

2013年下半年软件设计师考试下午真题（权威解析+标准答案）

卷面总分：75.0 分

答题时间：150 分钟

测试次数：9673 次

平均得分：33.4 分

是否需要批改：否

案例分析题

在下列各题中，请阅读说明材料，根据提问进行解答。

1 某大学欲开发一个基于Web的课程注册系统。该系统的主要功能如下：

1、验证输入信息

（1）检查学生信息：检查学生输入的所有注册所需信息。如果信息不合法，返回学生信息不合法提示；如果合法，输出合法学生信息。

（2）检查学位考试信息：检查学生提供的学位考试结果。如果不合法，返回学位考试结果不合法提示；如果合法，检查该学生注册资格。

（3）检查学生资格：根据合法学生信息和合法学位考试结果，检查该学生对欲选课程的注册资格。如果无资格，返回无注册资格提示；如果有注册资格，则输出注册学生信息（包含选课学生标识）和欲注册课程信息。

2、处理注册申请

（1）存储注册信息：将注册学生信息记录在学生库。

（2）存储所注册课程：将选课学生标识与欲注册课程进行关联，然后存入课程库。

（3）发送注册通知：从学生库中读取注册学生信息，从课程库中读取所注册课程信息，给学生发送接受提示；给教务人员发送所注册课程信息和已注册学生信息。

现采用结构化方法对课程注册系统进行分析 and 设计，获得如图1-1所示的0层数据流图和图1-2所示的1层数据流图。

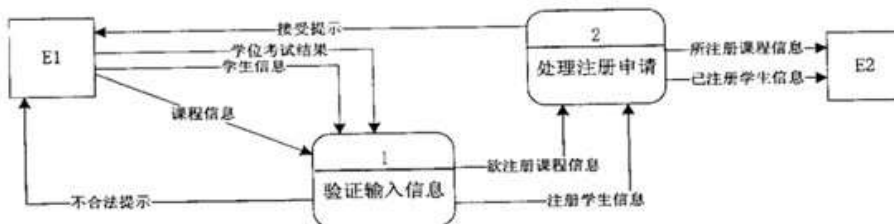


图 1-1 0 层数据流图

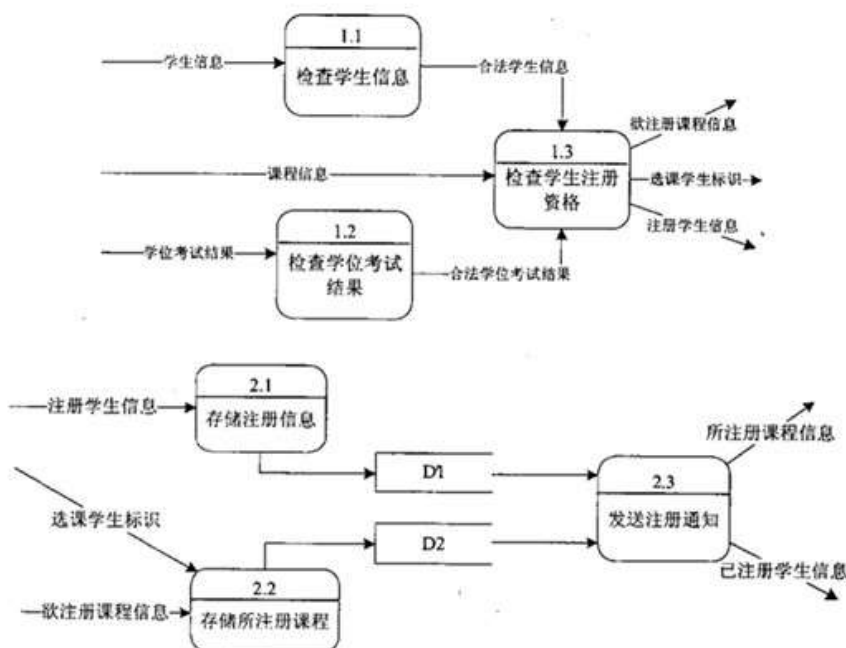


图 1-2 1 层数据流图

【问题1】（2分）

使用说明中的词语，给出图1-1中的实体E1和E2的名称。

【问题2】（2分）

使用说明中的词语，给出图1-2中的数据存储D1和D2的名称。

【问题3】（8分）

根据说明和图中术语，补充图1-2中缺失的数据流及其起点和终点。

【问题4】（3分）

根据补充完整的图1-1和图1-2，说明上层的哪些数据流是由下层的哪些数据流组合而成。

填写我的答案

段落格式

代码语言

元素路径:

2 某快递公司为了方便管理公司物品运送的各项业务活动，需要构建一个物品运送信息管理系统。

【需求分析结果】

（1）快递公司有多多个分公司，分公司信息包括分公司编号、名称、经理、办公电话和地址。每个分公司可以有多多名员工处理分公司的日常业务，每位员工只能在一个分公司工作。每个分公司由一名经理负责管理分公司的业务和员工，系统需要记录每个经理的任职时间。

（2）员工信息包括员工号、姓名、岗位、薪资、手机号和家庭地址。其中，员工号唯一标识员工信息的每一个元组。岗位包括经理、调度员、业务员等。业务员根据客户提交的快件申请单进行快件受理事宜，一个业务员可以受理多个客户的快件申请，一个快件申请只能由一个业务员受理。调度员根据已受理的申请单安排快件的承运事宜，例如：执行承运的业务员、运达时间等。一个业务员可以执行调度员安排的多个快件的承运业务。

（3）客户信息包括客户号、单位名称、通信地址、所属省份、联系人、联系电话、银行账号。其中，客户号唯一标识客户信息的每一个元组。当客户要寄快件时，先要提交快件申请单，申请号由系统自动生成。快件申请信息包括申请号、客户号、发件人、发件人电话、快件名称、运费、发出地、收件人、收件人电话、收件地址。其中，一个申请号对应唯一的一个快件申请，一个客户可以提交多个快件申请，但一个快件申请由唯一的一个客户提交。

【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（图2-1）和关系模型（不完整）如下：

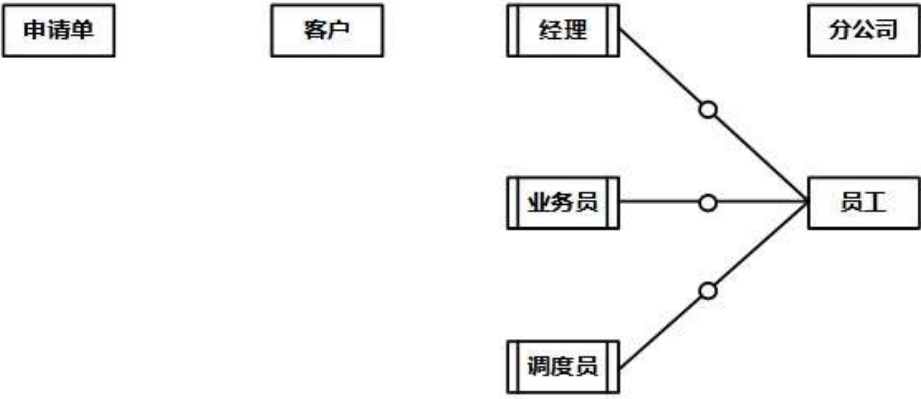


图 2-1 实体联系图

【关系模型设计】

分公司（分公司编号，名称，经理，办公电话，地址）

员工（员工号，姓名，（a），岗位，薪资，手机号，家庭地址）

客户（客户号，单位名称，通信地址，所属省份，联系人，联系电话，银行账号）

申请单（（b），发件人，发件人电话，发件人地址，快件名称，运费，收件人，收件人电话，收件地址，受理标志，业务员）

安排承运（（c），实际完成时间，调度员）

【问题1】

根据问题描述，补充五个联系，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、联系4和联系5代替，联系的类型分为1:1、1:n和m:n（或1:1、1:*和*:1）。

【问题2】

（1）根据实体联系图，将关系模式中的空（a）~(c)补充完整；

（2）给出员工、申请单和安排承运关系模式的主键和外键。

【问题3】

(1) 客户关系的通信地址可以进一步分为邮编、省、市、街道，那么该属性是否属于简单属性，为什么？请用100字以内的文字说明。

(2) 假设分公司需要增设一位经理的职位，那么分公司与经理之间的联系类型应修改为 (d)，分公司的主键应修改为 (e)。

填写我的答案

段落格式

代码语言

元素路径:

3 某航空公司会员积分系统 (CFrequentFlyer) 的主要功能描述如下：

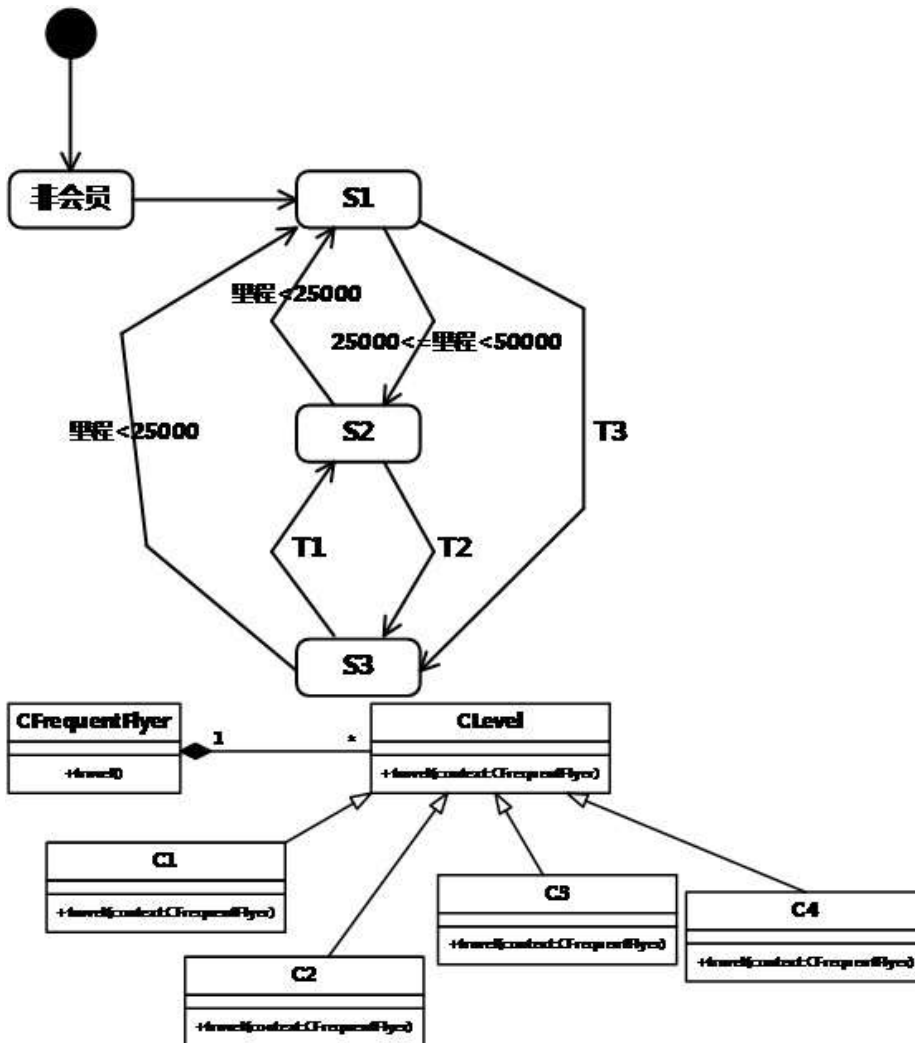
乘客只要办理该航空公司的会员卡，即可成为普卡会员 (CBasic)。随着飞行里程数的积累，可以从系统采用面向对象方法进行开发，在开发过程中认定出如表3-1所示的类。系统的用例图普卡会员升级到银卡会员 (CSilver) 或金卡会员 (CGold)。非会员 (CNonMember) 不能积累里程数。

每年年末，系统根据会员在本年度累积的里程数对下一年会员等级进行调整。

普卡会员在一年内累积的里程数若满25000英里但不足50000英里，则自动升级为银卡会员；若累积的里程数在50000英里以上，则自动升级为金卡会员。银卡会员在一年内累积的里程数若在50000英里以上，则自动升级为金卡会员。

若一年内没有达到对应级别要求的里程数，则自动降低会员等级。金卡会员一年内累积的里程数若不足25000英里，则自动降级为普卡会员；若累积的里程数达到25000英里，但是不足50000英里，则自动降级为银卡会员。银卡会员一年内累积的里程数若不足25000英里，则自动降级为普卡会员。

采用面向对象方法对会员积分系统进行分析与设计，得到如图3-1所示的状态图和图3-2所示的类图。



【问题1】(6分)

根据说明中的描述，给出图3-1中S1~S3处所对应的状态以及T1~T3处所对应的迁移的名称。

【问题2】(4分)

根据说明中的描述，给出图3-2中C1~C4所对应的类名(类名使用说明中给出的英文词汇)。

【问题3】(5分)

图3-2所示的类图中使用了那种设计模式？在这种设计模式下，类CFrequentFlyer必须具有的属性是什么？C1~C4中的travel方法应具有什么功能？

段落格式

代码语言

元素路径:

- 4
- 某工程计算中要完成多个矩阵相乘（链乘）的计算任务。

两个矩阵相乘要求第一个矩阵的列数等于第二个矩阵的行数，计算量主要由进行乘法运算的次数决定。采用标准的矩阵相乘算法，计算 $A_m \times n \times B_n \times p$ ，需要 $m \times n \times p$ 次乘法运算。

矩阵相乘满足结合律，多个矩阵相乘，不同的计算顺序会产生不同的计算量。以矩阵 $A_{110 \times 100}$ ， $A_{2100 \times 5}$ ， $A_{35 \times 50}$ 三个矩阵相乘为例，若按 $(A_1 \times A_2) \times A_3$ 计算，则需要进行 $10 \times 100 \times 5 + 10 \times 5 \times 50 = 7500$ 次乘法运算；若按 $A_1 \times (A_2 \times A_3)$ 计算，则需要进行 $100 \times 5 \times 50 + 10 \times 100 \times 50 = 75000$ 次乘法运算。可见不同的计算顺序对计算量有很大的影响。

矩阵链乘问题可描述为：给定 n 个矩阵 $\langle A_1, A_2, \dots, A_n \rangle$ ，矩阵 A_i 的维数为 $p_{i-1} \times p_i$ ，其中 $i = 1, 2, \dots, n$ 。确定一种乘法顺序，使得这 n 个矩阵相乘时进行乘法的运算次数最少。

由于可能的计算顺序数量非常庞大，对较大的 n ，用蛮力法确定计算顺序是不实际的。经过对问题进行分析，发现矩阵链乘问题具有最优子结构，即若 $A_1 \times A_2 \times \dots \times A_n$ 的一个最优计算顺序从第 k 个矩阵处断开，即分为 $A_1 \times A_2 \times \dots \times A_k$ 和 $A_{k+1} \times A_{k+2} \times \dots \times A_n$ 两个子问题，则该最优解应该包含 $A_1 \times A_2 \times \dots \times A_k$ 的一个最优计算顺序和 $A_{k+1} \times A_{k+2} \times \dots \times A_n$ 的一个最优计算顺序。据此构造递归式，

$$\text{cost}[i][j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \leq k < j} \text{cost}[i][k] + \text{cost}[k+1][j] + p_i * p_{k+1} * p_{j+1} & \text{if } i < j \end{cases}$$

其中， $\text{cost}[i][j]$ 表示 $A_{i+1} \times A_{i+2} \times \dots \times A_{j+1}$ 的最优计算的计算代价。最终需要求解 $\text{cost}[0][n-1]$ 。

【C代码】

算法实现采用自底向上的计算过程。首先计算两个矩阵相乘的计算量，然后依次计算3个矩阵、4个矩阵、...、 n 个矩阵相乘的最小计算量及最优计算顺序。下面是算法的C语言实现。

（1）主要变量说明

n ：矩阵数

$\text{seq}[]$ ：矩阵维数序列

$\text{cost}[][]$ ：二维数组，长度为 $n \times n$ ，其中元素 $\text{cost}[i][j]$ 表示 $A_{i+1} \times A_{i+2} \times \dots \times A_{j+1}$ 的最优计算的计算代价

$\text{trace}[][]$ ：二维数组，长度为 $n \times n$ ，其中元素 $\text{trace}[i][j]$ 表示 $A_{i+1} \times A_{i+2} \times A_{j+1}$ 的最优计算对应的划分位置，即 k

（2）函数 cmm

```
#define N 100
```

```
int cost[N][N];
```

```
int trace[N][N];
```

```
int cmm(int n,int seq[]){
```

```
int tempCost;
```

```
int tempTrace;
```

```
int i,j,k,p;
```

```
int temp;
```

```
for( i=0;i<n;i++){ cost[i][i] = 0;}
```

```
for(p=1;p<n;p++){
```

```
for(i=0; (1) ;i++){
```

```
    (2) ;
```

```
tempCost = -1;
```

```
for(k = i;k <j;k++){
```

```
    temp = (3) ;
```

```
        if(tempCost== -1||tempCost>temp){
```

```
            tempCost = temp;
```

```
        (4) ;
```

```
}
```

```
}
cost[i][j] = tempCost;
trace[i][j] = tempTrace;
}
}
return cost[0][n-1];
}
```

【问题1】（8分）

根据以上说明和C代码，填充C代码中的空（1）~（4）。

【问题2】（4分）

根据以上说明和C代码，该问题采用了（5）算法设计策略，时间复杂度（6）。（用O符号表示）

【问题3】（3分）

考虑实例n=6，各个矩阵的维数：A1为5*10，A2为10*3，A3为3*12，A4为12*5，A5为5*50，A6为50*6，即维数序列为5,10,3,12,5,50,6。则根据上述C代码得到的一个最优计算顺序为（7）（用加括号方式表示计算顺序），所需要的乘法运算次数为（8）。

填写我的答案

段落格式 代码语言

元素路径:



5 欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。以绘制直线和圆形为例，对应的绘图程序如表5-1所示。

表5-1 不同的绘图程序

	DP1	DP2
绘制直线	draw_a_line(x1,y1,x2,y2)	drawline(x1,x2,y1,y2)
绘制圆	draw_a_circle(x,y,r)	drawcircle(x,y,r)

该绘图软件的扩展性要求，将不断扩充新的图形和新的绘图程序。为了避免出现类爆炸的情况，现采用桥接（Bridge）模式来实现上述要求，得到如图5-1所示的类图。

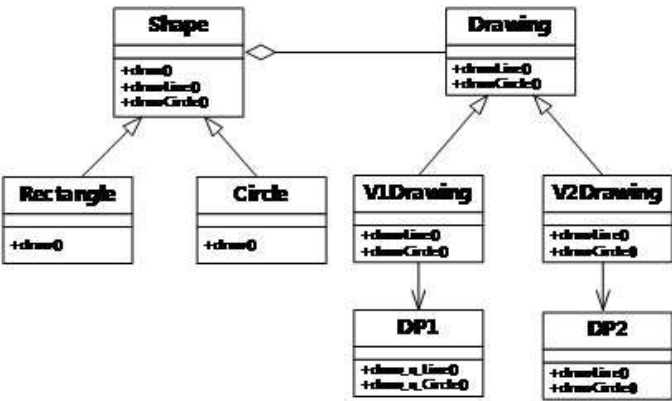


图5-1 类图

```
【C++代码】
class DP1{
public:
static void draw_a_line(double x1,double y1,double x2,double y2){/*代码省略*/}
static void draw_a_circle(double x,double y,double r){ /*代码省略*/}
};
class DP2{
```

```

public:
static void drawLine(double x1,double x2,double y1,double y2){/*代码省略*/}
static void drawCircle(double x,double y,double r){/*代码省略*/}
};

class Drawing{
public:
    (1) ;
    (2) ;
};

class V1Drawing:public Drawing{
public:
    void drawLine(double x1,double y1,double x2, double y2){/*代码省略*/}
    void drawCircle(double x, double y, double r){ (3) ;}
};

class V2Drawing:public Drawing{
public:
void drawLine(double x1,double y1,double x2, double y2){/*代码省略*/}
    void drawCircle(double x, double y, double r){ (4) ;}

};

class Shape{
public:
    (5) ;
Shape(Drawing *dp){_dp = dp;}
void drawLine(double x1, double y1,double x2, double y2){
_dp->drawLine(x1,y1,x2,y2);}
void drawCircle(double x, double y, double r){
_dp->drawCircle(x,y,r);}
private: Drawing * _dp;
};

class Rectangle:public Shape{
public:
void draw(){/*代码省略*/}
//其余代码省略
};

class Circle:public Shape{
private: double _x,_y,_r;
public:
    Circle(Drawing * dp, double x, double y, double r): (6) {_x = x; _y = y; _r = r;}
void draw(){ drawCircle(_x, _y, _r);}
};

int main(){
Resume * a = new Resume( "张三" );
a->SetPersonInfo( "男" , "29" );
a-> SetWorkExperience( "1998-2000" , "XXX公司" );
Resume * b = (5) ;
b-> SetWorkExperience( "2001-2006" , "YYY公司" );
return 0;
}

```

填写我的答案

段落格式

代码语言

表6-1 不同的绘图程序

	DP1	DP2
绘制直线	draw_a_line(x1,y1,x2,y2)	drawline(x1,x2,y1,y2)
绘制圆	draw_a_circle(x,y,r)	drawcircle(x,y,r)

该绘图软件的扩展性要求，将不断扩充新的图形和新的绘图程序。为了避免出现类爆炸的情况，现采用桥接（Bridge）模式来实现上述要求，得到如图6-1所示的类图。

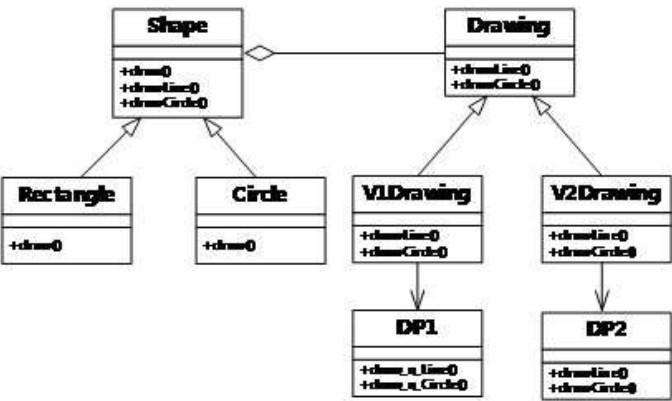


图6-1 类图

```
【Java代码】
(1) Drawing{
(2) ;
(3) ;
}
class DP1{
static public void draw_a_line(double x1, double y1, double x2, double y2)
/*代码省略*/
static public void draw_a_circle(double x, double y, double r)
/*代码省略*/
}

class DP2{
static public void drawline(double x1, double y1, double x2 ,double y2){/*代码省略*/}
static public void drawcircle (double x, double y, double r){/*代码省略*/}
}

class V1Drawing implements Drawing{
    public void drawLine(double x1, double y1, double x2 ,double y2){/*代码省略*/}
    public void drawCircle (double x, double y, double r){ ( 4 ) ;}
}

class V2Drawing implements Drawing{
    public void drawLine(double x1, double y1, double x2 ,double y2){/*代码省略*/}
    public void drawCircle (double x, double y, double r){ ( 5 ) ;}
}
```

```
Abstract class Shape{
    Private Drawing _dp;
    ( 6 );
    Shape(Drawing dp) {_dp=dp;}
    public void drawLine(double x1, double y1, double x2 ,double y2){
        _dp.drawLine(x1,y1,x2,y2); }
    public void drawCircle (double x, double y, double r){ _dp.drawCircle(x,y,r);}
}
```

```
class Rectangle extends Shape{
    private double _x1,_x2,_y1,_y2;
    public Rectangle(Drawing dp,double x1, double y1, double x2 ,double y2)
        {/*代码省略*/}
    public void draw(){/*代码省略*/}
}
```

```
class Circle extends Shape{
    private double _x, _y, _r;
    public Circle(Drawing dp,double x, double y, double r) {/*代码省略*/}
    public void draw(){drawCircle(_x,_y,_r);}
}
```

填写我的答案

段落格式

代码语言

元素路径: