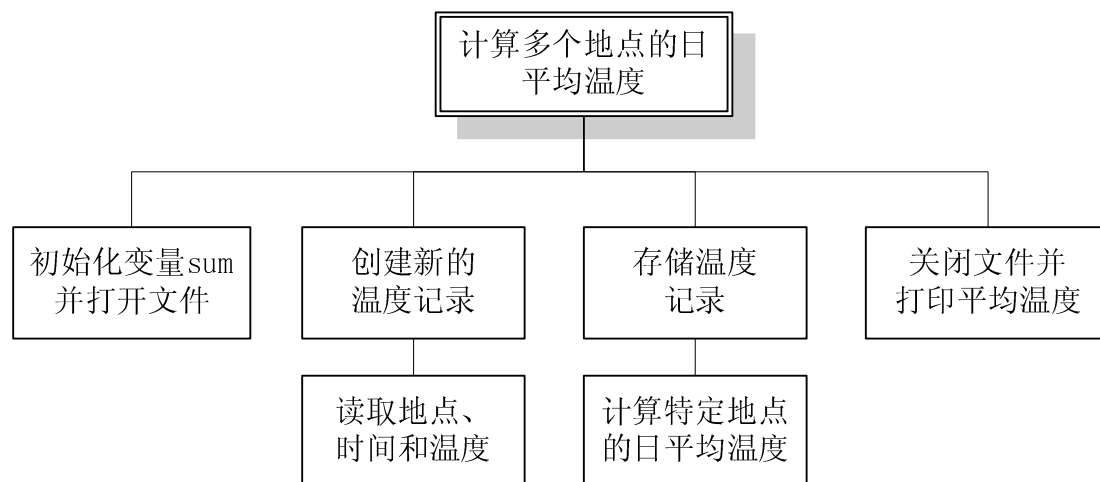


图 3.1 计算多地点日平均温度的程序

解析: 从图 3.1 所示的层次图可以看出, 这个程序的功能是计算若干个指定地点的每日平均温度。变量 `sum` 保存某地一天之内在指定的时间取样点的温度之和。程序运行时首先初始化变量 `sum` 并打开文件, 然后读取地点、时间和温度等原始数据, 创建用于保存这些数据的温度记录, 接下来计算特定地点的日平均温度, 存储温度记录。重复调用“读取地点、时间和温度”、“创建新的温度记录”、“计算特定地点的日平均温度”和“存储温度记录”等模块, 直至计算出并保存好所有指定地点的日平均温度。最后, 打印平均温度并关闭文件。

从上述叙述可知, “计算多个地点的日平均温度”、“读取地点、时间和温度”、“创建新的温度记录”、“计算特定地点的日平均温度”和“存储温度记录”等 5 个模块, 每个都完成一个单一的功能, 模块内所有处理元素都为完成同一个功能服务, 彼此结合得十分紧密, 因此这 5 个模块的内聚类型都是功能内聚。

初看起来，由于初始化变量 `sum` 和打开文件这两个操作都是程序运行的初始阶段完成的，“初始化变量 `sum` 并打开文件”这个模块的内聚类型似乎是时间内聚。但是，初始化这是 `sum` 是本程序特有的操作，而打开文件是硬件要求的操作，是任何使用文件的程序都包含的一个操作，并非本程序特有的操作。当可以分配两个或更多个不同级别的内聚类型给一个模块时，规则是分配最低级别的内聚类型给该模块。因此，“初始化变量 `sum` 并打开文件”这个模块的内聚类型是偶然内聚。同理，“关闭文件并打印平均温度”这个模块的内聚类型也是偶然内聚。

3. 分析图 3.2，确定模块之间的耦合类型。

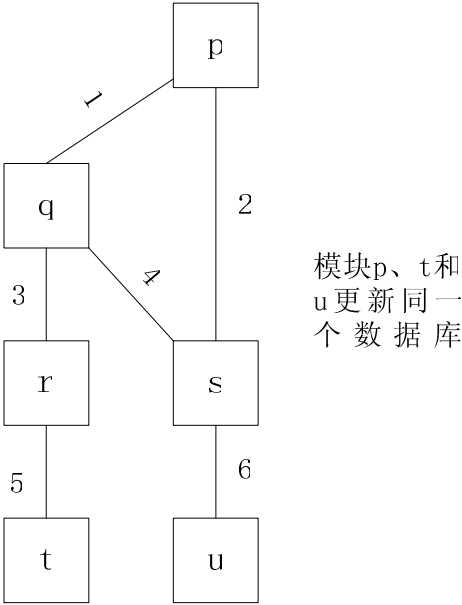


图 3.2 一个程序的模块互连图

在图 3.2 中已经给模块之间的接口编了号码，表 3.1 描述了模块间的接口。

表 3-1 模块接口描述

编号	输入	输出
1	飞机类型	状态标志
2	飞机零件清单	
3	功能代码	
4	飞机零件清单	
5	零件编号	零件制造商
6	零件编号	零件名称

解析：综合分析图 3.2 和表 3.1 所提供的信息可知各个模块之间的耦合情况。例如，当模块 `p` 调用模块 `q` 时(接口 1)，它传递了一个参数——飞机类型。当模块 `q` 把控制返还给模块 `p` 时，它传回一个状态标志。

某些模块之间的耦合类型是明显的，例如，模块 `p` 和 `q` 之间(接口 1)、模块 `r` 和 `t` 之间(接口 5)及模块 `s` 和 `u` 之间(接口 6)都是数据耦合，因为它们传递的都是一个简单变量。

如果两个模块中的一个模块给另一个模块传递控制元素，也就是说，如果一个模块明显地控制另一个模块的逻辑，则它们之间具有控制耦合。例如，当给具有逻辑内聚的模块传递功能代码时就传递了控制元素。另一个控制耦合的例子是把控制开关作为一个参数传递。图 3.2 中模块 `q` 调用模块 `r` 时(接口 3)传递一个功能代码，因此，这两个模块之间是控制耦合。

图 3.2 右侧文字说明，模块 `p`、`t` 和 `u` 更新同一个数据库，因此，它们之间具有公共环境耦合。

当模块 p 调用模块 s 时(接口 2)，如果模块 s 使用或更新模块 p 传递给它的零件清单中的所有元素，则模块 p 和 s 之间的耦合是数据耦合；但是，如果模块 s 只访问该清单中的一部分元素，则模块 p 和 s 之间的耦合是特征耦合。模块 q 和 s 之间(接口 4)的耦合情况与此类似。由于图 3.2 和表 3.1 中给出的信息尚不足以准确地描述各个模块的功能，所以不能确定这两对模块之间的耦合是数据耦合还是特征耦合。

7. 图 3.4 是用程序流程图描绘的程序算法，请把它改画为等价的盒图。

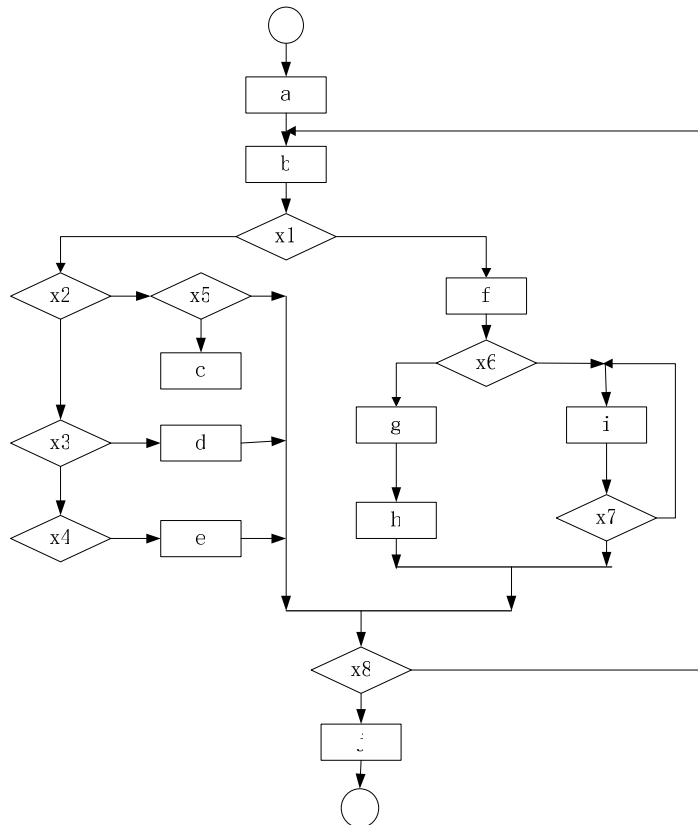


图 3.4 程序流程图

解析：分析图 3.4 可以看出，该处理过程由顺序执行的 3 个程序块组成：首先执行处理 a，然后执行一个 DO-UNTIL 型循环，最后执行处理 j。

DO-UNTIL 型循环的循环体是处理 b 和一个 IF-THEN-ELSE 型分支结构，循环结束条件为 x8。其中，IF-THEN-ELSE 型分支结构的分支条件是 x1，THEN 部分是处理 f 和另一个分支条件为 x6 的 IF-THEN-ELSE 型分支结构；ELSE 部分是一个 CASE 型多分支结构。这样一层一层地分析下去，可以画出图 3.19 所示的与图 3.4 等价的盒图。

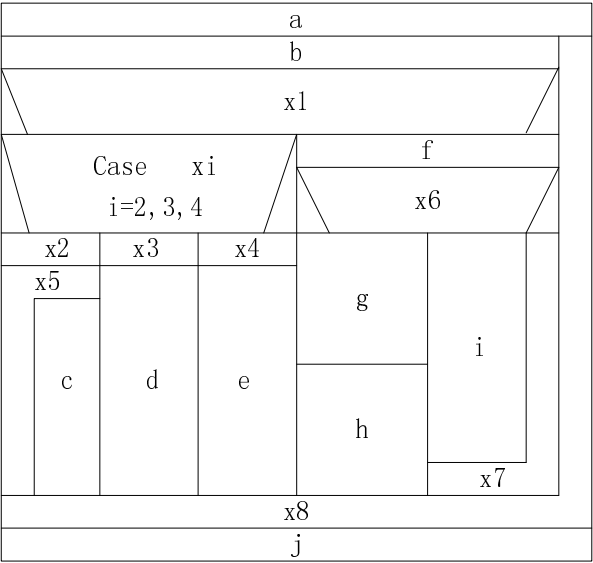


图 3.19 与图 3.4 等价的盒图

8. 某交易所规定给经纪人的手续费计算如下：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。如果交易总金额少于 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 元到 10000 元之间，则基本手续费为交易金额的 5%，再加 34 元；如果交易总金额超过 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为基本手续费的 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为基本手续费的 6%。当每股售价超过 25 元时，如果交易的股数零散（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) 用判定树表示手续费的计算方法。

解析：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。

判定表的每一列是一条计算规则。例如，第 1 列(规则 1)规定，当交易总金额 P 少于 1000 元，且每股售价 Q 低于 14 元，且交易的股数 n 是 100 的倍数时，给经纪人的手续费为

$$(1+0.05) \times 0.084P$$

第 16 列(规则 16)表明，当交易总金额 P 超过 10000 元，且每股售价 Q 在 14 到 25 元之间，且交易的股数 n 不是 100 的倍数时，手续费为

$$(1+0.06) \times (0.04P+134)$$

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
$1000 \leq P \leq 10000$	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
$14 \leq Q \leq 25$	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
$(1+0.05) \times 0.084P$	×																	
$(1+0.09) \times 0.084P$		×																
$(1+0.02) \times 0.084P$			×															
$(1+0.06) \times 0.084P$				×														
$(1+0.01) \times 0.084P$					×													
$(1+0.04) \times 0.084P$						×												
$(1+0.05) \times (0.05P+34)$							×											
$(1+0.09) \times (0.05P+34)$								×										
$(1+0.02) \times (0.05P+34)$									×									
$(1+0.06) \times (0.05P+34)$										×								
$(1+0.01) \times (0.05P+34)$											×							
$(1+0.04) \times (0.05P+34)$												×						
$(1+0.05) \times (0.04P+134)$													×					
$(1+0.09) \times (0.04P+134)$														×				
$(1+0.02) \times (0.04P+134)$															×			
$(1+0.06) \times (0.04P+134)$																×		
$(1+0.01) \times (0.04P+134)$																	×	
$(1+0.04) \times (0.04P+134)$																		×

图 3.20 描述手续费计算方法的判定表

(2) 表示手续费计算方法的判定树如图所示。

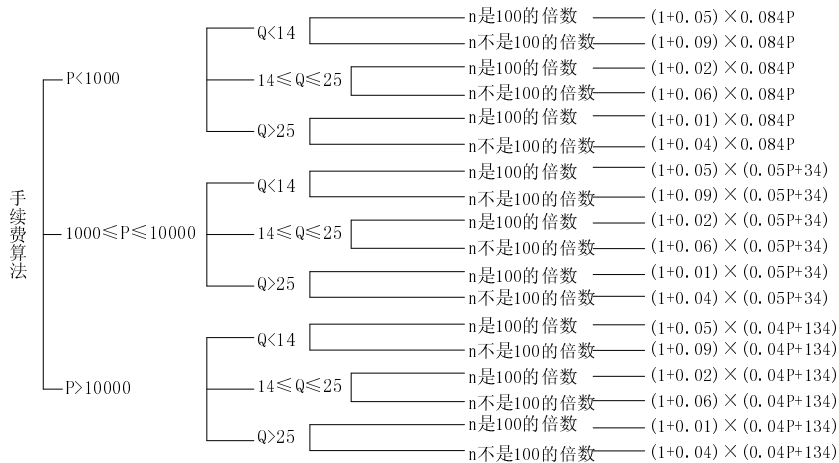


图 3.21 描述手续费计算方法的判定树

9. 画出下列伪码程序的程序流程图和 NS 图。

```

START
IF P THEN

```

```

    WHILE q DO
        F
    END DO
ELSE
    BLOCK
        g
        n
    END BLOCK
END IF
STOP

```

解：从该伪码程序转换成的程序流程图示于图 3.22 中。

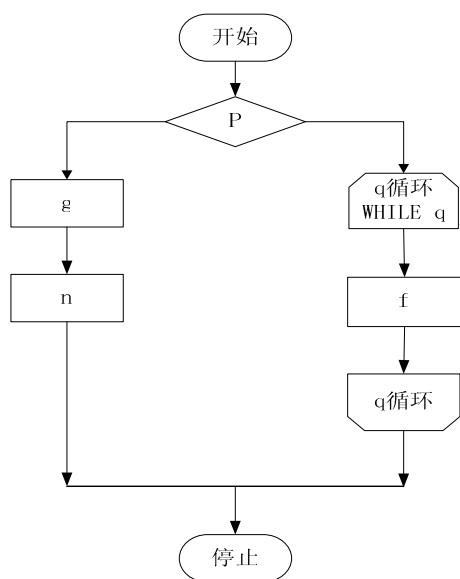


图 3.22 从伪码转变成的程序流程图

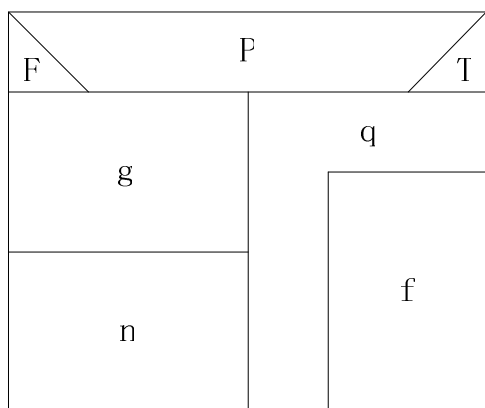


图 3.23 由伪码转变成的盒图。

10.图 3.5 给出的程序流程图代表一个非结构化的程序,请问:

- (1) 为什么说它是非结构化的?
- (2) 设计一个与它等价的结构化程序。
- (3) 在(2)题的设计中你使用附加的标志变量 **flag** 了吗? 若没用,请再设计一个使用 **flag** 的程序;若用了,请再设计一个不用 **flag** 的程序。

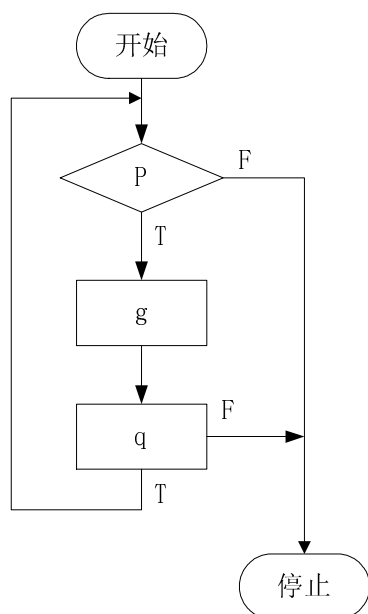
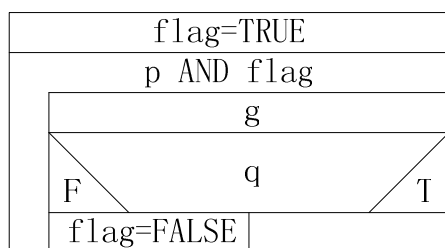


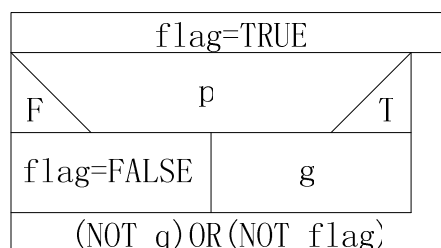
图 3.5 一个非结构化程序

解析：(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。图 3.5 所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 3.24 所示盒图描绘了等价的结构化程序。



(a) 解法1



(b) 解法2

图 3.24 与图 3.5 等价的结构化程序(用 flag)

(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 3.25 所示。

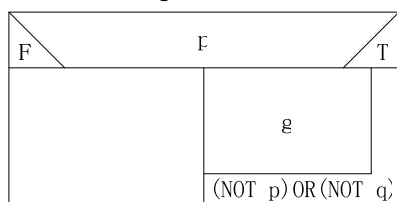


图 3.25 与图 3.5 等价的结构化程序(不用 flag)