

## 软件工程复习整理

### 1. 软件生命期各阶段的任务是什么？

答：软件生命期分为 7 个阶段：

- 1、问题定义：要解决的问题是什么
- 2、可行性研究：确定问题是否值得解，技术可行性、经济可行性、操作可行性
- 3、需求分析：系统必须做什么
- 4、总体设计：系统如何实现，包括系统设计和结构设计
- 5、详细设计：具体实现设计的系统
- 6、实现：编码和测试
- 7、运行维护：保证软件正常运行。

### 2、软件重用的效益是什么？

答：1、软件重用可以显著地改善软件的质量和可靠性。

- 2、软件重用可以极大地提高软件开发的效率。
- 3、节省软件开发的成本，避免不必要的重复劳动和人力、财力的浪费。

### 3、自顶而下渐增测试与自底而上渐增测试各有何优、缺点？

答：

#### ① 自顶而下渐增测试

优点：不需要测试驱动程序，能够在测试阶段的早期实现并验证系统的主要功能，而且能够尽早发现上层模块的接口错误。

缺点：需要存根程序，底层错误发现较晚。

#### ② 自底而上渐增测试

优点与缺点和自顶而下渐增测试相反。

### 4、提高可维护性的方法有哪些？

答：在软件工程的每一阶段都应该努力提高系统的可维护性，在每个阶段结束前的审查和复审中，应着重对可维护性进行复审。

在需求分析阶段的复审中，应对将来要扩充和修改的部分加以注明。在讨论软件可移植性问题时，要考虑可能要影响软件维护的系统界面。

在软件设计的复审中，因从便于修改、模块化和功能独立的目标出发，评价软件的结构和过程，还应对将来可能修改的部分预先做准备。

在软件代码复审中，应强调编码风格和内部说明这两个影响可维护性的因素。

在软件系统交付使用前的每一测试步骤中都应给出需要进行预防性维护部分的提示。

在完成每项维护工作后，都应对软件维护本身进行仔细认真的复审。

为了从根本上提高软件系统的可维护性，人们正试图通过直接维护软件规格说明来维护软件，同时也在大力发展软件重用技术。

**简述软件测试要经过哪几个步骤，每个步骤与什么文档有关。**

**【解答】**

测试过程按 4 个步骤进行，即单元测试（模块测试）、集成测试（子系统测试和系统测试）、确认测试（验收测试）和平行运行。

单元测试集中对用源代码实现的每一个程序单元进行测试，与其相关的文档是单元测试计划和详细设计说明书。

集成测试把已测试过的模块组装起来，主要对与设计相关的软件体系结构的构造进行测试。与其相关的文档是集成测试计划和软件需求说明书。

确认测试则是要检查已实现的软件是否满足了需求规格说明中确定的各种需求，以及软件配置是否完全、正确。与其相关的文档是确认测试计划和软件需求说明书。

平行运行把已经经过确认的软件纳入实际运行环境中，与其他系统成份组合在一起进行测试。与其相关的文档：用户指南、使用手册等。

### 36. 简述容错技术的四种主要手段，并解释。

结构冗余：包括静态冗余、动态冗余和混合冗余。

信息冗余：为检测或纠正信息在运算或传输中的错误，须外加一部分信息。

时间冗余：指重复执行指令或程序来消除瞬时错误带来的影响。

冗余附加技术：指为实现上述冗余技术所需的资源和技术。

### 36. 可行性报告的主要内容有哪些？

解：一个可行性研究报告的主要内容如下：

（1）引言：说明编写本文档的目的；项目的名称、背景；本文档用到的专门术语和参考资料。

（2）可行性研究前提：说明开发项目的功能、性能和基本要求；达到的目标；各种限制条件；可行性研究方法和决定可行性的主要因素。

（3）对现有系统的分析：说明现有系统的处理流程和数据流程；工作负荷；各项费用支出；所需要各类专业技术人员的数量；所需要各种设备；现有系统存在什么问题。

（4）所建议系统的技术可行性分析：所建议系统的简要说明；处理流程和数据流程；与现有的系统比较的优越性；采用所建议系统对用户的影响；对各种设备、现有软件、开发环境、运行环境的影响；对经费支出的影响；对技术可行性的评价。

（5）所建议系统的经济可行性分析：说明所建议系统的各种支出，各种效益；收益投资比；投资回收周期。

（6）社会因素可行性分析：说明法律因素，对合同责任、侵犯专利权、侵犯版权等问题的分析；说明用户使用可行性，是否满足用户行政管理、工作制度、人员素质的要求。

（7）其他可供选择方案：逐一说明其他可供选择的方案，并说明未被推荐的理由。

（8）结论意见：说明项目是否能开发；还需要什么条件才能开发；对项目目标有什么变动等。

[解析]做软件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确定该软件项目是否能够开发，是否值得去开发，其中的问题能否解决，报告共分为八个基本内容。

### 37. 系统设计的内容是什么？

解：系统设计阶段先从高层入手，然后细化。系统设计要决定整个结构及风格，这种结构为后面设计阶段的更详细策略的设计提供了基础。

（1）系统分解。系统中主要的组成部分称为子系统，子系统既不是一个对象也不是一个功能，而是类、关联、操作、时间和约束的集合。每次分解的各子系统数目不能太多，最底层

子系统称为模块。

(2) 确定并发性。分析模型、现实世界及硬件中不少对象均是并发的。系统设计的一个重要目标就是确定哪些是必须同时动作的对象,哪些不是同时动作的对象。后者可以放在一起,而综合成单个控制线或任务。

(3) 处理器及任务分配。各并发子系统必须分配给单个硬件单元,要么是一个一般的处理器,要么是一个具体的功能单元,必须完成下面的工作:估计性能要求和资源需求,选择实现子系统的硬软件,将软件子系统分配给各处理器以满足性能要求和极小化处理器之间的通信,决定实现各子系统的各物理单元的连接。

(4) 数据存储管理。系统中的内部数据和外部数据的存储管理是一项重要的任务。通常各数据存储可以将数据结构、文件、数据库组合在一起,不同数据存储要在费用、访问时间、容量以及可靠性之间做折中考虑。

(5) 全局资源的处理。必须确定全局资源,并且制定访问全局资源的策略。全局资源包括:物理资源,如处理器、驱动器等;空间,如盘空间、工作站屏等;逻辑名字,如对象标识符、类名、文件名等。

如果资源是物理对象,则可以通过建立协议实现对并发系统的访问,以达到自身控制;如果资源是逻辑实体,如对象标识符,那么在共享环境中存在冲突访问的可能,如独立的事务可能同时使用同一个对象标识符,则各个全局资源都必须有一个保护对象,由保护对象来控制对该资源的访问。

(6) 选择软件控制机制。分析模型中所有交互行为都表示为对象之间的事件。系统设计必须从多种方法中选择某种方法来实现软件的控制。

(7) 人机交互接口设计。设计中的大部分工作都与稳定的状态行为有关,但必须考虑用户使用系统的交互接口。

### 38. 什么是软件危机? 软件危机的表现是什么? 其产生的原因是什么?

解: 软件发展第二阶段的末期,由于计算机硬件技术的进步,计算机运行速度、容量、可靠性有显著的提高,生产成本显著下降,这为计算机的广泛应用创造了条件。一些复杂的、大型的软件开发项目提出来了,但是,软件开发技术的进步一直未能满足发展的需要。在软件开发中遇到的问题找不到解决办法,使问题积累起来,形成了尖锐的矛盾,因而导致了软件危机。

软件危机表现在以下四个方面:

(1) 经费预算经常突破,完成时间一再拖延。由于缺乏软件开发的经验和软件开发数据的积累,使得开发工作的计划很难制定。主观盲目制定计划,执行起来与实际情况有很大差距,使得开发经费一再突破。由于对工作量估计不足,对开发难度估计不足,进度计划无法按时完成,开发时间一再拖延。

(2) 开发的软件不能满足用户要求。开发初期对用户的要求了解不够明确,未能得到明确的表达。开发工作开始后,软件人员和用户又未能及时交换意见,使得一些问题不能及时解决,导致开发的软件不能满足用户的要求,因而导致开发失败。

(3) 开发的软件可维护性差。开发过程中没有同意的、公认规范,软件开发人员按各自的风格工作,各行其是,开发过程无完整、规范的文档,发现问题后进行杂乱无章的修改。程序结构不好,运行时发现错误也很难修改,导致维护性差。

(4) 开发的软件可靠性差。由于在开发过程中,没有确保软件质量的体系和措施,在软件测试时,又没有严格的、充分的、完全的测试,提交给用户的软件质量差,在运行中暴露出大量的问题。

造成软件危机的原因是:

(1) 软件的规模越来越大，结构越来越复杂。(2) 软件开发管理困难而复杂。(3) 软件开发费用不断增加。(4) 软件开发技术落后。(5) 生产方式落后。(6) 开发工具落后，生产率提高缓慢。

### 39. 软件质量保证应做好哪几方面的工作？

解：软件质量保证是软件工程管理的重要内容，软件质量保证应做好以下几方面的工作：

(1) 采用技术手段和工具。质量保证活动要贯彻开发过程始终，必须采用技术手段和工具，尤其是使用软件开发环境来进行软件开发。

(2) 组织正式技术评审。在软件开发的每一个阶段结束时，都要组织正式的技术评审。国家标准要求单位必须采用审查、文档评审、设计评审、审计和测试等具体手段来保证质量。

(3) 加强软件测试。软件测试是质量保证的重要手段，因为测试可发现软件中大多数潜在错误。

(4) 推行软件工程规范（标准）。用户可以自己制定软件工程规范（标准），但标准一旦确认就应贯彻执行。

(5) 对软件的变更进行控制。软件的修改和变更常常会引起潜伏的错误，因此必须严格控制软件的修改和变更。

(6) 对软件质量进行度量。即对软件质量进行跟踪，及时记录和报告软件质量情况。

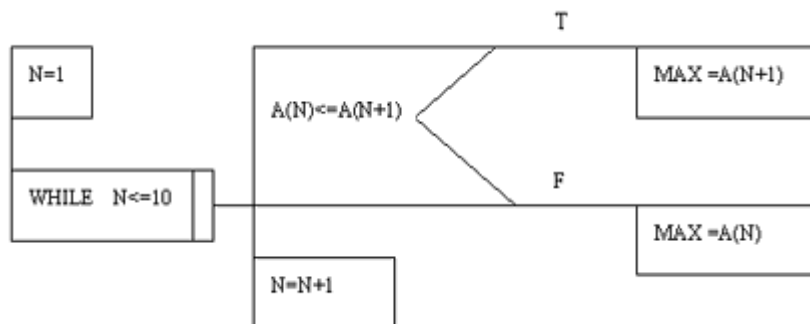
[解析]软件的质量保证是向用户及社会提供满意的高质量的产品，确保软件产品从诞生到消亡为止的所有阶段的质量的活动，是软件工程管理中的重要内容。

### 40. 请使用 PAD 图和 PDL 语言描述在数组 A (1) ~A (10) 中找最大数的算法。

解：PDL 语言：

```
N=1
WHILE N<=10 DO
  IF A (N) <=A (N+1) MAX =A (N+1) ;
  ELSE MAX =A (N) ENDIF;
  N=N+1;
ENDWHILE;
```

PAD 图：



[解析]人工查找时，是从第一个元素开始查找，用当前元素与下一个元素比较，将较大者作为当前元素又与下一元素比较，如此循环，直到数组末尾。

## 四、应用题（每题 8 分，共 40 分）

42. 某电器集团公司下属的厂包括技术科、生产科等基层单位。现在想建立一个计算机辅助企业管理系统，其中：

生产科的任务是：

(1) 根据销售公司转来的内部合同（产品型号、规格、数量、交获日期）制定车间月生产计划。

(2) 根据车间实际生产日报表、周报表调整月生产计划

3) 以月生产计划为以及，制定产品设计（结构、工艺）及产品组装月计划。

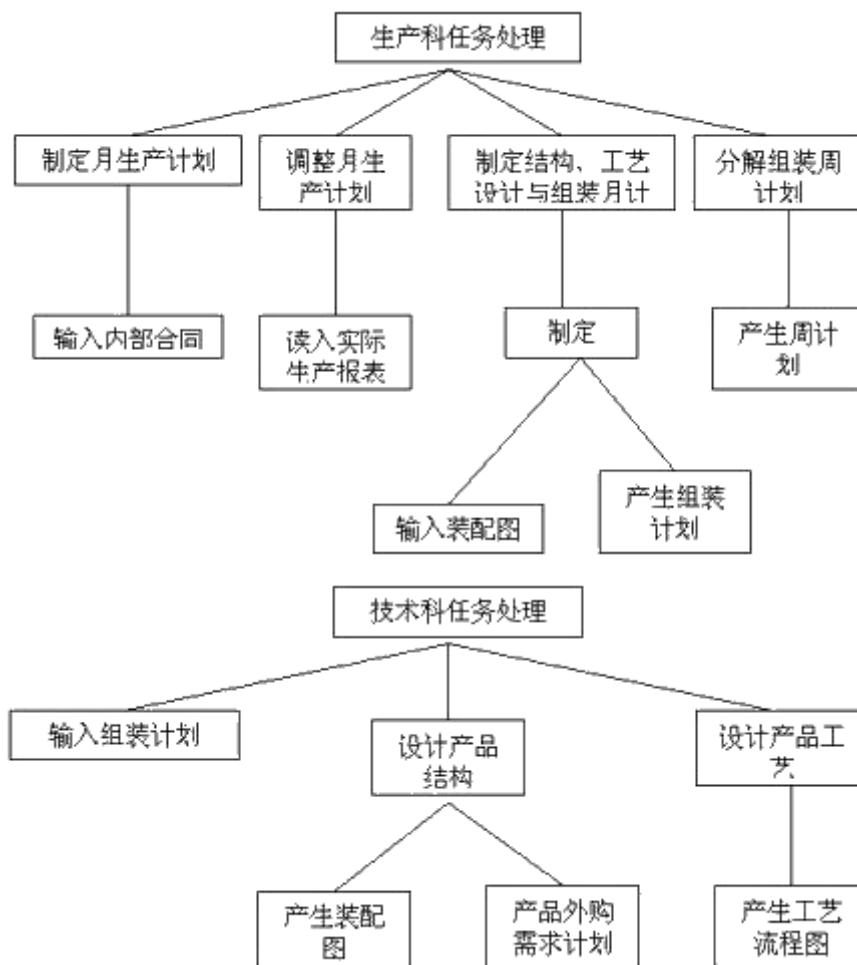
(4) 将产品的组装计划传达到各科，将组装月计划分解为周计划，下达给车间

技术科的任务是：

(1) 根据生产科转来的组装计划进行产品结构图设计，产生产品装配图给生产科，产生外购需求计划给供应科，并产生产品自制物料清单。

(2) 根据组装计划进行产品工艺设计，根据产品自制物料清单产生工艺流程图给零件厂。试写出以上系统中生产科和技术科处理的软件结构图。

解：



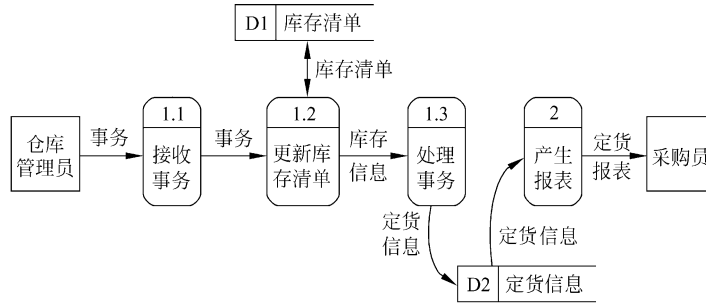
画出生产科图的给 6 分，画出技术科的给 4 分。

[解析] 软件结构图是软件系统的模块层次结构，反映了整个系统的功能实现，即将来程序的控制层次体系，软件结构往往用树状或网状结构的图形来表示，其主要内容有模块及模块的控制关系，根据题意，可绘制出生产科和技术科的软件结构图，其中生产科的结构图深度和宽度均为 4，技术科的结构图的深度和宽度均为 3。

1、假设一家工厂的采购部每天需要一张定货报表，报表按零件编号排序，表中列出所有需要再次定货的零件。对于每个需要再次定货的零件应该列出下述数据：零件编号，零件

名称，定货数量，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 CRT 终端把事务报告给定货系统。当某种零件的库存数量少于库存量临界值时就应该再次定货。要求：画出系统的数据流图。

2、  
答：

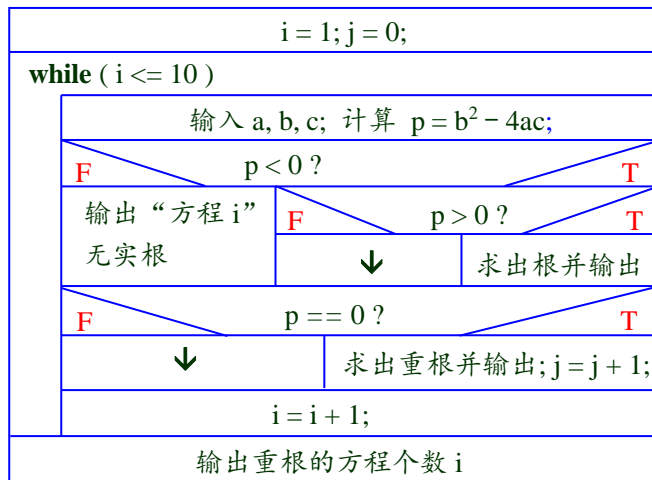


2、将下面给出的伪码转换为 N-S 图和 PAD 图。

```
void root ( float root1, float root2 ) {
    i = 1; j = 0;
    while ( i <= 10 ) {
        输入一元二次方程的系数 a, b, c;
        p = b*b - 4*a*c;
        if ( p < 0 ) 输出“方程 i 无实数根” ;
        else if ( p > 0 ) 求出根并输出;
        if ( p == 0 ) {
            求出重根并输出;
            j = j + 1;
        }
        i = i + 1;
    }
    输出重根的方程的个数 j;
}
```

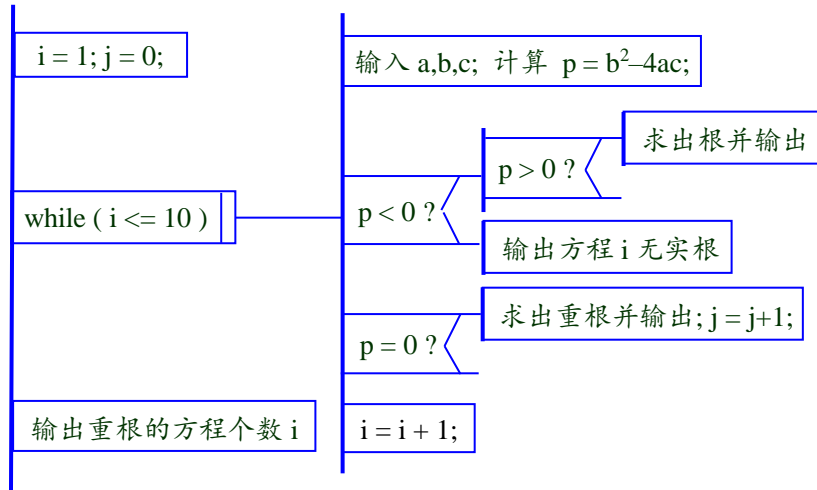
【解答】

(1) N-S 图



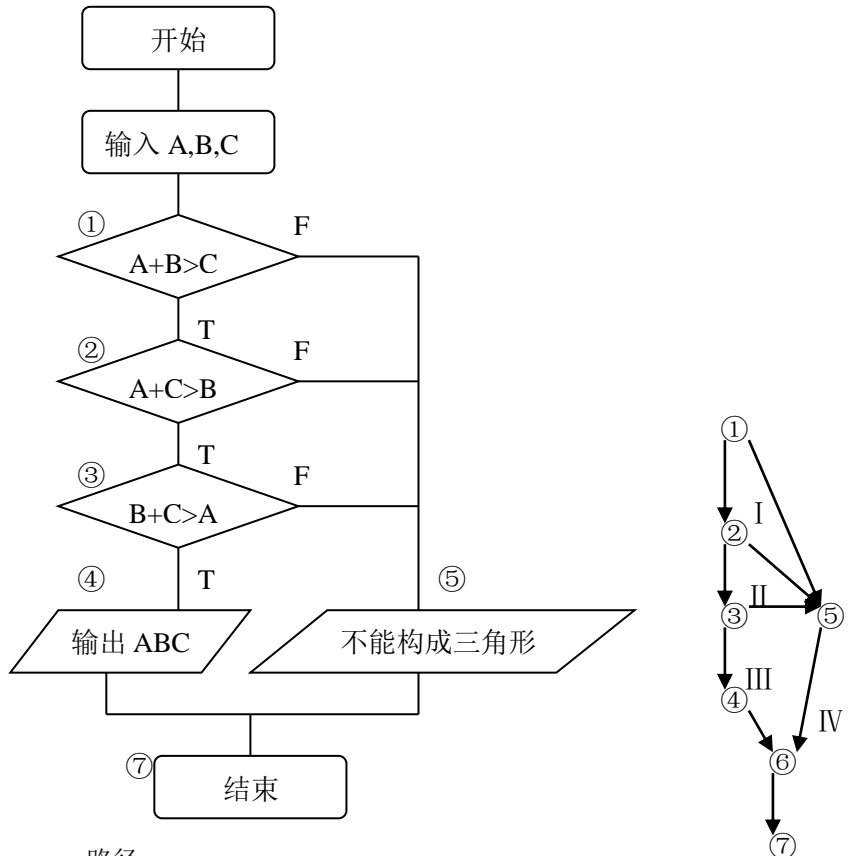


(2) PAD 图



3、:输入三整数,判断是否构成三角形,如构成三角形,则输出三条边的值,否则输出”不能构成三角形”。要求:1.用程序流程图表示该问题的算法; 2.计算程序复杂度; 3.设计路径覆盖的测试用例。

答:



路径:

1. ①—②—③—④—⑥—⑦
2. ①—②—⑤—⑥—⑦
3. ①—⑤—⑥—⑦
4. ①—②—③—⑤—⑥—⑦

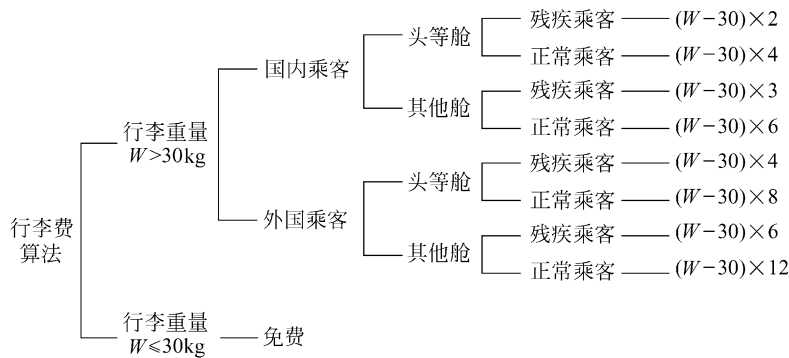
程序复杂度=3+1=4

测试用例

路径	测试用例 (A, B, C)	结果
1	A=3, B=4, C=5	A=3, B=4, C=5
2	A=5, B=8, C=2	不能构成三角形
3	A=2, B=2, C=5	不能构成三角形
4	A=5, B=2, C=2	不能构成三角形

4、某航空公司规定，乘客可以免费托运重量不超过 30kg 的行李。当行李重量超过 30kg 时，对头等舱的国内乘客超重部分每公斤收费 4 元，对其他舱的国内乘客超重部分每公斤收费 6 元，对外国乘客超重部分每公斤收费比国内乘客多一倍，对残疾乘客超重部分每公斤收费比正常乘客少一半。用判定树表示与上述每种条件组合相对应的计算行李费的算法。

答案：

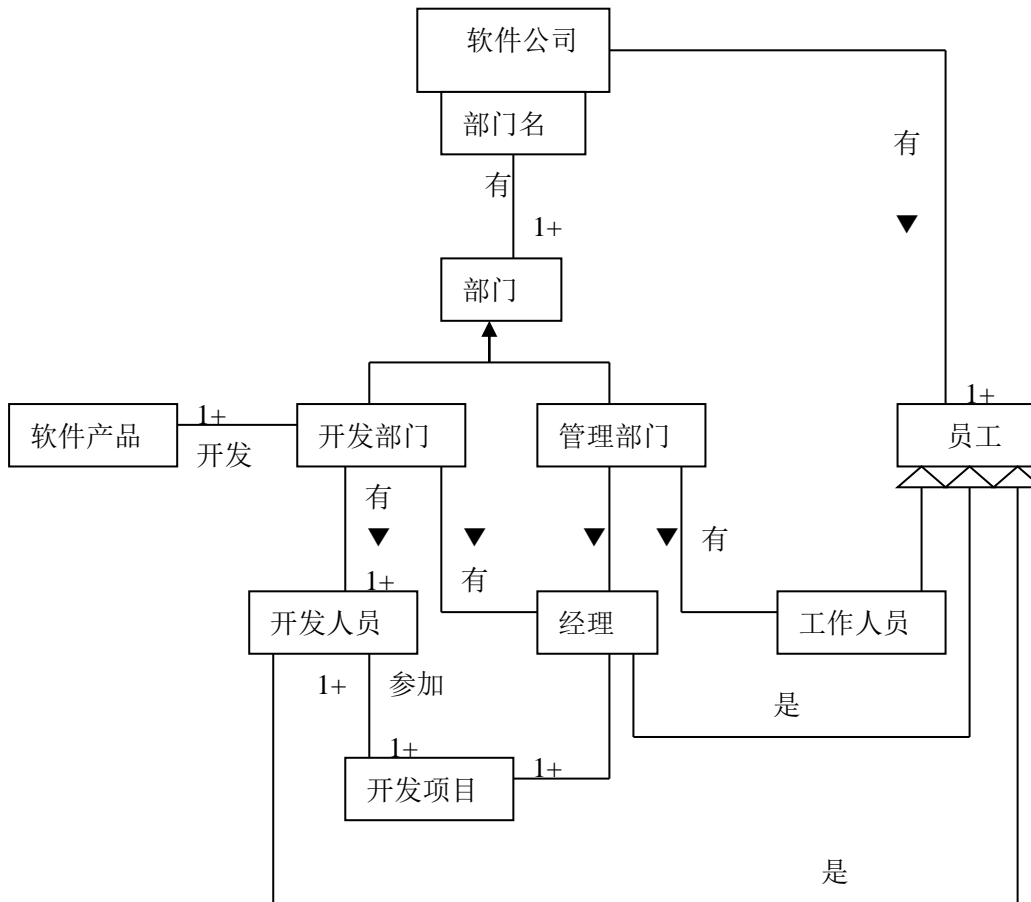


5、一个软件公司有许多部门，分为开发部门和管理部门两种，每个开发部门开发多个软件产品，每个部门由部门名字唯一确定。该公司有许多员工，员工分为经理，工作人员和开发人员。

开发部门有经理和开发人员，管理部门有经理和工作人员。每个开发人员可参加多个开发项目，每个开发项目需要多个开发人员，每位经理可主持多个开发项目，建立该公司的对象模型。



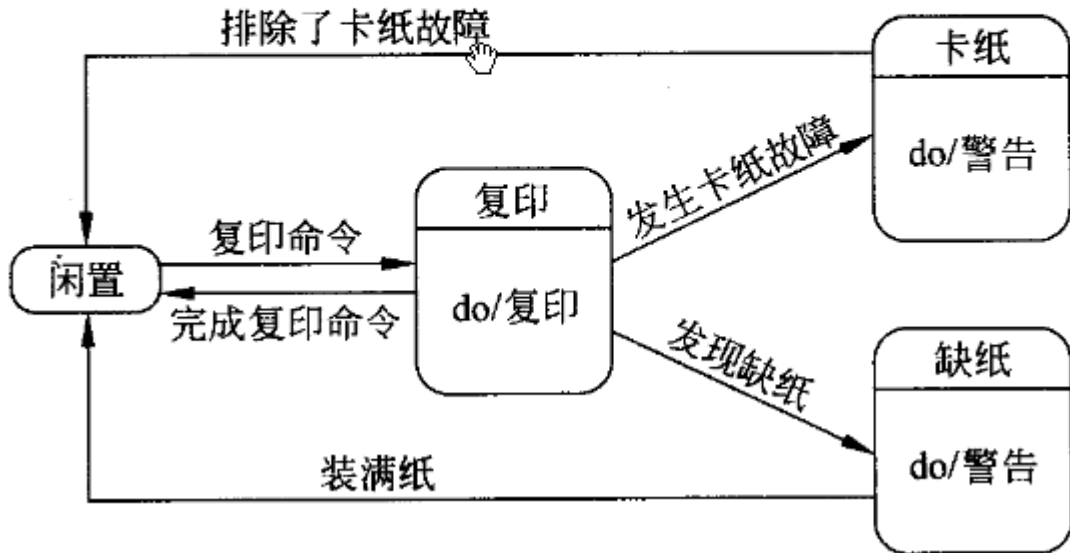
答案:



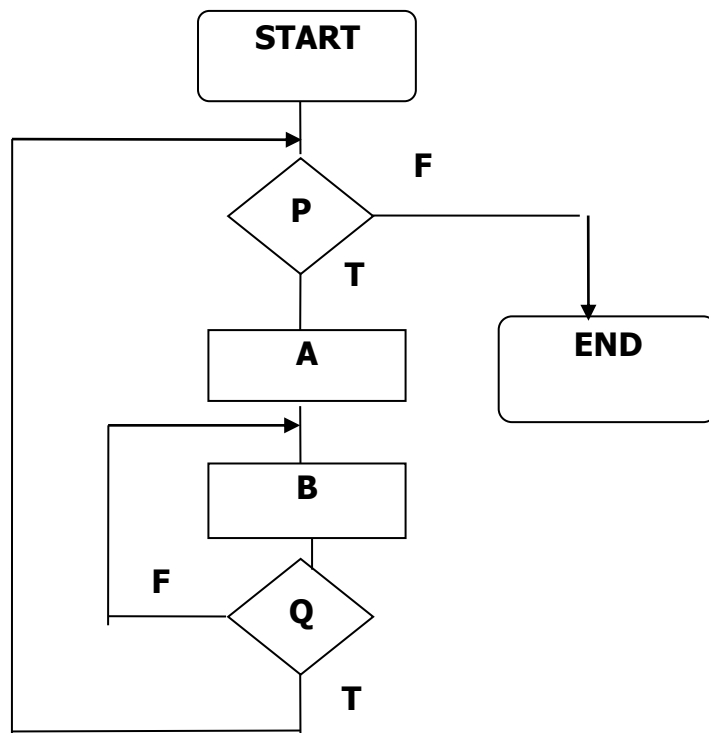
2. 复印机的工作过程大致如下：未接到复印命令时处于闲置状态，一旦接到复印命令则进入复印状态，完成一个复印命令规定的工作后又回到闲置状态，等待下一个复印命令；如果执行复印命令时发现没纸，则进入缺纸状态，发出警告，等待装纸，装满纸后进入闲置状态，准备接收复印命令；如果复印时发生卡纸故障，则进入卡纸状态，发出警告等待维修人员来排除故障，故障排除后回到闲置状态。

请用状态转换图描绘复印机的行为。

- 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”。



3. 根据下边的程序流程图，画出相应的N-S图，并计算它的环形复杂度。



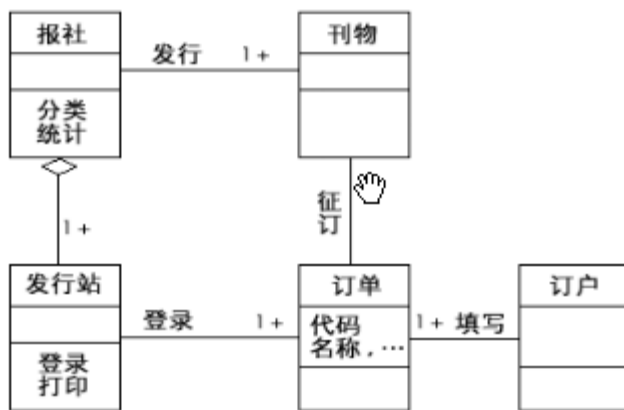
5. 某报社采用面向对象技术实现报刊征订的计算机管理系统,该系统基本需求如下:

(1) 报社发行多种刊物, 每种刊物通过订单来征订, 订单中有代码, 名称, 订期, 单价, 份数等项目, 订户通过填写订单来订阅报刊。

(2) 报社下属多个发行站, 每个站负责收集登录订单、打印收款凭证等事务。

(3) 报社负责分类并统计各个发行站送来的报刊订阅信息。

请就此需求建立对象模型。(15 分)



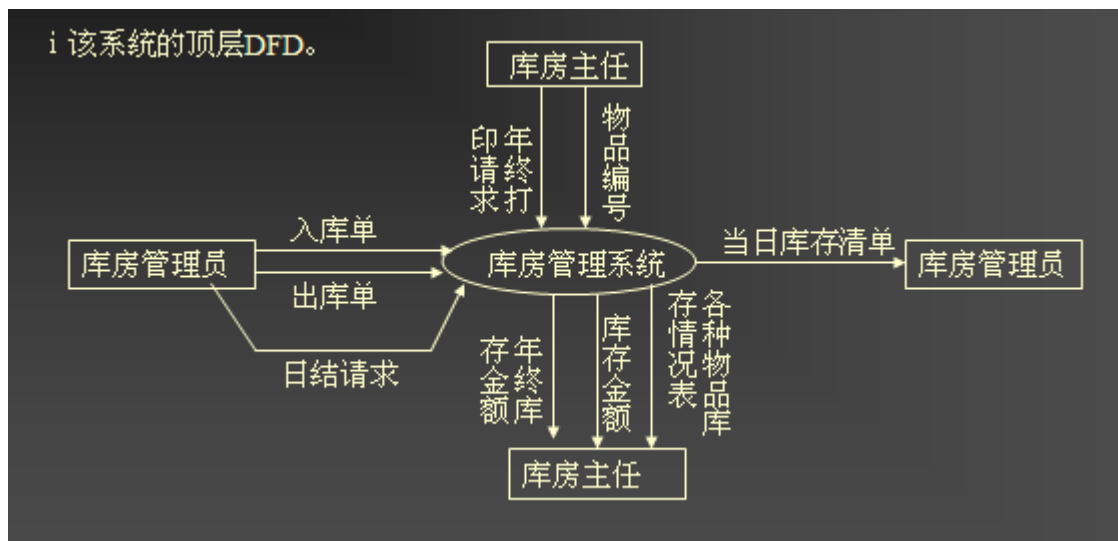
6. 在要建立的某库房简化的信息管理系统中, 库房管理员负责:

1) 当新的货物需要入库时, 根据入库单 (品名, 编号, 生产厂家, 数量, 单价, 日期) 录入物品信息。

2) 当接到出库单 (品名, 编号, 数量, 日期) 时, 进行数量的修改。

3) 当日结时, 打印库房商品库存清单 (品名, 编号, 库存量, 库存金额)。

就以上系统功能画出0层，1层的DFD图。



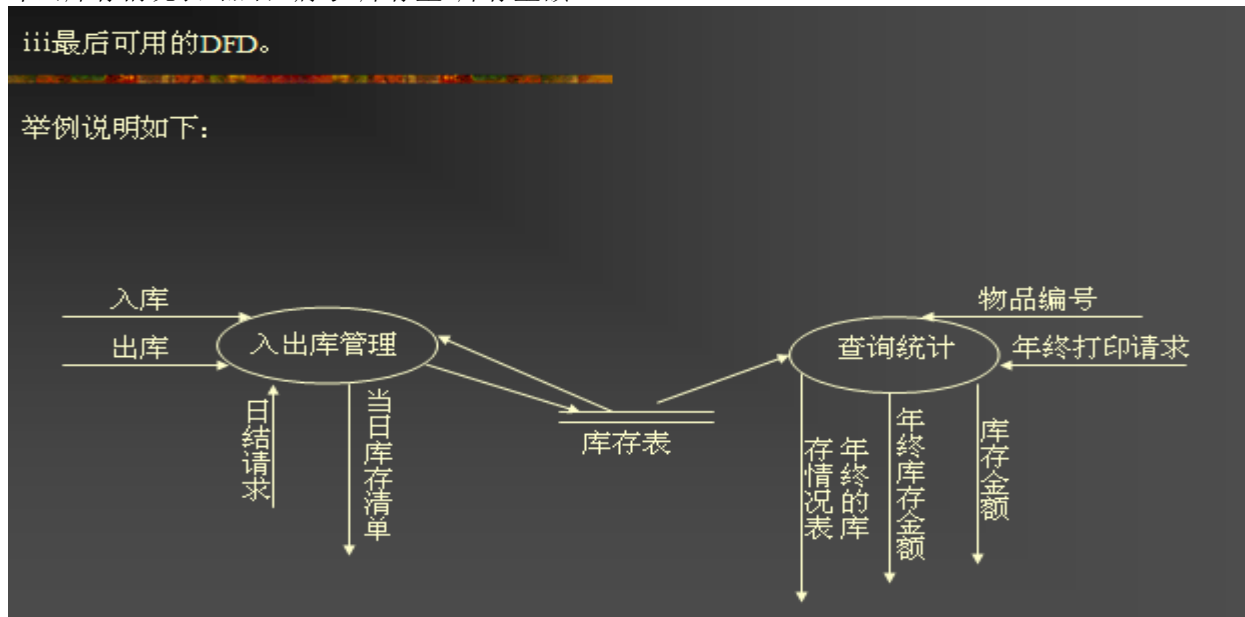
ii 顶层 DFD 的数据词典。

入库单=品名+编号+生产厂家+数量+单价+日期

出库单=品名+编号+数量+日期

库存清单=品名+编号+库存量+库存余额

年终库存情况表=品名+编号+库存量+库存金额



**简答题**

1. 软件工程的七条基本原理是什么？

答：软件工程的七条基本原理是：

- (1) 用分阶段的生存周期计划严格管理
- (2) 坚持进行阶段评审；
- (3) 严格实施的产品控制
- (4) 采用现代程序技术
- (5) 结果应能清楚地审查；
- (6) 开发小组的成员应该少而精
- (7) 承认不断改进软件工程的必要性。

2. 良好的编码风格应具备哪些条件？

答：应具备以下条件：

- (1) 使用标准的控制结构
- (2) 有限制地使用 GOTO 语句
- (3) 源程序的文档化 (应具备以下内容)
  - ①有意义的变量名称 —— “匈牙利命名规则”
  - ②适当的注释 —— “注释规范”
  - ③标准的书写格式:
    - 用分层缩进的写法显示嵌套结构的层次 (锯齿形风格)
    - 在注释段的周围加上边框
    - 在注释段与程序段、以及不同程序段之间插入空行
    - 每行只写一条语句
    - 书写表达式时, 适当使用空格或圆括号等作隔离符
- (4) 满足运行工程学的输入输出风格

### 3. 简述文档在软件工程中的作用

答: (1) 提高软件开发过程的能见度

- (2) 提高开发效率
- (3) 作为开发人员阶段工作成果和结束标志
- (4) 记录开发过程的有关信息便于使用与维护
- (5) 提供软件运行、维护和培训有关资料
- (6) 便于用户了解软件功能、性能

### 9. 什么是模块化? 模块设计的准则?

**模块化**是按规定的原则将一个大型软件划分为一个个较小的、相对独立但又相关的模块  
模块设计的准则:

- (1) 改进软件结构, 提高模块独立性: 在对初步模块进行合并、分解和移动的分析、精化过程中力求提高模块的内聚, 降低耦合
- (2) 模块大小要适中: 大约 50 行语句的代码, 过大的模块应分解以提高理解性和可维护性; 过小的模块, 合并到上级模块中
- (3) 软件结构图的深度、宽度、扇入和扇出要适当一般模块的调用个数不要超过 5 个
- (4) 尽量降低模块接口的复杂程度
- (5) 设计单入口、单出口的模块
- (6) 模块的作用域应在控制域之内

### 10. 什么是模块独立性? 用什么度量?

答: 模块独立性概括了把软件划分为模块时要遵守的准则, 也是判断模块构造是不是合理  
的标准。独立性可以从两个方面来度量: 即模块本身的内聚和模块之间的耦合。

### 11. 变换型数据流由哪几部分组成?

变换型结构由三部分组成: 传入路径、变换(加工)中心 和 传出路径

### 12. 变换分析设计的步骤?

- (1) 区分传入、传出和变换中心三部分, 划分 DFD 图的分界线
- (2) 完成第一级分解: 建立初始 SC 图的框架
- (3) 完成第二级分解: 分解 SC 图的各个分支
- (4) 对初始结构图按照设计准则进行精化与改进

### 14. 事务分析设计的步骤?

- (1) 在 DFD 图中确定事务中心、接收部分 (包含全部接收路径) 和 发送部分 (包含全部动作路径)
- (2) 画出 SC 图框架, 把 DFD 图三部分, 分别映射为事务控制模块, 接收模块和动作发送模块. 一般得到 SC 图的顶层和第一层 (如果第一层简单可以并入顶层)

(3) 分解和细化接收分支和动作分支,完成初始的 SC 图

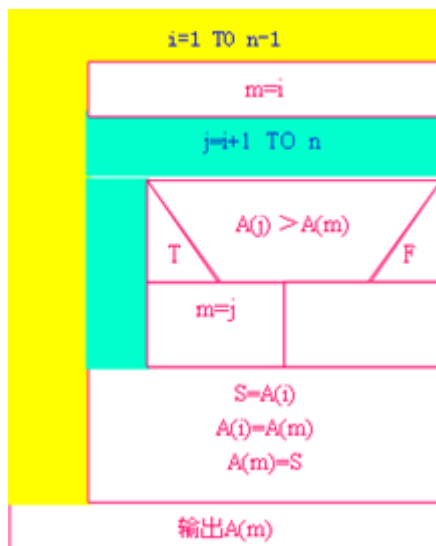
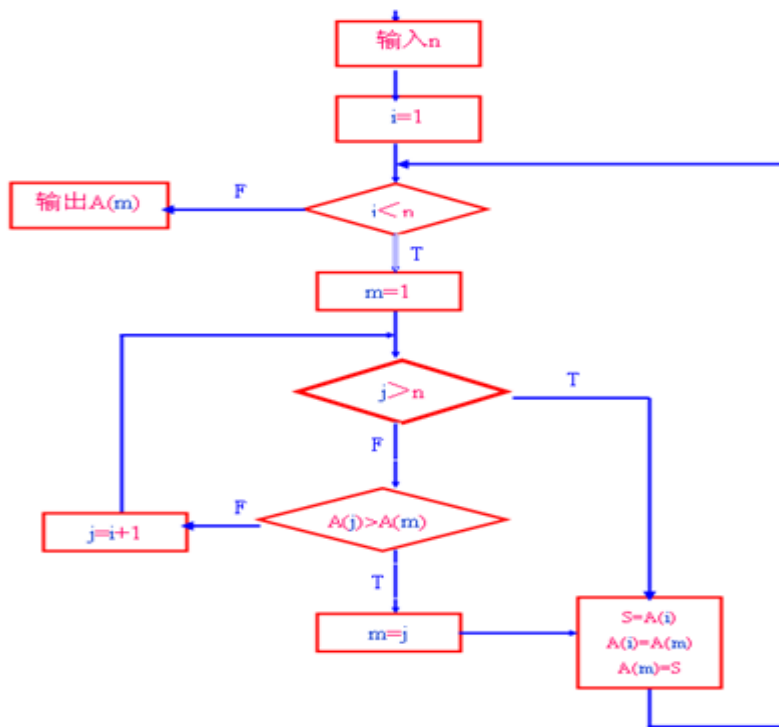
(4) 对初始结构图按照设计准则进行精化与改进

15. 数据字典包括哪些内容?

数据字典是描述数据流图中数据的信息的集合。它对数据流图上每一个成分：数据项、文件(数据结构)、数据流、数据存储、加工和外部项等给以定义和说明；它主要由数据流描述、加工描述和文件描述三部分组成

16. 给出一组数从小到大的排序算法，分别用下列工具描述其详细过程：

(1)流程图；(2)N-S 图；(3) PDL 语言



17. 欲开发一个银行的活期存取款业务的处理系统：储户将填好的存/取款单和存折交给银行工作人员，然后由系统作以下处理

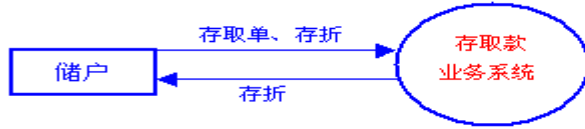
(1) 业务分类处理：系统首先根据储户所填的存/取款单，确定本次业务的性质，并将存/取款单和存折交下一步处理

(2) 存款处理：系统将存款单上的存款金额分别记录在存折和帐目文件中，并将现金存入现金库；最后将存折还给储户

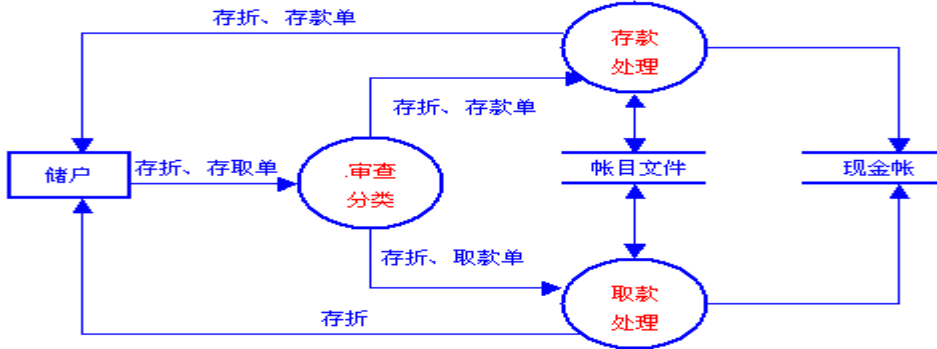
(3) 取款处理：系统将取款单上的取款金额分别记录在存折和帐目文件中，并从现金库提取现金；最后将现金和存折还给储户

绘制该系统的数据流图和软件结构图

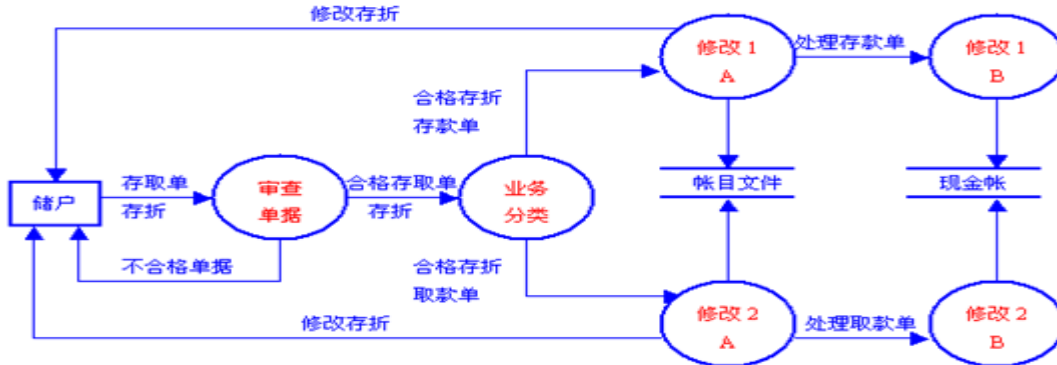
顶层



一层



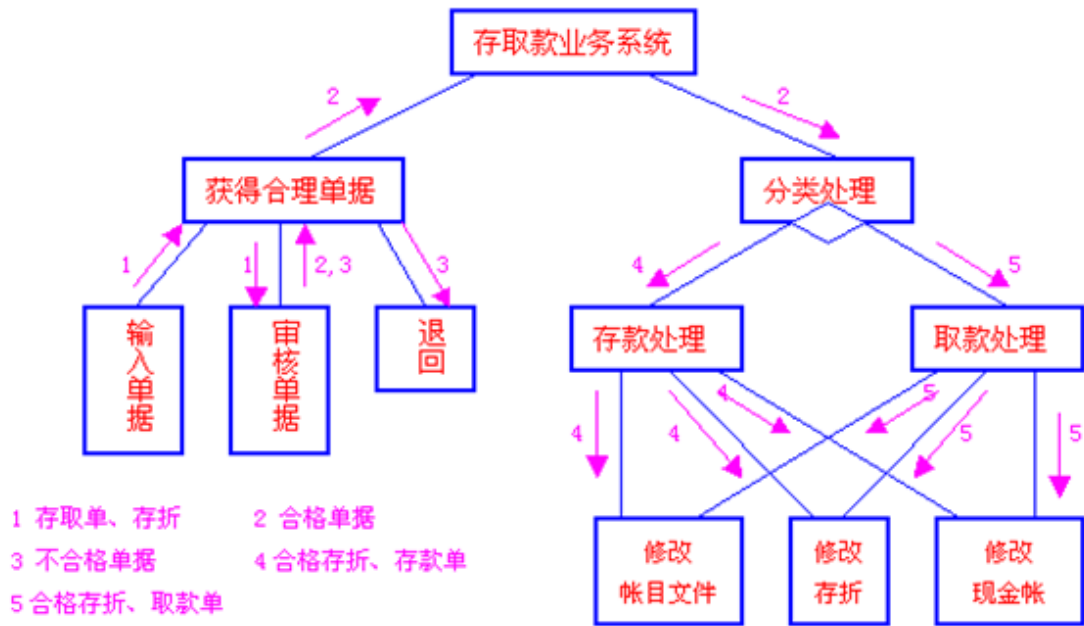
二层



注：A—文件帐目及存折；B—现金帐



软件结构图

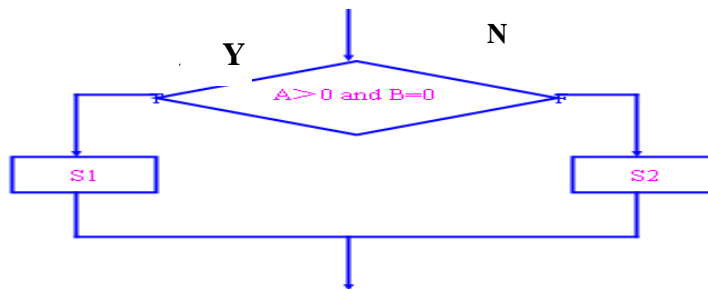


26. 等价分类法的测试技术采用的一般方法？举例说明？

- (1) 为每个等价类编号
- (2) 设计一个新的测试方案, 以尽可能多的覆盖尚未被覆盖的有效等价类, 重复这一步骤, 直到所有有效等价类被覆盖为止
- (3) 设计一个新的测试方案, 使它覆盖一个尚未被覆盖的无效等价类, 重复这一步骤, 直到所有无效等价类被覆盖为止 (例略)

27. 如图显示某程序的逻辑结构

试为它设计足够的测试用例, 分别实现对程序的判定覆概、条件覆概和条件组合覆概。



设计测试方案

覆盖种类	需满足的条件	测试数据	期望结果	
判定覆盖	A > 1, B = 0	A = 2, B = 0	执行 S1	
	A > 1, B ≠ 0 或 A ≤ 1, B = 0 或 A ≤ 1, B ≠ 0	A = 2, B = 1 或 A = 1, B = 0 或 A = 1, B = 1	执行 S2	
条件覆盖	以下四种情况各出现一次			
	A > 1	B = 0	A = 2, B = 0	执行 S1
	A ≤ 1	B ≠ 0	A = 1, B = 1	执行 S2
	A > 1, B = 0	A = 2, B = 0	执行 S1	

条件组合 覆盖	A>1, B≠0	A=2, B=1	执行 S2
	A≤1, B=0	A=1, B=0	执行 S2
	A≤1, B≠0	A=1, B=1	执行 S2

28. 某城市电话号码由三部分组成。它们的名称和内容分别是：

地区码：空白或三位数字

前 缀：非‘0’或‘1’的三位数字

后 缀：4 位数字

假定被测程序能接受一切符合上述规定的电话号码，拒绝所有不符合规定的电话号码。

根据该程序的规格说明，作等价类的划分，并设计测试方案。

划分等价类

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

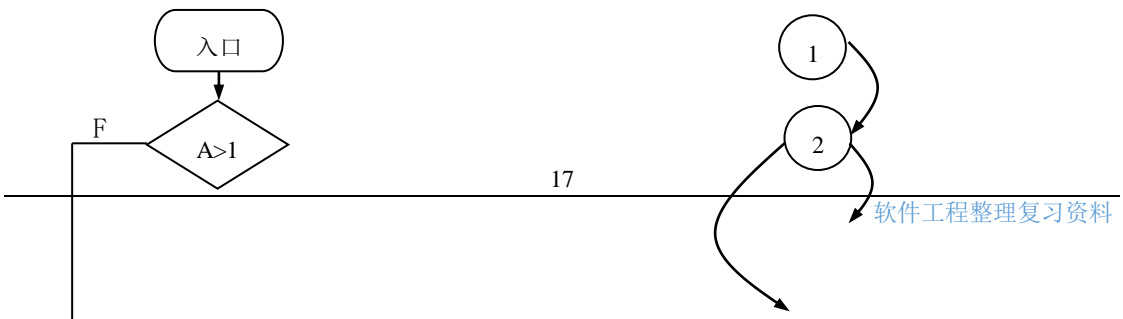
设计测试方案

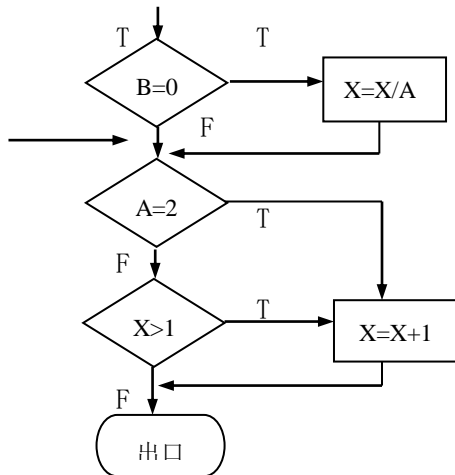
方 案	内 容			输 入	预期输出
	地区码	前缀	后缀		
1	空白	200~999 之间的三位数字	四位数字	( )276-2345	有效
2	三位数字		四位数字	(635)805-9321	有效
3	有非数字字符			(20A)723-4567	无效
4	少于三位数字			(33 )234-5678	无效
5	多于三位数字			(5555)345-6789	无效
6		有非数字字符		(345)5A2-3456	无效
7		起始位为‘0’		(345)012-3456	无效
8		起始位为‘1’		(345)132-3456	无效
9		少于三位数字		(345) 92-3456	无效
10		多于三位数字		(345)4562-3456	无效
11			有非数字字符	(345)342-3A56	无效
12			少于四位数字	(345)342- 356	无效
13			多于四位数字	(345)562-34567	无效

29. 为什么说软件的维护是不可避免的？

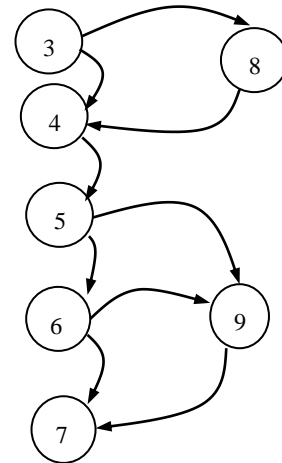
因为软件的开发过程中，一般很难检测到所有的错误，其次软件在应用过程中需要随用户新的要求或运行环境的变化而进行软件的修改或完成功能的增删等，为了提高软件的应用水平和使用寿命，软件的维护是不可避免的。

22. 试由程序流程图导出程序图





(a) 程序流程图



(b) 导出程序图

### 30. 软件的维护一般分为哪几类？

**完善性维护：**满足用户在使用过程中提出增加新的功能或修改已有功能，以满足用户日益增长的需要而进行的工作

**适应性维护：**是软件适应新的运行环境而进行的工作

**纠错性维护：**纠正开发期间未发现的遗留错误，即在程序使用期间发现的程序错误进行诊断和改正的过程

**预防性维护：**为了改善未来的可维护性或可靠性而修改软件的工作

### 31. 决定软件可维护性的因素？

- (1) 软件的可理解性、可测试性、可修改性；
- (2) 文档描述符合要求、用户文档简洁明确、系统文档完整并且标准

### 33. 说明结构化程序设计的主要思想是什么？

- 答 (1) 自顶向下、逐步求精的程序设计方法  
 (2) 使用 3 种基本控制结构、单入口、单出口来构造程序

### 34. 软件测试的基本任务是什么？测试与纠错有什么区别？

答：软件测试的基本任务是通过在计算机上执行程序，暴露程序中潜在的错误。测试和纠错目的和任务都不相同，测试是为了发现程序的错误，纠错是为了定位和纠正错误，从而消除软件故障，保证程序的可靠运行。

### 35. 软件测试包括哪些步骤？同时说明这些步骤的测试目的是什么？

- 答 (1) 单元测试，目的是代码达到模块说明书的要求  
 (2) 集成测试，目的是将经过单元测试的模块逐步组装成具有良好一致性的完整的程序  
 (3) 确认测试，目的是确认程序系统是否满足软件需求规格说明书的要求  
 (4) 系统测试，目的是检查能否与系统的其余部分协调运行，并且完成 SRS 对他的要求

### 36. 需求分析与软件设计二个阶段任务的主要区别是什么？

答：需求分析定义软件的用户需求，即定义待开发软件能做什么  
 软件设计定义软件的实现细节以满足用户需求，即研究如何实现软件

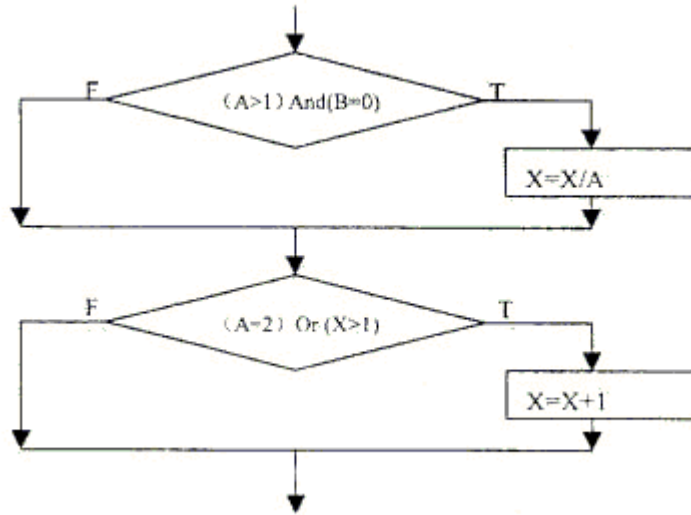
### 37. 说明软件测试和调试的目的有何区别？

答：测试的目的是判断和发现软件是否有错误      调试的目的是定位软件错误并纠

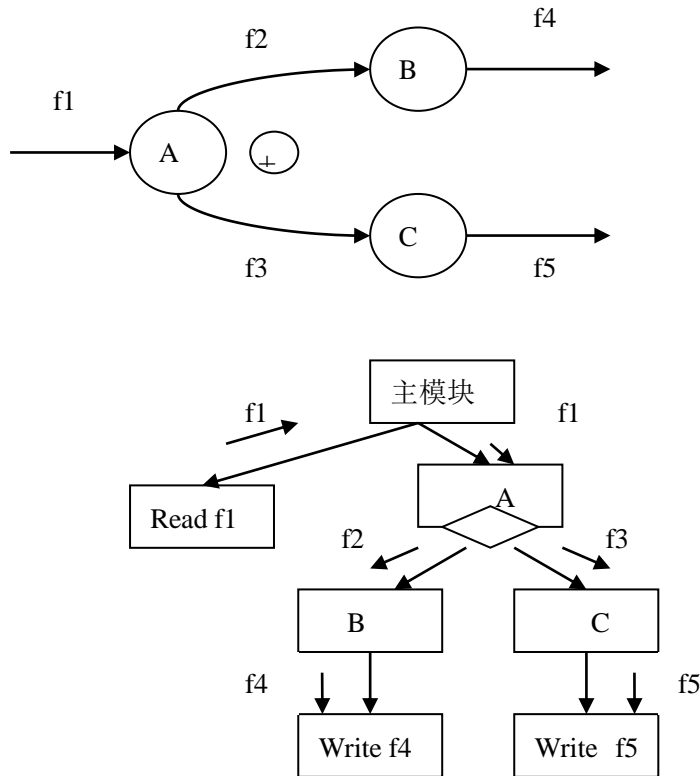
38. 根据以下被测试的流程图，选择相应的答案

- (1) A=2, B=0, X=4      (2) A=2, B=1, X=1      (3) A=2, B=0, X=1  
 A=1, B=1, X=1      A=3, B=0, X=3      A=1, B=1, X=2

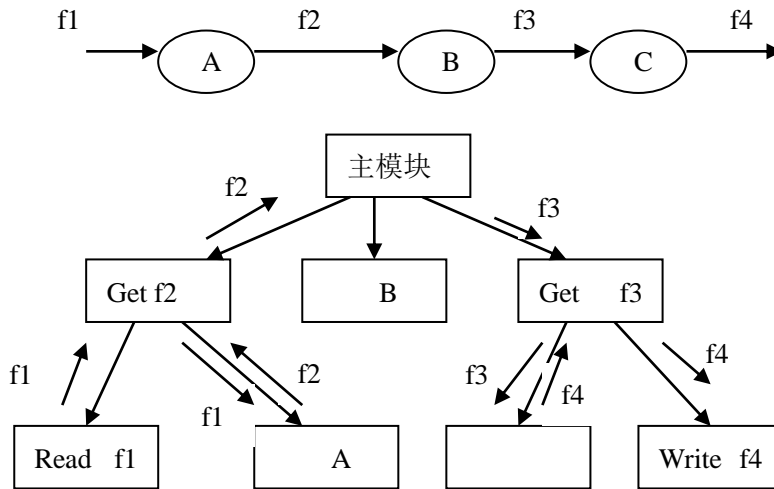
- ①在 1-3 的答案中满足判定覆盖的测试用例是\_\_1\_\_  
 ②在 1-3 的答案中满足条件覆盖的测试用例是\_\_3\_\_  
 ③在 1-3 的答案中满足判定/条件覆盖的测试用例是\_\_2\_\_



41. 把事务型数据流图映射成软件结构图：



42. 把变换型数据流图映射成软件结构图：



43. 什么是软件复用？简述软件复用和软件共享、软件移植的差别。

答：软件复用是指在构造新的软件系统的过程中，对已存在的软件人工制品的使用技术。如果是在一个系统中多次使用一个相同的软件成分，这叫软件共享；对一个软件进行修改，使它能运行于新的软、硬件平台这称作软件移植

44. 什么是软件再工程？软件再工程的主要活动有哪些？

答：软件再工程是运用逆向工程、重构等技术，在充分理解原有软件的基础上，进行分解、综合，并重新构建软件，用以提高软件的可理解性、可维护性、可复用性或演化性  
主要活动：信息库分析、文档重构、逆向工程、代码重构、数据重构、正向工程