全国计算机技术与软件专业技术资格（水平）考试

2016 年下半年 软件设计师 下午试卷

案例分析题 试题一

（共 15 分）

阅读下列说明，回答问题 l 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某证券交易所为了方便提供证券交易服务，欲开发一证券交易平台，该平台的主要功能如下：

（1）开户。根据客户服务助理提交的开户信息，进行开户，并将客户信息存入客户记录中，账户信息（余额等）存入账户记录中；

（2）存款。客户可以向其账户中存款，根据存款金额修改账户余额；

（3）取款。客户可以从其账户中取款，根据取款金额修改账户余额；

（4）证券交易。客户和经纪人均可以进行证券交易（客户通过在线方式，经纪人通过电话），将交易信息存入交易记录中；

（5）检查交易。平台从交易记录中读取交易信息，将交易明细返回给客户。 现采用结构化方法对该证券交易平台进行分析与设计 获得如图 1-1 所示的上下 文数一据流图和图 1-2 所示的 O 层数据流图。

【问题 1]（3 分）

使用说明中的词语，给出图 1-1 中的实体 E1-E3 的名称。

【问题 2】（3 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1-D3 的名称。

【问题 3】(4 分)

根据说明和图中的术语，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】(5 分)

实际的证券交易通常是在证券交易中心完成的，因此，该平台的“证券交易”功能需将交易信息传递给证券交易中心。针对这个功能需求，需要对图 1-1 和图 1-2 进行哪些修改，请用 200 字以内的文字加以说明。

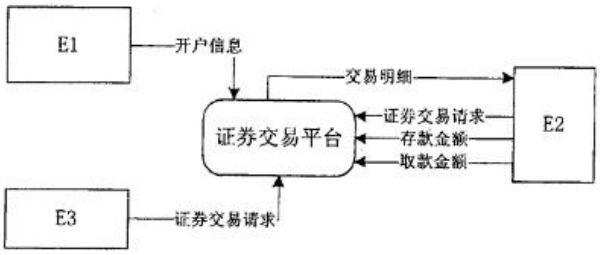


图 1-1 上下文数据流图

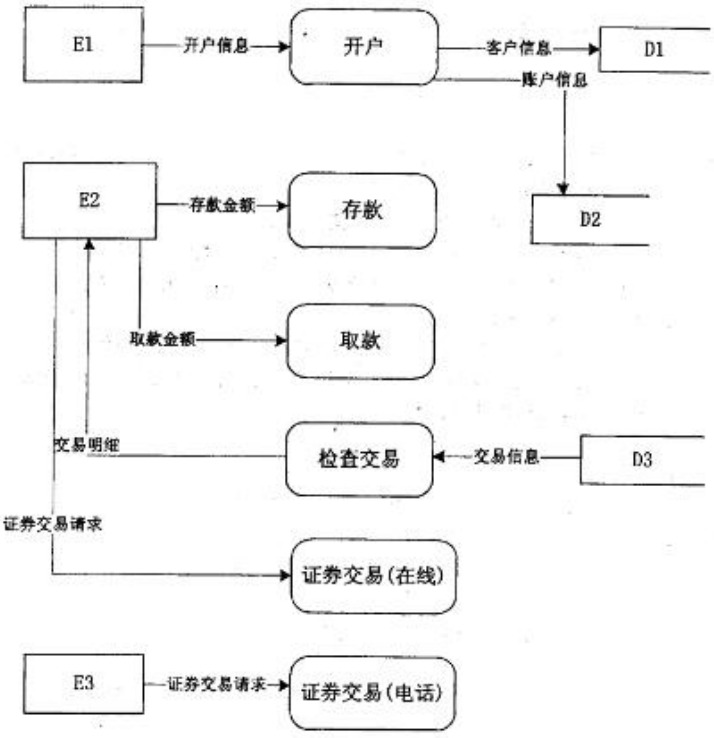


图 1-2 0 层数据流图

试题二（共 15 分）

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某宾馆为了有效地管理客房资源，满足不同客户需求，拟构建一套宾馆信息管理系统，以方便宾馆管理及客房预订等业务活动。

【需求分析结果】

该系统的部分功能及初步需求分析的结果如下：

（1）宾馆有多个部门，部门信息包括部门号、部门名称、电话、经理。每个部门可以有多名员工，每名员工只属于一个部门；每个部门只有一名经理，负责管理本部门。

（2）员工信息包括员工号、姓名、岗位、电话、工资，其中，员工号唯一标识员工关系中的一个元组，岗位有经理、业务员。

（3）客房信息包括客房号（如 1301、1302 等）、客房类型、收费标准、入住状态（已入住 / 未入住），其中客房号唯一标识客房关系中的一个元组，不同客房类型具有不同的收费标准。

（4）客户信息包括客户号、单位名称、联系人、联系电话、联系地址，其中客户号唯一标识客户关系中的一个元组。

（5）客户预订客房时，需要填写预订申请。预订申请信息包括申请号、客户号、入住时间、入住天数、客房类型、客房数量，其中，一个申请号唯一标识预订申请中的一个元组；一位客户可烈有多个预订申请，但一个预订申请对应唯一的二位客户。

（6）当客户入住时，业务员根据客户的预订申请负责安排入住客房事宜。

安排信息包括客房号、姓名、性别、身份证号、入住时间、天数、电话，其中客房号、身份证号和入住时间唯一标识一次安排。一名业务员可以安排多个预订申请，一个预订申请只由一名业务员安排，而且可安排多间同类型的客房。

【概念模型设计】 根据需求阶段收集的信息，设计的实体联系图如图 2-1 所示。

图 2-1 实体联系图

【关系模式设计】 部门（部门号，部门名称，经理，电话）员工（员工号，a，姓名，岗位，电话，工资）客户 b 联系人，. 联系电话，联系地址． 客房（客房号，客房类型，收费标准，入住状态）

预订申请（ c ）入住时间,天数，客房类型，客房数量）安排（申请号，客房号，姓名,性别，(d ） ，天数，电话，业务员）

【问题 1】(4 分)

根据问题描述，补充四个联系，完善图，2-1，的实体联系图.联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型为 1:1、1:n 和 m:n （或 1:1，和 1：*和*：*)。

【问题 2】（8 分）

（1）根据题意、，将关系模式中的空（a）~.（d）补充完整,并填入答题纸对应的位置上。

（2）给出“预订申请”和“安排”关系模式的主键和外键。

【问题 3】（3 分）

【关系模式设计】中的“客房”关系模式是否存在规范性问题，请用丑 100 字以内文字解释你的观点（若存在问题，应说明如何修改“客房”关系模式）.

试题三（共 15 分）

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某种出售罐装饮料的自动售货机.（ Vending Machine）的工作过程描述如下：

（l）顾客选择所需购买的饮料及数量。

（2）顾客从投币口向自动售货机中投入硬币（该自动售货机只接收硬币）。硬币器收集投入的硬币并计算其对应的价值。如果所投入的硬币足够购买所需数量的这种饮料且饮料数量足够，则推出饮料，计算找零，顾客取走饮料和找回的硬币；如果投入的硬币不够或者所选购的饮料数量不足，则提示用户继续投入硬币或重新选择饮料及数量。

（3）一次购买结束之后，将硬币器中的硬币移走（清空硬币器），等待下一次交易。自动售货机还设有一个退币按钮，用于退还顾客所投入的硬币。已经成功购买饮料的

钱是不会被退回的。

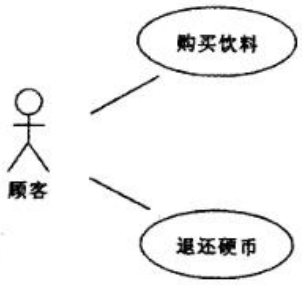


图 3-1 用例图

现采用面向对象方法分析和设计该自动售货机的软件系统，得到如图 3-1 所示的用例图，其中，用例“购买饮料”的用例规约描述如下。

参与者：顾客。

主要事件流： 1．顾客选择需要购买的饮料和数

量，投入硬币；

2．自动售货机检查顾客是否投入足够的硬币；

3．自动售货机检查饮料信存仓中所选购的饮料是否足够；

4．自动售货机推出饮料；

5．自动售货机返回找零。

各选事件流：

2a．若投入的硬币不足，则给出提示并退回到 1；

3a．若所选购的饮料数量不足 j?则给出提示并退回到 1 。 根据用例“购买饮料”得到自动售货机的 4 个状态：“空闲”状态、“准备服

务”状态、“可购买”状态以及“饮料出售”状态，对应的状态图如图 3-2 所示。

所设计的类图如图 3-3 所示。

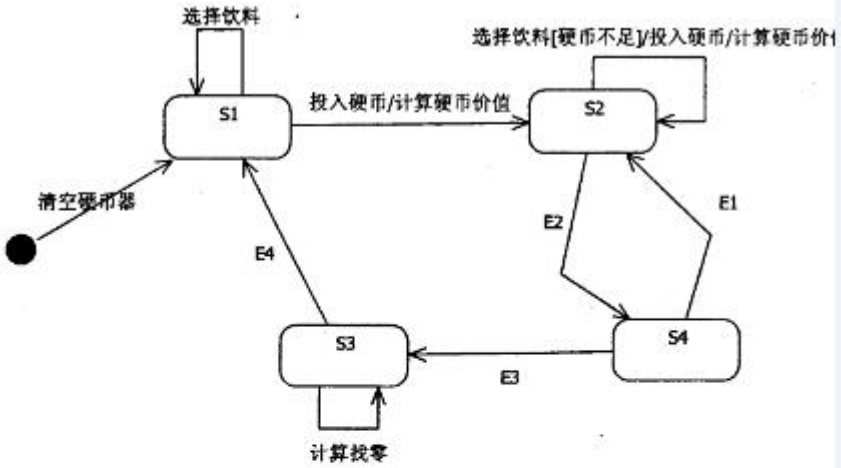


图 3-2 状态图

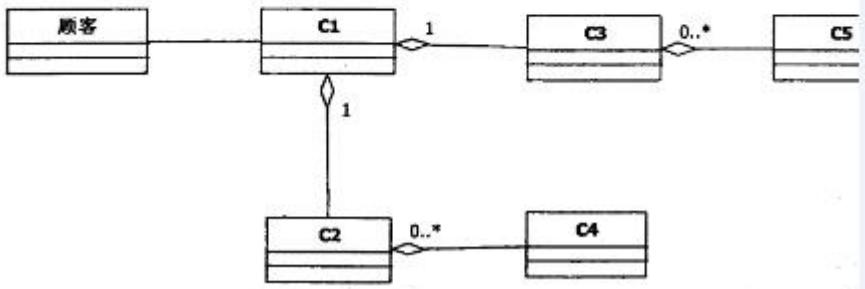


图 3-3 类图

【问题 1】(6 分)

根据说明中的描述，使用说明中的术语，给出图 3-2 中的 S1～S4 所对应的状态名。

【问题 2】(4 分)

根据说明中的描述，使用说明中的术语，给出图 3-2 中的 E1～E4 所对应的事件名。

【问题 3】(5 分)

根据说明中的描述，使用说明中的术语，给出图 3-3 中 CI～C5 所对应的类

名。

试题四（共 15 分）

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

模式匹配是指给定主串 t 和子串 s，在主串 t 中寻找子串 s 的过程，其中 s 称为模式。

如果匹配成功，返回 s 在 t 中的位置，否则返回-1。

KMP 算法用 next 数组对匹配过程进行了优化。KMP 算法的伪代码描述如下：

1．在串 t 和串 s 中，分别设比较的起始下标 i=J=O

2．如果串 t 和串 s 都还有字符，则循环执行下列操作：

（1）如果 j=-l 或者 t[i]-s[j]，则将 i 和 j 分别加 1，继续比较 t 和 s 的下一个字符；

（2）否则，将 j 向右滑动到 next[j]的位置，即 j =next[J]

3．如果 s 中所有字符均已比较完毕，则返回匹配的起始位置（从 1 开始）；否则返回一 1．

其中，next 数组根据子串 s 求解。求解 next 数组的代码已由 get_next 函数给出。

【C 代码】

（1）常量和变量说明

t，s：长度为悯铂 Is 的字符串

next:next 数组，长度为 Is

（2）C 程序

```
#include <stdio．h>
#nclude   <stdliB．h>
#include <string．h>
/*求 next【】的值*/
void get_next（  int *next, char *s,int Is）
                                        {
    int i=0，j=-1；
    next[0]=-1；/*初始化 next[0]* /
    while（i< ils）{/*还有字符* / if
    （j=-1l ls[i]=s[ j]）{ / *匹配* /
    j++；
    i++；
    if（  s[i]— s[ jl）
    next [i]- next[j];
    else
   Next[i]=j；
  }
else
  J= next[ j]；
```

```
  }
}
 int kmp (   int *next, char *t ,char *s, int.lt, int Is  )
 {
        inti= 0,j =0 ;
        while  ( i<lt &&   (     1     )
           { if (  j=-1 II     2_ ) {
                    i++ ;
                  j ++;
             } else
                        (3)  :
}
if  ( j>= ls )
     Retum    ( 4 )
else .
retum-1 ;
```

【问题 1】(8 分)

根据题干说明，填充 C 代码中的空（1）~（4）.

【问题 2】(2 分)

根据题干说明和 C 代码，分析出 kmp 算法的时间复杂度为 （5）(主串和子的长度分别为 It 和 Is，用 O 符号表示)。

【问题 3】（5 分）

根据 C 代码，字符串“BBABBCAC”的 next 数组元素值为（6）(直接写素值，之间用逗号隔开)。若主串为“AABBCBBABBCACCD”，子串为“BBABBCAC 则函数 Kmp 的返回值是（7）.

从下列的 2 道试题（试题五至试题六）中任选 1 道解答。请在答题纸上的指定位置处将所选择试题的题号框涂黑。若多涂或者未涂题号框，则对题号最小的一道试题进行评分。

试题五（共 15 分）

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。

【说明】

某发票（Invoice）由抬头（Head）部分、正文部分和脚注（Foot）部分构成。现采用装饰（Decorator）模式实现打印发票的功能，得到如图 5-1 所示的类图。

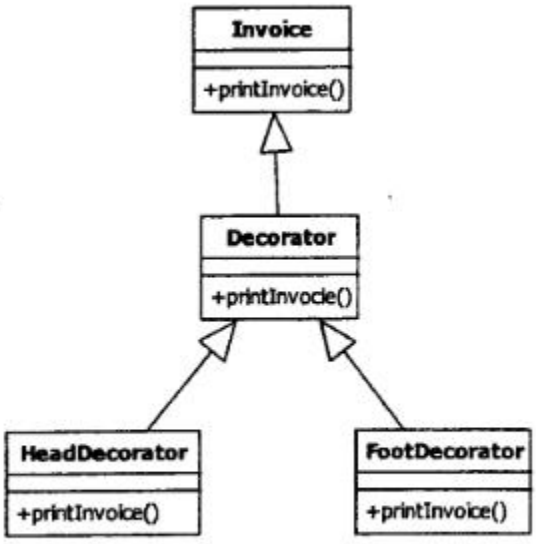


图 5-1　类图

【C++代码】

```
#include <iostream>
using namespace std ;
class invoice{
public :
   ( 1 )   {
    cout《 "This is the content of the invoice!"《 endl;
     }
};
class Decorator : public invoice
       { Invoice *ticket;
public:
      Decorator ( Invoice *t )          { ticket = t; }
       void printinvoice ( ) {
```

```
            if ( ticket != NULL )
            ( 2 );
        }
    };
    class HeadDecorator : public
    Decorator{ public:
        HeadDecorator ( Invoice*t ) : Decorator ( t ) { }
         void printinvoice0 {
                cout《 "This is the header of the invoice! " << endl;
        ( 3 )        ;
        }
} ;
class FootDecorator : public
  Decorator{ public:
      FootDecorator ( invoice *t ) : Decorator ( t ) { }
         void printInvoice ( ) {
                  ( 4 )   ;
                 cout《 "This is the footnote of the invoice!"《 endl;
        }
};
int main ( void )
        { Invoice t;
```

```
        FootDecorator f ( &t ) ;
        HeadDecorator h ( &f ) ;
        H.printlnvoice ( ) ;
          cout< <    "____" < < endl;

FootDecorator a ( NULL ) ;
        HeadDecorator b (   ( 5 )     ) ;
        B . printinvoice ( ) ;
        return 0;
    }
```

程序的输出结果为：

```
This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
----------------------------
This is the header of the invoice!
This is the footnote of the invoice!
```

试题六（共 15 分）

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】

某发票（Invoice）由抬头（Head）部分、正文部分和脚注（Foot）部分构成。现采用装饰（ Decorator）模式实现打印发票的功能，得到如图 6-1 所示的类图。

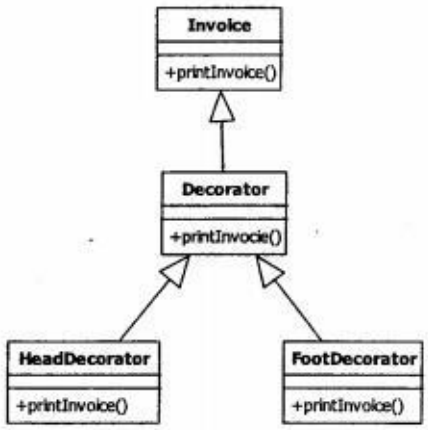


图 6-1　类图

【java 代码】

```
class invoice{
        public void printInvoice(){ :
                System.out.println ( "This is the content of the invoice!");
        }
 }

 class Decorator : extends Invoice
       { protected Invoice ticket;
```

```
        public Decorator ( Invoice t ) {
        ticket = t;
        }
        public void printinvoice (  )
                { if ( ticket != NULL )
                        ( 1 );
        }
    }
    class FootDecorator extends
            Decorator{ public FootDecorator
             ( Invoice t ) {
                    super ( t ) ;
             }
            public void printinvoice (){
                    Systent.out.println( "This is the header of the invoice! ");
                        ( 2 )    ;
             }
}
    class FootDecorator extends Decorator {
      public FootDecorator ( invoice
       t ) :{ super(t);
    }
```

```
        public void printInvoice (  ) {
                    (3)    ;
                        Systent.out.println( "This is the header of the invoice! ");
        }
    }
    Class test {
        public static void main ( string[] args )
        { Invoice t =new invioce();
        Invoice ticket;
        Ticket=      (4)    ;
        Ticket. Printinvoice();
        Systent.out.println( "------------- ")
        Ticket=      (5)    ;
        Ticket. Printinvoice();
    }
}
```

程序的输出结果为：

```
This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
  ---------------------------
 This is the header of the invoice!
```

This is the footnote of the invoice!

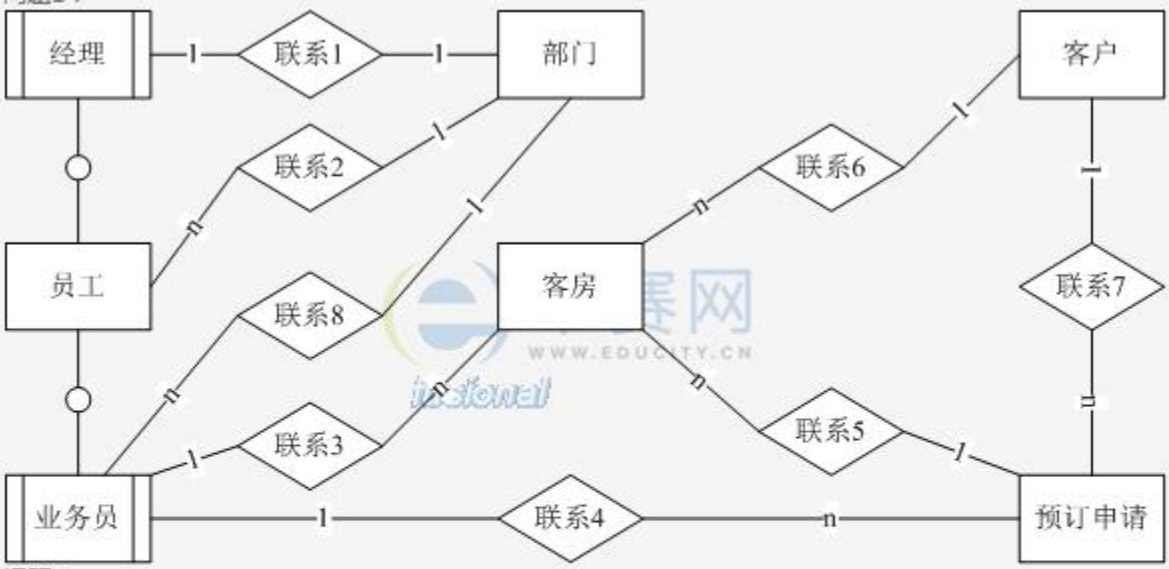
参考答案及解析

问题1 E1：客户服务助理，E2：客户，E3：经纪人。
问题2 D1：客户记录，D2：账户记录，D3：交易记录。
问题3
数据流名称：修改账户余额，起点：存款，终点：D2。
数据流名称：修改账户余额，起点：取款，终点：D2。
数据流名称：交易信息存入交易记录，起点：证券交易，终点：D3。
问题4 增加外部实体“证券交易中心”，原来“证券交易中的交易信息”的数据流终点改为“证券交易中心”，数据流“检测交心”。

试题分析：

本题问题1要求识别E1-E3具体为哪个外部实体，通读试题说明，可以了解到适合充当外部实体的包括：客户、客户服务助理、经过将顶层图与题目说明进行匹配得知。如：从图中可看出E1会向交易平台发出数据流“开户信息”；而从试题说明“根据客户服户，并将客户信息存入客户记录中，账户信息存入账户记录中”可以看出，E1对应是客户服务助理。E2、E3同理可得。
本题问题2要求识别存储，解决这类问题，以图的分析为主，配合说明给存储命名，因为存储相关的数据流一般展现了这个存储可以看到D1中有客户信息，而D2中有账户信息，题目说明中又有“根据客户服务助理提交的开户信息，进行开户，并将客户信账户记录中。”自然D1应为客户记录，D2应为账户记录。同理，D3为交易记录。
问题3分析：
缺失数据流1
名称：修改账户余额，起点：存款，终点：D2。
理由：从试题说明“客户可以向其账户中存款，根据存款金额修改账户余额”可以看出，这个功能有操作“根据存款金额修改账功能应有数据流“存款”至D2，而0层图没有。
缺失数据流2:
名称：修改账户余额，起点：取款，终点：D2。
理由：从试题说明“客户可以从其账户中取款，根据取款金额修改账户余额”可以看出，这个功能有操作“根据取款金额修改账功能应有数据流“取款”至D2，而0层图没有。
缺失数据流3
名称：交易信息存入交易记录，起点：证券交易，终点：D3。
理由：从试题说明“客户和经纪人均可以进行证券交易，将交易信息存入交易记录中”可以看出，这个功能有操作“将交易信息解到从该功能应有数据流“证券交易”至D3，而0层图没有。
本题问题4强调实际的证券交易通常是在“证券交易中心”完成，这个“证券交易中心”属于典型的外部实体，所以需要增加外平台的“证券交易”功能需将交易信息传递给证券交易中心，因此将原来“证券交易中的交易信息”的数据流终点改为“证券交易”中的起点改为“证券交易中心”。

问题1：



问题2：

（a）部门名称。

（b）客户号、单位名称、申请号、客房号。

（c）申请号、客户号、业务员。

（d）身份证号、入信时间、客户号。

“预订申请”关系模式中的主键是申请号，外键是申请号、客户号、入住时间。

“安排”关系模式中的主键是申请号、客户号、身份证号、入住时间，外键是申请号、客户号、客房号。

问题3：

根据试题中的描述，客房信息中客房号是唯一标识客房关系的一个元组，即可以作为唯一的主键。在客房关系模式中，不存在

范性问题。

参考答案：

问题1：S1：空闲，S2：准备服务，S3：饮料出售，S4：可购买。

问题2：E1：饮料数量不足，E2：硬币数量足够，E3：推出饮料，E4：返回找零。

问题3：C1：自动售货机，C2：硬币器，C3：饮料储存仓，C4：硬币，C5：饮料。

试题分析：

本题问题1系统中的状态图，是对状态转换的图形化表达。从题目的说明部分可知，在状态转换过程中，涉及到的状态一共有

料出售。从状态图涉及的转换可知S1~S4分别为：空闲、准备服务、饮料出售、可购买。关于状态转换的分析如下：

（1）清空硬币器后，自动售货机等待下一次交易，进入空闲状态。此时可任意的进行饮料选择数量，一旦顾客投入硬币，

（2）当自动售货机进行准备服务状态时，开始计算硬币价值，如果硬币不够则提示顾客继续投入硬币。如果硬币足够，则

（3）进行可购买状态后，自动售货机判断饮料数量。如果数量不够，则返回准备服务状态提示用户重新选择饮料。如果数

态。

（4）进行饮料出售状态后，自动售货机计算找零，并返回进入空闲状态等待下一次交易。

本题问题2主要是分析四种状态中的跳转事件。根据状态图和试题主要事件流的描述可以推出事件E1是饮料数量不足，事件E

料，事件E4是返回找零。

本题问题3根据主要事件流的描述，可以推断出C1~C5的类名分别对应自动售货机、硬币器、饮料储存仓、硬币、饮料。

参考答案：

问题1：
（1）：j<ls;
（2）：t[i]==s[j];
（3）：get_next(next, s, ls);
　　j=next[j];
（4）：i+1-ls;
问题2：
问题3：（6）：[-1,-1,1,-1,-1,2,0,0]，（7）5。

试题分析：

本题问题1根据KMP算法的伪代码描述进行推导。
根据伪代码中第2步可以推导（1）是判断字符串s是否还有字符，即j<ls。i表示字符串t的下标，j表示字符串s的下标。
根据伪代码第2.1步可以推导（2）是判断字符串t和字符串s当前位置的字符是否相同，即t[i]==s[j]。
根据伪代码第2.2步可以推导（3）是当第2.1步判断条件不满足时，改变j所指向的字符位置。即调用函数get_next(ne
根据伪代码第3步可以推导（4）是返回匹配的起始位置。由于当前i所指向字符串中匹配子串的最后一个字符的位置，
ls。
本题问题2是计算KMP算法的复杂度。
本题问题3中已知字符串“BBABBCAC”，则根据get_next()函数可以求得next数组的元素值为[-1,-1,1,-1,-1,2,0,0]。

参考答案：

（1）virtual void printInvoice()
（2）ticket->printInvoice()
（3）Decorator::printInvoice()
（4）Decorator::PrintInvoice()
（5）&a

试题分析：

参考答案：

（1）ticket.printInvoice()
（2）super.printInvoice()
（3）new HeadDecorator(ticket).printInvoice()
（4）new FootDecorator(t)
（5）new FootDecorator(null)

试题分析：