

2021年上半年软件设计师

下午案例分析真题与答案解析

本资料由信管网(www.cnitpm.com)整理发布，欢迎到信管网免费下载学习资料

信管网是专业软件设计师网站。提供了考试资讯、考试报名、成绩查询、资料下载、在线答题、考试培训、软件设计师人才交流、企业内训等服务。

信管网提供了备考软件设计师的精品学习资料；信管网案例分析频道和论文频道拥有丰富的案例范例和论文范例，信管网考试中心拥有软件设计师历年真题和模拟试题，并提供免费在线答题服务；信管网每年服务考生超100000人。

信管网——专业、专注、专心，成就你的软件设计师梦想！

信管网：www.cnitpm.com

信管网考试中心：www.cnitpm.com/exam/

信管网培训中心：www.cnitpm.com/wx/

信管网APP：www.cnitpm.com/app/

注：本资料由信管网整理后共享给各位考生，如果有侵犯版权行为，请来信告知。

信管网微信公众号



信管网客服微信号



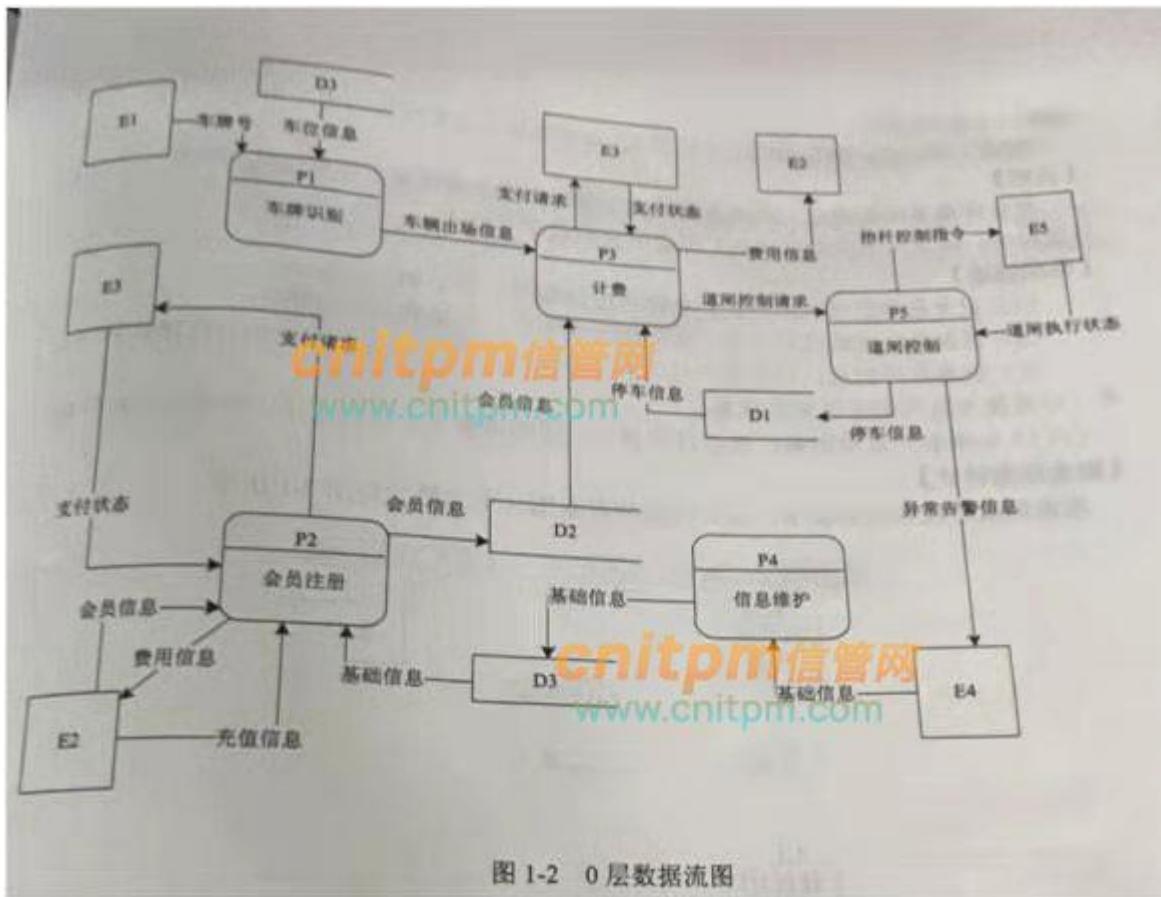
1、阅读下列说明和图回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】某停车场运营方为了降低运营成本，减员增效，提供良好的停车体验，欲开发无人值守停车系统，该系统的主要功能是：

- 1、信息维护。管理人员对车位（总数、空余车位数等）计费规则等基础信息进行设置。
- 2、会员注册。车主提供手机号、车牌号等信息进行注册，提交充值信息（等级、绑定并授权支付系统进行充值或交费的支付账号）不同级别和充值额度享受不同停车折扣点。
- 3、车牌识别。当车辆进入停车场时，若有（空余车位数大于 1），自动识别车牌号后进行道闸控制，当车主离开停车场时，识别车牌号，计费成功后，请求道闸控制。
- 4、计费。更新车辆离场时间，根据计费规则计算出停车费用，若车主是会员，提示停车费用：若储存余额够本次停车费用，自动扣费，更新余额，若储值余额不足，自动使用授权缴费账号请求支付系统进行支付，获取支付状态。若非会员临时停车，提示停车费用，车主通过扫描费用信息中的支付码调用支付系统自助交费，获取支付状态。
- 5、道闸控制。根据道闸控制请求向道闸控制系统发送抬杆指令和接收道闸执行状态。若道闸执行状态为正常放行时，对入场车辆，将车牌号及其入场时间信息存入停车记录，修改空余车位数；对出厂车辆更新停车状态，修改空余车位数。当因道闸重置系统出现问题（断网断电或是故障为抬杠等情况），而无法在规定的时间内接收到其返回的执行状态正常放行时，系统向管理人员发送异常告警信息，之后管理人员安排故障排查处理，确保车辆有序出入停车场。

现采用结构化方法对无人值守停车系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和图 1-2 所示的 0 层数据流图。





【问题 1】 (5 分)

使用说明中的词语，给出图 1-1 中的实体 E1-E5 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)

根据说明，采用结构化语言对“道闸控制”的加工逻辑进行描述。

信管网参考答案（最终答案以信管网题库为准）：

查看解析：www.cnitpm.com/st/4550117074.html

2、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】

某社区蔬菜团购网站，为规范商品收发流程，便于查询客户订单情况，需要开发个信息系统。请根据下述需求描述完成该系统的数据库设计。

【需求描述】

- (1) 记录蔬菜供应商的信息，包括供应商编号、地址和一个电话。
- (2) 记录社区团购点的信息，包括团购点编号、地址和一个电话。
- (3) 记录客户信息，包括客户姓名和一个电话。客户可以在不同的社区团购点下订单，不直接与蔬菜供应商发生联系。
- (4) 记录客户订单信息，包括订单编号、团购点编号客户电话，订单内容和日期。

【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

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式：

蔬菜供货商（供货商编号，地址，电话）

社区团购点（团购点编号，地址，电话）

供货（供货商编号， (a) ）

客户（姓名，客户电话）

订单（订单编号，团购点编号，订单内容，日期，

问题内容：

【问题 1】 (6 分)

根据问题描述，补充图 1-1 的实体联系图

【问题 2】 (4 分)

补充逻辑结构设计结果中的 (a) 、 (b) 两处空缺及完整性约束关系。

【问题 3】 (5 分)

若社区蔬菜团购网站还兼有代收快递的业务，请增加新的“快递”实体，并给出客户实体和快递实体之间的“收取”联系，对图 1 进行补充。“快递”关系模式包括快递编号、客户电话和日期。

信管网参考答案（最终答案以信管网题库为准）：

查看解析：www.cnitpm.com/st/4550227699.html

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

[说明]

某中医医院拟开发一套线上抓药 APP，允许患者凭借该医院医生开具的处方线上抓药，并提供免费送药上门服务。该系统的主要功能描述如下：

(1) 注册。患者扫描医院提供的二维码进行注册，注册过程中，患者需提供其病历号，系统根据病历号自动获取患者基本信息。

(2) 登录。已注册的患者可以登录系统进行线上抓药，未注册的患者系统拒绝其登陆。

(3) 确认处方。患者登录后，可以查看医生开具的所有处方。患者选择需要抓药的处方和数量(需要抓几副药)，同时说明是否需要煎制。选择取药方式：自行到店取药或者送药上门，若选择送药上门，患者需要提供收货人姓名、联系方式和收货地址。系统自动计算本次抓药的费用，患者可以使用微信或支付宝等支付方式支付费用。支付成功之后，处方被发送给药师进行药品配制。



图 3-1 用例图



图 3-2 类图

[问题 1] (7 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~U4 处所对应的用例名称。

[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] (3 分)

简要解释用例之间的 include、extend 和 generalize 关系的内涵。

信管网参考答案（最终答案以信管网题库为准）：

查看解析：www.cnitpm.com/st/4550322566.html

4、阅读下列说明和 C 代码，回答问题 1 和问题 2，将解答填入答题纸的对应栏内。

[说明]

凸多边形是指多边形的任意两点的连线均落在多边形的边界或者内部。相邻的点连线落在多边形边上，称为边，不相邻的点连线落在多边形内部。称为弦。假设任意两点连线上均有权重，凸多边形最优三剖分问题定义为：求将凸多边形划分为不相交的三角形集合，且各三角形权重之和最小的剖分方案。每个三角形的权重为三条边权重之和。假设 N 个点的凸多边形点编号为 V1,V2,……,VN,若在 VK 处将原凸多边形划分为一个三角形 V1VkVN，两个子多边形 V1,V2,⋯,V_k 和 V_k,V_{k+1},⋯,V_N，得到一个最优的剖分方案，则该最优剖分方案应该包含这两个子凸边形的最优剖分方案。用 m[i][j] 表示带你 V_{i-1},V_i,⋯,V_j 构成的凸多边形的最优剖分方案的权重，S[i][j] 记录剖分该凸多边形的 k 值。

则

$$m[i][j] = \begin{cases} 0, & i \geq j \\ \min_{i \leq k < j} \{m[i][k] + m[k+1][j] + W(V_{i-1}V_kV_j)\}, & i < j \end{cases}$$

其中：

W_{j,i-1} 分别为该三角形三条边的权重。求解凸多边形的最优剖分方案，即求解最小剖分的权重及对应的三角形集。

[C 代码]

```
#include
#define N 6
//凸多边形规模
int m[N+1][N+1]; //m[i][j]表示多边形 Vi-1 到 Vj 最优三角剖分的权值
int S[N+1][N+1]; //S[i][j]记录多边形 Vi-1 到 Vj 最优三角剖分的 k 值
int W[N+1][N+1]; //凸多边形的权重矩阵，在 main 函数中输入
/*三角形的权重 a, b, c, 三角形的顶点下标*/
int get_triangle_weight(int a, int b, int c){
    return W[a][b]+W[b][c]+W[c][a];
}
/*求解最优值*/
void triangle_partition(){
    int i,r,k,j;
    int temp;
    /*初始化*/
    for(i=1;i<=N;i++){
        m[i][i]=0;
    }
    /*自底向上计算 m, S*/
    for(r=2;(1);r++){/*r 为子问题规模*/ //r<=N
        for(i=1;k<=N-r+1;i++){
            (2); //int j=i+r-1
            m[i][j]= m[i][j]+m[i+1][j]+get_triangle_weight(i-1,i,j); /*k=j*/
            S[i][j]=i;
            for(k=j+1;k
                temp=m[i][k]+m[k+1][j]+ge_triangle_weight(i-1,k,j);
            if((3)){/*判断是否最小值*/ //temp
                m[i][j]=temp;
                S[i][j]=k;
            }
        }
    }
}
```

```
}

}

}

/*输出剖分的三角形 i, j: 凸多边形的起始点下标*/
void print_triangle(int i,int j){
if(i==j) return;
print_triangle(i,S[i][j]);
print_
triangle((4)); //S[i][j]+1,j
print( "V%d- -V%d-
-V%d\n ",i-1,S[i][j],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] (7 分)

根据说明和 C 代码，该算法采用的设计策略为(5)，算法的时间复杂度为(6)，空间复杂度为(7) (用 0 表示)。

信管网参考答案（最终答案以信管网题库为准）：

查看解析：www.cnitpm.com/st/455048357.html

5、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】

层叠菜单是窗口风格的软件系统中经常采用的一种系统功能组织方式。层叠菜单（如图 5-1 示例）中包含的可能是一个菜单项（直接对应某个功能），也可能是一个子菜单。现采用组合（Composite）设计模式实现层叠菜单，得到如图 5-2 所示的类图。

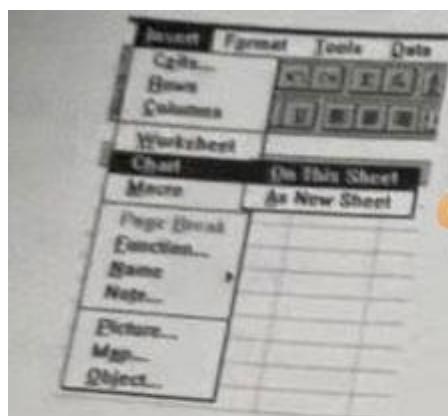


图 5-1 层叠菜单示例



图 5-2 类图

【C++代码】

```
#include <list>
#include <iostream>
#include <string>
using namespace std;

class MenuComponent { // 构成层叠菜单的元素
public:
    string name; // 菜单项或子菜单名称
    virtual void printMenu() { cout << name; }
    virtual void removeMenuItem(MenuComponent *element) = 0;
};

class MenuItem : public MenuComponents {
public:
    MenuItem(string name) { this->name = name; }
    void addMenuItem(MenuComponent *element) { return; }
    void removeMenuItem(MenuComponent *element) { return; }
    list<MenuComponent*> * getElement() { return NULL; }
};

class Menu : public MenuComponent {
private:
public:
    Menu(string name) { this->name = name; }
    void addMenuItem(MenuComponent *element) { elementList.push_back(element); }
    void removeMenuItem(MenuComponent *element) { elementList.remove(element); }
    list<MenuComponent*> * getElement() { return &elementList; }
};

int main() {
    MenuComponent *mainMenu = new Menu("Insert");
    MenuComponent *subMenu = new Menu("Chart");
    MenuComponent *element = new MenuItem("On This Sheet");
    subMenu->addMenuItem(element);
    return 0;
}
```

(1) :

(2) :

(3) :

(4) :

(5) :

查看解析: www.cnitpm.com/st/4550523215.html

6、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】层叠菜单是窗口风格的软件系统中经常采用的一种系统功能组织方式。层叠菜单（如图 6-1 示例）中包含的可能是一个菜单项（直接对应某个功能），也可能是一个子菜单，现在采用组合（composite）设计模式实现层叠菜单，得到如图 6-2 所示的类图。

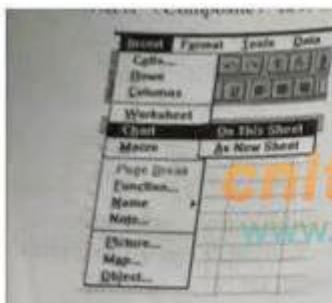


图 6-1 层叠菜单示例

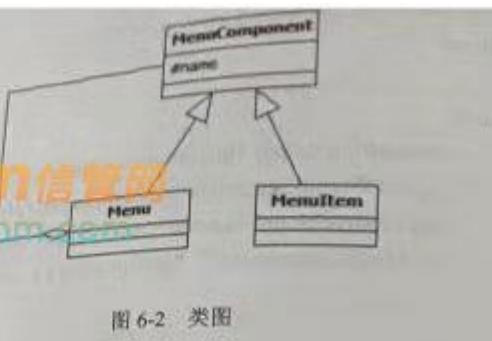


图 6-2 类图

【Java 代码】

```
import java.util.*;  
  
abstract class MenuComponent { // 构成层叠菜单的元素  
    _____(1) String name; // 菜单项或子菜单名称  
    public void printName() { System.out.println(name); }  
    public _____(2) ;  
    public abstract boolean removeMenuItem(MenuComponent element);  
    public _____(3) ;  
}
```

```
class MenuItem extends MenuComponent {
```

```
    public MenuItem(String name) { this.name = name; }  
    public boolean addMenuItem(MenuComponent element) { return false; }  
    public boolean removeMenuItem(MenuComponent element) { return false; }  
    public List<MenuComponent> getElement() { return null; }
```

```
class Menu extends MenuComponent {
    private _____ (4)
    public Menu(String name) {
        this.name = name;
        this.elementList = new ArrayList<MenuComponent>();
    }
    public boolean addMenuElement(MenuComponent element) {
        return elementList.add(element);
    }
    public boolean removeMenuElement(MenuComponent element) {
        return elementList.remove(element);
    }
    public List<MenuComponent> getElement() { return elementList; }
}

class CompositeTest {
    public static void main(String[] args) {
        MenuComponent mainMenu = new Menu("Insert");
        MenuComponent subMenu = new Menu("Chart");
        MenuComponent element = new MenuItem("On This Sheet");
        _____ (5);
        subMenu.addMenuElement(element);
        printMenus(mainMenu);
    }
}

private static void printMenus(MenuComponent ifile) {
    ifile.printName();
    List<MenuComponent> children = ifile.getElement();
    if(children == null) return;
    for (MenuComponent element:children) {
        printMenus(element);
    }
}
```

信管网参考答案（最终答案以信管网题库为准）：

查看解析：www.cnitpm.com/st/4550610845.html