

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

中级软件设计师下午试题模拟60

试题一

阅读以下说明和数据流图，根据要求回答下列问题。

[说明]

现准备为某银行开发一个信用卡管理系统CCMS，该系统的基本功能如下。

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。

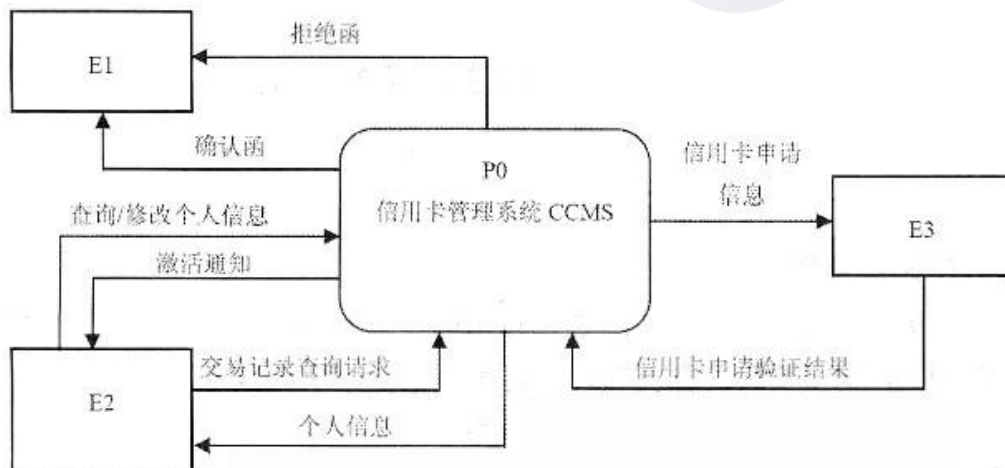
2. 信用卡激活。信用卡客户向CCMS提交激活请求，用信用卡号和密码激活该信用卡。激活操作结束后，CCMS将激活通知发送给客户，告知客户其信用卡是否被成功激活。

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)。

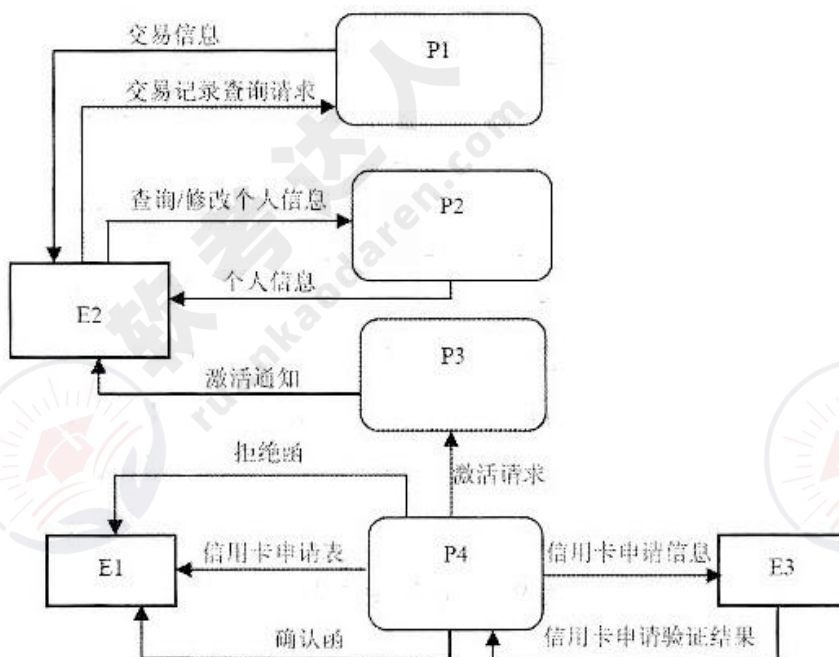
图1和图2分别给出了该系统的顶层数据流图和0层数据流图的初稿。

图1



信用卡管理系统的顶层数据流图

图2



信用卡管理系统的0层数据流图

1、[问题1]

根据[说明]，将图1中的E1~E3填充完整。

2、[问题2]

图1中缺少3条数据流，根据[说明]，分别指出这3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。)

3、[问题3]

图2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。)

4、[问题4]

根据[说明]，将图2中P1~P4的处理名称填充完整。

试题二

5、【说明】

学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授各带有若干研究生；每个班有若干学生，每个学生选修若干课程，每门课可由若干学生选修。

【问题1】

用E-R图画出此学校的概念模型，用文字写出各实体和联系的属性。

【问题2】

将E-R图转换成关系模型。

【问题3】

指出各关系模型的候选键。

试题三

6、【程序说明】

定义一个多边形结构：struct polygon实现以下内容：(1)建立该结构的链表：create函数是创建链表，每输入一个结点的数据，就把该结点加入到链表当中，它返回创建的链表的头指针。(2)显示链表的各个结点数据：结点数据包括：多边形顶点数、各顶点的纵横坐标、当多边形顶点数为0时，链表创建结束。(3)编写一个函数disp，删除链表中的所有结点。需要注意的是：要先释放结点数据内存，再删除结点，如果在释放结点数据内存单元之前删除结点，则无法找到结点数据内存单元的地址，也就无法释放数据的内存单元。

【程序】

```
#include "iomanip.h"
struct polygon
{
    int n;
    int* x;
    int *y;
    polygon * next;
};
void Push(polygon * & head, int n)
{
    polygon * newNde=newpolygon;
    newNde=newpo, Ygon;
    newNde->next=__ (1) __;
    newNde->x=new int [n];
    newNde->y=new int[n];
    newNde->n=__ (2) __;
```

```

for(int i=0; i<=__(3)__; i++){
    cout<<"请输入多边形各顶点x、y坐标，坐标值之间用空格分隔：";
    cin>>newNode->x[i]>>newNode->y[i];
}
__(4)____=head;    //在head前不需要额外的。
head=newNode;
}
polygon * create()
{
    polygon * head=NULL;
    polygon * tail;
    int n;
    cout<<"请输入多边形顶点的个数(顶点个数为0时结束)：";
    cin>>n;
    if(n==0) return __(5)____;
    Push(head, __(6)____);
    tail=head;
    cout<<"请输入多边形顶点的个数(顶点个数为0时结束)：";
    cin>>n;
    while(n!=0)
    {
        Push(tail->next, __(7)____);    //在tail->next增加结点
        tail =tail->next;    //advance tail to point to last node
        cout<<"请输入多边形顶点的个数(顶点个数为0时结束)：";
        cin>>n;
    }
    return head;
}
void disp(polygon * head)
{
    int i, No=1;
    cout<<setw(10)<<"x" <<setw(6)<<"y"<<endl;
    while(head!=NULL)
    {
        cout<<"第" <<No<<"结点：" <<endl;
        for(i=0; i<=head->n-1; i++)
            cout<<setw(10)<<head->x[i]<<setw(6)<<head->y[i]<<endl;
        __(8)____;
        head=__(9)____;
    }//Match while statement
}
void del(polygon * head)
{
    polygon * p;
    while(head!=NULL)
    {
        p=__(10)____;
        head=head->next;
    }
}

```



```

delete p->x;
delete P->y;
deletep;
} //Match while statement
}
void main()
{
    polygon * head;
    head=create();
    disp(head);
    del(head);
}

```

试题四

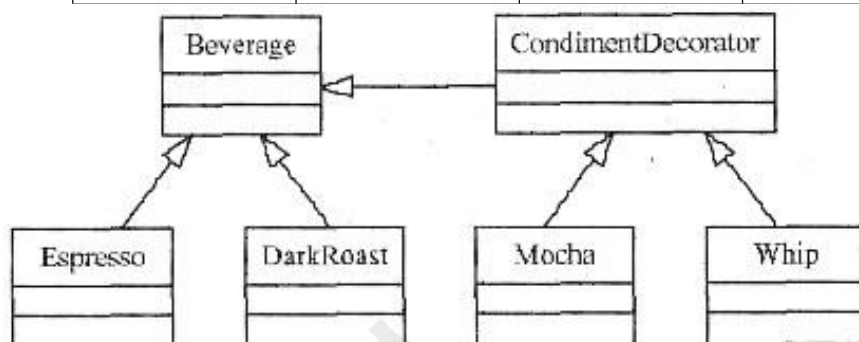
7、阅读下列说明和C++代码，回答下列问题。

[说明]

某咖啡店卖咖啡时，可以根据顾客的要求在其中加入各种配料，咖啡店会根据所加入的配料来计算费用。咖啡店所供应的咖啡及配料的种类和价格如表所示。

咖啡及配料的种类和价格

咖啡	价格/杯(¥)		配料	价格/份(¥)
蒸馏咖啡 (Espresso)	25		摩卡(Mocha)	10
深度烘焙咖啡 (DarkRoast)	20		奶泡(Whip)	8



咖啡计费的类图

[C++代码]

```

#include <iostream>
#include <string>
using namespace std;
const int ESPRESSO_PRICE = 25;
const int DRAKROAST_PRICE = 20;
const int MOCHA_PRICE = 10;
const int WHIP_PRICE = 8;
class Beverage { //饮料
    1: string description;
    public: 2() {return description;}
    3;};
class CondimentDecorator: public Beverage { //配料
    protected: 4;};
class Espresso: public Beverage { //蒸馏咖啡

```

```

public:
    Espresso () { description="Espresso"; }
    int cost () { return ESPRESSO_PRICE; }
};
class DarkRoast: public Beverage { //深度烘培咖啡
    public: DarkRoast() { description = "DardRoast"; }
    int cost(){ return DRAKROAST_PRICE; }
};
class Mocha: public CondimentDecorator { //摩卡
    public: Mocha (Beverage*beverage) { this->beverage=beverage; }
    string getDescription(){ return beverage->
getDescription()+"Mocha"; }
    int cost(){ return MOCHA_PRICE+beverage->cost(); }
};
class Whip: public CondimentDecorator { //奶泡
    public: Whip (Beverage*beverage) { this->beverage=beverage; }
    string getDescription() { return beverage->
getDescription()+"Whip"; }
    int cost(){ return WHIP_PRICE+beverage->cost(); }
}
int main(){
    Beverage* beverage = new DarkRoast() ;
    beverage=new Mocha( 5 );
    beverage=new Whip ( 6 );
    cout<<beverage->getDescription()<<"¥"<<beverage->cost()
endl;
    return 0;
}

```

编译运行上述程序，其输出结果为：DarkRoast, Mocha, Whip ¥38。

答案：

试题一

1、E1：非信用卡客户 E2：信用卡客户 E3：银行

由题干关键信息“1. 信用卡申请。非信用卡客户填写信用卡申请表.....CCMS将记录该客户的基本信息，并发送确认函给该客户.....否则该客户将会收到一封拒绝函”，以及图1中数据流“确认函”、“拒绝函”等信息可知，外部实体E1的名称是“非信用卡客户”。

由题干关键信息“2.信用卡客户向CCMS提交激活请求.....CCMS将激活通知发送给客户”和图1中数据流“激活通知”，题干信息“3.每位信用卡客户可以在线查询和修改个人信息”和图1中数据流“查询/修改个人信息”、“个人信息”，题干信息“4.信用卡客户可以通过CCMS查询并核实其交易信息”和图8-15中数据流“交易记录查询请求”可知，外部实体E2的名称是“信用卡客户”。

由题干说明中“1. 信用卡申请。.....如果信用卡申请被银行接受”、图1中数据流“信用卡申请信息”、“信用卡申请验证结果”和常识等关键信息可知，外部实体E3的名称是“银行”。

起点	终点	名称
E1	P0	信用卡申请表
E2	P0	激活请求
P0或信用卡管理系统CCMS	E2	信用卡交易信息

将[问题1]的分析结果填写到图1中，由题干关键信息“1.非信用卡客户填写信用卡申请表.....提交CCMS”，图1中数据流“确认函”、“拒绝函”及其流向等综合信息可知，外部实体E1“非信用卡客户”

有一条输出数据流“信用卡申请表”，流向加工P0“信用卡管理系统CCMS”。换言之，图1中缺少了1条名称为“信用卡申请表”的数据流，其起点是E1，终点为P0。

同理，由题干关键信息“2. 信用卡激活。信用卡客户向CCMS提交激活请求.....”和图1中数据流“激活通知”及其流向等综合信息可知，外部实体E2“信用卡客户”有一条输出数据流“激活请求”，流向加工P0“信用卡管理系统CCMS”。换言之，图1中缺少了一条名称为“激活请求”的数据流，其起点是E2，终点为P0。

由题干关键信息“4. 交易信息查询.....信用卡客户可以通过CCMS查询并核实其交易信息”、图1中数据流“交易记录查询请求”及其流向和生活常识等综合信息可知，加工P0“信用卡管理系统CCMS”有一条输出数据流“信用卡交易信息”，流向外部实体E2“信用卡客户”。换言之，图1中缺少了一条名称为“信用卡交易信息”的数据流，其起点是P0，终点为E2。

3、错误的数据流名称：激活请求和信用卡申请表

改正后的数据流：

数据流名称	改正后数据流起点	改正后数据流终点
激活请求	E2	P3
信用卡申请表	E1	P4

将[问题1]和[问题2]的分析结果填写到图1中。题干说明中“信用卡申请”、“信用卡激活”、“信用卡客户信息管理”和“交易信息查询”是典型的加工名称。根据数据流图父图与子图数据平衡原则，对照图2和信息填充后的图1可以直观地发现，图2中存在的一条错误数据流是“信用卡申请表”。数据流“信用卡申请表”在图1中是从外部实体E1流向CCMS系统，而在图8-16中是从加工P4流向外部实体E1，应予以更正。

根据题干说明中“1. 信用卡申请”描述信息，结合图2中与加工P4相关的“信用卡申请表”、“信用卡申请信息”、“信用卡申请验证结果”等关键数据流信息可得，加工P4的名称是“信用卡申请”。图2中数据流“信用卡申请表”的起点应改正为E1，终点应修改为P4。

同理，在图1中数据流“激活请求”从外部实体E2流向CCMS系统，而在图2中是从加工P4流向加工P3，应予以更正。根据题干说明中“2. 信用卡激活”描述信息，结合图2中加工P3的输出数据流“激活通知”等信息可得，加工P3的名称是“信用卡激活”。图2中数据流“激活请求”的起点应改正为E2，终点应修改为P3。

4、P1：交易信息查询 P2：(信用卡)客户信息管理 P3：信用卡激活 P4：信用卡申请
根据题干说明中“3. 信用卡客户信息管理”描述信息，结合图2中与加工P2相关的“查询/修改个人信息”、“个人信息”等关键数据流信息可得，加工P2的名称是“信用卡客户信息管理”。

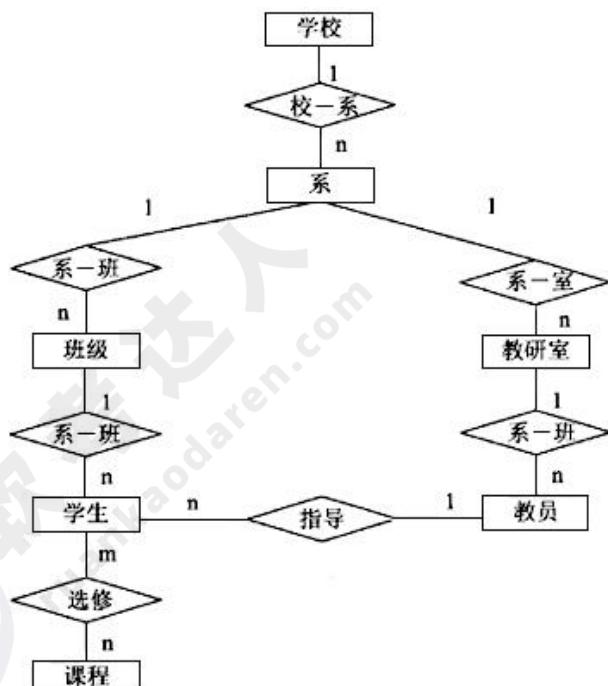
根据题干说明中“4. 交易信息查询”描述信息，结合图2中与加工P1相关的“交易记录查询请求”、“交易信息”等关键数据流信息可得，加工P1的名称是“交易信息查询”。

根据[问题3]的分析结果可知，加工P3的名称是“信用卡激活”，加工P4的名称是“信用卡申请”。

试题二

5、【问题1】

用E-R图表示的概念模型如下所示。



各实体的属性如下所述。

系：系编号，系名。

班级：班级编号，班级名。

教研室：教研室编号，教研室名。

学生：学号，姓名，学历。

课程：课程编号，课程名。

教员：职工号，姓名，职称。

各联系的属性为：

选修课：成绩。

其他联系无属性。

【问题2】

其关系模型为：

系(系编号，系名，学校名)

班级(班级编号，班级名，系编号)

教研室(教研室编号，教研室，系编号)

学生(学号，姓名，学历，班级编号，导师职工号)

课程(课程编号，课程名)

选课(学号，课程编号，成绩)

【问题3】

各关系模型的候选键为：

系—系编号；班级—班级编号，教研室—教研室编号，学生—学号；课程—课程编号；教员—职工号；选课—学号，课程编号。

[解析] 在画E-R图时，可以按照对问题的描述一步一步画出每一句话中涉及的实体，再根据给出的实际语义，画出实体之间的联系。例如，每个教研室有若干教员，每个班有若干学生，可以画出教研室和教员、班级和学生之间一对多的联系。再比如，有的教授和副教授每人各带若干研究生，而一个研究生一般指定一个导师，这是通常的规则，所以可以画出教员和学生之间一对多的关系。

按照上述的分析方法，从题的说明中，可以得到如下实体和联系。

·实体：学校、系、教研室、班级、教员、学生、课程。

·联系：校—系、系—室、系—班、室—教、班—生、指导、选修。

具体说明：

·实体“学校”和实体“系”是“一对多”的“校—系”联系。

·实体“系”和实体“班级”是“一对多”的“系—班”联系。

- 实体“系”和实体“教研室”是“一对多”的“系—室”联系。
- 实体“班级”和实体“学生”是“一对多”的“班—生”联系。
- 实体“教研室”和实体“教员”是“一对多”的“室—教”联系。
- 实体“学生”和实体“课程”是“多对多”的“选修”联系。
- 实体“教员”和实体“学生”是“一对多”的“指导”联系。

试题三

6、(1) NULL (2) n (3) n-1 (4) newNode->next
(5) head (6) n (7) n (8) No++ (9) head->next (10) head

[解析] 如果掌握了链表的创建、遍历和删除的方法，解决本题应该并不困难。要显示链表各结点的数据，就是要把各结点找到，然后把该结点的每一个x、y坐标打印出来。不过，与普通的链表也有不同的地方：就是该链表的结点数据是指针。要在链表结点中存入数据，必须先动态分配存储数据的内存单元；要删除链表中的各个结点，必须先释放结点数据的内存单元，否则会造成内存泄露。

试题四

7、(1) protected (2) virtual string getDescription
(3) virtual int cost()=0 (4) Beverage* beverage
(5) beverage (6) beverage