

2024 年上半年 软件设计师 案例分析

第一批次 案例分析（11.04 下午）

第一题：数据流图

人工智能问答系统（类似 chatGPT）的数据流图设计

问题：

1. 补充外部实体
2. 补充数据存储
3. 补充缺失的数据流（起点终点）
4. 什么是数据流图设计的父类、子类平衡？如何保持？（子图的输入输出流同父类相应的输入输出流保持一致）

第二题：数据库设计

汽车维修公司对汽车的维修数据库设计

问题：

1. 补充缺失的联系和联系类型
2. 根据实体联系，补充关系模式的空缺
3. 给出关系模式的主键和外键
4. 如果客户为汽车购买了多张保险，涉及到维修赔偿费率的问题。加入“保险”，补充对应的关系及联系类型。

第三题：UML 建模

类似淘宝的网上购物系统设计

问题：

1. 补充用例名
2. 补充类名
3. 给出关系模式 $(0..*, *, 1..*, *)$

第四题：C 语言算法

最大子段和，给定 n 个整数（可能有负数）组成的序列 $a[1], a[2], \dots, a[n]$ ，求该序列如 $a[i] + a[i+1] + \dots + a[j]$ 的子段和的最大值。依次定义，所求最优值为： $\max\{0, a[i] + a[i+1] + \dots + a[j]\}$

动态规划法 $O(n)$

第五题：面向对象程序设计

command 命令模式。

第二批次 案例分析（11.05 下午）

第一题：数据流图

学校的疫情防控系统（日报/返校审批系统）。实体有学生，教师，行政管理人员，系统管理员等。

- 1) 补充实体 E1 - E5
- 2) 补充存储名称 3 个表
- 3) 补充缺失的数据流图
- 4) 用结构化语言描述审批流程

第二题：数据库设计

快递系统。给出了快递公司，快递员，客户，包裹（联系）的图要求补充。（参考如下2020年软考数据库系统工程师下午真题）

某小区快递驿站代为收发各家快递公司的包裹，为规范包裹收发流程，提升效率，需要开发一个信息系统。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) 记录快递公司和快递员的信息。快递公司信息包括：公司名称、地址和一个电话；快递员信息包括：姓名、手机号码和所属公司名称。一个快递公司可以有若干快递员，一个快递员只能属于一家快递公司。

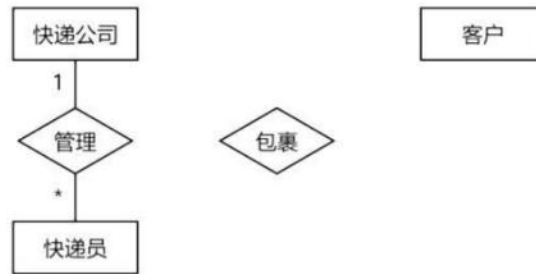
(2) 记录客户信息，客户信息包括：姓名、手机号码和客户等级。驿站对客户进行等级评定，等级高的客户在驿站投递包裹有相应的优惠。

(3) 记录包裹信息，便于快速查找和管理。包裹信息包括：包裹编号，包裹到达驿站时间、客户手机号码和快递员手机号码。快递驿站每个月根据收发的包裹数量，与各快递公司结算代收发的费用。



软考题库 - 微信搜一搜

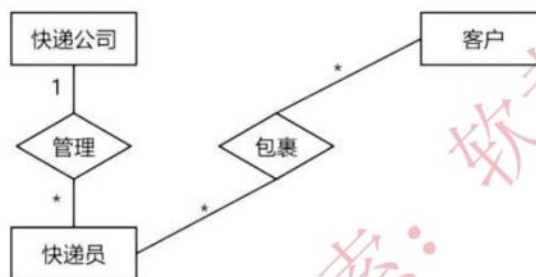
Q 软考达人



问题 1 (6 分)

1) 根据需求阶段收集的信息，设计实体联系图。

参考答案：如图：



问题 2 (4 分)

补充下列逻辑结构设计中的(a)、(b)两处空缺，并描述完整性约束关系。

【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(不完整)：

快递公司(公司名称，地址，电话)

快递员(姓名，快递员手机号码，(a))

客户(姓名，客户手机号码，客户等级)

包裹(编号，到达时间，(b)，快递员手机号码)

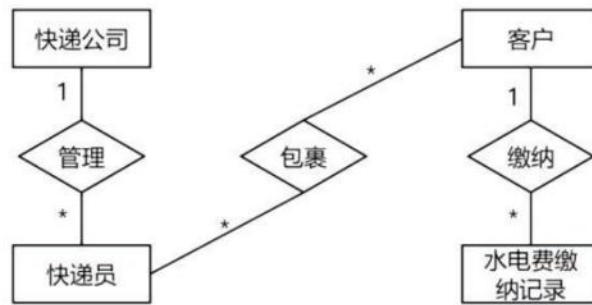
参考答案：(a) 所属公司名称。外键：公司名称 主键：快递员手机号

(b) 客户手机号码。外键：快递员手机号、客户手机号 主键：编号，

问题 3 (5 分)

若快递驿站还兼有代缴水电费业务，请增加新的“水电费缴费记录”实体，并给出客户和水电费缴费记录之间的“缴纳”联系，对图 1-1 进行补充。“水电费缴费记录”实体包括：编号、客户手机号码、缴费类型、金额和时间，请给出“水电费缴费记录”的关系模式，并说明其完整性约束。

参考答案：如图：

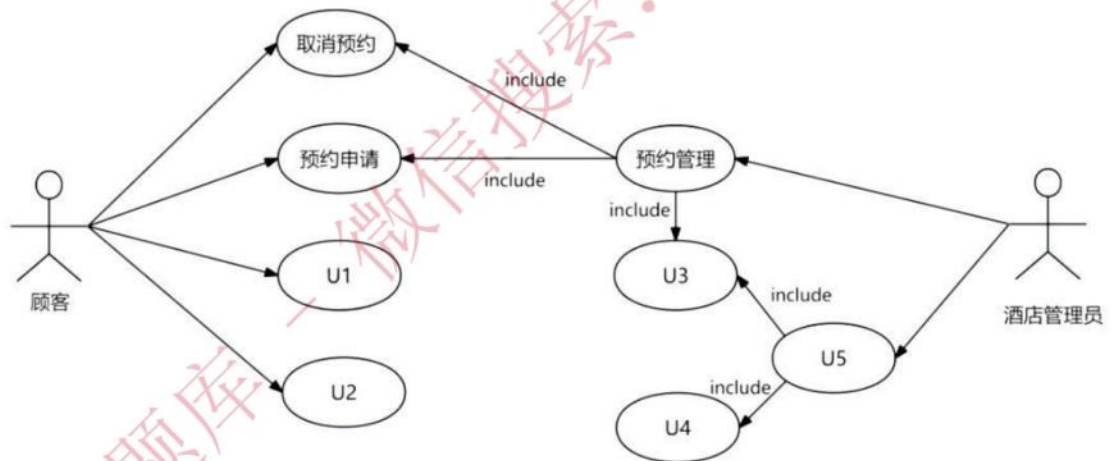


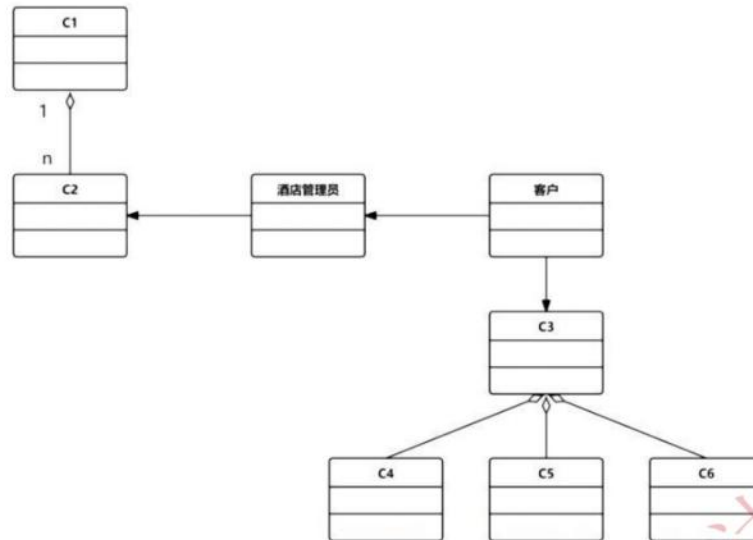
水电费缴费记录（编号，客户手机号码，缴费类型，金额，时间）

主键：编号，外键：客户手机号

第三题：UML 建模

度假村预定系统。图 1 是用例图，包括预定，支付（泛化为不同支付方式）；图 2 是类图，类名有客户，管理员，支付（继承为不同支付方式）。





1) 补充 U1 - U5

2) 补充 C1-C5

3) 说明关键字含义：包含(include)、扩展(extend)和泛化(generalization)

第四题：C 语言算法

最大子段和（分治法）。数组有正负数，类似{23, 15, 16, -1, -5-3}，求和最大的最长子结构。（不过代码内容可能不一样） $O(n\log n)$

第五题：面向对象程序设计

command 模式。

Java 题命令模式（与此代码类似不过也有些不同仅供参考）

```

abstract class Command {
    public abstract void execute();
}

class ConcreteCommand1 extends Command {
    private Receiver receiver;
    public ConcreteCommand1() {
        this.receiver = new ConcreteReceiver1();
    }

    public void setReceiver(Receiver receiver) {
        this.receiver = receiver;
    }

    public void execute() {
        if(receiver!=null){
            receiver.doSomething();
        }
    }
}
    
```

```
    }  
    }  
}  
  
abstract class Receiver {  
    public abstract void doSomething();  
}  
  
class ConcreteReceiver1 extends Receiver {  
    @Override  
    public void doSomething() {  
        System.out.println("ConcreteReceiver1 do...");  
    }  
}  
  
class Invoker {  
    private Command command;  
    public void setCommand(Command command) {  
        this.command = command;  
    }  
    public void action() {  
        this.command.execute();  
    }  
}  
  
class Test {  
    public static void main(String args[]) {  
        Invoker invoker = new Invoker();  
        Receiver receiver = new ConcreteReceiver1();  
  
        //这个位置真题循环的是 List  
        //List<Invoker> invokerList = new ArrayList<>();  
        //类似于 invokerList.get(i).action();  
  
        Command command = new ConcreteCommand1();  
        invoker.setCommand(command);  
        invoker.action();  
    }  
}
```

C 语言算法难度降低，题目跟随时代前沿，时间足够，历年真题仍然是重中之重。11.4 号和 11.5 号下午案例分析最后两道大题雷同（不过代码不一样）。(5 号比 4 号简单)