软件设计师

2016年下半年试题

本试卷为：**样式1**

样式1：适用于模拟考试，所有答案在最后面。

样式2：适用于复习，每道题的题目和答案在一起。

本试卷由**跨步软考**提供

我们目前提供的免费服务有：

- 手机APP刷题
- 网页版刷题
- 真题pdf版下载
- 视频课程下载
- 其他资料下载

更多免费服务请访问我们的官网：https://kuabu.xyz

你也可以关注我们的微信公众号：**跨步软考**

如果您发现试题有错误，您可以通过以下方式联系我们

-
- 客服邮箱：kuabu@outlook.com
- 您也可以在微信公众号后台留言

上午综合试卷

第1题：在程序运行过程中，CPU需要将指令从内存中取出并加以分析和执行。CPU依据（ ）来区分在内存中以二进制编码形式存放的指令和数据。

A. 指令周期的不同阶段

B. 指令和数据的寻址方式

C. 指令操作码的译码结果

D. 指令和数据所在的存储单元

第2题：计算机在一个指令周期的过程中，为从内存读取指令操作码，首先要将（ ）的内容送到地址总线上。

A. 指令寄存器（IR）

B. 通用寄存器（GR）

C. 程序计数器（PC）

D. 状态寄存器（PSW）

第3题：设16位浮点数，其中阶符1位、阶码值6位、数符1位、尾数8位。若阶码用移码表示，尾数用补码表示，则该浮点数所能表示的数值范围是（ ）。

A. -2^{64} ~ $(1-2^{-8})2^{64}$

B. -2^{63} ~ $(1-2^{-8})2^{63}$

C. -2^{64} ~ $(1-2-(1-2^{-8})2^{64}$ ~ $(1-2^{-8})2^{64}$

D. $-(1-2^{-8})2^{63}$ ~ $(1-2^{-8})2^{63}$

第4题：已知数据信息为16位，最少应附加（ ）位校验位，以实现海明码纠错。

A. 3

B. 4

C. 5

D. 6

第5题：将一条指令的执行过程分解为取址、分析和执行三步，按照流水方式执行，若取指时间t取址=4△t、分析时间t分析=2△t、执行时间t执行=3△t，则执行完100条指令，需要的时间为（ ）△t。

A. 200

B. 300

C. 400

D. 405

第6题：以下关于Cache与主存间地址映射的叙述中，正确的是（ ）。

A. 操作系统负责管理Cache与主存之间的地址映射

B. 程序员需要通过编程来处理Cache与主存之间的地址映射

C. 应用软件对Cache与主存之间的地址映射进行调度

D. 由硬件自动完成Cache与主存之间的地址映射

第7题：可用于数字签名的算法是（ ）。

A. RSA

B. IDEA

C. RC4

D. MD5

第8题：（ ）不是数字签名的作用。

A. 接收者可验证消息来源的真实性

B. 发送者无法否认发送过该消息

C. 接收者无法伪造或篡改消息

D. 可验证接收者合法性

第9题：在网络设计和实施过程中要采取多种安全措施，其中（ ）是针对系统安全需求的措施。

A. 设备防雷击

B. 入侵检测

C. 漏洞发现与补丁管理

D. 流量控制

第10题：（ ）的保护期限是可以延长的。

A. 专利权

B. 商标权

C. 著作权

D. 商业秘密权

第11题：甲公司软件设计师完成了一项涉及计算机程序的发明。之后，乙公司软件设计师也完成了与甲公司软件设计师相同的涉及计算机程序的发明。甲、乙公司于同一天向专利局申请发明专利。此情形下，（ ）是专利权申请人。

A. 甲公司

B. 甲、乙两公司

C. 乙公司

D. 由甲、乙公司协商确定的公司

第12题：甲、乙两厂生产的产品类似，且产品都使用“B"商标。两厂于同一天向商标局申请商标注册，且申请注册前两厂均未使用“B"商标。此情形下，（ ）能核准注册。

A. 甲厂

B. 由甲、乙厂抽签确定的厂

C. 乙厂

D. 甲、乙两厂

第13题：在FM方式的数字音乐合成器中，改变数字载波频率可以改变乐音的（13），改变它的信号幅度可以改变乐音的（14）。

A. 音调

B. 音色

C. 音高

D. 音质

第14题：在FM方式的数字音乐合成器中，改变数字载波频率可以改变乐音的（13），改变它的信号幅度可以改变乐音的（14）。

A. 音调

B. 音域

C. 音高

D. 带宽

第15题：结构化开发方法中，（ ）主要包含对数据结构和算法的设计。

A. 体系结构设计

B. 数据设计

C. 接口设计

D. 过程设计

第16题：在敏捷过程的开发方法中，（ ）使用了迭代的方法，其中，把每段时间（30天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。

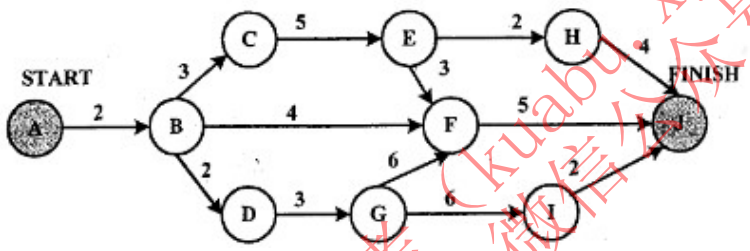
A. 极限编程XP

B. 水晶法

C. 并列争球法

D. 自适应软件开发

第17题：某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间（天），则完成该项目的最少时间为（17）天。活动BC和BF最多可以晚开始（18）天而不会影响整个项目的进度。



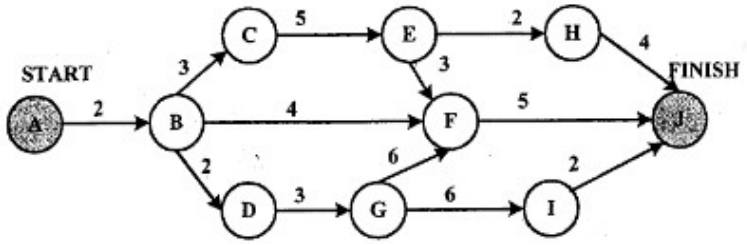
A. 11

B. 15

C. 16

D. 18

第18题：某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间（天），则完成该项目的最少时间为（17）天。活动BC和BF最多可以晚开始（18）天而不会影响整个项目的进度。



A. 0和7

B. 0和11

C. 2和7

D. 2和11

第19题：成本估算时，（ ）方法以规模作为成本的主要因素，考虑多个成本驱动因子。该方法包括三个阶段性模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。

A. 专家估算

B. Wolverton

C. COCOMO

D. COCOMO Ⅱ

第20题：逻辑表达式求值时常采用短路计算方式。“&&"、“||”、“！”分别表示逻辑与、或、非运算，“&&”、“||”为左结合，“！”为右结合，优先级从高到低为“！”、“&&”、“||”。对逻辑表达式“x&&（y II!z）”进行短路计算方式求值时，（ ）。

A. x为真，则整个表达式的值即为真，不需要计算y和z的值

B. x为假，则整个表达式的值即为假，不需要计算y和z的值

C. x为真，再根据z的值决定是否需要计算y的值

D. x为假，再根据y的值决定是否需要计算z的值

第21题：常用的函数参数传递方式有传值与传引用两种。（ ）。

A. 在传值方式下，形参与实参之间互相传值

B. 在传值方式下，实参不能是变量

C. 在传引用方式下，修改形参实质上改变了实参的值。

D. 在传引用方式下，实参可以是任意的变量和表达式。

第22题：二维数组a[1..N，1..N]可以按行存储或按列存储。对于数组元素a[i,j]（1<=i,j<=N），当（ ）时，在按行和按列两种存储方式下，其偏移量相同。

A. i≠j

B. i=j

C. i>j

D. i<j

第23题：实时操作系统主要用于有实时要求的过程控制等领域。实时系统对于来自外部的事件必须在（ ）。

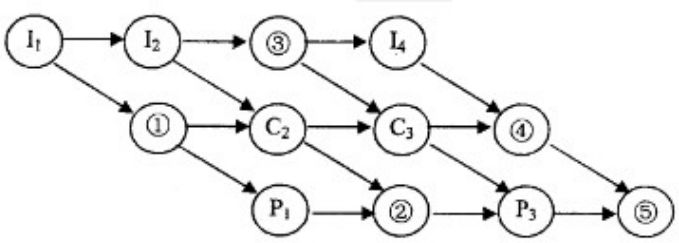
A. 一个时间片内进行处理

B. 一个周转时间内进行处理

C. 一个机器周期内进行处理

D. 被控对象规定的时间内做出及时响应并对其进行处理

第24题：假设某计算机系统中只有一个CPU、一台输入设备和一台输出设备，若系统中有四个作业T1、T2、T3和T4，系统采用优先级调度，且T1的优先级>T2的优先级>T3的优先级>T4的优先级。每个作业Ti具有三个程序段：输入Ii、计算Ci和输出Pi（i=1，2，3，4），其执行顺序为Ii→Ci→Pi。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为（24），③、④、⑤分别为（25）。



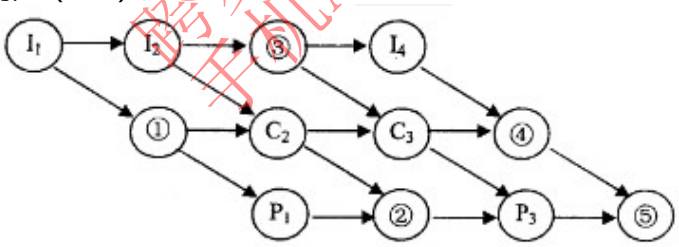
A. I2、P2

B. I2、C2

C. C1、P2

D. C1、P3

第25题：假设某计算机系统中只有一个CPU、一台输入设备和一台输出设备，若系统中有四个作业T1、T2、T3和T4，系统采用优先级调度，且T1的优先级>T2的优先级>T3的优先级>T4的优先级。每个作业Ti具有三个程序段：输入Ii、计算Ci和输出Pi（i=1，2，3，4），其执行顺序为Ii→Ci→Pi。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为（24），③、④、⑤分别为（25）。



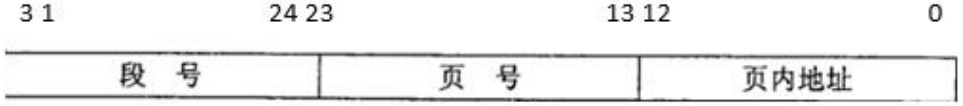
A. C2、C4、P4

B. I2、I3、C4

C. I3、P3、P4

D. I3、C4、P4

第26题：假设段页式存储管理系统中的地址结构如下图所示，则系统（ ）。



A. 最多可有256个段，每个段的大小均为2048个页，页的大小为8K

B. 最多可有256个段，每个段最大允许有2048个页，页的大小为8K

C. 最多可有512个段，每个段的大小均为1024个页，页的大小为4K

D. 最多可有512个段，每个段最大允许有1024个页，页的大小为4K

第27题：假设系统中有n个进程共享3台扫描仪，并采用PV操作实现进程同步与互斥。若系统信号量S的当前值为-1，进程P1、P2又分别执行了1次P（S）操作，那么信号量S的值应为（ ）。

A. 3

B. -3

C. 1

D. -1

第28题：某字长为32位的计算机的文件管理系统采用位示图（bitmap）记录磁盘的使用情况。若磁盘的容量为300GB，物理块的大小为1MB，那么位示图的大小为（ ）个字。

A. 1200

B. 3200

C. 6400

D. 9600

第29题：某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不适宜采用（29）模型，主要是因为这种模型（30）。

A. 瀑布

B. 原型

C. 增量

D. 喷泉

第30题：某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不适宜采用（29）模型，主要是因为这种模型（30）。

A. 不能解决风险

B. 不能快速提交软件

C. 难以适应变化的需求

D. 不能理解用户的需求

第31题：（ ）不属于软件质量特性中的可移植性。

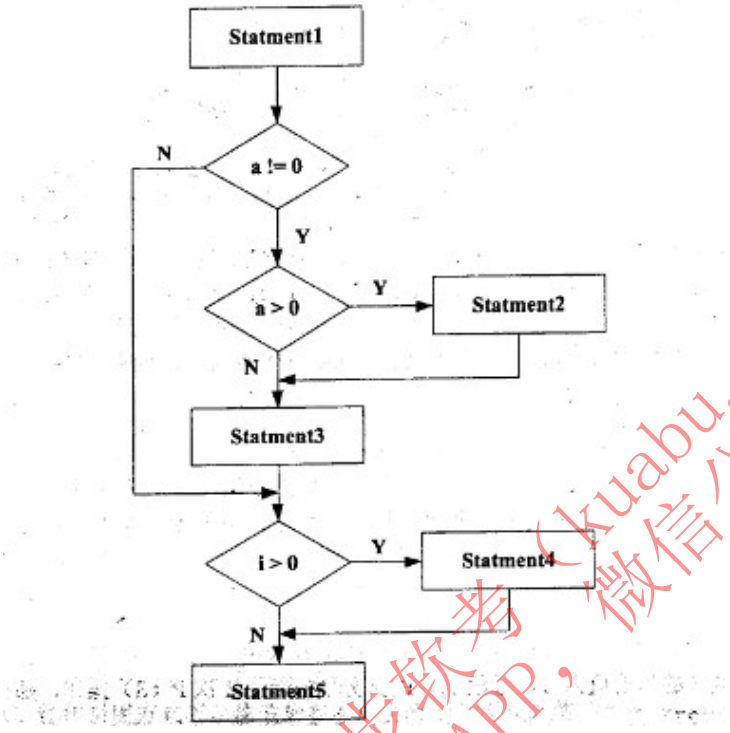
A. 适应性

B. 易安装性

C. 易替换性

D. 易理解性

第32题：对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要（32）个测试用例。采用McCabe度量法计算该程序的环路复杂性为（33）。



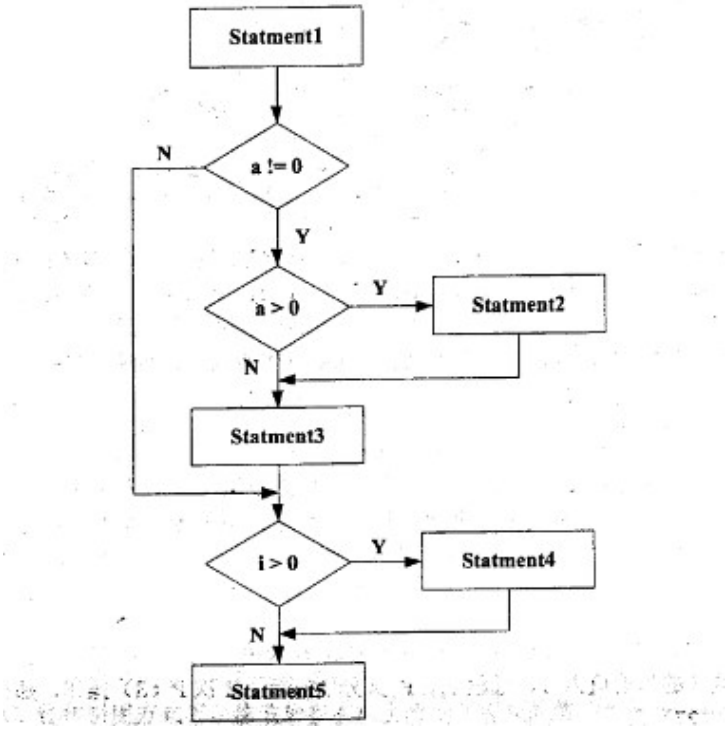
A. 3

B. 4

C. 6

D. 8

第33题：对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要（32）个测试用例。采用McCabe度量法计算该程序的环路复杂性为（33）。



A. 1

B. 2

C. 3

D. 4

第34题：计算机系统的（ ）可以用MTBF/（1+MTBF）来度量，其中MTBF为平均失效间隔时间。

A. 可靠性

B. 可用性

C. 可维护性

D. 健壮性

第35题：以下关于软件测试的叙述中，不正确的是（ ）。

A. 在设计测试用例时应考虑输入数据和预期输出结果

B. 软件测试的目的是证明软件的正确性

C. 在设计测试用例时，应该包括合理的输入条件

D. 在设计测试用例时，应该包括不合理的输入条件

第36题：某模块中有两个处理A和B，分别对数据结构X写数据和读数据，则该模块的内聚类型为（ ）内聚。

A. 逻辑

B. 过程

C. 通信

D. 内容

第37题：在面向对象方法中，不同对象收到同一消息可以产生完全不同的结果，这一现象称为（ ）。在使用时，用户可以发送一个通用的消息，而实现的细节则由接收对象自行决定。

A. 接口

B. 继承

C. 覆盖

D. 多态

第38题：在面向对象方法中，支持多态的是（ ）。

A. 静态分配

B. 动态分配

C. 静态类型

D. 动态绑定

第39题：面向对象分析的目的是为了获得对应用问题的理解，其主要活动不包括（ ）。

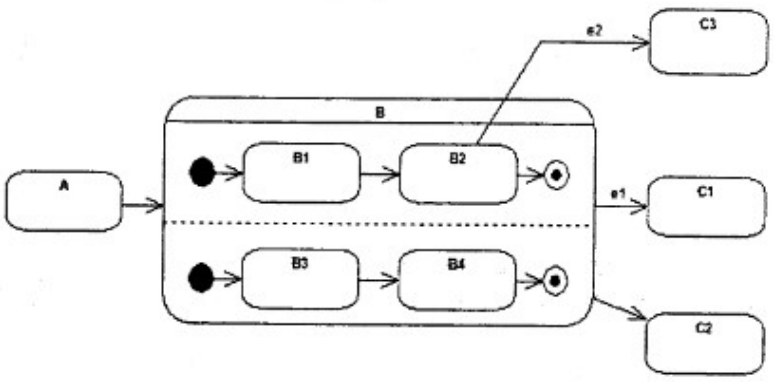
A. 认定并组织对象

B. 描述对象间的相互作用

C. 面向对象程序设计

D. 确定基于对象的操作

第40题：如下所示的UML状态图中，（ ）时，不一定会离开状态B。



A. 状态B中的两个结束状态均达到

B. 在当前状态为B2时，事件e2发生

C. 事件e2发生

D. 事件e1发生

第41题：以下关于UML状态图中转换（transition）的叙述中，不正确的是（ ）。

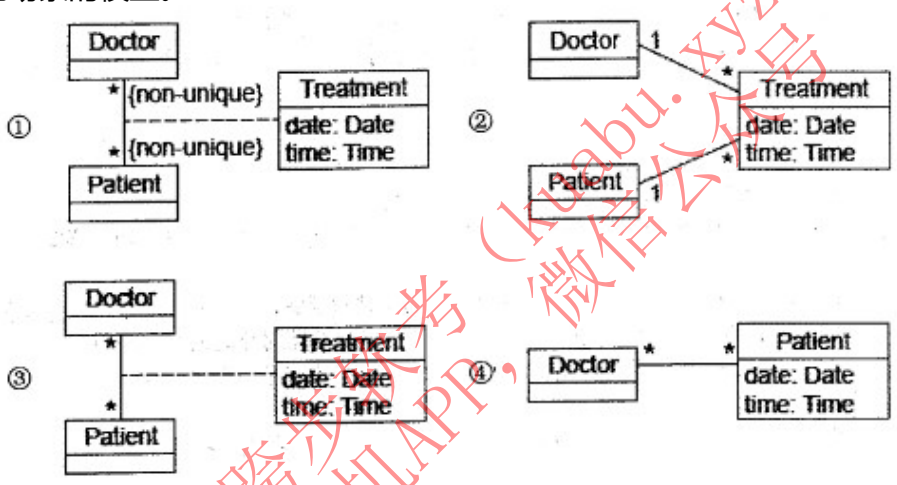
A. 活动可以在转换时执行也可以在状态内执行

B. 监护条件只有在相应的事件发生时才进行检查

C. 一个转换可以有事件触发器、监护条件和一个状态

D. 事件触发转换

第42题：下图①②③④所示是UML（42）。现有场景：一名医生（Doctor）可以治疗多位病人（Patient），一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时，需要存储治疗（Treatment）的日期和时间。以下①②③④图中（43）。是描述此场景的模型。



A. 用例图

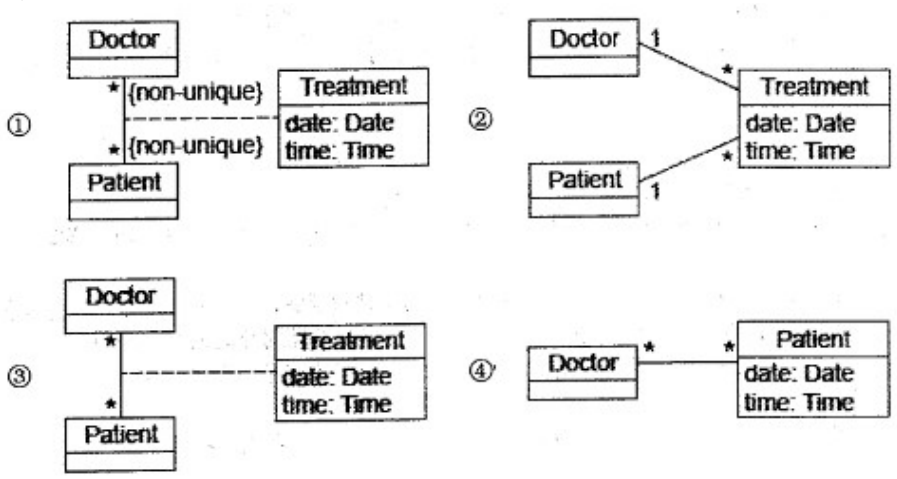
B. 对象图

C. 类图

D. 协作图

第43题：下图①②③④所示是UML（42）。现有场景：一名医生（Doctor）可以治疗多位病人（Patient），一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时，需要存储治疗（Treatment）的日期和时间。以下①②③④图中（43）。是描

述此场景的模型。



A. ①

B. ②

C. ③

D. ④

第44题：（44）模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下（45）情况适合选用该模式。
①一个客户需要使用一组相关对象
②一个对象的改变需要改变其它对象
③需要使用一个算法的不同变体
④许多相关的类仅仅是行为有异

A. 命令（Command）

B. 责任链（Chain of Responsibility）

C. 观察者（Observer）

D. 策略（Strategy）

第45题：（44）模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下（45）情况适合选用该模式。
①一个客户需要使用一组相关对象
②一个对象的改变需要改变其它对象
③需要使用一个算法的不同变体
④许多相关的类仅仅是行为有异

A. ①②

B. ②③

C. ③④

D. ①④

第46题：（46）模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创 建不同的表示。以下（47）情况适合选用该模式。
①抽象复杂对象的构建步骤
②基于构建过程的具体实现构建复杂对象的不同表示
③一个类仅有一个实例
④一个类的实例只能有几个不同状态组合中的一种

A. 生成器（Builder）

B. 工厂方法（Factory Method）

C. 原型（Prototype）

D. 单例（ Singleton）

第47题：（46）模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创 建不同的表示。以下（47）情况适合选用该模式。
①抽象复杂对象的构建步骤
②基于构建过程的具体实现构建复杂对象的不同表示
③一个类仅有一个实例
④一个类的实例只能有几个不同状态组合中的一种

A. ①②

B. ②③

C. ③④

D. ①④

第48题：由字符a、b构成的字符串中，若每个a后至少跟一个b，则该字符串集合可用正规式表示为（ ）。

A. （b|ab）*

B. （ab*）*

C. （a*b*）*

D. （a|b）*

第49题：乔姆斯基（Chomsky）将文法分为4种类型，程序设计语言的大多数语法现象可用其中的（ ）描述。

A. 上下文有关文法

B. 上下文无关文法

C. 正规文法

D. 短语结构文法

第50题：运行下面的C程序代码段，会出现（ ）错误。

```
int k=0;
for(;k<100;);
{k++;}
```

A. 变量未定义

B. 静态语义

C. 语法

D. 动态语义

第51题：在数据库系统中，一般由DBA使用DBMS提供的授权功能为不同用户授权，其主要目的是为了保证数据库的（ ）。

A. 正确性

B. 安全性

C. 一致性

D. 完整性

第52题：给定关系模式R（U,F），其中：U为关系模式R中的属性集，F是U上的一组函数依赖。假设U={A1，A2，A3，A4}，F={A1→A2，A1A2→A3，A1→A4，A2→A4}，那么关系R的主键应为（52）。函数依赖集F中的（53）是冗余的。

A. A1

B. A1A2

C. A1A3

D. A1A2A3

第53题：给定关系模式R（U,F），其中：U为关系模式R中的属性集，F是U上的一组函数依赖。假设U={A1，A2，A3，A4}，F={A1→A2，A1A2→A3，A1→A4，A2→A4}，那么关系R的主键应为（52）。函数依赖集F中的（53）是冗余的。

A. A1→A2

B. A1A2→A3

C. A1→A4

D. A2→A4

第54题：给定关系R（A，B，C，D）和关系S（A，C，E，F），对其进行自然连接运算R?S后的属性列为（54）个；与$\sigma_{R.B>S.E}(R?S)$等价的关系代数表达式为（55）。

A. 4

B. 5

C. 6

D. 8

第55题：给定关系R（A，B，C，D）和关系S（A，C，E，F），对其进行自然连接运算R?S后的属性列为（54）个；与$\sigma_{R.B>S.E}(R?S)$等价的关系代数表达式为（55）。

A. $\sigma_{2>7}(R \times S)$

B. $\pi_{1,2,3,4,7,8}(\sigma_{1=5\wedge 2>7\wedge 3=6}(R \times S))$

C. $\sigma_{2>'7'}(R \times S)$

D. $\pi_{1,2,3,4,7,8}(\sigma_{1=5\wedge 2>'\ 7'\ \wedge 3=6}(R \times S))$

第56题：下列查询B=“大数据”且F=“开发平台”，结果集属性列为A、B、C、F的关系代数表达式中，查询效率最高的是（ ）。

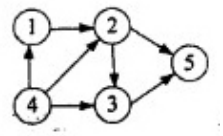
A. π1,2,3,8（σ2=" 大数据" ^ 1=5 ^ 3=6 ^ 8=" 开发平台"（R×S））

B. π1,2,3,8（σ1=5 ^ 3=6 ^ 8=" 开发平台"（σ2=" 大数据"（R）×S））

C. π1,2,3,8（σ2=" 大数据" ^ 1=5 ^ 3=6（R×σ4=" 开发平台"（S））

D. π1,2,3,8（σ1=5 ^ 3=6（σ2=" 大数据"（R）×σ4=" 开发平台"（S））

第57题：拓扑序列是有向无环图中所有顶点的一个线性序列，若有向图中存在弧<v，w>或存在从顶点v到w的路径，则在该有向图的任一拓扑序列中，v一定在w之前。下面有向图的拓扑序列是（ ）。



A. 41235

B. 43125

C. 42135

D. 41325

第58题：设有一个包含n个元素的有序线性表。在等概率情况下删除其中的一个元素，若采用顺序存储结构，则平均需要移动（58）个元素；若采用单链表存储，则平均需要移动（59）个元素。

A. 1

B. (n-1)/2

C. logn

D. n

第59题：设有一个包含n个元素的有序线性表。在等概率情况下删除其中的一个元素，若采用顺序存储结构，则平均需要移动（58）个元素；若采用单链表存储，则平均需要移动（59）个元素。

A. 0

B. 1

C. (n-1)/2

D. n/2

第60题：具有3个节点的二叉树有（ ）种形态。

A. 2

B. 3

C. 5

D. 7

第61题：以下关于二叉排序树（或二叉查找树、二叉搜索树）的叙述中，正确的是（ ）。

A. 对二叉排序树进行先序、中序和后序遍历，都得到结点关键字的有序序列

B. 含有n个结点的二叉排序树高度为（$\log_2 n$）+1

C. 从根到任意一个叶子结点的路径上，结点的关键字呈现有序排列的特点

D. 从左到右排列同层次的结点，其关键字呈现有序排列的特点

第62题：下表为某文件中字符的出现频率，采用霍夫曼编码对下列字符编码，则字符序列“bee”的编码为（62）；编码“110001001101”的对应的字符序列为（63）。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

A. 10111011101

B. 10111001100

C. 001100100

D. 110011011

第63题：下表为某文件中字符的出现频率，采用霍夫曼编码对下列字符编码，则字符序列“bee”的编码为（62）；编码“110001001101”的对应的字符序列为（63）。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

A. bad

B. bee

C. face

D. bace

第64题：两个矩阵A_{m*n}和B_{n*p}相乘，用基本的方法进行，则需要的乘法次数为m*n*p。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定M_i，$M_{(i+1)}$，…，Mj多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用m[i,j]表示，其递归式定义为：

$$m[i,j]=\begin{cases}0 & i \geq j \\ \min\limits_{i\leq k\leq j}\{m[i,k]+m[k+1,j]+p_{i-1}p_kp_j\} & i<j\end{cases}$$

其中i、j和k为矩阵下标，矩阵序列中Mi的维度为（pi-1）*pi采用自底向上的方法实现该算法来确定n个矩阵相乘的顺序，其时间复杂度为（64）。若四个矩阵M1、 M2、M3、M4相乘的维度序列为2、6、3、10、3，采用上述算法求解，则乘法次数为（65）。

A. O（n^2）

B. O（n^2lgn）

C. O（n^3）

D. O（n^3lgn）

第65题：两个矩阵A_{m*n}和B_{n*p}相乘，用基本的方法进行，则需要的乘法次数为m*n*p。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定M_i，$M_{(i+1)}$，…，Mj多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用m[i,j]表示，其递归式定义为：

$$m[i,j]=\begin{cases}0 & i \geq j \\ \min\limits_{i\leq k\leq j}\{m[i,k]+m[k+1,j]+p_{i-1}p_kp_j\} & i<j\end{cases}$$

其中i、j和k为矩阵下标，矩阵序列中Mi的维度为（pi-1）*pi采用自底向上的方法实现该算法来确定n个矩阵相乘的顺序，其时间复杂度为（64）。若四个矩阵M1、 M2、M3、M4相乘的维度序列为2、6、3、10、3，采用上述算法求解，则乘法次数为（65）。

A. 156

B. 144

C. 180

D. 360

第66题：以下协议中属于应用层协议的是（66），该协议的报文封装在（67）。

A. SNMP

B. ARP

C. ICMP

D. X.25

第67题：以下协议中属于应用层协议的是（66），该协议的报文封装在（67）。

A. TCP

B. IP

C. UDP

D. ICMP

第68题：某公司内部使用wb.xyz.com.cn作为访问某服务器的地址，其中wb是（ ）。

A. 主机名

B. 协议名

C. 目录名

D. 文件名

第69题：如果路由器收到了多个路由协议转发的关于某个目标的多条路由，那么决定采用哪条路由的策略是（ ）。

A. 选择与自己路由协议相同的

B. 选择路由费用最小的

C. 比较各个路由的管理距离

D. 比较各个路由协议的版本

第70题：与地址220.112.179.92匹配的路由表的表项是（ ）。

A. 220.112.145.32/22

B. 220.112.145.64/22

C. 220.112.147.64/22

D. 220.112.177.64/22

第71题：Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a（71）, open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.
Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard.

Software systems have orders of magnitude more （72）than computers do.
Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some（73）fashion,and the complexity of the whole increases much more than linearly.
The complexity of software is a(an)（74）property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities（75）in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.
Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. task

B. job

C. subroutine

D. program

第72题：Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a（71）, open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.
Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard.
Software systems have orders of magnitude more （72）than computers do.
Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some（73）fashion,and the complexity of the whole increases much more than linearly.
The complexity of software is a(an)（74）property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities（75）in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.
Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. states

B. parts

C. conditions

D. expressions

第73题：Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a（71）, open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.
Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more （72）than computers do.
Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some（73）fashion,and the complexity of the whole increases much more than linearly.
The complexity of software is a(an)（74）property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities（75）in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.
Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. linear

B. nonlinear

C. parallel

D. additive

第74题：Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a（71）, open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.
Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more （72）than computers do.
Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some（73）fashion,and the complexity of the whole increases much more than linearly.
The complexity of software is a(an)（74）property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the

models, and verifying those properties experimentally. This worked because the complexities (75) in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.
Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. surface

B. outside

C. exterior

D. essential

第75题：Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a (71) , open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.
Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more (72) than computers do.
Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some (73) fashion,and the complexity of the whole increases much more than linearly.
The complexity of software is a(an) (74) property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities (75) in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.
Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. fixed

B. included

C. ignored

D. stabilized

下午案例分析

第1题：阅读下列说明，回答问题1至问题4，将解答填入答题纸的对应栏内。

【说明】

某证券交易所为了方便提供证券交易服务，欲开发一证券交易平台，该平台的主要功能如下：

（1）开户。根据客户服务助理提交的开户信息，进行开户，并将客户信息存入客户记录中，账户信息（余额等）存入账户记录中；

（2）存款。客户可以向其账户中存款，根据存款金额修改账户余额；

（3）取款。客户可以从其账户中取款，根据取款金额修改账户余额；

（4）证券交易。客户和经纪人均可以进行证券交易（客户通过在线方式，经纪人通过电话），将交易信息存入交易记录中；

（5）检查交易。平台从交易记录中读取交易信息，将交易明细返回给客户。

现采用结构化方法对该证券交易平台进行分析与设计，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。

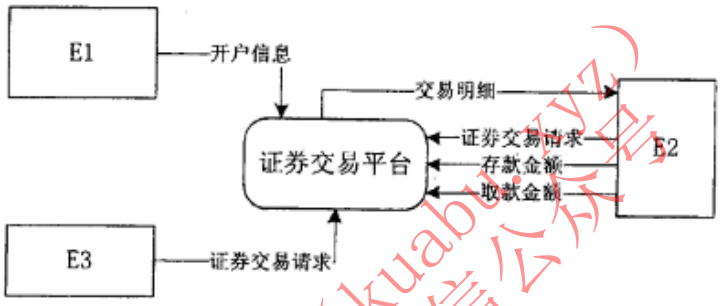


图1-1 上下文数据流图

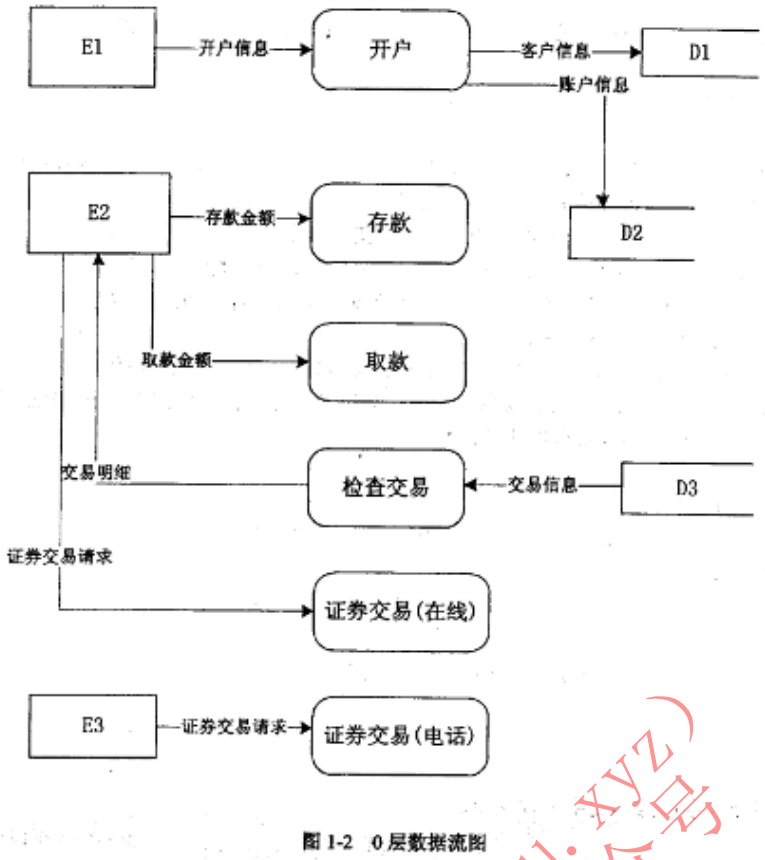


图 1-2 0 层数据流图

问题：1.1 （3分）
使用说明中的词语，给出图1-1中的实体E1-E3的名称。问题：1.2 （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：1.3 （4分）
根据说明和图中的术语，补充图1-2中缺失的数据流及其起点和终点。问题：1.4 （5分）
实际的证券交易通常是在证券交易中心完成的，因此，该平台的“证券交易”功能需将交易信息传递给证券交易中心。针对这个功能需求，需要对图1-1和图1-2进行哪些修改，请用200字以内的文字加以说明。

第2题：【说明】
某宾馆为了有效地管理客房资源，满足不同客户需求，拟构建一套宾馆信息管理系统，以方便宾馆管理及客房预订等业务活动。
【需求分析结果】
该系统的部分功能及初步需求分析的结果如下：
（1）宾馆有多个部门，部门信息包括部门号、部门名称、电话、经理。每个部门可以有多名员工，每名员工只属于一个部门；每个部门只有一名经理，负责管理本部门。
（2）员工信息包括员工号、姓名、岗位、电话、工资，其中，员工号唯一标识员工关系中的一个元组，岗位有经理、业务员。
（3）客房信息包括客房号（如1301、1302等）、客房类型、收费标准、入住状态（已入住/未入住），其中客房号唯一标识客房关系中的一个元组，不同客房类型具有不同的收费标准。
（4）客户信息包括客户号、单位名称、联系人、联系电话、联系地址，其中客户号唯一标识客户关系中的一个元组。
（5）客户预订客房时，需要填写预订申请。预订申请信息包括申请号、客户号、入住时间、入住天数、客房类型、客房数量，其中，一个申请号唯一标识预订申请中的一个元组；一位客户可以有多个预订申请，但一个预订申请对应唯一的一位客户。
（6）当客户入住时，业务员根据客户的预订申请负责安排入住客房事宜。安排信息包括客房号、

姓名、性别、身份证号、入住时间、天数、电话，其中客房号、身份证号和入住时间唯一标识一次安排。一名业务员可以安排多个预订申请，一个预订申请只由一名业务员安排，而且可安排多间同类型的客房。

【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。

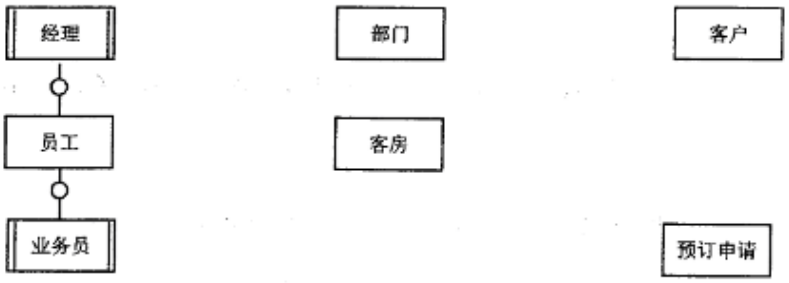


图 2-1 实体联系图

【关系模式设计】

部门（部门号，部门名称，经理，电话）

员工（员工号，（ a ），姓名，岗位，电话，工资）

客户（（ b ），联系人，联系电话，联系地址）

客房（客房号，客房类型，收费标准，入住状态）

预订申请（（ c ），入住时间，天数，客房类型，客房数量）

安排（申请号，客房号，姓名，性别，（ d ），天数，电话，业务员）问题：2.1 （4分）

根据问题描述，补充四个联系，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和联系4代替，联系的类型为1:1、1:n和m:n（或1:1，和1:*和*:*）。问题：2.2 （8分）

（1）根据题意，将关系模式中的空（a）~（d）补充完整，并填入答题纸对应的位置上。

（2）给出“预订申请”和“安排”关系模式的主键和外键。问题：2.3 （3分）

【关系模式设计】中的“客房”关系模式是否存在规范性问题，请用100字以内文字解释你的观点（若存在问题，应说明如何修改“客房”关系模式）。

第3题：【说明】

某种出售罐装饮料的自动售货机．（ Vending Machine）的工作过程描述如下：

（1）顾客选择所需购买的饮料及数量。

（2）顾客从投币口向自动售货机中投入硬币（该自动售货机只接收硬币）。硬币器收集投入的硬币并计算其对应的价值。如果所投入的硬币足够购买所需数量的这种饮料且饮料数量足够，则推出饮料，计算找零，顾客取走饮料和找回的硬币；如果投入的硬币不够或者所选购的饮料数量不足，则提示用户继续投入硬币或重新选择饮料及数量。

（3）一次购买结束之后，将硬币器中的硬币移走（清空硬币器），等待下一次交易。自动售货机还设有一个退币按钮，用于退还顾客所投入的硬币。已经成功购买饮料的钱是不会被退回的。

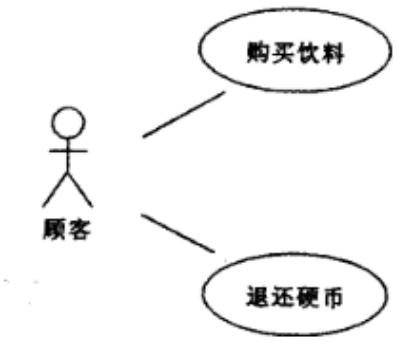


图 3-1 用例图

采用面向对象方法分析和设计该自动售货机的软件系统，得到如图3-1所示的用例图，其中，用例“购买饮料”的用例规约描述如下。
参与者：顾客。
主要事件流：
1．顾客选择需要购买的饮料和数量，投入硬币；
2．自动售货机检查顾客是否投入足够的硬币；
3．自动售货机检查饮料储存仓中所选购的饮料是否足够；
4．自动售货机推出饮料；
5．自动售货机返回找零。
各选事件流：
2a．若投入的硬币不足，则给出提示并退回到1；
3a．若所选购的饮料数量不足，则给出提示并退回到1。
根据用例“购买饮料”得到自动售货机的4个状态：“空闲”状态、“准备服务”状态、“可购买”状态以及“饮料出售”状态，对应的状态图如图3-2所示。
所设计的类图如图3-3所示。

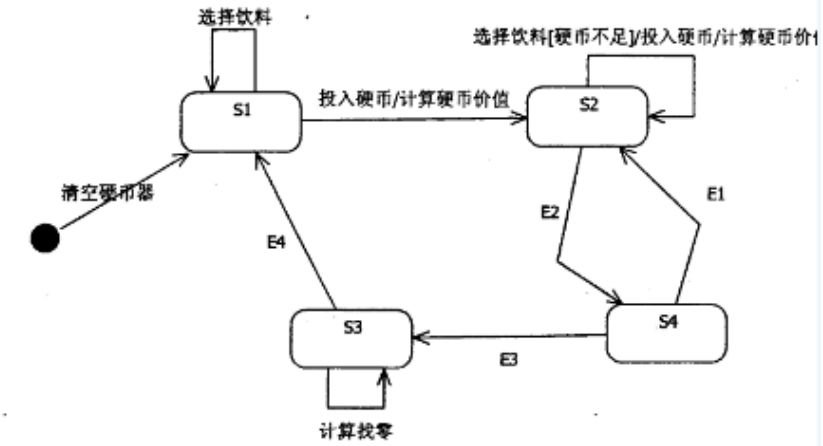


图 3-2 状态图

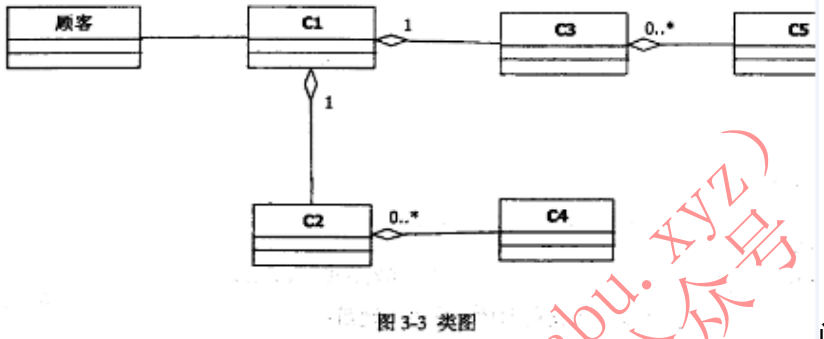


图 3-3 类图

问题：3.1 （6分）
根据说明中的描述，使用说明中的术语，给出图3-2中的S1～S4所对应的状态名。问题：3.2 （4分）
根据说明中的描述，使用说明中的术语，给出图3-2中的E1～E4所对应的事件名问题：3.3 （5分）
根据说明中的描述，使用说明中的术语，给出图3-3中C1～C5所对应的类名。

第4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】
模式匹配是指给定主串t和子串s，在主串t中寻找子串s的过程，其中s称为模式。如果匹配成功，返回s在t中的位置，否则返回-1 。
KMP算法用next数组对匹配过程进行了优化。KMP算法的伪代码描述如下：
1．在串t和串s中，分别设比较的起始下标i=j=0。
2．如果串t和串s都还有字符，则循环执行下列操作：
（1）如果j=-l或者t[i]=s[j]，则将i和j分别加1，继续比较t和s的下一个字符；
（2）否则，将j向右滑动到next[j]的位置，即j =next[j]。
3．如果s中所有字符均已比较完毕，则返回匹配的起始位置（从1开始）；否则返回-1．
其中，next数组根据子串s求解。求解next数组的代码已由get_next函数给出。
【C代码】
（1）常量和变量说明
t，s：长度为悯铂ls的字符串
next:next数组，长度为ls
（2）C程序
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```
/*求next[]的值*/
void get_next( int *next, char *s, int ls) {
int i=0，j=-1;
next[0]=-1;/*初始化next[0]*/
while(i < ls){/*还有字符*/
if(j==-1l ls[i]==s[j]){/*匹配*/
j++;
i++;
if( s[i]==s[j])
next[i] = next[j];
else
Next[i] = j;
}
else
j = next[j];
}
}
int kmp( int *next, char *t ,char *s, int lt, int ls )
{
Int i= 0,j =0 ;
while (i < lt && （1）){
if( j==-1 || （2）){
i ++ ;
j ++ ;
} else
（3）;
}
if (j >= ls)
return （4）;
else
return -1;
}
```

问题：4.1 （8分）

根据题干说明，填充C代码中的空（1）~（4）.问题：4.2 （2分）

根据题干说明和C代码，分析出kmp算法的时间复杂度为（5）（主串和子串的长度分别为lt和ls，用O符号表示）。问题：4.3 （5分）

根据C代码，字符串“BBABBCAC”的next数组元素值为（6）（直接写素值，之间用逗号隔开）。若主串为“AABBCBBABBCACCD”，子串为“BBABBCAC”，则函数Kmp的返回值是（7）。

第5题：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】

某发票（Invoice）由抬头（Head）部分、正文部分和脚注（Foot）部分构成。现采用装饰（Decorator）模式实现打印发票的功能，得到如图5-1所示的类图。

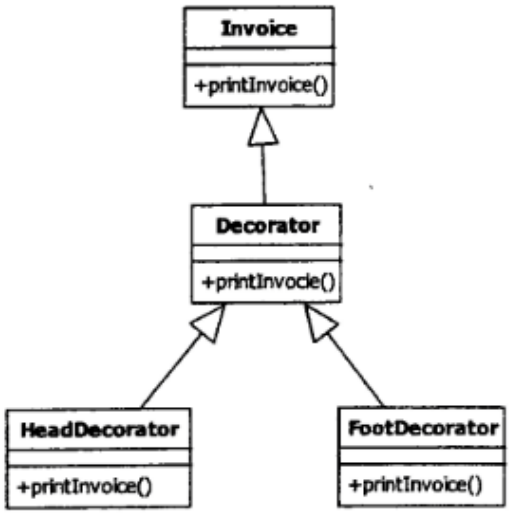


图 5-1 类图

问题：5.1 【C++代码】

```
#include <iostream>
using namespace std;
class Invoice{
public:
（1）{
cout<<"This is the content of the invoice!"<<endl;
}
};
class Decorator : public Invoice {
Invoice *ticket;
public:
Decorator(Invoice *t) { ticket = t; }
void printInvoice(){
if(ticket != NULL)
（2）;
}
};
class HeadDecorator : public Decorator{
public:
HeadDecorator(Invoice*t): Decorator(t) { }
void printInvoice() {
cout<< "This is the header of the invoice! "<< endl;
（3）;
}
};
class FootDecorator : public Decorator{
public:
FootDecorator(Invoice *t): Decorator(t) { }
void printInvoice(){
（4）;
cout<< "This is the footnote of the invoice!"<< endl;
}
};
```

```
int main(void) {
Invoice t;
FootDecorator f(&t);
HeadDecorator h(&f);
h.printInvoice();
cout<<" ------------------------" <<endl;
FootDecorator a(NULL) ;
HeadDecorator b( ( 5 ) );
b.printInvoice();
return 0;
}
```

程序的输出结果为：

```
This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
---------------------------
This is the header of the invoice!
This is the footnote of the invoice!
```

第6题：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】

某发票（Invoice）由抬头（Head）部分、正文部分和脚注（Foot）部分构成。现采用装饰（Decorator）模式实现打印发票的功能，得到如图6-1所示的类图。

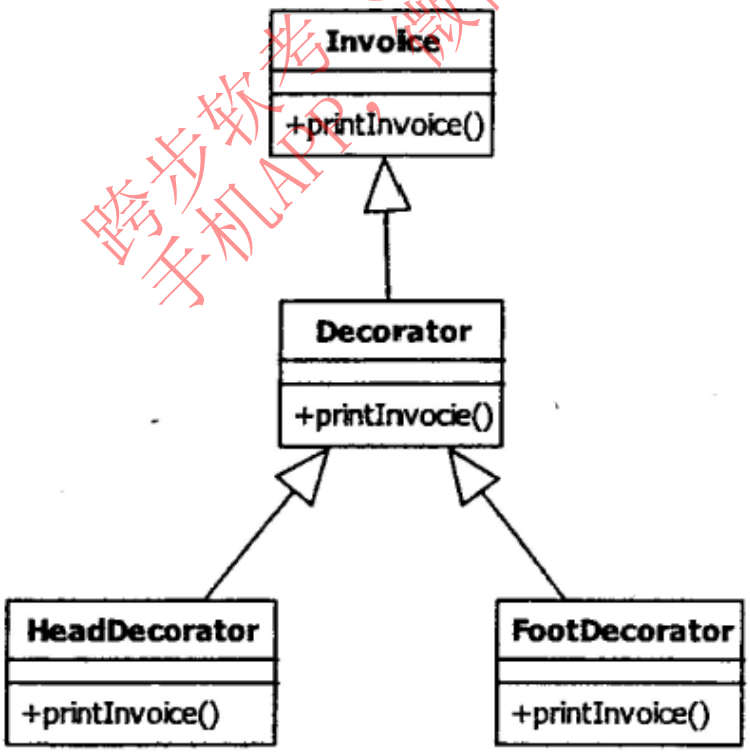


图 6-1　类图

问题：6.1　【java代码】

```
class invoice{
public void printInvoice(){
System.out.println ( "This is the content of the invoice!");
}
}
class Decorator extends Invoice {
protected Invoice ticket;
public Decorator(Invoice t){
ticket = t;
}
public void printInvoice(){
if(ticket != )
（1）；
}
}
class HeadDecorator extends Decorator{
public HeadDecorator(Invoice t){
super(t);
}
public void printInvoice (){
Systent.out.println( "This is the header of the invoice! ");
（2）；
}
}
class FootDecorator extends Decorator {
public FootDecorator(Invoice t){
super(t);
}
public void printInvoice(){
（ 3）；
Systent.out.println( "This is the footnote of the invoice! ");
}
}
Class test {
public static void main(String[] args){
Invoice t =new Invioce();
Invoice ticket;
ticket=（4）；
ticket.printInvoice();
Systent.out.println( “----------------- “);
ticket=（5）；
ticket.printInvoice();
}
}
```

程序的输出结果为：

```
This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
---------------------------
```

This is the header of the invoice!
This is the footnote of the invoice!

参考答案与解析

上午综合试卷答案与解析

第1题，参考答案：A

解析 ：

指令和数据是都存储在内存中，传统计算机CPU在执行过程中根据指令周期的不同阶段来区分是指令还是数据，取指周期取出的是指令，执行周期取出的是数据。

第2题，参考答案：C

解析 ：

PC（程序计数器）是用于存放下一条指令所在单元的地址。当执行一条指令时，处理器首先需要从PC中取出指令在内存中的地址，通过地址总线寻址获取。

第3题，参考答案：B

解析 ：

如果浮点数的阶码(包括1位阶符)用R位的移码表示，尾数(包括1位数符)用M位的补码表示，则浮点数表示的数值范围如下。

最大正数：$+\left(1-2^{-M+1}\right)\times 2^{(2^{R-1}-1)}$，最小负数$-1\times 2^{(2^{R-1}-1)}$

第4题，参考答案：C

解析 ：

海明码的构造方法是：在数据位之间插入k个校验位，通过扩大码距来实现检错和纠错。设数据

位是n位，校验位是k位，则n和k的必须满足以下的关系。
$2^K-1\geq n+k$
数据为16位时，至少需要5位校验位。
$2^5-1\geq 16+5$

第5题，参考答案：D

解析 ：

第一条指令执行时间+(指令数-1)*各指令段执行时间中最大的执行时间。
$4\triangle t + 3\triangle t + 2\triangle t + (100-1) X 4\triangle t = 405\triangle t$

第6题，参考答案：D

解析 ：

在程序的执行过程中，Cache与主存的地址映射是由硬件自动完成的。

第7题，参考答案：A

解析 ：

IDEA算法和RC4算法都对称加密算法，只能用来进行数据加密。MD5算法是消息摘要算法，只能用来生成消息摘要无法进行数字签名。
RSA算法是典型的非对称加密算法，主要具有数字签名和验签的功能。

第8题，参考答案：D

解析 ：

数字签名是信息的发送者才能产生的别人无法伪造的一段数字串，这段数字串同时也是对信息的发送者发送信息真实性的一个有效证明。不能验证接收者的合法性。

第9题，参考答案：C

第10题，参考答案：B

解析 ：

根据《中华人民共和国商标法》第三十八条：注册商标有效期满，需要继续使用的，应当在期满前六个月内申请续展注册。专利权和著作权到期后都无法延长，而商业秘密权无期限限制。

第11题，参考答案：D

解析 ：

专利审查指南的规定：
在审查过程中，对于不同的申请人同日（指申请日，有优先权的指优先权日）就同样的发明创

造分别提出专利申请，并且这两件申请符合授予专利权的其他条件的，应当根据专利法实施细则第四十一条第一款的规定，通知申请人自行协商确定申请人。

第12题，参考答案：B

解析 ：

按照商标法的规定，第29条，以及实施条例19条规定，同一天申请的，初步审定并公告使用在先的。驳回其他人的申请。均未使用获无法证明的，各自协商，不愿协商或者协商不成的，抽签决定，不抽签的，视为放弃。

第13题，参考答案：A

第14题，参考答案：C

第15题，参考答案：D

第16题，参考答案：C

解析 ：

极限编程（xp）：由价值观、原则、实践和行为四个部分组成。
水晶法：每一个不同的项目都需要一套不同的策略、约定和方法论。
并列争球法：使用了迭代的方法，其中，把每段时间（30天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。

第17题，参考答案：D

第18题，参考答案：A

第19题，参考答案：D

第20题，参考答案：B

解析 ：

在进行逻辑与“&&”运算时，只有当两个操作数的值为真，最后的结果才会为真。因此一旦x的值为假，整个运算表达式的值则为假。

第21题，参考答案：C

解析 ：

传值调用最显著的特征就是被调用的函数内部对形参的修改不影响实参的值。引用调用是将实参的地址传递给形参，使得形参的地址就是实参的地址。

第22题，参考答案：B

第23题，参考答案：D

解析 ：

实时操作系统是保证在一定时间限制内完成特定功能的操作系统。实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成操作，这是在操作系统设计时保证的；软实时则只要按照任务的优先级，尽可能快地完成操作即可。

第24题，参考答案：C

解析 ：

题目告诉我们一共有3个设备，分别是一个CPU、一台输入设备和一台输出设备，其实输入设备对应程序段输入Ii，而CPU对应程序段计算Ci，输出设备对应程序段输出Pi。而每个作业都分为这三段，各段间有个顺序关系。再结合图中已经给出的结点，我们不难发现，第一行是输入，第二行是计算，而第三行的结点数输出结点。因此可以知道①、②分别为C1、P3，③、④、⑤分别为I3、C4、P4。

第25题，参考答案：D

解析 ：

略题目告诉我们一共有3个设备，分别是一个CPU、一台输入设备和一台输出设备，其实输入设备对应程序段输入Ii，而CPU对应程序段计算Ci，输出设备对应程序段输出Pi。而每个作业都分为这三段，各段间有个顺序关系。再结合图中已经给出的结点，我们不难发现，第一行是输入，第二行是计算，而第三行的结点数输出结点。因此可以知道①、②分别为C1、P3，③、④、⑤分别为I3、C4、P4。

第26题，参考答案：B

解析 ：

页内地址为13位，页号地址为11位，段号地址为8位。根据公式 ，可以分别计算段号，页号以及页内地址最大的寻址空间。存储管理系统中的地址长度均表示为最大的寻址空间。

第27题，参考答案：B

解析 ：

当有进程运行时，其他进程访问信号量，信号量就会减1。S=-1-2。

第28题，参考答案：D

解析 ：

磁盘的容量为300GB，物理块的大小为1MB，则磁盘共300×1024/1个物理块，位示图的大小为300×1024/（32）=9600个字。

第29题，参考答案：A

解析 ：

对于较大型软件系统的需求往往难以在前期确定，所以瀑布模型最不适合。

第30题，参考答案：C

解析 ：

对于较大型软件系统的需求往往难以在前期确定，所以瀑布模型最不适合。

第31题，参考答案：D

解析 ：

可移植性包含：适应性、易安装性、共存性和易替换性四个特性。

第32题，参考答案：C

第33题，参考答案：D

解析 ：

环形复杂度V(G)=E-N+2，其中，E是流图中边的条数，N是结点数。
V(G)=E-N+2=10-8+2=4。

第34题，参考答案：A

第35题，参考答案：B

解析 ：

软件测试的目的在于希望以最少的人力和时间发现潜在的各种错误和缺陷。

第36题，参考答案：C

解析 ：

如果一个模块的所有成分都操作同一数据集或生成同一数据集，则称为通信内聚。
内聚有一下几种：
功能内聚：完成一个单一功能，各个部分协同工作，缺一不可。
顺序内聚：处理元素相关，而且必须顺序执行。
通信内聚：所有处理元素集中在一个数据结构的区域上。
过程内聚：处理元素相关，而且必须按特定的次序执行。
瞬时内聚：所包含的任务必须在同一时间间隔内执行（如初始化模块）。
逻辑内聚：完成逻辑上相关的一组任务。
偶然内聚：完成一组没有关系或松散关系的任务。

第37题，参考答案：D

解析 ：

本题考察面向对象多态的概念。
多态实质上是将子类的指针对象或者引用对象传递给父类指针对象后，通过这个父类指针对象调用的函数（此函数在父类中声明为虚函数，且在各个子类中重写这个函数），不是父类中定义的，而是传递进来的子类对象中重写的函数。

第38题，参考答案：D

解析 ：

动态绑定是实现多态的基础。

第39题，参考答案：C

解析 ：

面向对象分析的任务是了解问题域所涉及的对象、对象间的关系和操作，然后构造问题的对象模型。

第40题，参考答案：C

解析 ：

当e2发生时，如果当前状态是B2，则会离开B；如果当前状态不是B2，则不会离开。

第41题，参考答案：C

解析 ：

转换的五要素：
源状态：即受转换影响的状态
目标状态：当转换完成后对象的状态
触发事件：用来为转换定义一个事件，包括调用、改变、信号、时间四类事件
监护条件：布尔表达式，决定是否激活转换、
动作：转换激活时的操作

第42题，参考答案：C

解析 ：

类图描述的是类与类之间的关系
对象图描述的是某个具体的对象。
本图描述的是类与类之间的关系。

第43题，参考答案：C

第44题，参考答案：D

解析 ：

策略模式定义了一系列的算法，并将每一个算法封装起来，而且使它们还可以相互替换。策略模式让算法独立于使用它的客户而独立变化。
应用场景：
1、 多个类只区别在表现行为不同，可以使用Strategy模式，在运行时动态选择具体要执行的行为。
2、 需要在不同情况下使用不同的策略(算法)，或者策略还可能在未来用其它方式来实现。
3、 对客户隐藏具体策略(算法)的实现细节，彼此完全独立。

第45题，参考答案：C

解析 ：

策略模式定义了一系列的算法，并将每一个算法封装起来，而且使它们还可以相互替换。策略模式让算法独立于使用它的客户而独立变化。
应用场景：
1、 多个类只区别在表现行为不同，可以使用Strategy模式，在运行时动态选择具体要执行的行为。
2、 需要在不同情况下使用不同的策略(算法)，或者策略还可能在未来用其它方式来实现。
3、 对客户隐藏具体策略(算法)的实现细节，彼此完全独立。

第46题，参考答案：A

解析 ：

生成器模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。
实用范围
1 当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时。
2 当构造过程必须允许被构造的对象有不同表示时。

第47题，参考答案：A

解析 ：

生成器模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。
实用范围
1 当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时。
2 当构造过程必须允许被构造的对象有不同表示时。

第48题，参考答案：A

解析 ：

规式(a | b)*表示字符a和b组成的任何长度的字符串(a和b的位置任意)。a* | b*表示由若干个a组成的字符串，或者是由若干个b组成的任何长度的字符串。a*b*萨表示由若干个a后跟若干个b所组成的任何长度的字符串(a在b前面)。(ab)*表示每个ab所组成的任何长度的字符串(ab不能分

离)。(a*b*)*表示由字符a和b组成的任何长度的字符串(若干个a后面跟若干个b，b后面再跟若干个a)。只有(a*b*)*与(a｜b)*含义相同，因此正规式(a｜b)*与(a*b*)*是等价的。

第49题，参考答案：B

解析 ：

上下文无关文法：形式语言理论中一种重要的变换文法，用来描述上下文无关语言，在乔姆斯基分层中称为2型文法。由于程序设计语言的语法基本上都是上下文无关文法，因此应用十分广泛。

第50题，参考答案：D

解析 ：

在本题中，for语句后有“;”号，说明该循环语句的语句体为空，此时，循环会是一个死循环，所以存在语义错误。

第51题，参考答案：B

解析 ：

DBMS是数据库管理系统，主要用来保证数据库的安全性和完整性。而DBA通过授权功能为不同用户授权，主要的目的是为了保证数据的安全性。

第52题，参考答案：A

解析 ：

本题中U1={A1、A2、A3、A4}，构造出依赖关系图之后，A1是入度为0的结点，且从A1出发能遍历全图，因此A1为主键。
A1→A2，A2→A4利用传递率：A1→A4，因此A1→A4是冗余。

第53题，参考答案：C

解析 ：

本题中U1={A1、A2、A3、A4}，构造出依赖关系图之后，A1是入度为0的结点，且从A1出发能遍历全图，因此A1为主键。
A1→A2，A2→A4利用传递率：A1→A4，因此A1→A4是冗余。

第54题，参考答案：C

解析 ：

关系R（A,B,C,D）和S（A,C,E,F）做自然连接时，会以两个关系公共字段做等值连接，然后将操作结果集中重复列去除，所以运算后属性列有6个。

第55题，参考答案：B

解析 ：

关系R（A,B,C,D）和S（A,C,E,F）做自然连接时，会以两个关系公共字段做等值连接，然后将操作结果集中重复列去除，所以运算后属性列有6个。

第56题，参考答案：D

第57题，参考答案：A

解析 ：

拓扑排序通俗一点来讲，其实就是依次遍历没有前驱结点的结点。而某一时刻没有前驱结点的结点有可能存在多个，所以一个图的拓扑排序可能有多个。
4号结点没有前戏，所以拓扑排序的第一个元素是4。当4访问完了就可以访问1，1号访问完了就可以访问2，2号访问完了就可以访问3或5。所以拓扑排序结果为：412(35)。

第58题，参考答案：B

解析 ：

若用顺序表存储，则最好情况是删除最后一个元素，此时不用移动任何元素，直接删除，最差的情况是删除第一个元素，此时需要移动n-1个元素，所以平均状态是移动(n-1)/2。
若用链表存储，直接将需要删除元素的前趋next指针指向后继元素即可，不需要移动元素，所以移动元素个数为0。

第59题，参考答案：A

解析 ：

若用顺序表存储，则最好情况是删除最后一个元素，此时不用移动任何元素，直接删除，最差的情况是删除第一个元素，此时需要移动n-1个元素，所以平均状态是移动(n-1)/2。
若用链表存储，直接将需要删除元素的前趋next指针指向后继元素即可，不需要移动元素，所以移动元素个数为0。

第60题，参考答案：C

第61题，参考答案：D

第62题，参考答案：A

第63题，参考答案：C

解析 ：

110001001101 中：f(1100) a(0) c(100) e(1101)。

第64题，参考答案：C

解析 ：

四个矩阵分别为：
2*6 6*3 3*10 10*3
先计算：M1*M2 及M3*M4，计算次数分别为：
2*6*3=36，3*10*3=90。
然后结果相乘，计算次数为：
2*3*3=18。
36+90+18=144。

第65题，参考答案：B

解析 ：

四个矩阵分别为：
2*6 6*3 3*10 10*3
先计算：M1*M2 及M3*M4，计算次数分别为：
2*6*3=36，3*10*3=90。
然后结果相乘，计算次数为：
2*3*3=18。
36+90+18=144。

第66题，参考答案：A

解析 ：

ARP和ICMP是网络层协议，X.25是数据链路层协议，只有SNMP是应用层协议。
SNMP协议的报文是封装在UDP协议中传送。

第67题，参考答案：C

解析 ：

ARP和ICMP是网络层协议，X.25是数据链路层协议，只有SNMP是应用层协议。
SNMP协议的报文是封装在UDP协议中传送。

第68题，参考答案：A

第69题，参考答案：C

解析 ：

对于多种不同的路由协议到一个目的地的路由信息，路由器首先根据管理距离决定相信哪一个协议。

第70题，参考答案：D

解析 ：

地址220.112.179.92中179的二制码为1011 0011，假如网络号采用22位，与该地址匹配的路由表项则为220.112.177.64/22。

第71题，参考答案：C

第72题，参考答案：A

第73题，参考答案：B

第74题，参考答案：D

第75题，参考答案：C

下午案例分析答案与解析

第1题：跨步软考[www.kuabu.xyz]答案解析：
E1：客户服务助理，E2：客户，E3：经纪人。

本题要求识别E1-E3具体为哪个外部实体，通读试题说明，可以了解到适合充当外部实体的包括：客户、客户服务助理、经记人。具体的对应关系，可以通过将顶层图与题目说明进行匹配得知。如：从图中可看出E1会向交易平台发出数据流开户信息;；而从试题说明根据客户服务助理提交的开户信息，进行开户，并将客户信息存入客户记录中，账户信息存入账户记录中可以看出，E1对应是客户服务助理。E2、E3同理可得。

跨步软考[www.kuabu.xyz]答案解析：
D1：客户记录，D2：账户记录，D3：交易记录。

本题要求识别存储，解决这类问题，以图的分析为主，配合说明给存储命名，因为存储相关的数据流一般展现了这个存储中到底存了些什么信息，如从图中可以看到D1中有客户信息，而D2中有账户信息，题目说明中又有根据客户服务助理提交的开户信息，进行开户，并将客户信息存入客户记录中，账户信息存入账户记录中。自然D1应为客户记录，D2应为账户记录。同理，D3为交易记录。

跨步软考[www.kuabu.xyz]答案解析：
数据流名称：修改账户余额，起点：存款，终点：D2。
数据流名称：修改账户余额，起点：取款，终点：D2。

数据流名称：交易信息存入交易记录，起点：证券交易，终点：D3。

缺失数据流1
名称：修改账户余额，起点：存款，终点：D2。
理由：从试题说明客户可以向其账户中存款，根据存款金额修改账户余额可以看出，这个功能有操作根据存款金额修改账户余额。据此可以了解到从该功能应有数据流存款至D2，而0层图没有。
缺失数据流2:
名称：修改账户余额，起点：取款，终点：D2。
理由：从试题说明客户可以从其账户中取款，根据取款金额修改账户余额可以看出，这个功能有操作根据取款金额修改账户余额。据此可以了解到从该功能应有数据流取款至D2，而0层图没有。
缺失数据流3
名称：交易信息存入交易记录，起点：证券交易，终点：D3。
理由：从试题说明客户和经纪人均可以进行证券交易，将交易信息存入交易记录中可以看出，这个功能有操作将交易信息存入交易记录中。据此可以了解到从该功能应有数据流证券交易至D3，而0层图没有。

跨步软考[www.kuabu.xyz]答案解析：
增加外部实体证券交易中心，原来证券交易中的交易信息的数据流终点改为证券交易中心，数据流检测交易中的起点改为证券交易中心。

本题强调实际的证券交易通常是在证券交易中心完成，这个证券交易中心属于典型的外部实体，所以需要增加外部实体证券交易中心。由于该平台的证券交易功能需将交易信息传递给证券交易中心，因此将原来证券交易中的交易信息的数据流终点改为证券交易中心，数据流检测交易中的起点改为证券交易中心。

第2题：跨步软考[www.kuabu.xyz]答案解析：
1、经理与部门 之间 存在1:1的联系。
2、部门与员工 之间 存在1:n的联系。
3、客户与预订申请 之间 存在 1:n的联系。
4、业务员、客房、预订申请 之间存在1:m:n的联系。

跨步软考[www.kuabu.xyz]答案解析：
（a）部门号。
（b）客户号、单位名称
（c）申请号、客户号。
（d）身份证号、入住时间。
预订申请关系模式中的主键是申请号，外键是申请号、客户号。
安排关系模式中的主键是：（客房号、身份证号、入住时间），外键是：申请号、客房号、业务员。

跨步软考[www.kuabu.xyz]答案解析：
根据试题中的描述，客房信息中客房号是唯一标识客房关系的一个元组，即可以作为唯一的主键。在客房关系模式中，不存在其他部分依赖关系，但客房号->类型->收费标准，存在传递函数依赖，所以冗余，添加异常，修改异常，删除异常均存在。

第3题：跨步软考[www.kuabu.xyz]答案解析：
S1：空闲，S2：准备服务，S3：饮料出售，S4：可购买。

本题系统中的状态图，是对状态转换的图形化表达。从题目的说明部分可知，在状态转换过程中，涉及到的状态一共有四种：空闲、准备服务、可购买、饮料出售。从状态图涉及的转换可知S1~S4分别为：空闲、准备服务、饮料出售、可购买。关于状态转换的分析如下：
（1）清空硬币器后，自动售货机等待下一次交易，进入空闲状态。此时可任意的进行饮料选择数量，一旦顾客投入硬币，自动售货机便进入准备服务状态。
（2）当自动售货机进行准备服务状态时，开始计算硬币价值，如果硬币不够则提示顾客继续投入硬币。如果硬币足够，则进入可购买状态。
（3）进行可购买状态后，自动售货机判断饮料数量。如果数量不够，则返回准备服务状态提示用户重新选择饮料。如果数量足够，则推出饮料进入饮料出售状态。
（4）进行饮料出售状态后，自动售货机计算找零，并返回进入空闲状态等待下一次交易。

跨步软考[www.kuabu.xyz]答案解析：
E1：饮料数量不足，E2：硬币数量足够，E3：推出饮料，E4：返回找零。

跨步软考[www.kuabu.xyz]答案解析：
C1：自动售货机，C2：硬币器，C3：饮料储存仓，C4：硬币，C5：饮料。

第4题：跨步软考[www.kuabu.xyz]答案解析：
（1）：j<ls
（2）：t[i]==s[j];
（3）：get_next(next, s, ls);
　　j=next[j]
（4）：i+1-ls

跨步软考[www.kuabu.xyz]答案解析：
(5)O(ls+lt)

跨步软考[www.kuabu.xyz]答案解析：
（6）[-1,-1,1,-1,-1,2,0,0]
（7）6

第5题：跨步软考[www.kuabu.xyz]答案解析：
（1）virtual void printInvoice()
（2）ticket->&gt::printvoice();
（3）Decorator::printInvoice()
（4）Decorator::printInvoice()
（5）&a

第6题：跨步软考[www.kuabu.xyz]答案解析：
（1）ticket.printInvoice()
（2）super.printInvoice()
（3）super.printInvoice()
（4）new HeadDecorator(new FootDecorator(t))
（5）new HeadDecorator(new FootDecorator(NULL))