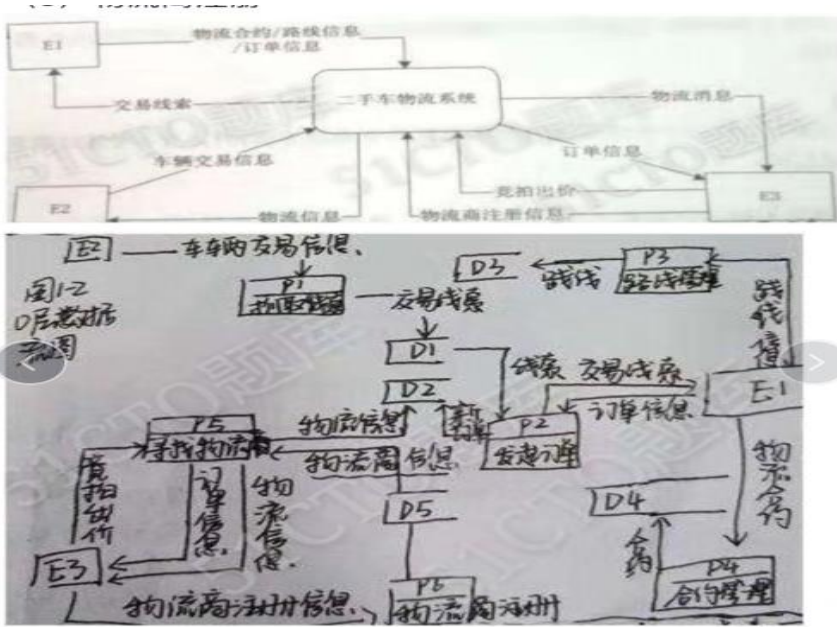


2019 软件设计师下半年下午答案

问答题(共 6 题， 共 60 分)

第一题:

1. [说明]某公司欲开发一款二手车物流系统，以有效提升物流分发效率，系统功能是：
- (1). 订单管理。系统抓取线索，将车辆交易系统中的交易信息抓取为线索，帮买家看到有买车线索后，电话询问买家是否需要物流，若需要。线索发起订单，并在存储系统中存储，系统帮买家找物流商运；
 - (2). 路线管理。对物流商路线管理，存储路线信息，包括：路线类型，物流商，起止地点。路线分为：固定路线，包车路线，竞拍体系。固定路线和包车路线是合约制，包车路线，发车时间由公司自行管理，订单优先采用合约制。
 - (3). 合约管理。合约信息包括物流商信息，路线起止城市，价格有效期；
 - (4). 寻找物流商。根据订单类型，起止城市，服务模式进行自动分发或竞拍体系方式选择合适物流商。
 - (5). 物流商注册



- 问题 1:使用说明中的词语，给出图 1-1 中的实体 E1-E3 名称；
- 问题 2:使用说明中的词语，给出图 1-2 中的数据存储,D1-D5 名称；
- 问题 3: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:根据说明采用结构化语言对“P5 寻找物流商”的加工逻辑进行描述。

参考答案

问题 1:

E1:系统帮买商; E2:车辆交易系统; E3:物流商

问题 2:

D1:交易线索信息表; D2:订单信息表

D3:路线信息表; D4:合约信息表; D5:物流商信息表

问题 3:

✓ 发送物流信息:起点 P5 终点 E2

✓ 自动分配:起点 P5 终点 E3

✓ 更新物流信息:起点 P5 终点 D2

问题 4: 略

第二题:

2. [说明]公司拟开发新入职员工的技能培训管理系统以便使新员工快速胜任新岗位。

(1). 部门信息包括:部门号、名称、部门负责人、电话等,部门号唯一标识部门关系中的每一个元组,一个部门有多个员工,但一名员工只属于一个部门,每个部门只有一名负责人,负责部门工作。

(2). 员工信息包括员工号、姓名、部门号、岗位、基本工资、电话、家庭住址等,其中员工号是唯一标示员工关系中的每一个元组。岗位有新入职员工,培训师、部门负责人等不同岗位设置不同的基本工资,新入职员工要选择多门课程进行培训,并通过考试取得课程成绩,名培训师可以讲授多门课程、一门课程可由多名培训师讲授。

(3). 课程信息包括课程号,课程名称、学时等;其中课程号唯一标识课程关系的每一个元组。

关系模式设计:

部门(部门号, 部门名, 部门负责人, 电话)

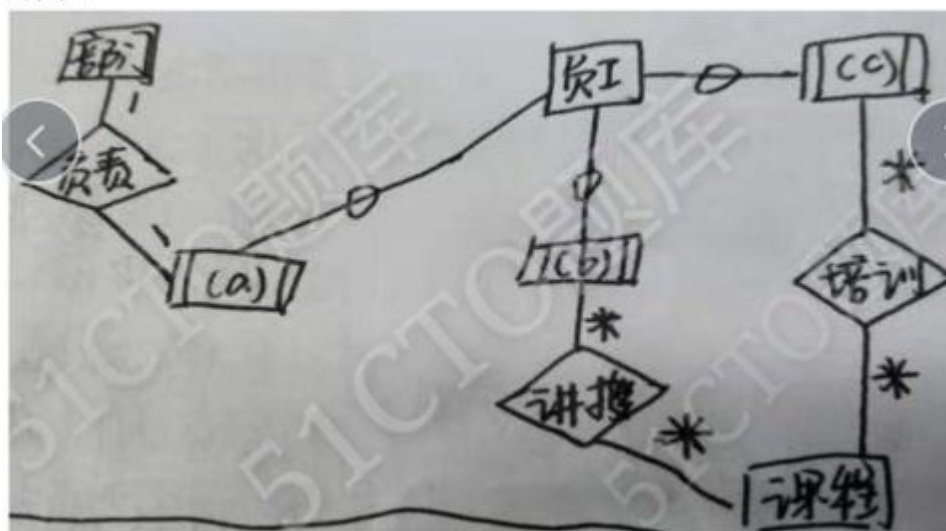
员工(员工号, 姓名, 部门号, d, 电话, 家庭住址)

课程(c, e, 课程名称, 学时)

讲授(课程号, 培训师, 培训地点)

培训(课程号, (f))

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。



(1) 补充图 2-1 中的空 (a) - (c)

(2) 图 2-1 中是否存在缺失联系, 若存在, 则说明所缺失的联系和联系类型

问题 2:

根据题意，将关系模式中的空(d) - (f) 补充完整

问题 3:

员工关系模式的主键为(g) ， 外键为(h) ， 讲授关系模式的主键为(i) ， 外键为(j)

问题 4:

员工关系是否存在传递依赖?用 100 字以内的文字说明理由

参考答案

问题 1:

(1) a:部门负责人 b:培训师 c:新员工

(2) 存在缺失数据流, 部门和员工之间: 1:m(一个部门对应多有员工)

问题 2:

d:岗位基本工资 e:课程号 f:员工号

问题 3:

员工关系表: g:主键:员工号外键:部门号

讲授关系模式: i 主键:课程号培训师培训地点 外键: 课程号

问题 4: 不存在传递依赖

第三题:

试题三(共 15 分)

[说明]

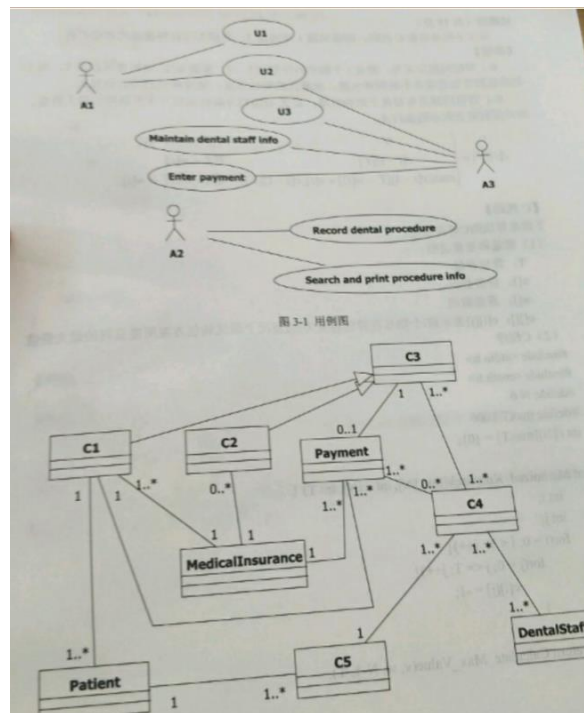
说明了列说明和图，回答问题 1 至问题 3, 将解答填入答题纸的对应栏内。

某牙科诊所拟开发一套信息系统， 用于管理病人的基本信息和就医信息。诊所工作人员包括:医护人员(DentalStaff)、 接待员(Receptionist) 和办公人员(OfficeStaff) 等。

系统主要功能需求描述如下

- (1). 记录病人基本信息(Maintain patient info)。初次就诊的病人，由接待员将病人基本资料录入系统。病人基本信息包括病人姓名、身份证号、出生日期、性别、首次就诊时间和最后次就诊时间等。每位病人与其医保信息(MedicalInsurance) 关联。
- (2). 记录就诊信息(Record Office Visit info)录入系统病人在诊所的每一一次就诊，由接待员将就诊信息(Office Visit) 录入系统。就诊信息包括就诊时间、就诊费用、支付代码、病人支付费用和医生支付费用等。
- (3). 记录治疗信息(Record dental procedure)。病人在就诊时，可能需要接受多项治疗，每项治疗(Procedure)可能由多位医护人员为其服务。治疗信息包括:治疗项目名称、治疗项目描述、治疗的牙齿和费用等。治疗信息由每位参与治疗的医护人员分别向系统中录入。
- (4). 打印发票(Print invoices)。 发票(Invoice) 由办公人员打印。发票分为两种:给医保机构的发票(InsuranceInvoice) 和给病人的发票(PatientInvoice)。两种发票内容相同，只是支付的费用不同。当收到治疗费用后，办公人员在系统中更新支付状态(Enterpayment)。
- (5). 记录医护人员信息(Maintain dental staff info)。办公人员将医护人员信息录入系统。医护人员信息包括姓名、职位、身份证号、家庭住址和联系电话等。
- (6). 医护人员可以查询并打印其参与的治疗项目相关信息(Search and print procedure info)。

现采用面向对象方法开发该系统，得到如图 3-1 所示的用例图和 3-2 所示的初始类图。



[问题 1] (6 分)

根据说明中的描述，给出图 3-1 中 A1~A3 所对应的参与者名称和 U1~U3 所对应的用例名称。

[问题 2] (5 分)

根据说明中的描述，给出图 3-2 中 C1~C5 所对应的类名。

[问题 3] (4 分)

根据说明中的描述，给出图 3-2 中类 C4、C5、Patient 和 DentalStaff 的必要属性。

A1:Receptionist(接待人员)

A2:Dentalstaff(医护人员) .

A3:offhcestaff(办公人员)

U1: Maintain patient info (记录病人基本信息)

U2: Record office visit info (记录就诊信息)

U3: Print invoices (打印发票)

问题 2:

C1: PatientInvoice (病人发票)

C2:InsuranceInvoice (机构发票)

C3: Invoice (发票)

C4::Procedure (治疗信息)

C5: OfnceVisit (就诊信息)

问题 3:

C4:项目名称、项目描述、治疗的牙齿和费用

C5:就诊时间、就诊费用、支付代码、病人支付费用和医保支付费用

Patient:病人姓名、身份证号、出生日期、性别、首次

就诊时间、最后就诊时间等

Dentalstaff:病人姓名、身份证号、出生日期、性别、首次就诊时间、最后就诊时间

第四题:

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]

0-1 背包问题定义为: 给定 i 个物品的价值: $\dots i$ 小. 重量 w_i 和背包容量 T , 每个物品装到背包里或者不装到背包里。求最优的装包方案, 使得所得到的价值最大。

0-1 背包问题具有最优子结构性质。定义 $el[i][T]$ 为最优装包方案所获得的最大价值, 则可以得到如下所示的递归式。

$$c[i][T] = \begin{cases} 0 & \text{若 } i = 0 \text{ 或 } T = 0 \\ c[i-1][T] & \text{若 } T < w[i] \\ \max(c[i-1][T - w[i]] + v[i], c[i-1][T]) & \text{若 } i > 0 \text{ 且 } T \geq w[i] \end{cases}$$

[C 代码]

下面是算法的 C 语言实现。

(1) 常量和变量说明

T: 背包容量

v[]: 价值数组

w[]: 重量数组

c[i][j]: c[i][j] 表示前 i 个物品在背包容量为 j 的情况下最优装包方案所能获得的最大价值

(2) C 程序

```
#include <stdio.h>
#include <math.h>
#define N 6
#define maxT 1000
int c[N][maxT] = {0};
int Memoized Knapsack(int v[N], int w[N], int T) {
    int i;
    int j;
    for(i=0; i<N; i++){
        for(j=0; j<= T; j++){
            c[i][j] = -1;
        }
        return Calculate_Max_Value(v, w, N-i, T);
    }
    int Calculate_Max_Value(int v[N], int w[N], int l, int j){
        int temp=0;
        if(c[l][j] != -1){
            return (1);
        }
        if(l==0 || j==0){
            c[l][j]=0;
        }
    }
```

```

else {
c[i][i]=Calculate_Max_Value(v, w, i-1,j);
if( (2) ){
temp= (3) ;
if(c[i][j]<temp){
(4) ;
}
}
}
return c[i][j];
}

```

[问题 1] (8 分)

根据说明和 C 代码，填充 C 代码中的空(1)~(4)。

[问题 2] (4 分)

根据说明和 C 代码，算法采用了 (5) 设计策略。在求解过程中，采用了 (6) (自底向上或者自顶向下)的方式。

[问题 3] (3 分)

若 5 项物品的价值数组和重量数组分别为 $v[] = (0.1, 6, 18, 22, 28)$ 和 $w[] = (0, 1, 2, 5, 6, 7)$

背包容量为 $T = 11$, 则获得的最大价值为 (7) 。

参考答案:

问题 1:

1. $c[i][j]$

2. $w[i] \leq j$

3. $\text{Calculate_Max_Value}(v, w, i-1, j-w[i]+v[j]);$

4. $c[i][j] = \text{temp}$

问题 2:

动态规划 自顶向下

问题 3: 40

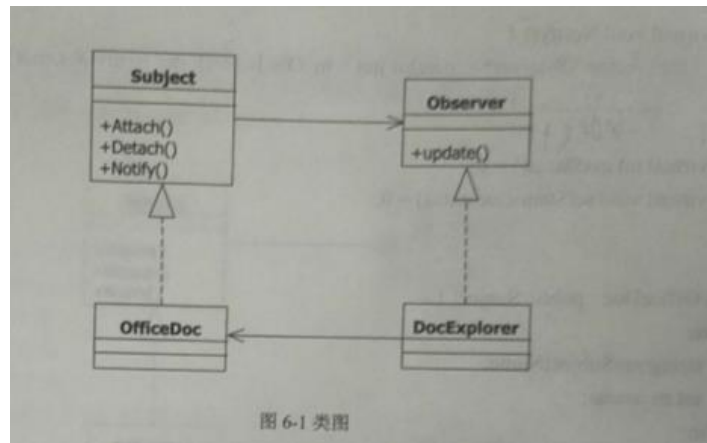
第六题:

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]

某文件管理系统中定义了类 `OfficeDoc` 和 `DocExplorer`，当 `OfficeDoc` 发生变化时，类 `DocExplorer` 的所有对象都要更新其自身的状态。现采用观察者(Observer)设计模式来实现该需求，所设计的类图如图 6-1 所示。



[Java 代码]

```

import java.util.*;
interface Observer {
    public (1);
};
interface Subject {
    public void Attach(Observer obs); doxentdue
    public void Detach(Observer obs);
    public void Notify();
    public void setStatus(int status);
    public int getStatus();
    class OfficeDoc implements Subject {
        private List<(2)> myObs;
        private String mySubjectName;
        private int m status;
        public OfficeDoc(String name) {
            mySubjectName = name;
            this.myObs = new ArrayList<Observer>0;
            m_status=0;
            public void Attach(Observer obs) { this.myObs add(obs); }
            public void Detach(Observer obs) { this.myObs remove(obs); }
            public void Notify() {
                for (Observer obs: this.myObs) { (3); }
            }
            public void setStatus(int status) {
                m_status = status;
                System. out,println("SetStatus subject[" + mySubjectName + "]status." + status);
            }
            public int getStatus() { return m status; }
        };
        class DocExplorer implements Observer {
            private String myObsName;
            public DocExplorer(String name, (4) sub){
                myObsName = name;
                sub.(5);
            }
        }
    }
  
```

```
public void update() {  
    System.out.println("update observer[" + myObsName + "I]");  
}  
};  
class ObserverTest{  
    public static void main(String[] args) {  
        Subject subjectA = new OfficeDc("subjectA");  
        Observer observerA = new DocExplorer("observer A", subjectA);  
        subjectA.setStatus(1);  
        subjectA.Notify0;  
    }  
}
```

参考答案：

1. void update();
2. Observer
3. obs.update();
4. Subject
5. Attach(this)