

# 2013年上半年软件设计师考试下午真题（权威解析+标准答案）

注:红色部字体为其他提问:

## 1. 参考答案:

### 【问题 1】

E1 志愿者 E2 捐赠人 E3 募捐机构 E4 场馆

### 【问题 2】

P1 确定活动时间范围 P2 搜索场馆 P3 推广募捐活动缺少的数据流:

名称: 活动请求 起点: 确定募款需求收集所募捐款 终点: P1

### 【问题 3】

D1 捐赠人信息表 D2 已联系的捐赠人信息表 D3 捐赠表 D4 已经处理的捐赠表。

试题分析:

解答这类题目有两个原则:

1.第一个原则是紧扣试题系统说明部分,数据流图与系统说明有着严格的对应关系,系统说明部分的每一句话都能对应到图中来,解题时一句一句的对照图来分析。

2.第二个原则即数据平衡原则,这一点在解题过程中也是至关重要的。数据平衡原则有两方面的含义,一方面是分层数据流图父子图之间的数据流平衡原则,另一方面是每张数据流图中输入与输出数据流的平衡原则。

### 【问题 1】

根据 0 层数据流管理志愿者中的募捐任务给志愿者发送加入邀请,邀请跟进,工作任务和管理志愿者提供的邀请响应可知 E1 为志愿者;从录入捐款人信息,向捐赠人发送募捐请求,;向捐赠人发送致谢函等可知 E2 为捐赠人;从根据说明中从募捐机构获取资金并向其发放赠品可知 E3 为募捐机构;根据向场馆发送可用性请求和获得场所可用性可知 E4 为场馆。

### 【问题 2】

根据 1 层数据流图中 P1 的输出流活动时间再结合说明可知 P1 为确定活动时间范围;从加工 P2 的输入流活动时间和输出流场馆可用性请求和活动时间和地点可知 P2 为搜索场馆;说明中根据活动时间和地点推广募捐活动,根据相应的活动信息举办活动,再结合 P3 的输入输出流可知 P3 为推广募捐活动。比较 0 层和 1 层中的数据流可知, P1 加工只有输出流,故缺少输出流,根据说明可知需要根据活动请求才能确定 P1,故该数据流为活动请求,在 0 层数据流中活动请求的起始加工为确定募款需求收集所募捐赠,故可知答案。

### 【问题 3】

根据最后的说明和 2 层数据流可知 D1 为捐赠人信息表, D2 为已联系的捐赠人信息表, D3 为捐赠表, D4 为已经处理的捐赠表。

1-3 中有缺的数据流么? 问题 2 的答案是只有一个数据流么?

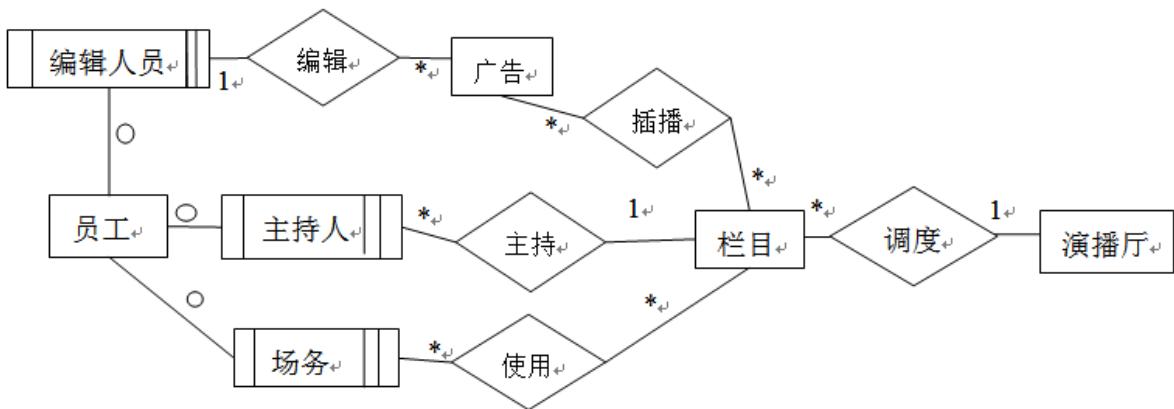
你好

试题之前编辑的时候,可能把缺少的数据流给截掉了。先已修正过来,给您带来的不便,烦请见谅。

谢谢

## 2. 参考答案:

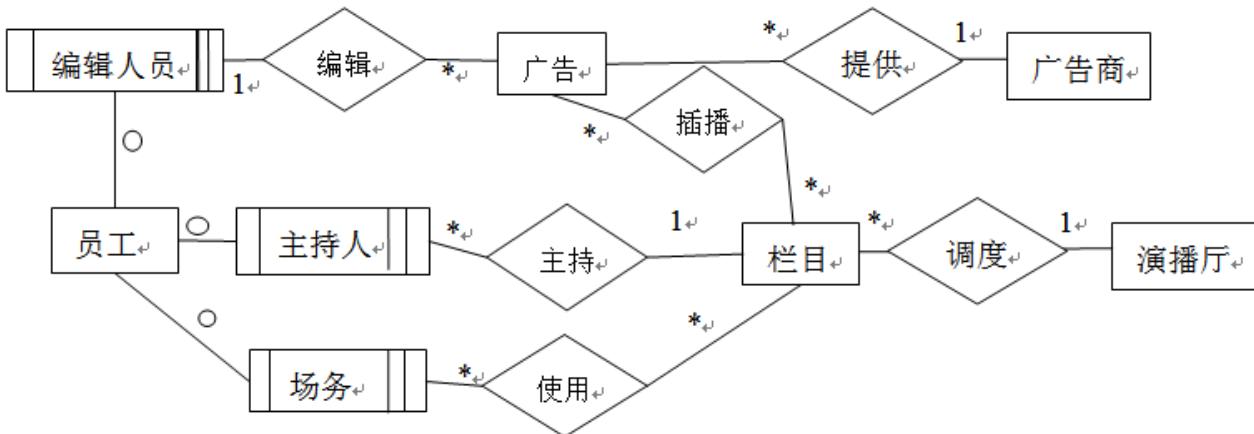
### 【问题 1】



## 【问题 2】

- (1) 编辑人员工号 主键: 广告编号
- (2) 栏目名称 主键: 主持人工号
- (3) 栏目名称、广告编号 主键: 栏目名称、广告编号
- (4) 栏目名称、房间号、场务工号 主键: 栏目名称、场务工号

## 【问题 3】



试题分析：

本题考查数据库设计，设计考点有：数据库的概念结构设计和逻辑结构设计。

## 【问题 1】

由说明每档栏目可以插播多条广告，每条广告可以在多档栏目中插播，可知广告和栏目之间是插播关系且为多比多；一个主持人可以主持

一个栏目，一个栏目可以有多个主持人，故主持人和栏目之间是多比一的关系；多名场务来进行栏目的演出协调，而场务又可以被多个栏目循环使用，故其关系为多对多。补充关系如图 1 所示。

## 【问题 2】

逻辑结构设计中，广告实体中缺少编辑人员工号，主键为广告编号；主持人实体与栏目实体为多对一的关系，故将栏目中主键栏目名称加

入到主持人实体中，主键为主持人工号；插播单位栏目实体和广告实体这种多对多的关系所派生出的实体，其中记录了栏目和广告的主键

信息，故插播单中缺少栏目名称和广告编号信息，主键为栏目名称和广告编号；调度单位场务、栏目和演播厅实体这种多对多的关系所派生的实体，故其记录了栏目名称，房间号，场务工号，主键为栏目名称、房间号和场务工号。

## 【问题 3】

## 2013年上半年软件设计师考试下午真题（权威解析+标准答案）

因为一个广告商可以提供多条广告，一条广告只能由一个广告商提供，故广告商和广告之间的关系为一对多，器关系下图所示。

**问题2 第（1）空：题中没有任何地方提广告时长。怎么能填出告时长？**

题目中确实没有提及有关广告时长的任何信息，所以我们只能根据平常的生活经验来推断出一个合理的答案。

场务为何不与调度有联系，题中也说了“调度单来协调格挡栏目、演播厅和场务”

栏目的主键是：栏目名和播出时间吗？因为某个栏目一天可多次播放啊。

此外调度单中怎么可能会没有播出时间呢？

题目中没有指出调度包括调度时间所以不要去考虑

栏目的主键理解是对的

### 3. 参考答案：

**【问题1】** A1 网络用户； A2 授权用户； UC1 更新信息； UC2 认证；（1）<<include>>

**【问题2】** C1: InternetClient; C2: CustomerList; C3: Administor; C4: RegisteredClient

C5: Customer; (2) 1 ; (3) 0…\*; (4) 0…1; (5) 0…1

**【问题3】**

候选类的选择运用了良性依赖原则“不会在实际中造成危害的依赖关系，都是良性依赖”和接口隔离原则（ISP）。

试题分析：

本题考查面向对象分析中的类图、用例图。

**【问题1】**

用例图中，A1 可以搜索信息，A2 由 A1 派生且 A2 参与了两个用例，根据题中的说明（1）和（2），可知 A1 为网络用户，A2 为授权用户；由用例 UC1 和登录用例之间存在关系，可知 UC1 为更新信息，因为更新信息前必须登录，所以更新信息用例包含登录用例，它们之间的关系为 include。

**【问题2】**

本问题考查类图。考查类图的层次结构和多重度。图中更有两个非常明显的继承结构，即 C3 和 C4 继承与 C1，且 C1 与 C2 是多对一的关系，

根据说明（1）中任何网络用户都可以搜索客户信息，即 C1 为网络用户，C2 为客户提供信息，由此很明显得出 C3 和 C4 在授权用户和系统管理员中选取。根据 C2 和 C5 之间存在聚合关系，且 C2 为客户提供信息，可以推断 C5 为客户提供集。再由 C4 和 C5 之间的关联关系，且 C5 为客户提供集，能对客户提供集进行批量操作的用户 C4 显然就是系统管理员，由此得出 C3 为授权用户。由此（2）~（5）的多重度就显而易见，（2）为 1，（3）为 0…\*，（4）为 1，（5）为 0…\*。

**【问题3】**

候选类的选择运用了良性依赖原则“不会在实际中造成危害的依赖关系，都是良性依赖”和接口隔离原则（ISP）。ISP：使用多个专门的接口比使用单一的总接口要好。一个类对另外一个类的依赖性应当是建立在最小的接口上的。一个接口代表一个角色，不应当将不同的角色都交给一个接口。没有关系的接口合并在一起，形成一个臃肿的大接口，这是对角色和接口的污染。“不应该强迫客户依赖于它们不用的方法。接口属于客户，不属于它所在的类层次结构。”这个说得很明白了，再通俗点说，不要强迫客户使用它们不用的方法，如果强迫

用户使用它们不使用的方法，那么这些客户就会面临由于这些不使用的方法的改变所带来的改变。

**uc1 应该是更新信息吧，uc2 是认证**

题目中明确提到了，更新信息前要先登录，这么明显的提示了

认证并未说明是先登录，而是说需通过系统认证，认证和登录有着本质的区别

## 2013年上半年软件设计师考试下午真题（权威解析+标准答案）

你好

由用例 UC1 和登录用例之间存在关系，可知 UC1 为认证用例，因为用户登录必须先认证，所以登录用例是认证扩展，所以他们之间

的关系为 extend。对于授权用户还可以更新信息，故 UC2 为更新信息用例。

谢谢

问题 2 里面的 C2 和 C5 类的聚合关系，为什么是客户集指向客户类？这两个类分别是什么意思？聚合关系不是应该从部分指向整体吗？

你好

参考答案存在问题，现已修正过来。由于时间精力等方面的原因，没能仔细校对答案的正确性，由此给您带来的不便，深表歉意。

问题 2 的答案和解释不一样啊

你好

答案解析还是之前的，暂时没有更新过来。

谢谢

问题 1 的答案是错误的是吧，问题 2 的答案是正确的对不对？

如果 C2 是客户集，C5 是客户信息，那么他们之间的多重度为何不是（2）1；（3）1..\*

呢？因为如果客户集里面一条客户信息都没有，那就没有客户集这个概念了；与此类似的还有“图书目录和图书”、“订单列表和订单”、“订单和

订单明细”等等，烦请老师讲讲他们之间的多重度，到底该怎么填？是 1 对 1..\* 还是 1 对 0..\* 呢？

答案已经修改过来了

问题(2)当然是 1，问题 3 应该为 0...\*，可能没有客户信息表，或者写成\*，因为客户信息的改删等可能碰巧没有客户信息了，其他情况类似

4. 参考答案：

【问题 1】

（1） $d[i] = t[i]$  （2） $i=m$  （3） $s[k][count[k]] = i$  （4） $Max < d[i]$

【问题 2】

（5）贪心 （6） $O(mn)$

【问题 3】

（7）0 （8）1、5 （9）2、3、4、6 （10）17

试题分析：

本题考查算法的设计和分析技术中的贪心算法。

贪婪算法（Greedy algorithm）是一种对某些求最优解问题的更简单、更迅速的设计技术。用贪婪法设计算法的特点是一步一步地进行，常以当前情况为基础根据某个优化测度作最优选择，而不考虑各种可能的整体情况，它省去了为找最优解要穷尽所有可能而必须耗费的大量时间，它采用自顶向下，以迭代的方法做出相继的贪心选择，每做一次贪心选择就将所求问题简化为一个规模更小的子问题，通过每一步贪心选择，可得到问题的一个最优解，虽然每一步上都要保证能获得局部最优解，但由此产生的全局解有时不一定是最优的，所以贪婪

法不要回溯。

【问题 1】

根据上述思想和题中的说明，首先将  $s[]$  和  $d[]$  数组初始化为 0，然后要做的就是按要求“算法基于最长运行时间作业优先的策略，按顺序先把每个任务分配到一台机器上”，可以推断（1）处为  $d[i] = t[i]$ ，此后需将剩下的  $n-m$  个任务按顺序分配给空闲的机器，故（2）处将  $i$  初始化为以  $m$  为起始的任务，即  $i=m$ ，（3）处所在的位置是分配后  $n-m$  个任务，在这个过程中，必须要对  $s$  矩阵的内容进行修改，但目前已经出的代码没有这个内容，所以此处必然是对  $s$  的修改。从对  $s$  矩阵的注释可以了解到， $s[i][j]$  表示

## 2013年上半年软件设计师考试下午真题（权威解析+标准答案）

机器  $i$  运行的任务  $j$  的编号，此时涉及任务的机器号为  $k$ ，而待分配的任务  $i$  是机器的第  $count[k]$  个任务，即  $s[k][count[k]] = i$ ，(4) 处已经完成了任务的运行，此处需要统计所有机器所运行任务的最长时间，对于每个机器  $i$  的运行时间为  $d[i]$ ，存在  $d[i]$  大于当前的最大时间  $Max$ ，就将当前机器的运行时间  $d[i]$  赋给  $Max$ ，即  $Max < d[i]$ 。

### 【问题 2】

根据以上分析，(5) 处采用了贪心算法的策略，而时间复杂度由算法中的两个嵌套 **for** 循环和两个非嵌套 **for** 循环确定，即为  $O(mn)$ ，

### 【问题 3】

根据题中算法的思想将任务的前三个任务分给三个机器，再将接下来的任务分给最先空闲的机器，故可知机器 0 运行任务 0，机器 1 运行任务 1、5，机器 3 运行任务 2、3、4、6；且运行的最长时间为 17。

题目说明中最后一句话的“将剩余任务一次放入。”的意思是分配了  $M$  台机器后剩下的任务都放在一台机器上吗？ $s[]$  的数组在算完成所有任务的时间有什么作用？

你好

问题：“将剩余任务一次放入。”的意思是分配了  $M$  台机器后剩下的任务都放在一台机器上吗

答：这里不能这么理解，而是先将任务按照时间长度从大到小排序，每台机器依次从任务序列中运行一个任务，那台机器先运行完，就再从任务序列中运行一个任务；这个时候哪台机器先运行完，就再从任务序列中运行一个任务；总之哪台机器空闲并且还有未运行的任务，就从任务序列中取出一个任务运行，而不是全部。题目中的“一次”，可能是编辑错误，应该为“依次”。

问题： $s[]$  的数组在算完成所有任务的时间有什么作用？

答：实际上二维数组就是记录每台机器运行了哪几个任务，在统计完成任务的时间时，就可以通过获取各个机器运行的任务数，来得到相应的运行时间，比较各个时间的大小就可以得到答案。

谢谢

时间复杂度嵌套 **for** 循环有两个和没有嵌套的 **for** 循环，按最高的两个 **for** 循环应该为  $O(n^2)$ ，这里的  $O(mn)$  又是如何来的呢，有时候双有  $O(lgn)$  之类的，这里我见书上的答案又是  $O(2m*n+2m)$

$O(2m*n+2m)$  就等于  $O(m*n)$ ，这是渐进式，查询下书上的资料彻底理解下吧

在两个 **for** 循环里面一个是  $m$ ，一个是  $n$ ，当然是  $m*n$ ，其他两个循环跟它不是一个数量级，故可以忽略

### 5. 参考答案：

- (1) virtual Cloneable \* Clone()=0
- (2) WorkExperience \* obj = new WorkExperience()
- (3) (WorkExperience \* ) work->Clone()
- (4) Resume \* obj = new Resume(this->work)
- (6) (Resume \*)a->Clone()

试题分析：

本题考查原型模型的概念及应用。

原型模型的主要思想：先借用已有系统作为原型模型，通过“样品”不断改进，使得最后的产品就是用户所需要的。原型模型通过向用户提供原型获取用户的反馈，使开发出的软件能够真正反映用户的需求。同时，原型模型采用逐步求精的方法完善原型，使得原型能够“快速”开发，避免了像瀑布模型一样在冗长的开发过程中难以对用户的反馈作出快速的响应。相对瀑布模型而言，原型模型更符合人们开发软件的习惯，使目前较流行的一种实用软件生存期模型。**Prototype** 模式其实就是常说的“虚拟构造函数”一个实现，C++的实现机制中并没有支持这个特性，但是通过不同派生类实现的 **Clone** 接口函数可以完成与“虚拟构造函数”同样的效果。

题中声明一个虚拟基类，所有的原型都是从这个基类继承，(1) 所代表的就是这个基类中的纯虚函数，

## 2013年上半年软件设计师考试下午真题（权威解析+标准答案）

需要供继承者自行实现，即为 `virtual Cloneable * Clone()=0`，(1) 声明一个抽象基类，并定义 `clone()` 函数为纯虚函数。然后根据基类实例化各个子类，并且实现赋复制构造函数，并实现 `clone()` 函数，由此可知(2) 处为 `WorkExperience * Obj`，(3) 处为 `Work`，(4) 处为 `Resume * Obj`。在 `main` 函数中实现 `Resume * b` 对 `*a` 的复制，故根据 C++ 语法(5) 中为 `a->Clone()`。

该题的试题分析和答案不一样啊，到底答案是什么以哪个为准？

(4) 为何不和(2)一样呢？烦请老师讲讲

您好，这是原型模式的特点，第二空和第四空是不一样了，都是定义了 `obj`，但是看下面的属性也是不一样的，2 下面是 `workdate` 什么的，自然是 `Workexperience`,4 下面是 `name` 什么的，自然是 `Resume`。

老师：请问下面答案

(3) `(WorkExperience * ) work->Clone()`

(4) `Resume * obj = new Resume(this->work)`

(6) `(Resume *)a->Clone()`

比如 `(WorkExperience * ) work->Clone()` 这里所表示的意思是什么，`(WorkExperience * ) work` 这里为什么要这样用呢。还有 "`->Clone()`" 又是指什么呢。什么情况下要这样用呢。

在另外一个页面回答过了，这里不重复了

### 6. 参考答案：

(1) `implements`

(2) `WorkExperience obj = new Workexperience()`

(3) `implements`

(4) `(WorkExperience)work.Clone()`

(5) `new Resume(this.work)`

(6) `(Resume)a.Clone()`

试题分析：

本题考查原型模型的概念及应用。

原型模型的主要思想：先借用已有系统作为原型模型，通过“样品”不断改进，使得最后的产品就是用户所需要的。原型模型通过向用户提供原型获取用户的反馈，使开发出的软件能够真正反映用户的需求。同时，原型模型采用逐步求精的方法完善原型，使得原型能够“快速”开发，避免了像瀑布模型一样在冗长的开发过程中难以对用户的反馈作出快速的响应。相对瀑布模型而言，原型模型更符合人们开发

软件的习惯，使目前较流行的一种实用软件生存期模型。

所有 java 类都继承自 `java.lang.Object`，而 `object` 类提供一个 `clone()` 方法，可以将一个 java 对象复制一份。因此在 java 中可以直接使用 `Object` 提供的 `clone()` 方法来实现对象的克隆。能够实现克隆的 java 类必须实现一个标识接口 `Cloneable`，表示这个 java 支持复制。

题中 `WorkExperience` 类和 `Resume` 类需要实现 `Cloneable` 接口，故(1)和(3)为 `implements`，`WorkExperience` 中需要实现 `Clone()` 方法，并将自身复制一份，有下面代码可知(2) 为 `WorkExperience obj = new WorkExperience()`。`Resume` 类中的私有构造方法实现 `WorkExperience` 深复制，故(4)中为 `(WorkExperience) work.clone()`，而 `Resume` 类中 `Clone` 方法实现自身的复制，故(5) 中为 `new Resume(this.work)`。

在 `main` 中实现 `Resume b` 对 `a` 的复制，故(6) 中为 `a.Clone()`。

这边有两个(3)，请问答案呢

你好

`Resume(WorkExperience * work){`

`this->work = (3);`

## 2013年上半年软件设计师考试下午真题（权威解析+标准答案）

这段代码是多余的，是 C++ 代码中的，题目编辑的时候可能没注意，已经改过来了，给您带来的不便还请见谅。

这图的意思不是继承吗？接口实现不是用虚线表示吗？

你好

你的理解是正确的，接口实现在 UML 图中是使用虚线表示的，此题在图的表示上存在不严谨之处，但解答时我们还是需要结合题目所给的代码，在 java 中 Cloneable 是一个接口，因此 (1) 处填写 extends 显然不妥。实际上参看以前的考题，图的表示都存在这一问题，这也提醒我们考题中图仅供参考，答题仍然要以题目所给代码语言为准。

你好，请教两个问题：1、如何判定 JAVA 程序填空 (1) 是填实现接口还是继承父类？

2、能否逐行解析一下 JAVA 程序中 RESUME 类？

你好

问题：1、如何判定 JAVA 程序填空 (1) 是填实现接口还是继承父类？

答：因为 Cloneable 在 java 中是一个接口，(1) 空你填 extends，语法上不对，这里需要你有 java 基础，如果不确定 Cloneable 是一个接口还是抽象类，可以选作 c++ 题。

问题：2、能否逐行解析一下 JAVA 程序中 RESUME 类？

答： private String name; private String sex; private String age; private WorkExperience work;  
定义 4 个成员属性 name、sex、age、work。

public Resume(string name){ this.name = name; work = new WorkExperience();}

根据传入的参数 name 构造 Resume (初始化成员属性 name、work 的值)。

private Resume(WorkExperience work){ this.work = (4) ;}

根据传入的参数 work，克隆此 work 对象构造 Resume (初始化成员属性 work 的值)。

public void SetPersonInfo(string sex,string age){/\*实现省略\*/}

用于设置成员属性 sex、age 的值 public void SetWorkExperience(string workDate,string company){/\*实现省略\*/}

根据传入的 workDate, company 参数设置成员属性 work 的值。

public Object clone(){Resume Obj = (5) ; return Obj;}

实现 Cloneable 接口的 clone() 方法。调用第二个构造方法 Resume(WorkExperience work) 克隆对象。

比如：Resume b = (Resume)a.Clone() ; 此处需要强转是因为 a.clone() 传回的是 Object 对象 【public Object clone()】，对象 a 调用实现 Cloneable 接口的 Clone() 方法，将自身克隆一份赋值给对象 b；而在 Clone() 方法中，会调用 Resume(WorkExperience work) 构造方法，将 a 的成员属性 work 克隆一份，这样对象 b 中的 work 成员属性的值就和 a 中的 work 的值一样。