

第一章

4.什么是软件工程？它与软件工程方法学有何关系？

软件过程是为了开发高质量的软件产品所需完成的一系列任务的框架，它规定了完成各项任务的工作步骤。

软件过程定义了运用技术方法的顺序，应该交付的文档资料，为保证软件质量和协调软件变化必须采取的管理措施，以及标志完成了相应开发活动的里程碑。软件过程是软件工程方法学的 3 个重要组成部分之一。

2、假设要求你开发一个软件，该软件的功能是把读入的浮点数开平方，所得到的结果要求精确到小数点后 4 位。一旦实现并测试完成后，该产品将被抛弃。你打算选用哪种软件生命周期模型？说明你做出选择的理由。

该软件的要求很明确，实现开平方算法的功能也很成熟，因此既无需通过原型来分析需求也无需用原型来验证设计方法。此外，一旦实现并测试完成后，该产品将被抛弃，因此无需使用有助于提高软件可维护性的增量模型或螺旋模型来开发该软件。

综上所述，为了开发这个软件，使用瀑布模型就可以了。

8.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的开发。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？

答：对这个项目的一个重要要求是，严格按照已对外公布的日期完成产品开发工作，因此，选择生命周期模型时应该着重考虑哪种模型有助于加快产品开发的进度。使用增量模型开发软件时可以并行完成开发工作，因此能够加快开发进度。

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三点结论：第一，旧版本相当于一个原型，通过收集用户对旧版本的反应，较容易确定对新版本的需求，没有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本字处理软件所需要的技术水平；第三，该软件收到广大用户的喜爱，今后很可能还要开发更新版本，因此，应该把该软件的体系结构设计成开放式，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较恰当。

第三章

3.29

用 Jackson 图描绘下述的一列火车的构成：

一列火车最多有两个火车头。只有一个火车头时则位于列车最前面，若还有第二个火车头时，则第二个火车头位于列车最后面。火车头既可能是内燃机车也可能是电器机车。车厢分为硬座车厢、硬卧车厢和软卧车厢等 3 种。硬座车厢在所有车厢的前面部分，软卧车厢在所有车厢的后面部分。此外，在硬卧车厢和软卧车厢之间还有一节餐车。

答：jackson 图善于描绘复杂事物的组成。用 jackson 图描绘一列火车的构成的方法至少有两种，一种方法是把火车分为一个车头和两个车头两类，另一种方法是把后车头作为可选的。图 3.29 给出了描绘一列火车的构成的 jackson 图。

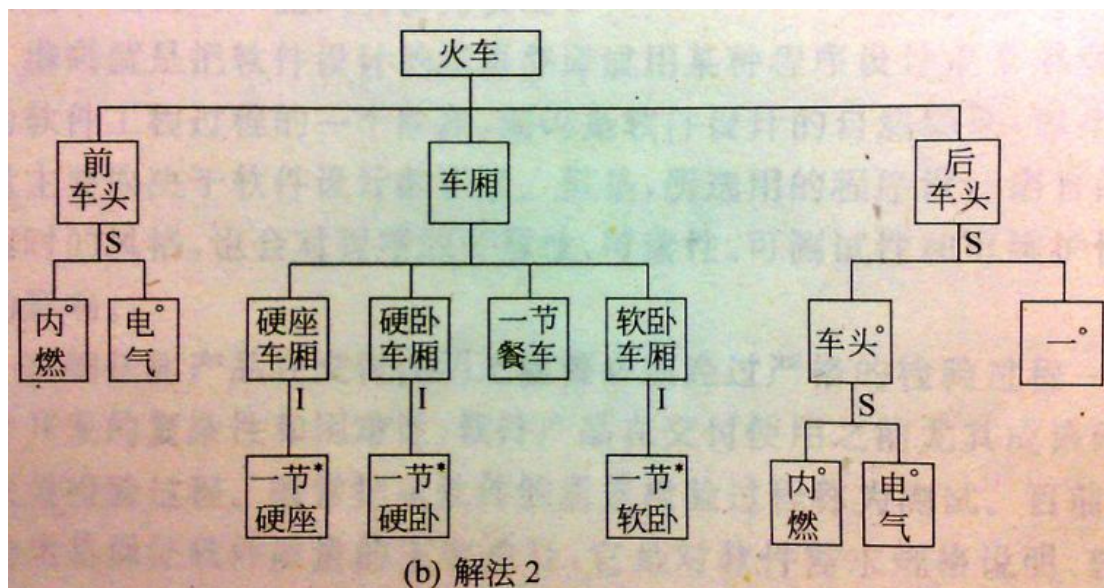
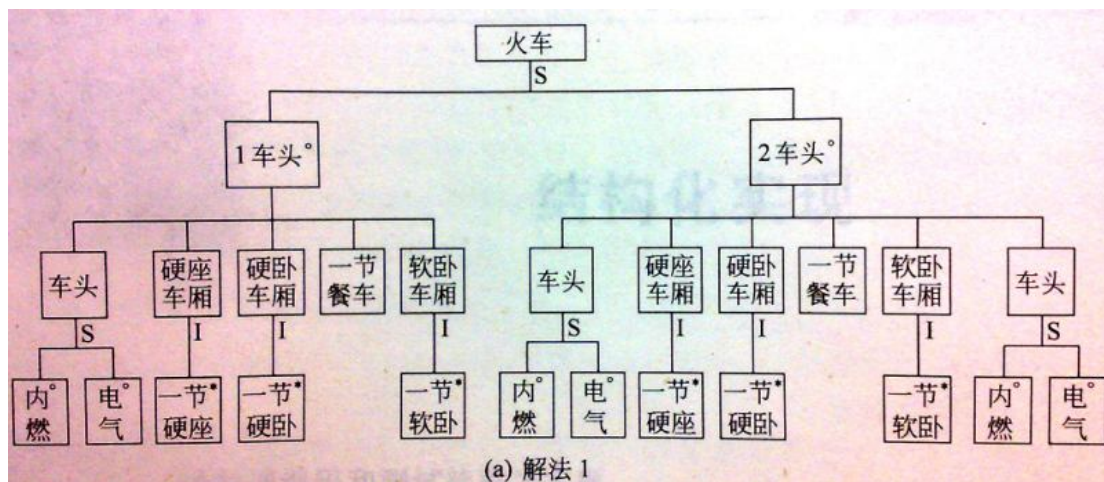


图 3.29 用 Jackson 图描绘一列火车的构成

第四章

3. 假设有一个由 5000 行 FORTRAN 语句构成的程序（经编译后大约有 25000 条机器指令），你估计在对它进行测试期间将发生多少个错误？为什么？

答：经验表明，在类似的程序中，单位长度里的错误数 Er/Ir 近似为常数。美国的一些统计数字告诉我们，通常

$$0.5 \times 10^{-2} \leq Er/Ir \leq 2 \times 10^{-2}$$

也就是说，在测试之前每 1000 条指令中大约有 5-20 个错误。

假设在该程序的每 1000 条指令中大约有 10 个错误，则估计在对它进行测试期间将发现的错误数为

$$25000 \times 10/1000 = 250$$

第五章

3、某软件公司拟采取下述措施提高他们开发出的软件产品的可维护性。请判断那些措施是正确的，那些措施不正确。

(1) 在分析用户需求时同时考虑维护的问题。

正确。在分析用户需求的同时考虑维护问题，列出将来可能变更或增加的需求，就可以在设计时为将来可能做的修改预先做一些准备，使得在用户确实提出这些维护要求时，实现起来比较容易一些。

(2) 测试完程序后，删去程序中的注解以缩短源程序的长度。

不正确。程序中的注解是提高程序可理解性的关键的内部文档，删去程序中的注解必然会降低程序的可读性和可理解性，从而降低软件的可维护性。

(3) 在软件开发过程中尽量保证各阶段文档的正确性。

正确。完整准确的文档对提高软件的可理解性有重要贡献，保证文档的正确性是提高软件可维护性的关键。

(4) 编码时尽量多用全局变量。

不正确。程序中使用的全局变量多，不仅违背局部化原理而且会使得具有公共环境耦合的模块数量增多，从而降低程序的可理解性，可修改性和可测试性，因此，这样的软件可维护性较差。

(5) 选用时间效率和空间效率尽可能高的算法。

不正确。一般说来，效率高的算法的可理解性较差，选用效率尽可能高的算法将降低软件的可维护性。事实上，程序的效率能够满足用户的需求就可以了，没有必要盲目地追求尽可能高的效率。

(6) 尽可能利用硬件特点以提高程序效率。

不正确。程序对硬件特点依赖越多，运行程序的硬件变更适应性维护的工作量也就越大。

(7) 尽可能使用高级语言编写程序。

正确。用高级语言编程时，用户可以给程序变量和程序模块赋予含义鲜明的名字，通过名字能够比较容易的把程序对象和它们所代表的实体联系起来。此外，高级语言使用的概念和符号更符合人的习惯。上述事实都使得用高级语言编写的程序更容易阅读，因此也就更容易维护。

(8) 进行总体设计时加强模块间的联系。

不正确。模块间耦合越紧密，程序就越难理解和修改，修改后测试也比较困难。因此，加强模块间的联系将降低软件的可维护性。

(9) 尽量减少程序模块的规模。

不正确。程序模块的规模很小，就会使程序中包含的模块很多，这将使模块间的接口数量大大增加，从而增加了理解，修改和测试程序的难度，降低了软件的可维护性。

(10) 用数据库系统代替文件系统来存储需要长期保存的信息。

正确。数据库系统比文件系统使用起来更方便，更安全，用数据库系统代替文件系统来存储需要长期保存的信息，可减少差错，降低改正性维护需求的数量。此外，使用数据库系统的程序比使用文件系统的程序更容易修改。上述事实表明，用数据库系统代替文件系统来存储需要长期保存的信息，将提高软件的可维护性。

(11) 用 CASE 环境或程序自动生成工具来自动生成一部分程序。

正确。自动生成的程序段没有差错，对软件的改正性维护需求自然将减少。当因用户的需求变更而需要修改程序时，可以先修改相应部分的规格说明，然后用 CASE 环境或程序自动生成工具自动生成需改动的程序，显然，这样做可以降低维护的工作量。

(12) 尽量用可重用的软件构成来组装程序。

正确。可重用的软件构件基本上没有错误，用这样的构件组装成的程序可靠性高，改正性维护需求自然就比较少。此外，可重用的软件构件适应性强，应用范围广，容易使它适应新需求，因此，用这样的构建组装的程序比较容易实现适应性或完善性维护。

(13) 使用先进的软件开发技术。

正确。用先进的软件技术开发出来的软件容易理解容易修改容易重用，因此，可维护性较好。

(14) 采用防错程序设计技术，在程序中引入自检能力。

正确。在程序中引入自检能力可以显著提高软件的可靠性，因此将明显减少改正性维护需求的数量。

(15) 把与硬件及操作系统有关的代码放到某些特定程序模块中。

正确。把和硬件及操作系统有关的代码放到某些特定的程序模块中，可以把因环境变化而必须修改的程序代码局限在少数模块内，从而更容易修改和测试。

6.代码重构与正向工程有何相同之处？有何不同之处？

答：代码重构和正向工程都需要重新设计数据结构和算法，并且需要重新编写程序代码，这些是代码重构和正向工程的相同的地方。

通常代码重构并不修改程序的体系结构，它只修改某些模块的设计细节和模块中使用的局部数据结构，并重新编写这些模块的代码。如果修改的范围扩展到模块边界之外并涉及到程序的体系结构，则代码重构变成了正向工程。

第六章习题

3.为什么说夏利牌汽车是小汽车类的特化，而发动机不是小汽车类的特化？

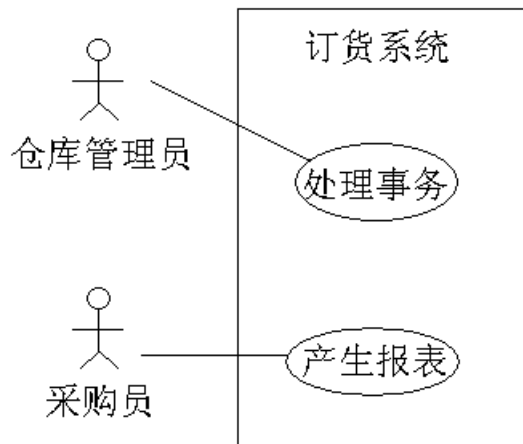
答：夏利牌汽车具有小汽车的全部属性和行为，它只不过是一种特定品牌的小汽车，因此，夏利牌汽车可以从基类“小汽车”派生出来，也就是说，夏利牌汽车是小汽车类的特化。

发动机是组成小汽车的一种零件，小汽车还有车身、车灯、轮子等许多种其他零件，小汽车所具有的许多属性和行为发动机都不具有，因此，发动机不能从小汽车类派生出来，它不是小汽车类的特化。

9.试简历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。当某种零件库存数量少于库存量临界值时就应该再次订货。

答：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，系统接收到事务信息之后应该处理事务；采购员需要使用订货系统提供的产生报表功能，以获取订货报表。综上所述，可以画出下图所示的用例图。



订货系统用例图

第七章习题

2.应该依据什么准则来评价脚本？

答：脚本必须从用户的观点来描述每个重要的功能序列，因此，脚本应该能够说明系统的一类重要功能或具体的使用方法。

3.应该依据什么准则来评价状态图？

答：状态图应该描述所有可能的状态转换。图中每条弧都要有一个引起状态转换的事件。从开始结点（初态）到每个结点（中间状态），以及从每个结点到最终结点（终态），都必须有一条路径。

第八章

3、在面向对象设计过程中为什么会调整对目标系统的需求？怎样调整需求？

答：有两种情况会导致修改由面向对象分析确定下来的系统需求：一是客户需求或系统外部环境发生了变化；二是分析员对问题域理解不透彻或缺乏领域专家帮助，以致面向对象分析模型不能完整、准确地反映客户的真实需求。

为了调整对目标系统的需求，通常只需简单地修改面向对象分析的结果（例如，增添或删除一些类，从已有类派生出新类派，调整某些类之间的关系），然后把这些修改反映到问题域子系统中。

6、从面向对象分析阶段到面向对象设计阶段，对象模型有何变化？

答：在面向对象分析阶段建立的对象模型中，对象是对问题空间中实体的抽象。随着软件开发过程进入面向对象设计阶段，这些对象逐渐变成了解空间的实体。

第九章

2.什么是强类型语言？这类语言有哪些优点？

答：按照编译时对程序中使用的数据进行类型检查的严格程度，可以把程序设计语言划分成两类。如果语言仅要求每个变量或属性属于一个对象，则是弱类型的；如果语法规则规定每个变量或属性必须准确地属于某个特定的类，则这样的语言是强类型的。

强类型的语言主要有两个优点：一是有利于在编译时发现程序错误，二是增加了优化的可能性。因此，强类型语言有助于提高程序的可靠性和运行效率。

4.为什么说参数化类有助于提高可重用性？

答：在实际的应用程序中，往往有这样一些软件元素（即函数、类等软件成分），从它们的逻辑功能看，彼此是相同的，所不同的主要是处理的对象类型不同。

所谓参数化类，就是使用一个或多个类型去参数化一个类的机制，有了这种机制，程序员就可以先定义一个参数化的类模板（即在类定义中包含以参数形式出现的一个或多个类型），然后在使用时把数据类型作为参数传递进来，从而把这个类模板在不同的应用程序中重复使用，或在同一程序的不同部分重复使用。

—

1. 填空(每空 0.5 分，共 20 分)

(1) 软件生命周期可划分为定义、开发和维护3个时期，通常把这3个时期再细分为8个阶段，它们是①问题定义、②可行性研究、③需求分析、④总体设计、⑤详细设计、⑥编码和单元测试、⑦综合测试和⑧维护，其中维护阶段的工作量是8个阶段中最大的。

(2) 可行性研究的任务是从技术、经济和操作等三个方面研究软件项目的可行性。

(3) 至少应该从一致性、现实性、完整性和有效性等4个方面研究软件需求的正确性，其中完整性和有效性这两个方面的正确性必须有用户的积极参与才能验证，而且为了验证这两个方面的正确性往往需要开发原型系统。

(4) 软件总体设计时应遵循模块化、抽象、逐步求精、信息隐藏、局部化和模块独立等6条基本原理。详细设计通常以结构程序设计技术为逻辑基础，因为从软件工程观点看，可理解性是软件最重要的质量标准之一。

(5) 软件测试的目的是发现错误，通常把测试方法分为黑盒测试和白盒测试两大类。因为通常不可能做到穷尽测试，所以精心设计测试方案是保证达到测试目的所必须的。

(6) 面向对象方法用对象分解取代了传统方法的功能分解。

(7) 在面向对象的软件中，类是对具有相同数据和相同操作的一组相似对象的定义；实例是由某个特定的类所描述的一个具体对象。

2. 按下述要求完成给出的程序流程图，即在答案栏内写出图中 A、B、C、D、E 的正确内容。(每栏 3 分，共 15 分)

给程序输入二维数组 $W(I, J)$ ，其中 $I \leq M$ ， $J \leq N$ (M 和 N 均为正整数)。程序打印出数组中绝对值最小的元素值 $Q=W(K, L)$ ，及其下标 K, L 的值。假定数组中仅有一个绝对值最小的元素。

应该完成的程序流程图如图 A.1 所示。

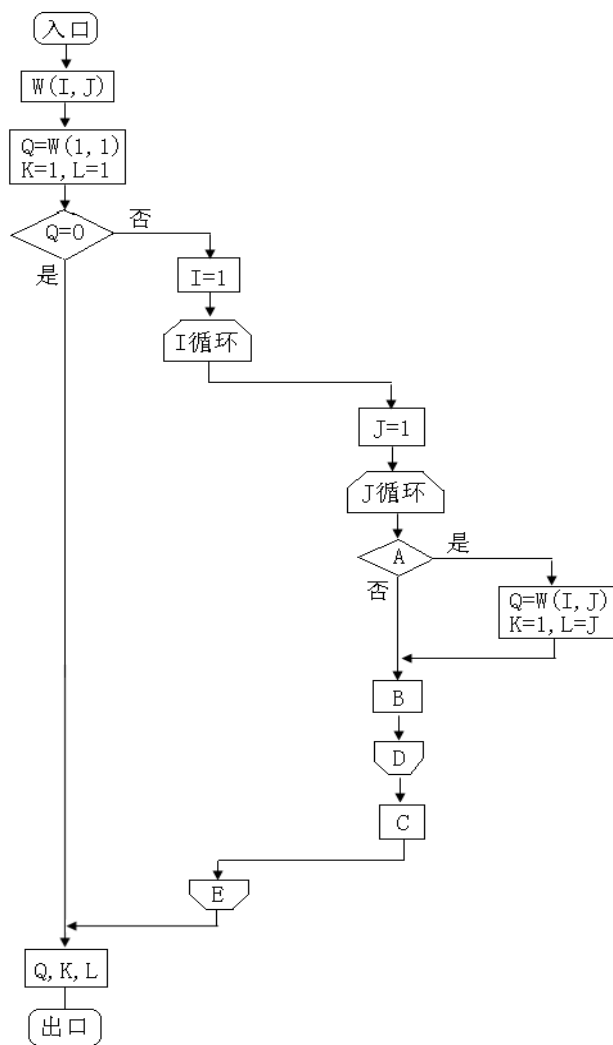


图 A.1 要求完成的程序流程图

[答案栏]

A: $|Q|=|W(I, J)|$

B: $J=J+1$

C: $I=I+1$

D: J 循环 UNTIL $J>N$

E: I 循环 UNTIL $I>M$

3. 下面给出了用盒图(见图 A.2)描绘的一个程序的算法, 请用逻辑覆盖法设计测试方案, 要求做到语句覆盖和路径覆盖。(共 15 分)

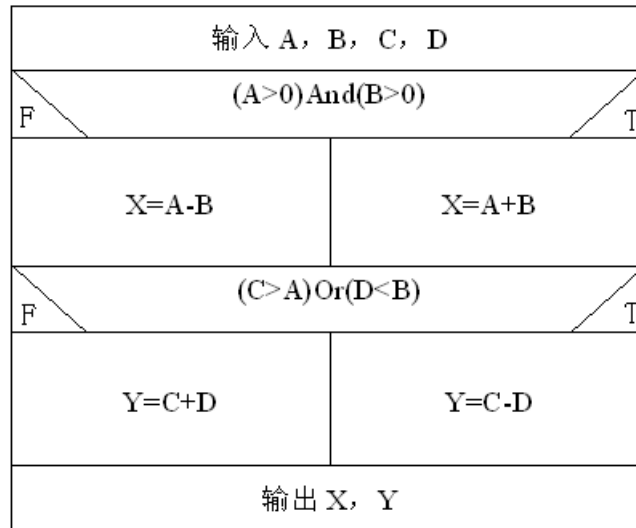


图 A.2 用盒图描绘的算法

答:

(1) 实现语句覆盖的测试方案如下

①覆盖 THEN 部分的语句, 即使得条件(A>0)And(B>0)和条件(C>A)Or(D<B)均为真

输入: A=1, B=1, C=2, D=0

预期输出: X=2, Y=2

②覆盖 ELSE 部分的语句, 即使得条件(A>0)And(B>0)和条件(C>A)Or(D<B)均为假

输入: A=0, B=-1, C=0, D=1

预期输出: X=1, Y=1

(2) 实现路径覆盖的测试方案如下

①覆盖两个条件均为假的路径

输入: A=-1, B=-2, C=-3, D=3

预期输出: X=1, Y=0

②覆盖第一个条件为假第二个条件为真的路径

输入: A=-1, B=-2, C=1, D=-3

预期输出: X=1, Y=4

③覆盖第一个条件为真第二个条件为假的路径

输入: A=1, B=1, C=0, D=2

预期输出: X=2, Y=2

④覆盖两个条件均为真的路径

输入: A=1, B=2, C=2, D=1

预期输出: X=3, Y=1

4. 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 1 位数字不是 0; 校外电话又分为本市电话和外地电话两类, 拨校外电话需先拨 0, 如果是本市电话再接着拨 8 位电话号码(第 1 位不是 0), 如果是外地电话则先拨区码(3~5 位数字), 再拨当地电话号码(7 或 8 位数字, 第 1 位不是 0)。

请定义上述的电话号码(共 15 分)

答:

电话号码=[校内号码|校外号码]

校内号码=非零数字+3 位数字
 校外号码=[本市号码|外地号码]
 本市号码=数字零+非零数字+7 位数字
 外地号码=数字零+区码+当地号码
 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
 7 位数字=7{数字}7
 区码=3{数字}5
 当地号码=非零数字+6{数字}7
 数字=[0|1|2|3|4|5|6|7|8|9]

5. 请说明多态重用与继承重用的关系。(共 15 分)

答：多态重用实际上是一种特殊的继承重用，是充分利用多态性机制支持的继承重用。一般说来，使用多态重用方式重用已有的类构件时，在子类中需要重新定义的操作比较少，因此，多态重用方式的成本比继承重用方式的成本低。

6. 请建立下述的图书馆馆藏出版物的对象模型。(共 20 分)

一家图书馆藏有书籍、杂志、小册子、电影录像带、音乐 CD、录音图书磁带和报纸等出版物，供读者借阅。这些出版物有出版物名、出版者、获得日期、目录编号、书架位置、借出状态和借出限制等属性，并有借出、收回等服务。

答：图书馆馆藏出版物的对象模型示于图 B.1 中。

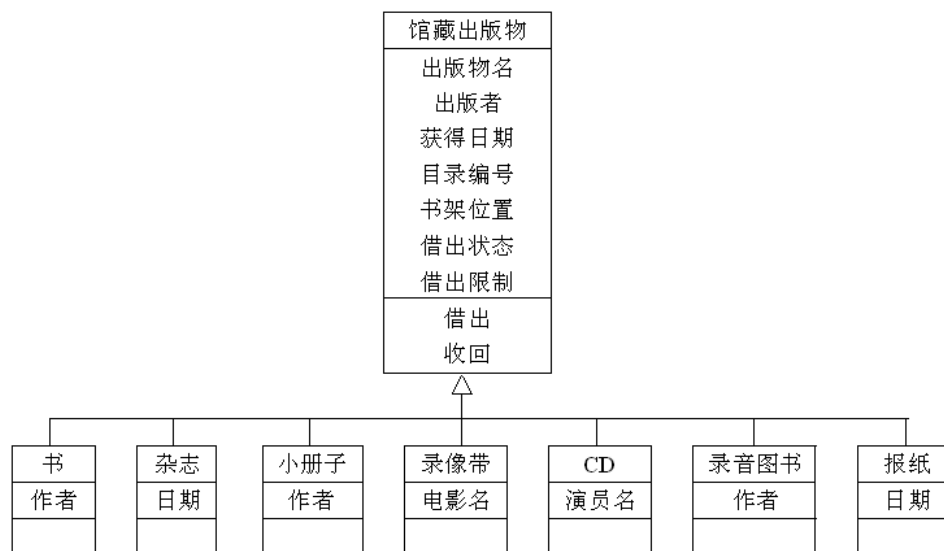


图 B.1 图书馆馆藏出版物的对象模型

二

1. 将下列各对事物之间的关系（继承、聚集或一般关联）写在括弧内。（共 10 分）

- (1) 小汽车——富康牌小汽车 (继承)
- (2) 人员——雇员 (继承)
- (3) 图书馆——期刊阅览室 (聚集)
- (4) 书——图书馆馆员 (一般关联)
- (5) 小汽车——司机 (一般关联)
- (6) 读者——借出的书 (一般关联)
- (7) 班级——学生 (聚集)
- (8) 教师——教授 (继承)
- (9) 丈夫——妻子 (一般关联)
- (10) 列车——餐车 (聚集)

2. 判断下列各项是类还是类的实例，答案写在括弧内。（共 10 分）

- (1) 我的小汽车 (实例)
- (2) 人员 (类)
- (3) 王晓明 (实例)
- (4) 交通工具 (类)
- (5) 教授 (类)
- (6) 计算机系 (实例)
- (7) 中国工人 (类)
- (8) 清华大学学生 (类)
- (9) 日本国 (实例)
- (10) 喷气式战机 (类)

3. 从供选择的答案中选出在下列叙述中括弧内字母应该代表的正确内容，把答案写在答案栏内。（共 10 分）

(1) 一组语句在程序的多处出现，为了节省内存空间把这些语句放在一个模块中，该模块的内聚度是 (A) 的。

(2) 将几个逻辑上相似的成分放在一个模块中，该模块的内聚度是 (B) 的。

(3) 模块中所有成分都使用共同的数据，该模块的内聚度是 (C) 的。

(4) 模块内某些成分的输出是另一些成分的输入，该模块的内聚度是 (D) 的。

(5) 模块中所有成分结合起来完成单独一项任务，该模块的内聚度是 (E) 的。它具有简明的外部界面，由它构成的软件易于理解，测试和维护。

【供选择的答案】

A~E 1. 功能性； 2. 顺序性； 3. 通信性； 4. 过程性； 5. 偶然性； 6. 瞬时性；
7. 逻辑性

【答案栏】

A: 5 B: 7 C: 3
D: 2 E: 1

4. 图 A.3 是用程序流程图描绘的处理算法，请把它改画为等价的盒图。（共 10 分）

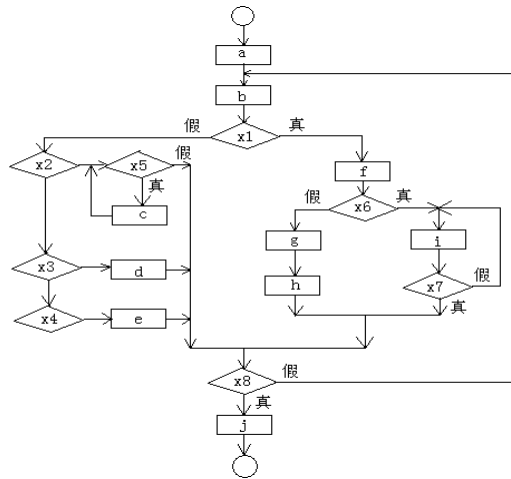
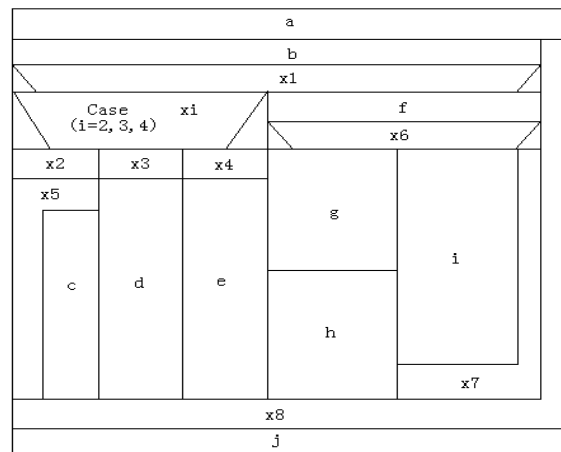


图 A.3

答：如图所示：



5. 有一个长度为 48000 条机器指令的程序，第一个月由甲、乙二人分别测试它。甲改正了 20 个错误，使程序的平均无故障时间达到 8 小时。乙在测试该程序的另一个副本时改正了 24 个错误，其中 6 个错误与甲改正的相同。然后，由甲一个人继续测试这个程序。请问（共 20 分）

（1）刚开始测试时程序中的错误总数 E_T 是多少？

（2）为使平均无故障时间达到 240h，如果甲不利用乙的工作成果，则他还需再改正多少个错误？

（3）为使平均无故障时间到达 480h，如果甲利用了乙的工作成果，则他还需再改正多少个错误？

答：

(1) 刚开始测试时程序中错误总数为

$$E_T = (24/6) * 20 = 80$$

(2) 由方程

$$8 = 48000 / (K(E_T - 20))$$

解得

$$K = 48000 / 8(80 - 20) = 100$$

由方程

$$240=48000/100(80-E_{C1})$$

解得

$$E_{C1}=78$$

$$78-20=58$$

即，为使平均无故障时间到达 240h，如果甲不利用乙的工作成果，则他还需再改正 58 个错误。

(3) 由方程

$$480=48000/100(80-E_{C2})$$

解得

$$E_{C2}=79$$

$$79-20-(24-6)=41$$

即，为使平均无故障时间到达 480h，如果甲利用了乙的工作成果，则他还需要再改正 41 个错误。

6. 办公室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缺纸，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，准备接受复印命令；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。（共 20 分）

答：描绘复印机行为的状态转换图如图 B. 3 所示。

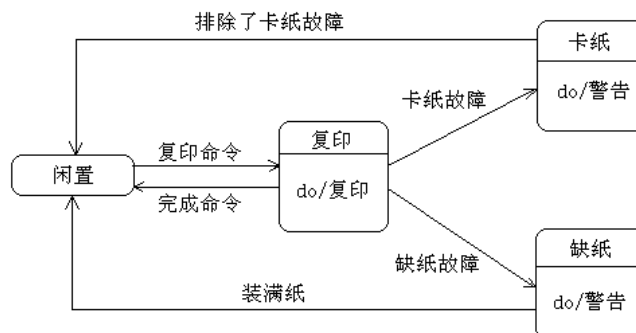


图 B. 3

7. 请建立下述杂货店问题的对象模型。（共 20 分）

一家杂货店想使其库存管理自动化。这家杂货店拥有能记录顾客购买的所有商品的名称和数量的销售终端。顾客服务台也有类似的终端，以处理顾客的退货。它在码头有另一个终端用于处理供应商发货。肉食部和农产品部有终端用于输入由于损耗导致的损失和折扣。

答：杂货店问题的对象模型如图 B. 4 所示。

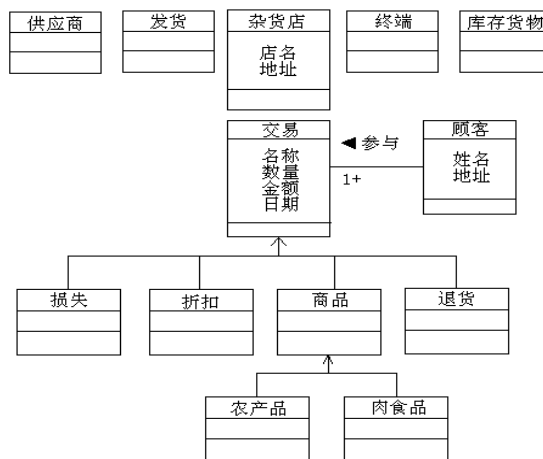


图 B. 4

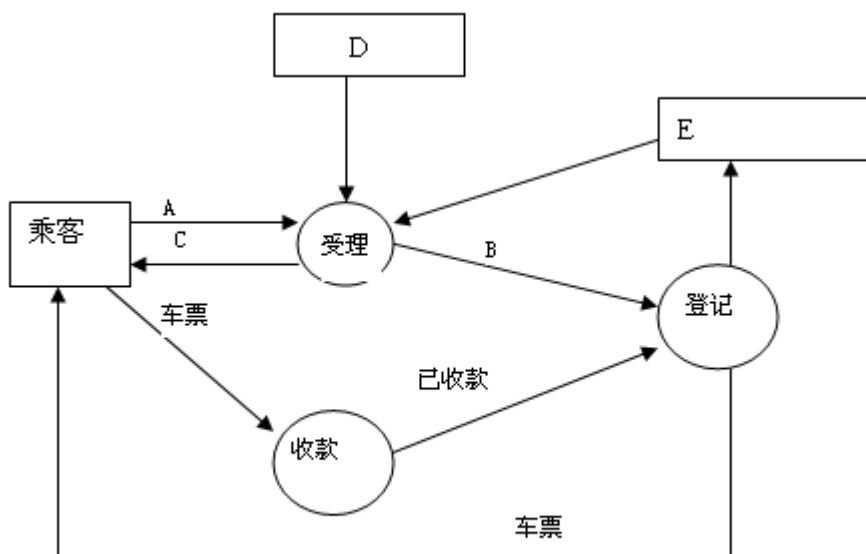
三

1.判断下述提高软件可维护性的措施是否正确，正确的在括弧内写对，错的写错。（共 10 分）

- (1) 在进行需求分析时同时考虑维护问题。 (对)
- (2) 完成测试后，为缩短源程序长度而删去程序中的注解。 (错)
- (3) 尽可能在软件开发过程中保证各阶段文档的正确性。 (对)
- (4) 编码时尽可能使用全局变量。 (错)
- (5) 选择时间效率和空间效率尽可能高的算法。 (错)
- (6) 尽可能利用硬件的特点以提高效率。 (错)
- (7) 重视程序结构的设计，使程序具有较好的层次结构。 (对)
- (8) 使用维护工具或软件工程环境。 (对)
- (9) 进行概要设计时尽量加强模块间的联系。 (错)
- (10) 提高程序可读性，尽量使用高级语言编程。 (对)

2. 为开发一个铁路自动售票系统（该系统预计从 2004 年使用到 2014 年），请完成下面的数据图（见图 A.4）和数据字典，即从供选择的答案中选出 A、B、C、D 和 E 的内容，并给出 F、G 和 H 的内容，填在答案栏中。（共 20 分）

(1) 数据流图



图A.4 铁路自动售票系统数据流图

【供选择的答案】

A~E ①车次表；②接受；③售票记录；④购票请求；⑤拒绝。

(2) 数据字典

购票请求=F

乘车日期=G

到站=4{字母}20

字母=【“A” . . “Z” | “a” . . “z”】

车次=“001” . . “999”

拒绝=【无车次|无票】

无车次=“no train”

无票=“no ticket”

接受=“to sale”

已收款=“yes”

车次表={起站+止站+车次}

起站=止站=到站

售票记录={乘车日期+起站+止站+车次+座号}

车厢号=“01” . . “20”

座位号=H

注：

① “01” . . “20” 表示数字范围从 01 到 20；

② 乘车日期应给出年、月、日，例如，2004/06/21；

③ 假设每个车厢有 100 个座位。

【答案栏】

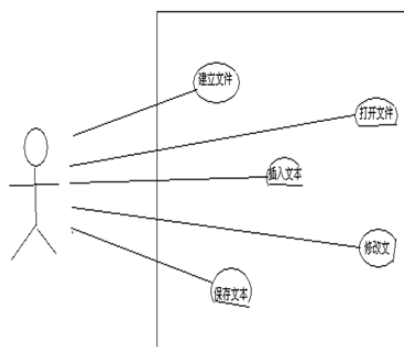
A: 购票请求 B: 接受 C: 拒绝

D: 车次表 E: 售票记录 F: 乘车日期+到站+车次

G : “2004” . . “2014” + “/” +1” . . ” 12” +” /” +” 01” +” 31”

H: “001 “.” 100 “

3. 画出简化的文本编辑程序的用例图，该编辑程序的主要功能有，建立文件、打开文件、插入文件、修改文件和保存文件。（共 10 分）

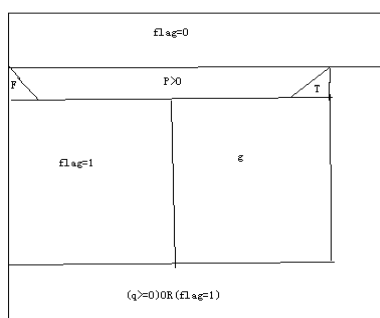


答：简化的文件编辑程序的用例图如图 B. 5

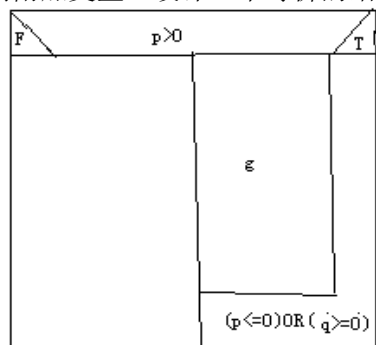
4. 图 A . 5 所示的程序流程描绘了一个非结构化的程序。（共 20 分）

(1) 为什么说它是非结构化的？

(2) 利用附加变量 flag，设计一个等价的结构化程序，用盒图描绘你的设计结果。



(3)不用附加变量，设计一个等价的结构化程序，用盒图描绘你的设计结果。

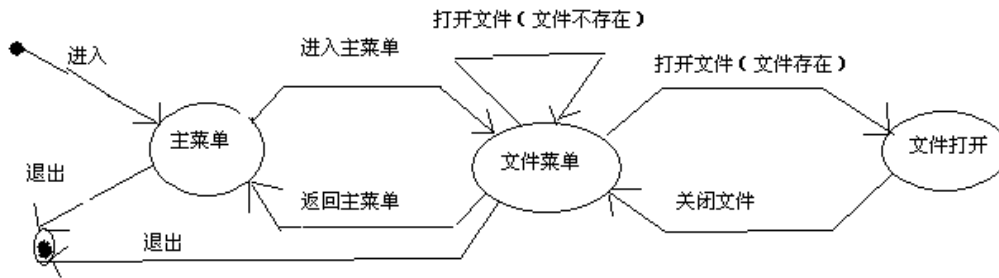


(1) 图 A. 5 所示的程序循环控制结构有两个出口，因此是非结构化的程序。

(2) 利用附加变量 flag 设计的等价的结构化程序如图 B. 6

(3) 不用附加变量 flag 设计的等价的程序如图 B. 7

5. 画出图形用户界面（GUI）的状态转换图，该界面具有一个主菜单和一个带有文件打开命令的文件菜单，在每个菜单上都有一条退出命令。假设每次只有一个文件能够打开。（共 20 分）



图形用户界面的状态转换图如图 B. 8

6. 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，再跟上一个符号 E，在加上另一个可选符号（+或-）及一个或多个二进制位。浮点二进制的形式化定义如下：

$\langle \text{floating-point binary} \rangle ::= [\langle \text{sign} \rangle] \langle \text{bitstring} \rangle E [\langle \text{sign} \rangle] \langle \text{bitstring} \rangle$
 $\langle \text{sign} \rangle ::= + \mid -$
 $\langle \text{bitstring} \rangle ::= \langle \text{bit} \rangle [\langle \text{bitstring} \rangle]$
 $\langle \text{bit} \rangle ::= 0 \mid 1$

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。

假设一个有穷状态机以一串字符为输入，根据浮点二进制的定义判断输入的字符

答：判断浮点二进制数的有穷状态机如图 B. 9

