**全国计算机技术与软件专业技术资格（水平）考试**

**2009年下半年 软件设计师 下午试卷**

（考试时间 14:00～16:30 共 150 分钟）

|  |
| --- |
| **请按下述要求正确填写答题纸** |

1.在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。

2.在答题纸的指定位置填写准考证号、出生年月日和姓名。

3.答题纸上除填写上述内容外只能写解答。

4.本试卷共6道题，试题一至试题四是必答题，试题五至试题六选答 1 道。每

题 15 分，满分 75 分。

5.解答时字迹务必清楚，字迹不清时，将不评分。

6.仿照下面例题，将解答写在答题纸的对应栏内。

**例题**

2009 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是（1）

月（2）日。

因为正确的解答是“11 月 4 日”，故在答题纸的对应栏内写上“11”和“4”

（参看下表）。

|  |  |
| --- | --- |
| 例题 | 解答栏 |
| （1） | 11 |
| （2） | 4 |

|  |
| --- |
| 试题一至试题四是必答题 |

**试题一**

现准备为某银行开发一个信用卡管理系统CCMS,该系统的基本功能为：

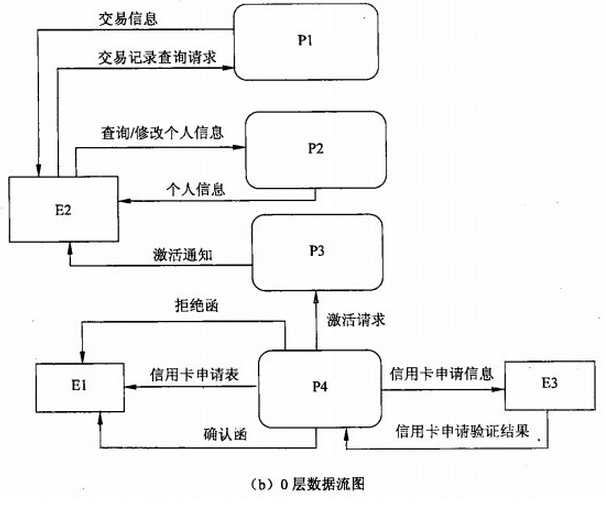
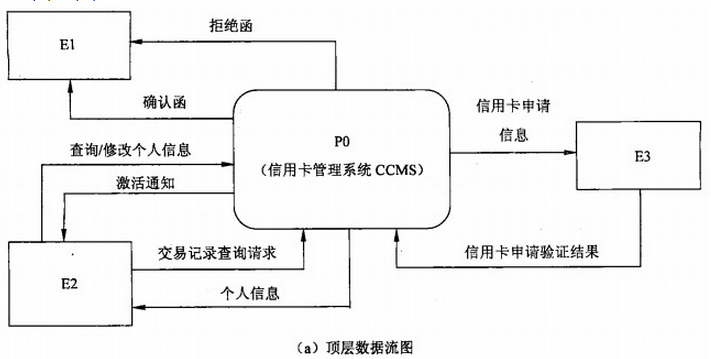
1. 信用卡申请。非信用卡客户填写信用卡申请表，说明所要申请的信用卡类型及申请者的基本信息，提交CCMS。如果信用卡申请被银行接受，CCMS将记录该客户的基本信息，并发送确认函给该客户，告知客户信用卡的有效期及信贷限额；否则该客户 将会收到一封拒绝函。非信用卡客户收到确认函后成为信用卡客户。

2. 信用卡激活。信用卡客户向CCMS提交激活请求，用信用卡号和密码激活该信用卡。激活操作结束后，CCMS将激活通知发送给客户，告知客户其信用卡是否被成功激活。

3. 信用卡客户信息管理。信用卡客户的个人信息可以在CCMS中进行在线管理。每位信用卡客户可以在线查询和修改个人信息。

4. 交易信息查询。信用卡客户使用信用卡进行的每一笔交易都会记录在CCMS中。信用卡客户可以通过CCMS查询并核实其交易信息（包括信用卡交易记录及交易额)。

下图（a)和（b)分别给出了该系统的顶层数据流图和0层数据流图的初稿。



**【问题1】**

根据说明，将图（a)中的E1〜E3填充完整。

**【问题2】**

图（a)中缺少三条数据流，根据说明，分别指出这三条数据流的起点和终点。（注： 数据流的起点和终点均采用图中的符号和描述）

**【问题3】**

图（b)中有两条数据流是错误的，请指出这两条数据流的名称，并改正。（注：数据流的起点和终点均采用图中的符号和描述）

**【问题4】**

根据说明，将图（b)中P1〜P4的处理名称填充完整。

**试题二**

某公司拟开发一多用户电子邮件客户端系统，部分功能的初步需求分析结果如下：

(1) 邮件客户端系统支持多个用户，用户信息主要包括用户名和用户密码，且系统 中的用户名不可重复。

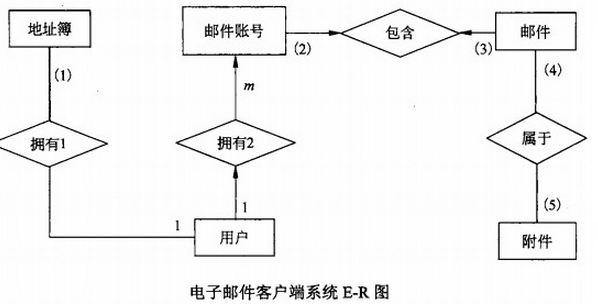
(2) 邮件账号信息包括邮件地址及其相应的密码，一个用户可以拥有多个邮件地址 (如userl@123.com)。

(3) 一个用户可拥有一个地址簿，地址簿信息包括联系人编号、姓名、电话、单位 地址、邮件地址1、邮件地址2、邮件地址3等信息。地址簿中一个联系人只能属于一个用户，且联系人编号唯一标识一个联系人。

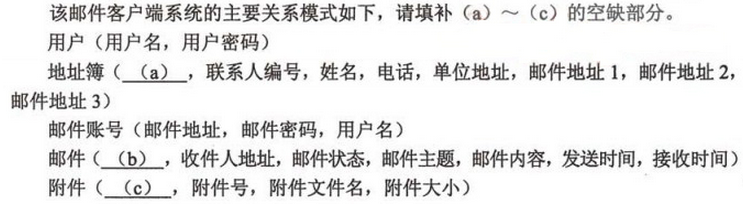
(4) 一个邮件账号可以含有多封邮件，一封邮件可以含有多个附件。邮件主要包括 邮件号、发件人地址、收件人地址、邮件状态、邮件主题、邮件内容、发送时间、接收时间。其中，邮件号在整个系统内唯一标识一封邮件，邮件状态有已接收、待发送、已发送和已删除4种，分别表示邮件是属于收件箱、发件箱、已发送箱和废件箱。一封邮件可以发送给多个用户。附件信息主要包括附件号、附件文件名、附件大小。一个附件 只属于一封邮件，附件号仅在一封邮件内唯一。

**【问题1】**

根据以上说明设计的E-R图如下图所示，请指出地址簿与用户、电子邮件账号与邮 件、邮件与附件之间的联系类型。



**【问题2】**



**【问题3】**

(1) 请指出问题2中给出的地址簿、邮件和附件关系模式的主键，如果关系模式存在外键请指出。

(2) 附件属于弱实体吗？请用50字以内的文字说明原因。

**试题三**

某企业为了方便员工用餐，为餐厅开发了一个订餐系统（COS: Cafeteria Ordering System),企业员工可通过企业内联网使用该系统。

企业的任何员工都可以查看菜单和今日特价。

系统的顾客是注册到系统的员工，可以订餐（如果未登录，需先登录)、注册工资支付、预约规律的订餐，在特殊情况下可以覆盖预订。

餐厅员工是特殊顾客，可以进行备餐、生成付费请求和请求送餐，其中对于注册工资支付的顾客生成付费请求并发送给工资系统。

菜单管理员是餐厅特定员工，可以管理菜单。

送餐员可以打印送餐说明，记录送餐信息（如送餐时间）以及记录收费（对于没有注册工资支付的顾客，由送餐员收取现金后记录）。

顾客订餐过程如下：

1. 顾客请求查看菜单；

2. 系统显示菜单和今日特价；

3. 顾客选菜；

4. 系统显示订单和价格；

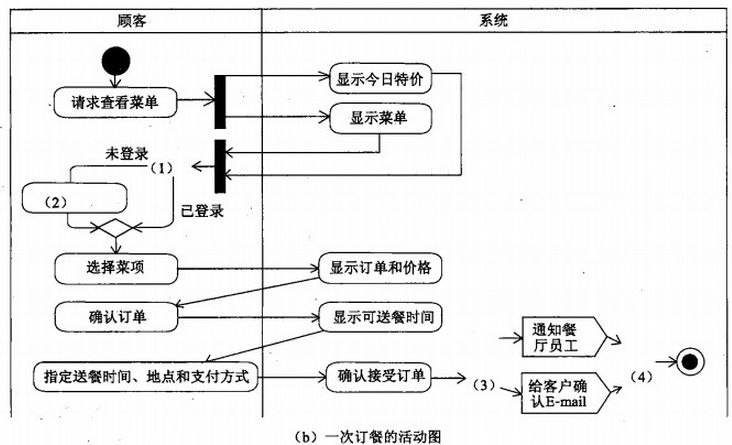
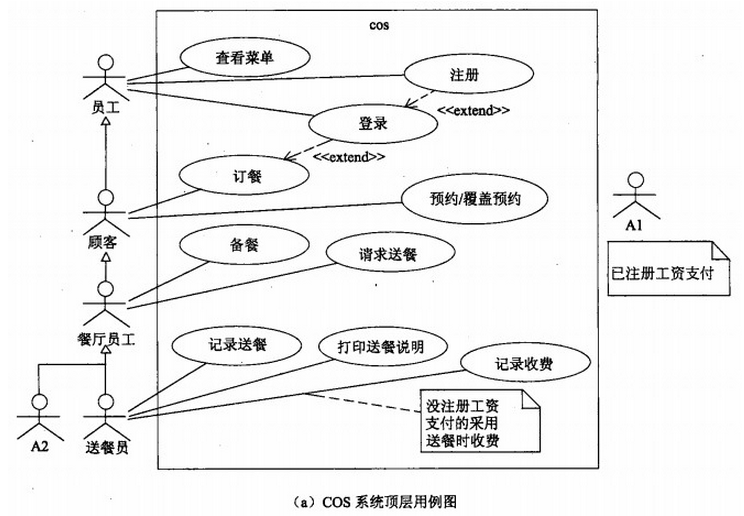
.5.顾客确认订单；

6. 系统显示可送餐时间；

7. 顾客指定送餐时间、地点和支付方式；

8. 系统确认接受订单，然后发送E-mail给顾客以确认订餐，同时发送相关订餐信息通知给餐厅员工。

系统采用面向对象方法开发，使用UML进行建模。系统的顶层用例图和一次订餐的活动图初稿分别如下图（a)和（b)所示。



**【问题1】**

根据说明中的描述，给出图 (a) 中A1和A2所对应的参与者。

**【问题2】**

根据说明中的描述，给出图 (a)中缺少的四个用例及其所对应的参与者。

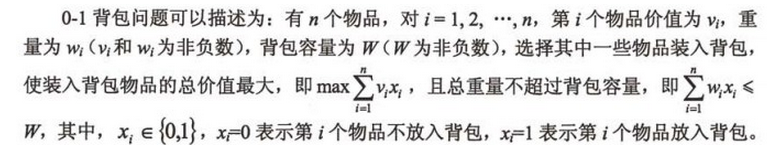
**【问题3】**

根据说明中的描述, 给出图 (b) 中（1)〜（4)处对应的活动名称或图形符号。

**【问题4】**

指出图（a)中员工和顾客之间是什么关系，并解释该关系的内涵。

**试题四**



**【问题1】**

用回溯法求解此0-1背包问题，请填充下面伪代码中（1)〜（4)处空缺。

回溯法是一种系统的搜索方法。在确定解空间后，回溯法从根结点开始，按照深度优先策略遍历解空间树，搜索满足约束条件的解。对每一个当前结点，若扩展该结点已经不满足约束条件，则不再继续扩展。为了进一步提髙算法的搜索效率，往往需要设计一个限界函数，判断并剪枝那些即使扩展了也不能得到最优解的结点。现在假设已经设计了BOUND（v,w,k,W）的函数，其中v、w、k和W分别表示当前已经获得的价值、当前背包的重量、已经确定是否选择的物品数和背包的总容量。对应于搜索树中的某个结点， 该函数值表示确定了部分物品是否选择之后，对剩下的物品在满足约束条件的前提下进行选择可能获得的最大价值，若该价值小于等于当前已经得到的最优解，则该结点无需再扩展。

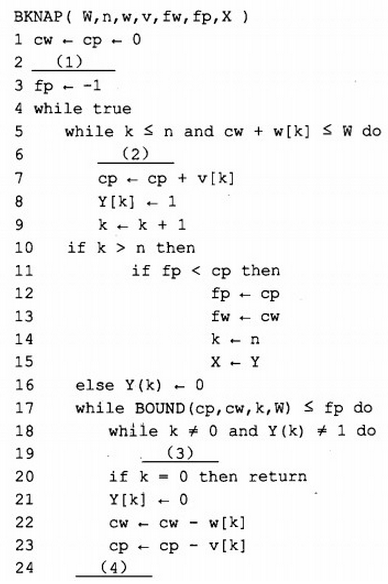
下面给出0-1背包问题的回溯算法伪代码。

函数参数说明如下：

W：背包容量；n：物品个数；w:重量数组；v:价值数组；fw:获得最大价值时背包的重量；fp:背包获得的最大价值；X：问题的最优解。

变量说明如下：

CW：当前的背包重量；cp：当前获#的价值；k：当前考虑的物品编号；Y:当前已获得的部分解。

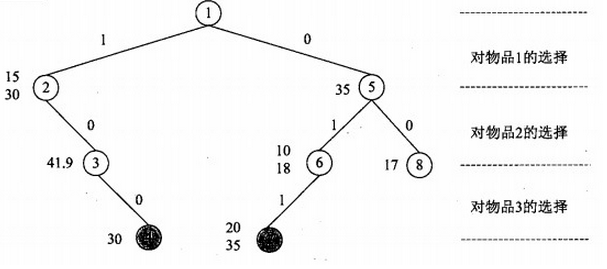


**【问题2】**

考虑下表所示的实例，假设有3个物品，背包容量为22。



下图是根据上述算法构造的搜索树，其中结点的编号表示了搜索树生成的顺序，边上的数字1/0分别表示选择/不选择对应物品。除了根结点之外，每个左孩子结点旁边的上下两个数字分别表示当前背包的重量和已获得的价值，右孩子结点旁边的数字表示扩展了该结点后最多可能获得的价值。为获得最优解，应该选择物品（5),获得的价值 为（6)。.



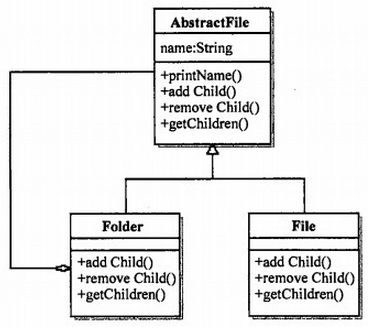
对于上述实例，若采用穷举法搜索骜个解空间，则搜索树的结点数为（7)，而用了上述回溯法，搜索树的结点数为（8)。

|  |
| --- |
| 从下列的 2 道试题（试题五至试题六）中任选 1 道解答。  如果解答的试题数超过 1 道，则题号小的 1 道解答有效。 |

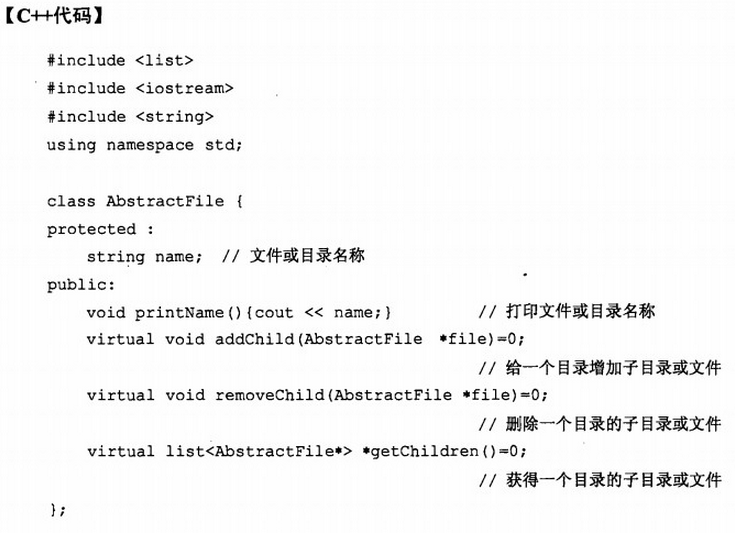
**试题五**

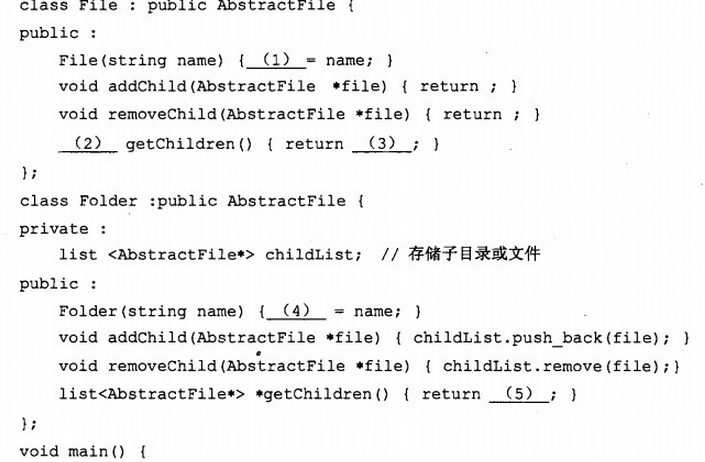
阅读下列说明和C++代码，将应填入（n)处的字句写在答题纸的对应栏内。

现欲构造一文件/目录树，采用组合（Composite)设计模式来设计，得到的类图如下图所示：



**【问题1】**



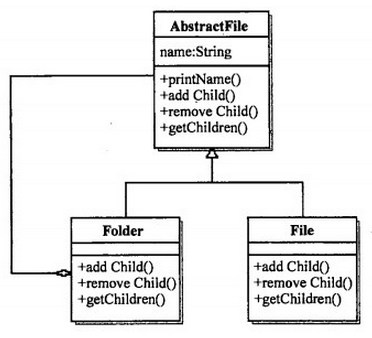




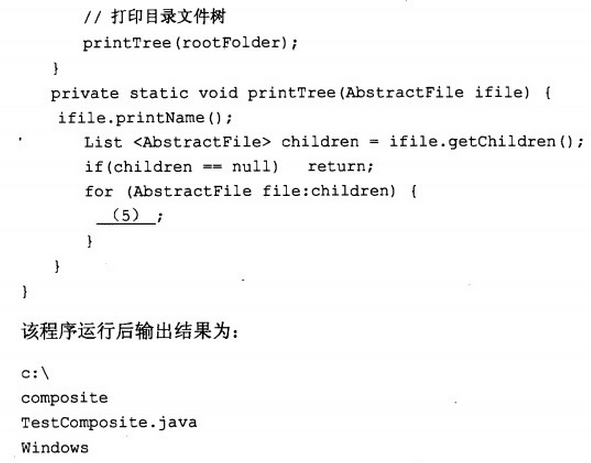
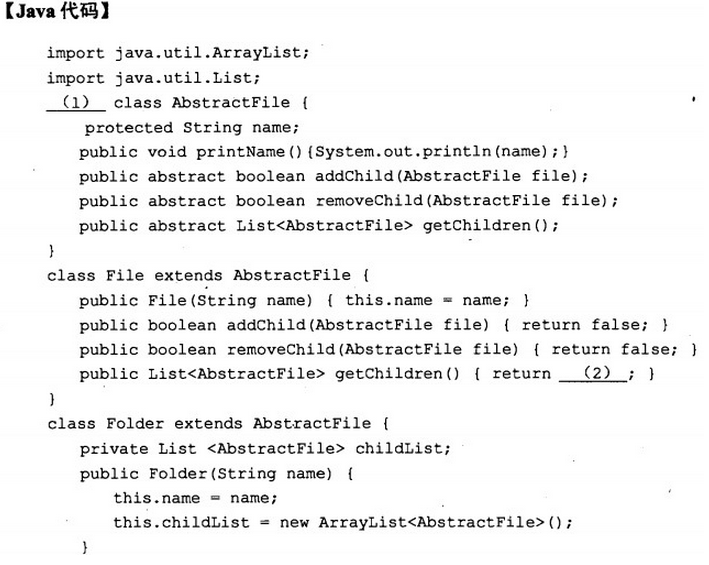
**试题六**

阅读下列说明和Java代码，将应填入（n)处的字句写在答题纸的对应栏内- 【说明】

现欲构造一文件/目录树，采用组合（Composite)设计模式来设计，得到的类图如下图所示：



**【问题1】**

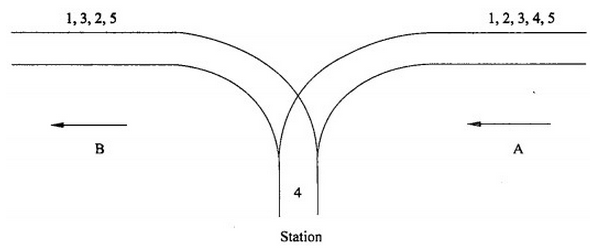


**【试题七】**

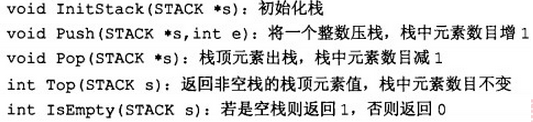
阅读以下说明和C程序，将应填入（n)处的字句写在答题纸的对应栏内。

**【说明】**

现有n(n<1000)节火车车厢，顺序编号为1，2, 3,…，n，按编号连续依次从A 方向的铁轨驶入，从B方向铁轨驶出，一旦车厢进入车站（Station)就不能再回到A方向的铁轨上；一旦车厢驶入B方向铁轨就不能再回到车站，如下图所示，其中Station 为栈结构，初始为空且最多能停放1000节车厢。



下面的C程序判断能否从B方向驶出预先指定的车厢序列，程序中使用了栈类型 STACK，关于栈基本操作的函数原型说明如下：



**【问题1】**

