

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

2017 年 5 月第 1 题

CPU 执行算术运算或者逻辑运算时，常将源操作数和结果暂存在(1)中。

- (1) A. 程序计数器 (PC) B. 累加器 (AC) C. 指令寄存器 (IR) D. 地址寄存器 (AR)

【答案】B

【解析】

程序计数器 (PC) 是用于存放下一条指令所在单元的地址的地方。

累加器 (AC) 全称累加寄存器，是一个通用寄存器。其功能是，当运算器的算术逻辑单元 (ALU) 执行算术或逻辑运算时，为 ALU 提供一工作区。累加寄存器暂时存放 ALU 运算的结果信息。

指令寄存器 (IR) 用于存放当前从主存储器读出的正在执行的一条指令。

地址寄存器 (AR) 用来保存当前 CPU 所访问的内存单元的地址。由于在内存和 CPU 之间存在着操作速度上的差别，所以必须使用地址寄存器来保持地址信息，直到内存的读/写操作完成为止。

2017 年 5 月第 2 题

要判断字长为 16 位的整数 a 的低四位是否全为 0，则(2)。

- (2) A. 将 a 与 0x000F 进行“逻辑与”运算，然后判断运算结果是否等于 0
B. 将 a 与 0x000F 进行“逻辑或”运算，然后判断运算结果是否等于 F
C. 将 a 与 0x000F 进行“逻辑异或”运算，然后判断运算结果是否等于 0
D. 将 a 与 0x000F 进行“逻辑与”运算，然后判断运算结果是否等于 F

【答案】A

【解析】判断是否为 1，让 1 和它做与运算，为 1 时为 1，不为 1 时为 0

2017 年 5 月第 3 题

计算机系统中常用的输入/输出控制方式有无条件传送、中断、程序查询和 DMA 方式等。

当采用(3)方式时，不需要 CPU 执行程序指令来传送数据。

- (3) A. 中断 B. 程序查询 C. 无条件传送 D. DMA

【答案】D

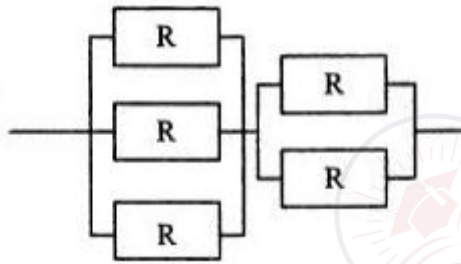
【解析】

DMA (直接存储器访问方式)，DMA 将数据从一个地址空间复制到另外一个地址空间时，

CPU 初始化这个传输动作，传输动作本身是由 DMA 控制器来执行和完成。

2017 年 5 月第 4 题

某系统由下图所示的冗余部件构成。若每个部件的千小时可靠度都为 R ，则该系统的千小时可靠度为 (4)。



- (4) A. $(1-R)^3 (1-R)^2$ B. $(1-(1-R)^3) (1-(1-R)^2)$
C. $(1-R)^3 + (1-R)^2$ D. $(1-(1-R)^3) + (1-(1-R)^2)$

【答案】B

【解析】

串行系统的可靠性 $R = R_1 \times R_2 \times \dots \times R_n$

并联系统的可靠性为 $R = 1 - (1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)$

2017 年 5 月第 5 题

已知数据信息为 16 位，最少应附加 (5) 位校验位，才能实现海明码纠错。

- (5) A. 3 B. 4 C. 5 D. 6

【答案】C

【解析】

$2^k - 1 \geq m + k$ (m 为信息位, k 为校验位) 当 $k=5$ 时, $31 \geq 21$ 等式成立。

2017 年 5 月第 6 题

以下关于 Cache (高速缓冲存储器) 的叙述中, 不正确的是 (6)。

- (6) A. Cache 的设置扩大了主存的容量
B. Cache 的内容是主存部分内容的拷贝
C. Cache 的命中率并不随其容量增大线性地提高
D. Cache 位于主存与 CPU 之间

【答案】A

【解析】

Cache 中储存的内容是主存部分内容的复本，是按程序的局部性原理选取出来的最常使用或不久将来仍将使用的内容。

2017 年 5 月第 7 题

HTTPS 使用 (7) 协议对报文进行封装

- (7) A. SSH B. SSL C. SHA-1 D. SET

【答案】B

【解析】

HTTPS (Hyper Text Transfer Protocol over Secure Socket Layer)，是以安全为目标的 HTTP 通道，是 HTTP 的安全版。HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议。

2017 年 5 月第 8 题

以下加密算法中适合对大量的明文消息进行加密传输的是 (8)

- (8) A. RSA B. SHA-1 C. MD5 D. RC5

【答案】D

【解析】对大量数据加密时一般都是使用快速的对称加密方法：如 RC

2017 年 5 月第 9 题

假定用户 A、B 分别在 I1 和 I2 两个 CA 处取得了各自的证书，下面 (9) 是 A、B 互信的必要条件。

- A. A、B 互换私钥 B. A、B 互换公钥 C. I1、I2 互换私钥 D. I1、I2 互换公钥

【答案】D

【解析】

如果用户数量很多，仅一个 CA 负责为所有用户签署证书可能不现实。通常应有多个 CA，每个 CA 为一部分用户发行和签署证书。

设用户 A 已从证书发放机构 X_1 处获取了证书，用户 B 已从 X_2 处获取了证书，如果 A 不如 X_2 的公钥，他虽然能读取 B 的证书，但却无法验证用户 B 证书中 X_2 的签名，因此 B 的证

书对 A 来说是没有用处的。然而，如果两个证书发放机构 X_1 和 X_2 彼此间已经安全地交换了公开密钥，则 A 可通过以下过程获取 B 的公开密钥：

(1) A 从目录中获取由 X_1 签署的 X_2 证书 $X_1 \langle X_2 \rangle$ ，因为 A 知道 X_1 的公开密钥，所以能验证 X_2 的证书，并从中得到 X_2 的公开密钥

(2) A 再从目录中获取由 X_2 签署的 B 的证书 $X_2 \langle B \rangle$ ，并由 X_2 的公开密钥对此加以验证，然后从中得到 B 的公开密钥。

2017 年 5 月第 10 题

甲软件公司受乙企业委托安排公司软件设计师开发了信息系统管理软件，由于在委托开发合同中未对软件著作权归属作出明确的约定，所以该信息系统管理软件的著作权由(10)享有。

- (10) A. 甲 B. 乙 C. 甲与乙共同 D. 软件设计师

【答案】A

【解析】

委托开发：如果是接受他人委托进行开发的软件，其著作权的归属应由委托人与受托人签订书面合同约定；如果没有签订合同，或合同中未规定的，则其著作权由受托人享有。

由国家机关下达任务开发的软件，著作权的归属由项目任务书或合同规定，若未明确规定，其著作权应归任务接受方所有。

2017 年 5 月第 11 题

根据我国商标法，下列商品中必须使用注册商标的是(11)。

- (11) A. 医疗仪器 B. 墙壁涂料 C. 无糖食品 D. 烟草制品

【答案】D

【解析】

根据我国法律规定：

1. 卷烟、雪茄烟和有包装的烟丝必须申请商标注册，未经核准注册的，不得生产、销售。
2. 除中药材和中药饮片以为的其他药品，都必须注册商标。

2017 年 5 月第 12 题

甲、乙两人在同一天就同样的发明创造提交了专利申请，专利局将分别向各申请人通报

有关情况，并提出多种可能采用的解决办法。下列说法中，不可能采用(12)。

- (12) A. 甲、乙作为共同申请人
B. 甲或乙一方放弃权利并从另一方得到适当的补偿
C. 甲、乙都不授予专利权
D. 甲、乙都授予专利权

【答案】D

【解析】同样的发明创造只能被授予一项专利的规定。在同一天两个不同的人就同样的发明创造申请专利的，专利局将分别向各申请人通报有关情况，请他们自己去协商解决这一问题，解决的办法一般有两种：一种是，两申请人作为一件申请的共同申请人，另一种是其中一方放弃权利，并从另一方得到适当的补偿。都授予专利权是不存在的。

2017年5月第13题

数字语音的采样频率定义为 8kHz，这是因为(13)。

- (13) A. 语音信号定义的频率最高值为 4kHz
B. 语音信号定义的频率最高值为 8kHz
C. 数字语音传输线路的带宽只有 8kHz
D. 一般声卡的采样频率最高为每秒 8k 次

【答案】A

【解析】

采样频率大于等于工作频率的二倍，才能在以后恢复出实际波形，防止信息的丢失。

2017年5月第14题

使用图像扫描仪以 300DPI 的分辨率扫描一幅 3×4 英寸的图片，可以得到(14)像素的数字图像。

- (14) A. 300×300 B. 300×400 C. 900×4 D. 900×1200

【答案】D

【解析】

DPI 为像素/英寸 $(3 \times 300) \times (4 \times 300) = 900 \times 1200$

2017 年 5 月第 15、16 题

在采用结构化开发方法进行软件开发时，设计阶段接口设计主要依据需求分析阶段的(15)。接口设计的任务主要是(16)。

(15) A. 数据流图 B. E-R 图 C. 状态-迁移图 D. 加工规格说明

(16) A. 定义软件的主要结构元素及其之间的关系

B. 确定软件涉及的文件系统的结构及数据库的表结构

C. 描述软件与外部环境之间的交互关系，软件内模块之间的调用关系

D. 确定软件各个模块内部的算法和数据结构

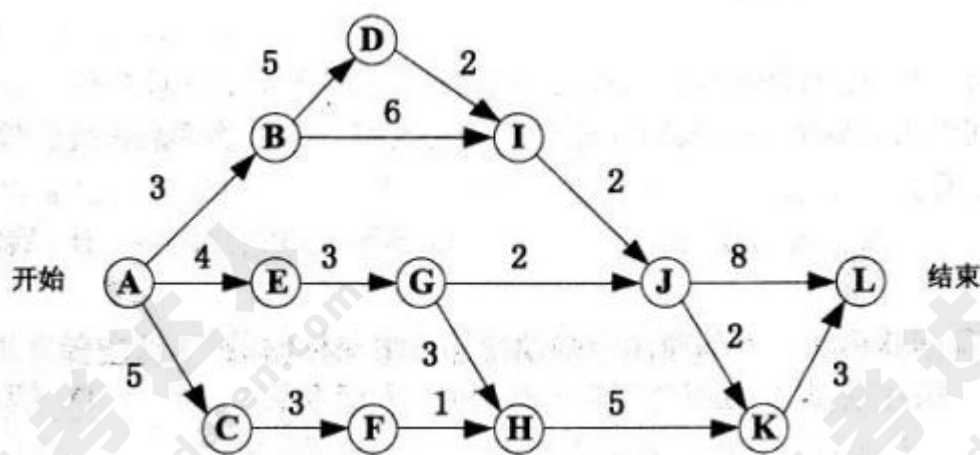
【答案】A C

【解析】

(16) 题 A 是架构定义任务，B 是数据存储设计任务，D 是详细设计任务

2017 年 5 月第 17、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，边上的数字表示活动的持续时间(天)，则完成该项目的最少时间为(17)天。活动 BD 和 HK 最早可以从第(18)天开始。(活动 AB、AE 和 AC 最早从第 1 天开始)



(17) A. 17

B. 18

C. 19

D. 20

(18) A. 3 和 10

B. 4 和 11

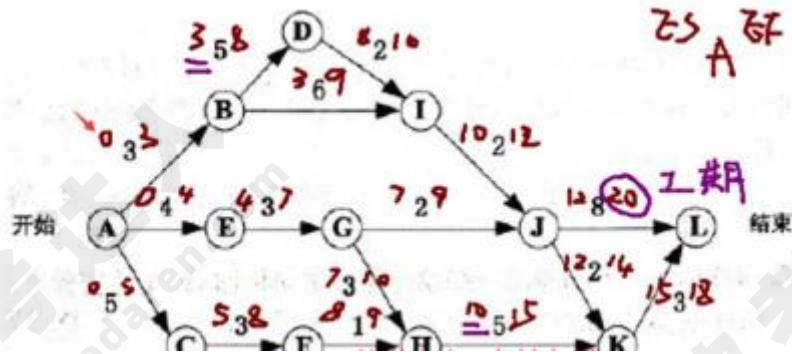
C. 3 和 9

D. 4 和 10

【答案】D B

【解析】

因为网络图是从 0 开始算的，按题目要求活动 AB 从第 1 天开始的话，就是 1、2、3，活动 BD 就是第 4 天开始，相应的活动 HK 就是第 11 天开始。



2017 年 5 月第 19 题

在进行软件开发时，采用无主程序员的开发小组，成员之间相互平等；而主程序员负责制的开发小组，由一个主程序员和若干成员组成，成员之间没有沟通。在一个由 8 名开发人员构成的小组中，无主程序员组和主程序员组的沟通路径分别是(19)。

- (19) A. 32 和 8 B. 32 和 7 C. 28 和 8 D. 28 和 7

【答案】D

【解析】

沟通渠道= $N(N-1)/2$ ，N 是指参与沟通者的人数。所以 $8*7/2=28$

成员之间没有沟通，只与主程序员沟通，所以沟通路径为 7。

2017 年 5 月第 20 题

在高级语言源程序中，常需要用户定义的标识符为程序中的对象命名，常见的命名对象有(20)。

①关键字（或保留字）②变量③函数④数据类型⑤注释

- (20) A. ①②③ B. ②③④ C. ①③⑤ D. ②④⑤

【答案】B

【解析】

在编程语言中，标识符是用户编程时使用的名字，对于变量、常量、函数、语句块也有名字，我们统统称之为标识符。关键字作为用户标识符。

2017 年 5 月第 21 题

在仅由字符 a、b 构成的所有字符串中，其中以 b 结尾的字符串集合可用正规式表示为(21)。

(21) A. $(b|ab)^*b$

B. $(ab^*)^*b$

C. a^*b^*b

D. $(a|b)^*b$

【答案】D

【解析】正规式 $(a|b)^*$ 对应的正则集为 $\{\epsilon, a, b, aa, ab, \dots\}$,所有由a和b组成的字符串}, 结尾为b

2017年5月第22题

在以阶段划分的编译过程中,判断程序语句的形式是否正确属于(22)阶段的工作。

(22) A. 词法分析

B. 语法分析

C. 语义分析

D. 代码生成

【答案】B

【解析】

词法分析阶段:输入源程序,对构成源程序的字符串进行扫描和分解,识别出一个个的单词,删掉无用信息,报告分析时的错误。

语法分析阶段:语法分析器以单词符号作为输入,分析单词符号串是否形成符合语法规则的语法单位,如表达式、赋值、循环等,按语法规则分析检查每条语句是否有正确的逻辑结构。

语义分析阶段主要是检查源程序是否存在语义错误,并收集类型信息供后面的代码生成阶段使用,如:赋值语句的右端和左端的类型不匹配。表达式的除数是否为零等。

2017年5月第23题

某文件管理系统在磁盘上建立了位示图(bitmap),记录磁盘的使用情况。若计算机系统字长为32位,磁盘的容量为300GB,物理块的大小为4MB,那么位示图的大小需要(23)个字。

(23) A. 1200

B. 2400

C. 6400

D. 9600

【答案】B

【解析】

$$300 \times 1024 / 4 / 32 = 2400$$

2017年5月第24题

某系统中有3个并发进程竞争资源R,每个进程都需要5个R,那么至少有(24)个R,才能保证系统不会发生死锁。

(24) A. 12

B. 13

C. 14

D. 15

【答案】B

【解析】

系统保证不发生死锁的资源数为：并发进程数 \times （进程需要资源数-1）+1。这样不管哪个进程得到最后这个资源都可以顺利执行完，释放他所占有的资源后，其它进程可以顺利执行。

2017 年 5 月第 25 题

某计算机系统页面大小为 4K，进程的页面变换表如下所示。若进程的逻辑地址为 2D16H。该地址经过变换后，其物理地址应为(25)。

| 页号 | 物理块号 |
|----|------|
| 0 | 1 |
| 1 | 3 |
| 2 | 4 |
| 3 | 6 |

(25) A. 2048H

B. 4096H

C. 4D16H

D. 6D16H

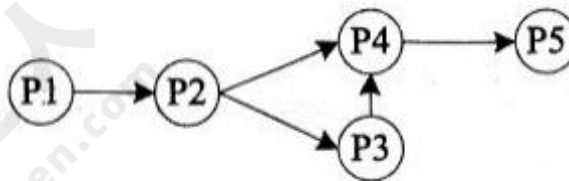
【答案】C

【解析】

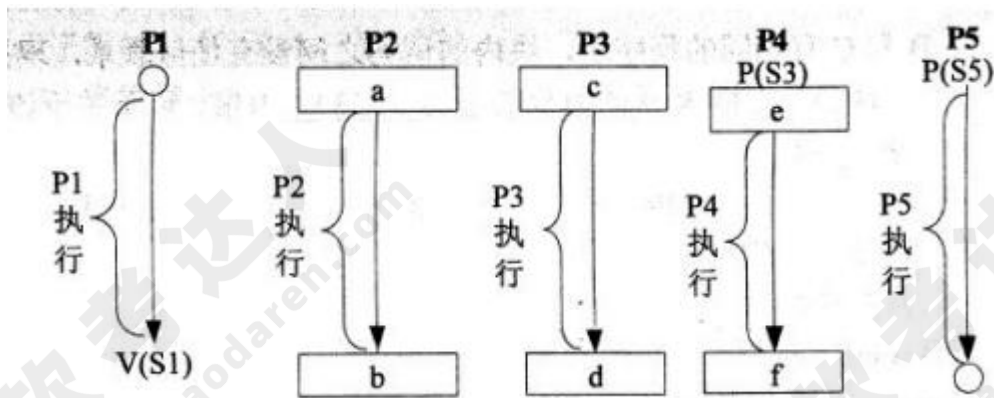
页面大小为 4K，说明需要 12 位（ $4K=4096=2^{12}$ ）来表示页内地址。2D16H 表示十六进制数 2D16，转成二进制为 0010 1101 0001 0110。从右向左划出 12 位表示页内地址，即 0010 1101 0001 0110。红色部分表示页内地址，剩余的蓝色部分表示页号，页号为 0010，转成十进制为 2，通过页表查询对应的物理块号为 4，4 的二进制为 0100，再连上页内地址，所以物理地址为：0100 1101 0001 0110，转成十六进制为：4D16H。

2017 年 5 月第 26、27、28 题

进程 P1、P2、P3、P4 和 P5 的前趋图如下所示：



若用 PV 操作控制进程 P1、P2、P3、P4 和 P5 并发执行的过程，需要设置 5 个信号量 S1、S2、S3、S4 和 S5，且信号量 S1~S5 的初值都等于零。如下的进程执行图中 a 和 b 处应分别填写(26)；c 和 d 处应分别填写(27)；e 和 f 处应分别填写(28)。



- (26) A. V(S1) 和 P(S2)V(S3) B. P(S1) 和 V(S2)V(S3)
 C. V(S1) 和 V(S2)V(S3) D. P(S1) 和 P(S2)V(S3)
- (27) A. P(S2) 和 P(S4) B. V(S2) 和 P(S4) C. P(S2) 和 V(S4) D. V(S2) 和 V(S4)
- (28) A. P(S4) 和 V(S5) B. V(S5) 和 P(S4) C. V(S4) 和 P(S5) D. V(S4) 和 V(S5)

【答案】B C A

【解析】箭头线开始位置为V，末端为P。P代表是请求资源，V代表释放资源。（教程103~105是详解，这里是我自己总结的）

2017年5月第29题

以下关于螺旋模型的叙述中，不正确的是(29)。

- (29) A. 它是风险驱动的，要求开发人员必须具有丰富的风险评估知识和经验
 B. 它可以降低过多测试或测试不足带来的风险
 C. 它包含维护周期，因此维护 and 开发之间没有本质区别
 D. 它不适用于大型软件开发

【答案】C

【解析】

2017年5月第30题

以下关于极限编程(XP)中结对编程的叙述中，不正确的是(30)。

- (30) A. 支持共同代码拥有和共同对系统负责 B. 承担了非正式的代码审查过程
 C. 代码质量更高 D. 编码速度更快

【答案】A

【解析】

支持共同代码拥有和共同对系统负责是指的极限编程(XP) 中代码集体所有权（代码共享）。

2017 年 5 月第 31 题

以下关于 C/S（客户机/服务器）体系结构的优点的叙述中，不正确的是（31）。

- (31) A. 允许合理地划分三层的功能，使之在逻辑上保持相对独立性
- B. 允许各层灵活地选用平台和软件
- C. 各层可以选择不同的开发语言进行并行开发
- D. 系统安装、修改和维护均只在服务器端进行

【答案】D

【解析】

系统安装、修改和维护均只在服务器端进行是 B/S 体系的优点。

2017 年 5 月第 32 题

在设计软件的模块结构时，（32）不能改进设计质量。

- (32) A. 尽量减少高扇出结构
- B. 模块的大小适中
- C. 将具有相似功能的模块合并
- D. 完善模块的功能

【答案】C

【解析】

2017 年 5 月第 33、34 题

模块 A、B 和 C 有相同的程序块，块内的语句之间没有任何联系，现把改程序块取出来，形成新的模块 D，则模块 D 的内聚类型为（33）内聚。以下关于该内聚类型的叙述中，不正确的是（34）。

- (33) A. 巧合
- B. 逻辑
- C. 时间
- D. 过程
- (34) A. 具有最低的内聚性
- B. 不易修改和维护
- C. 不易理解
- D. 不影响模块间的耦合关系

【答案】A D

【解析】

内聚：指模块内部各元素之间联系的紧密程度。模块的内聚类型分为 7 种，根据内聚度从高到低的排序。



- (37) A. 多重继承 B. 多态 C. 封装 D. 层次继承
- (38) A. 动态 B. 私有 C. 公共 D. 二义性

【答案】A D

【解析】

多重继承，指的是一个类可以同时继承多个父类的行为和特征功能。比如水上飞机继承了飞机和船的特征。多重继承指代可以导致某些令人混淆的情况，所以关于它的好处与风险之间孰轻孰重常常受人争论。

2017 年 5 月第 39 题

采用面向对象方法进行软件开发，在分析阶段，架构师主要关注系统的(39)。

- (39) A. 技术 B. 部署 C. 实现 D. 行为

【答案】D

【解析】

2017 年 5 月第 40 题

在面向对象方法中, 多态指的是(40)。

- (40) A. 客户类无需知道所调用方法的特定子类的实现
- B. 对象动态地修改类
- C. 一个对象对应多张数据库表
- D. 子类只能覆盖父类中非抽象的方法

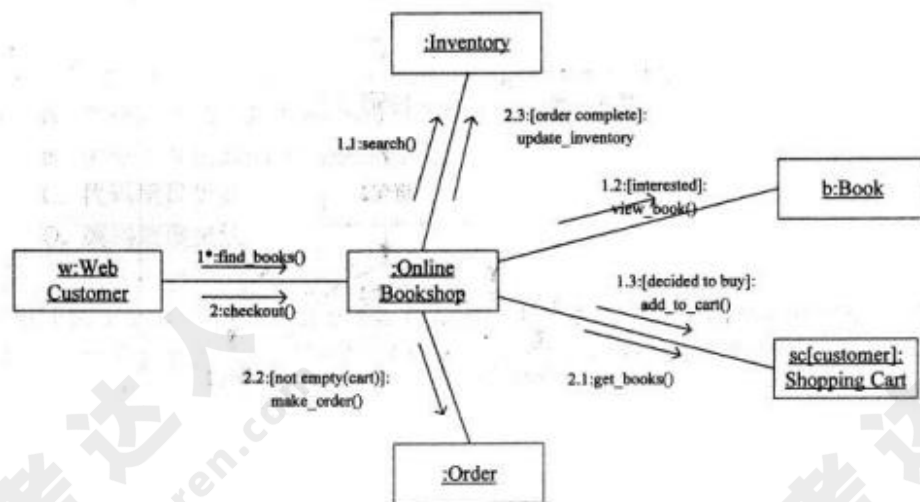
【答案】A

【解析】

多态是指同一操作作用于不同的对象, 可以有不同的解释, 产生不同的执行结果。比如: 上课铃响了, 上体育课的学生跑到操场上站好, 上语文课的学生在教室里坐好。在运行时, 可以通过指向基类的指针, 来调用实现派生类中的方法。也就是说客户类其实在调用方法时, 并不需要知道特定子类的实现, 都会用统一的方式来调用。

2017 年 5 月第 41、42、43 题

● 以下 UML 图是 (41), 图中 `:Order` 和 `b:Book` 表示 (42), `1*:find_books()` 和 `1.1:search()` 表示 (43)。



- (41) A. 序列图 B. 状态图 C. 通信图 D. 活动图
- (42) A. 类 B. 对象 C. 流名称 D. 消息
- (43) A. 类 B. 对象 C. 流名称 D. 消息

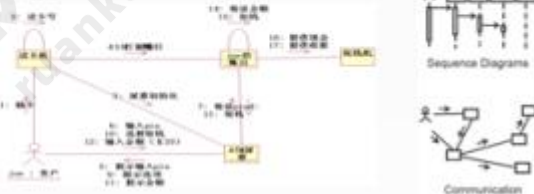
【答案】C B D

【解析】

从图示可以了解到，题目中的图是通信图。通信图描述的是对象和对象之间的关系，即一个类操作的实现。简而言之就是，对象和对象之间的调用关系，体现的是一种组织关系。该图明显表达的是对象与对象之间的关系。其中如果一个框中的名称中带有“：”，说明这表示的是一个对象，“：”号前的部分是对象名，“：”号后面的部分是类名。而对象之间连线上方的箭头所标识的对象之间通信的消息。

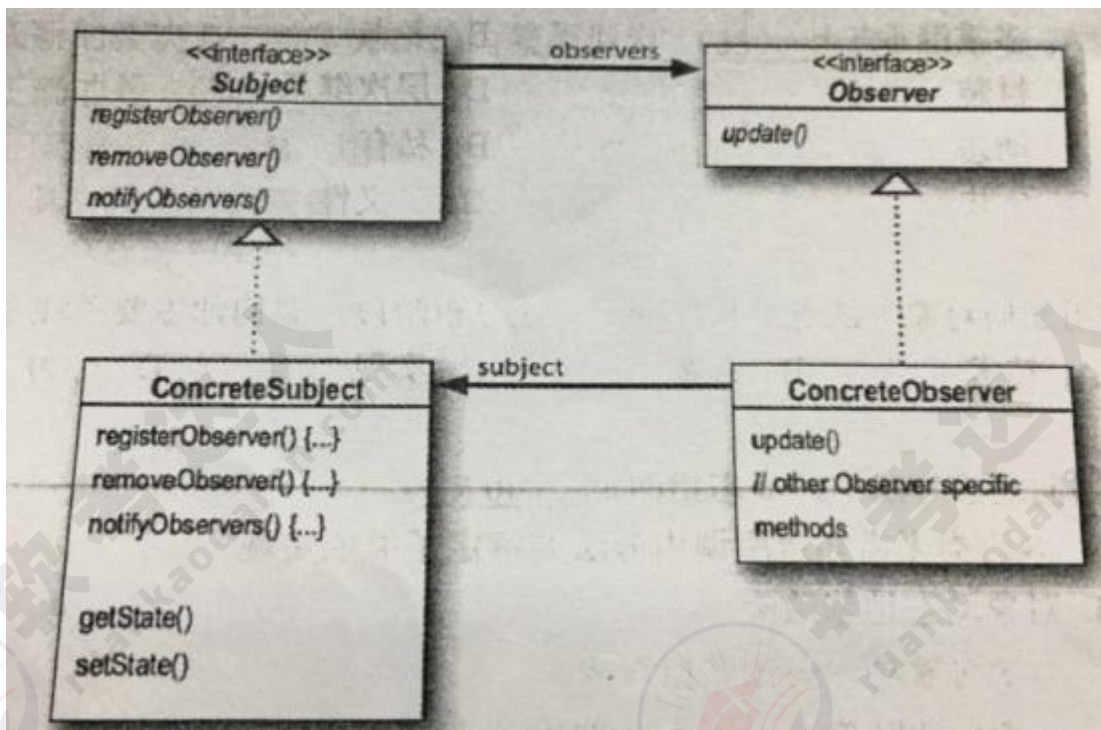
六、协作图（通信图，UML2.0后的名称）

是一种交互图，强调的是发送和接收消息的对象之间的组织结构。一个协作图显示了一系列的对象和在这些对象之间的联系以及对象间发送和接收的消息。



2017 年 5 月第 44、45 题

下图所示为观察者(Observer)模式的抽象示意图, 其中(44)知道其观察者, 可以有任意多个观察者观察同一个目标; 提供住处和删除观察者对象的接口。此模式体现的最主要的特征是(45)。



- (44) A. Subject B. Observer C. ConcreteSubject D. ConcreteObserver
- (45) A. 类应该对扩展开放，对修改关闭 B. 使所要交互的对象尽量松耦合
- C. 组合优先于继承使用 D. 仅与直接关联类交互

【答案】C A

【解析】

观察者将自己注册到事件，那么具体的事件就知道了自己的观察者。观察者和事件都有自己的抽象，当实现具体的观察者和事件的时候都要实现相应接口，所以对扩展是开放的。

2017 年 5 月第 46、47 题

装饰器（Decorator）模式用于(46);外观（Facade）模式用于(47)。

- ①将一个对象加以包装以给客户提供其希望的另外一个接口
- ②将一个对象加以包装以提供一些额外的行为
- ③将一个对象加以包装以控制对这个对象的访问
- ④将一系列对象加以包装以简化其接口

(46) A. ① B. ② C. ③ D. ④

(47) A. ① B. ② C. ③ D. ④

【答案】B D

【解析】

9.装饰模式（Decorator）

动态地给一个对象添加一些额外的职责。提供了用子类扩展功能的一个灵活的替代，但比生成子类更为灵活。

装饰模式可以在不需要创造更多子类的情况下，将对象的功能加以扩展。

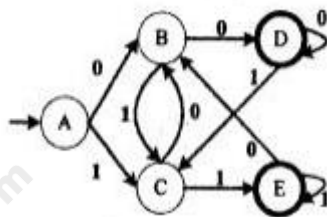
外观模式(Facade)

定义了一个高层接口，为子系统的一组接口提供一个一致的界面，从而简化子系统的使用。



2017 年 5 月第 48 题

某确定的有限自动机（DFA）的状态转换图如下图所示（A 是初态，D、E 是终态），则该 DFA 能识别(48)。



- (48) A. 00110 B. 10101 C. 11100 D. 11001

【答案】C

【解析】选项中，只用 C 中的字符串能被 DFA 解析。解析路径为 ACEEBDD。

2017 年 5 月第 49 题

函数 main()、f() 的定义如下所示，调用函数们 f() 时，第一个参数采用传值 (call by value) 方式，第二个参数采用传引用 (call by reference) 方式，main() 函数中 “print(x)” 执行后输出的值为(49)。

main()

```
int x = 5;
f(x+1, x);
print(x);
```

f(int x, int &a)

```
x = x*x - 1;
a = x + a;
return;
```

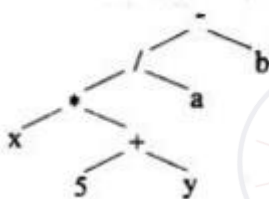
- (49) A. 11 B. 40 C. 45 D. 70

【答案】B

【解析】当值传递的时候，将原来的参数复制了一份，但是引用传递的时候是将变量的地址传来出去，会改变地址上的数据。因此，a 代表的其实就是 x 本身，f 函数里面的 x 是另一个变量，只用 a 的变化才能导致 main 函数里面的 x 值的变化

2017 年 5 月第 50 题

下图为一个表达式的语法树，该表达式的后缀形式为(50)。



- (50) A. $x \ 5 \ y \ + \ * \ a \ / \ b \ -$ B. $x \ 5 \ y \ a \ b \ * \ + \ / \ -$
C. $- \ / \ * \ x \ + \ 5 \ y \ a \ b$ D. $x \ 5 \ * \ y \ + \ a \ / \ b \ -$

【答案】A

【解析】要得到题目中的表达式语法树后缀式，只需要对树进行后序遍历即可。

2017 年 5 月第 51、52 题

若事务 T1 对数据 D1 加了共享锁，事务 T2、T3 分别对数据 D2、D3 加了排它锁，则事务 T1 对数据(51)；事务 T2 对数据(52)。

- (51)A. D2、D3 加排它锁都成功
B. D2、D3 加共享锁都成功
C. D2 加共享锁成功，D3 加排它锁失败
D. D2、D3 加排它锁和共享锁都失败

- (52)A. D1、D3 加共享锁都失败
B. D1、D3 加共享锁都成功
C. D1 加共享锁成功，D3 加排它锁失败
D. D1 加排它锁成功，D3 加共享锁失败

【答案】D C

【解析】

有共享锁可以再加共享锁，但不可以加排他锁。有排它锁，则共享锁和排它锁都不可以再加。

2017 年 5 月第 53 题

假设关系 $R\langle U, F \rangle$, $U = \{A1, A2, A3\}$, $F = \{A1A3 \rightarrow A2, A1A2 \rightarrow A3\}$ ，则关系 R 的各候选关键字中必定含有属性(53)。

- (53)A. A1 B. A2 C. A3 D. A2 A3

【答案】A

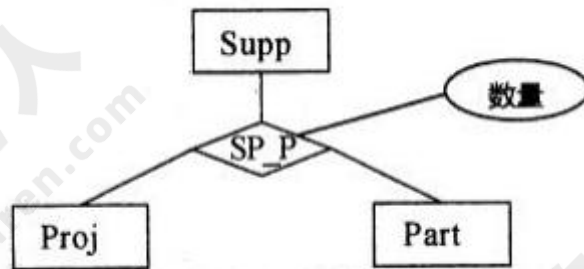
【解析】

解析：如果一个超关键字去掉其中任何一个字段后不再能唯一地确定记录，则称它为“候选关键字”(Candidate Key)。候选关键字既能唯一地确定记录，它包含的字段又是最精炼的。也就是说候选关键字是最简单的超关键字。

2017 年 5 月第 54、55、56 题

在某企业的工程项目管理系统的数据库中供应商关系 Supp、项目关系 Proj 和零件关

系 Part 的 E-R 模型和关系模式如下：



Supp（供应商号, 供应商名, 地址, 电话）

Proj（项目号, 项目名, 负责人, 电话）

Part（零件号, 零件名）

其中，每个供应商可以为多个项目供应多种零件，每个项目可由多个供应商供应多种零件。SP P 需要生成一个独立的关系模式，其联系类型为 (54)

给定关系模式 SP P（供应商号, 项目号, 零件号, 数量）查询至少供应了 3 个项目（包含 3 项）的供应商，输出其供应商号和供应零件数量的总和，并按供应商号降序排列。

SELECT 供应商号, SUM（数量） FROM (55)

GROUP BY 供应商号

(56)

ORDER BY 供应商号 DESC;

(54) A. *:*:~ B. 1:~:~ C. 1:1:~ D. 1:1:1

(55) A. Supp B. Proj C. Part D. SP P

(56) A. HAVING COUNT(项目号)>2 B. WHERE COUNT(项目号)>2

C. HAVING COUNT(DISTINCT(项目号))>2 D. WHERE COUNT(DISTINCT(项目号))>3

【答案】A D C

【解析】多个对多个，很显然是 A 选项。后两个空是 SQL 语言，大致浏览一遍就可以了，用排除法

2017 年 5 月第 57 题

以下关于字符串的叙述中，正确的是 (57)。

(57) A. 包含任意个空格字符的字符串称为空串

B. 字符串不是线性数据结构

C. 字符串的长度是指串中所含字符的个数

D. 字符串的长度是指串中所含非空格字符的个数

【答案】C

【解析】

A 为空格串。

C 字符串长度是指字符串所含的字符个数，但不包括最后的'\0'。

2017 年 5 月第 58 题

已知栈 S 初始为空，用 I 表示入栈、O 表示出栈，若入栈序列为 a1a2a3a4a5，则通过栈 S 得到出栈序列 a2a4a5a3a1 的合法操作序列(58)。

(58) A. IIOIIOIOO00 B. IOIOIOIOIO C. IOOIIOIOIO D. IIOOIOIOO00

【答案】A

【解析】

解析：操作序列：a1a2 入栈，a2 出栈，a3a4 入栈，a4 出栈，a5 入栈，a5 出栈，a3 出栈，a1 出栈

2017 年 5 月第 59 题

某二叉树的先序遍历序列为 ABCDEF，中序遍历序列为 BADCFE，则该二叉树的高度(即层数)为(59)。

(59) A. 3 B. 4 C. 5 D. 6

【答案】B

【解析】

根据先序遍历和中序遍历得此二叉树为：



2017 年 5 月第 60 题

对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$ ，当且仅当满足关系 $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ $\{i=1, 2, \dots, \lfloor n/2 \rfloor\}$ 时称其为小根堆(小顶堆)。以下序列中，(60)不是小根堆。

(60) A. 16, 25, 40, 55, 30, 50, 45

B. 16, 40, 25, 50, 45, 30, 55

C. 16, 25, 39, 41, 45, 43, 50

D. 16, 40, 25, 53, 39, 55, 45

【答案】D**【解析】**D 中第二个关键字小于第五个关键字，不满足小跟堆的条件。**2017 年 5 月第 61 题**

在 12 个互异元素构成的有序数组 $a[1..12]$ 中进行二分查找(即折半查找，向下取整)，若待查找的元素正好等于 $a[9]$ ，则在此过程中，依次与数组中的(61)比较后，查找成功结束。

(61) A. $a[6]$ 、 $a[7]$ 、 $a[8]$ 、 $a[9]$ B. $a[6]$ 、 $a[9]$ C. $a[6]$ 、 $a[7]$ 、 $a[9]$ D. $a[6]$ 、 $a[8]$ 、 $a[9]$ **【答案】B****【解析】****2、折半查找(二分查找)**

先给数据排序，形成有序表，把待查数据值与查找范围的中间元素值进行比较，会有四种情况出现：

- 1) 待查找数值与中间元素值相等，返回中间元素值的索引。
- 2) 待查找数值比中间元素值小，则以整个查找范围的前半部分作为新的查找范围，执行 1)，直到找到相等的值。
- 3) 待查找数值比中间元素值大，则以整个查找范围的后半部分作为新的查找范围，执行 1)，直到找到相等的值。
- 4) 如果最后找不到相等的值，则返回错误提示信息。

2017 年 5 月第 62、63、64、65 题

某汽车加工工厂有两条装配线 L_1 和 L_2 ，每条装配线的工位数均为 n (S_{ij} , $i=1$ 或 2 , $j=1, 2, \dots, n$)，两条装配线对应的工位完成同样的加工工作，但是所需要的时间可能不同 (a_{ij} , $i=1$ 或 2 , $j=1, 2, \dots, n$)。汽车底盘开始到进入两条装配线的时间 (e_1, e_2) 以及装配后到结束的时间 (X_1, X_2) 也可能不相同。从一个工位加工后流到下一个工位需要迁移时间 (t_{ij} , $i=1$ 或 2 , $j=2, \dots, n$)。现在要以最快的时间完成一辆汽车的装配，求最优的装配路线。

分析该问题，发现问题具有最优子结构。以 L_1 为例，除了第一个工位之外，经过第 j 个工位的最短时间包含了经过 L_1 的第 $j-1$ 个工位的最短时间或者经过 L_2 的第 $j-1$ 个工位的最短时间，如式(1)。装配后到结束的最短时间包含离开 L_1 的最短时间或者离开 L_2 的最短时间如式(2)。

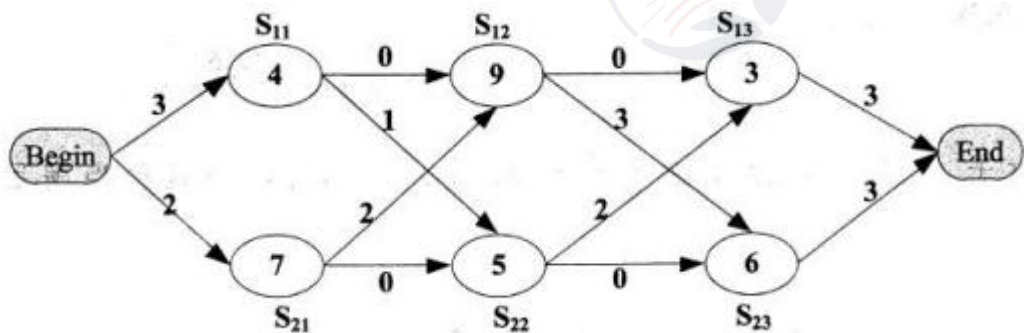
$$f_{1,j} = \begin{cases} e_1 + a_{1,j} & \text{若 } j=1 \\ \min(f_{1,j-1} + a_{1,j} + t_{1,j-1}, f_{2,j-1} + a_{1,j} + t_{2,j-1}) & \text{其他} \end{cases} \quad (1)$$

$$f_{\min} = \min(f_{1,n} + x_1, f_{2,n} + x_2) \quad (2)$$

由于在求解经过 L1 和 L2 的第 j 个工位的最短时间均包含了经过 L1 的第 j-1 个工位的最短时间或者经过 L2 的第 j-1 个工位的最短时间, 该问题具有重复子问题的性质, 故采用迭代方法求解。

该问题采用的算法设计策略是 (62), 算法的时间复杂度为 (63)

以下是一个装配调度实例, 其最短的装配时间为 (64), 装配路线为 (65)



(62) A. 分治 B. 动态规划 C. 贪心 D. 回溯

(63) A. $\Theta(\lg n)$ B. $\Theta(n)$ C. $\Theta(n^2)$ D. $\Theta(n \lg n)$

(64) A. 21 B. 23 C. 20 D. 26

(65) A. S11→S12→S13 B. S11→S22→S13
C. S21→S12→S23 D. S21→S22→S23

【答案】B B A B

【解析】求最优所以是动态规划, 不是分治法; 那么该算法的复杂度就是 B 选项 $O(n)$ 了; 后两个空比较最后一个选项中, 选出最短的。

2017 年 5 月第 66 题

在浏览器地址栏输入一个正确的网址后, 本地主机将首先在 (66) 查询该网址对应的 IP 地址。

(66) A. 本地 DNS 缓存 B. 本机 hosts 文件 C. 本地 DNS 服务器 D. 根域名服务器

【答案】B

【解析】

- 域名查询顺序：
- a. 浏览器缓存（本机 hosts 文件），浏览器会缓存 DNS 记录一段时间。
 - b. 系统缓存 c. 路由器缓存
 - d. 如果还是没有，那么就去检查 ISP 有没有吧
 - e. 递归搜索域名服务器

2017 年 5 月第 67 题

下面关于 Linux 目录的描述中，正确的是(67)。

- (67)A. Linux 只有一个根目录，用 “/root” 表示
- B. Linux 中有多个根目录，用 “/” 加相应目录名称表示
- C. Linux 中只有一个根目录，用 “/” 表示
- D. Linux 中有多个根目录，用相应目录名称表示

【答案】C

【解析】

2017 年 5 月第 68 题

以下关于 TCP/IP 协议栈中协议和层次的对应关系正确的是(68)。

(68)A.

| | |
|------|--------|
| TFTP | Telnet |
| UDP | TCP |
| ARP | |

B.

| | |
|-----|--------|
| RIP | Telnet |
| UDP | TCP |
| ARP | |

C.

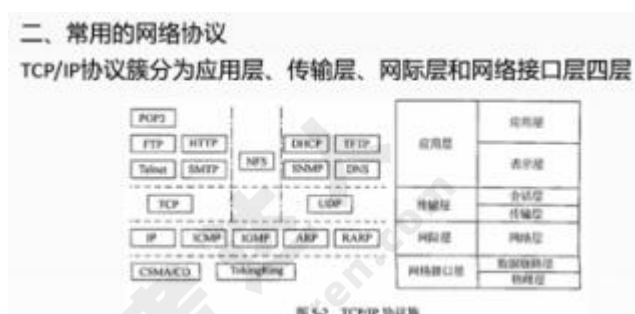
| | |
|------|------|
| HTTP | SNMP |
| TCP | UDP |
| IP | |

D.

| | |
|------|-----|
| SMTP | FTP |
| UDP | TCP |
| IP | |

【答案】C

【解析】



2017 年 5 月第 69 题

在异步通信中，每个字符包含 1 位起始位、7 位数据位和 2 位终止位，若每秒钟传送 500 个字符，则有效数据速率为(69)。

- (69) A. 500b/s B. 700b/s C. 3500b/s D. 5000b/s

【答案】C

【解析】

解析： $(1+7+2) * 500 * 7/10 = 3500\text{b/s}$

2017 年 5 月第 70 题

以下路由策略中，依据网络信息经常更新路由的是(70)。

- (70) A. 静态路由 B. 洪泛式 C. 随机路由 D. 自适应路由

【答案】D

【解析】

路由算法分为：

一、静态路由算法

- 泛射路由算法（扩散法）
- 固定路由算法
- 随机走动法（Random Walk）
- 最短路径法（Shortest Path, SP）

二、动态路由算法(自适应路由选择算法)

- 分布式路由选择
- 集中式路由选择。
- 混合式动态路由选择
- 链路状态路由算法

2017年5月第71、72、73、74、75题

The beauty of software is in its function, in its internal structure, and in the way in which it is created by a team. To a user, a program with just the right features presented through an intuitive and (71) interface is beautiful. To a software designer, an internal structure that is partitioned in a simple and intuitive manner, and that minimizes internal coupling is beautiful. To developers and managers, a motivated team of developers making significant progress every week, and producing defect-free code, is beautiful. There is beauty on all these levels.

Our world needs software—lots of software. Fifty years ago software was something that ran in a few big and expensive machines. Thirty years ago it was something that ran in most companies and industrial settings. Now there is software running in our cell phones, watches, appliances, automobiles, toys, and tools. And need for new and better software never (72). As our civilization grows and expands, as developing nations build their infrastructures, as developed nations strive to achieve ever greater efficiencies, the need for more and more Software (73) to increase. It would be a great shame if, in all that software, there was no beauty.

We know that software can be ugly. We know that it can be hard to use, unreliable, and carelessly structured. We know that there are software systems whose tangled and careless internal structures make them expensive and difficult to change. We know that there are software systems that present their features through an awkward and cumbersome interface. We know that there are software systems that crash and misbehave. These are (74) systems. Unfortunately, as a profession, software developers tend to create more ugly systems than beautiful ones.

There is a secret that the best software developers know. Beauty is cheaper than ugliness. Beauty is faster than ugliness. A beautiful software system can be built and maintained in less time, and for less money, than an ugly one. Novice software developers don't understand this. They think that they have to do everything fast and quick. They think that beauty is (75). No! By doing things fast and quick, they make messes that make the software stiff, and hard to understand. Beautiful systems are flexible and easy to understand. Building them and maintaining them is a joy. It

is ugliness that is impractical. Ugliness will slow you down and make your software expensive and brittle. Beautiful systems cost the least build and maintain, and are delivered soonest.

(71) A. Simple B. Hard C. Complex D. Duplicated

(72) A. happens B. exists C. stops D. starts

(73) A. starts B. continues C. appears D. stops

(74) A. practical B. useful C. beautiful D. ugly

(75) A. impractical B. perfect C. time-wasting D. practical

【答案】 A C B D A

【解析】

试题一（15 分）

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】

某医疗器械公司作为复杂医疗产品的集成商，必须保持高质量部件的及时供应。为了实现这一目标，该公司欲开发一采购系统。系统的主要功能如下：

1. 检查库存水平。采购部门每天检查部件库存量，当特定部件的库存量降至其订货点时，返回低存量部件及库存量。
2. 下达采购订单。采购部门针对低存量部件及库存量提交采购请求，向其供应商(通过供应商文件访问供应商数据)下达采购订单，并存储于采购订单文件中。
3. 交运部件。当供应商提交提单并交运部件时，运输和接收(S/R)部门通过执行以下三步过程接收货物：

(1) 验证装运部件。通过访问采购订单并将其与提单进行比较来验证装运的部件，并将提单信息发给 S/R 职员。如果收货部件项目出现在采购订单和提单上，则已验证的提单和收货部件项目将被送去检验。否则，将 S/R 职员提交的装运错误信息生成装运错误通知发送给供应商。

(2) 检验部件质量。通过访问质量标准来检查装运部件的质量，并将已验证的提单发给检验员。如果部件满足所有质量标准，则将其添加到接受的部件列表用于更新部件库存。如果部件未通过检查，则将检验员创建的缺陷装运信息生成缺陷装运通知发送给供应商。

(3) 更新部件库存。库管员根据收到的接受的部件列表添加本次采购数量，与原有库存量累加来更新库存部件中的库存量。标记订单采购完成。现采用结构化方法对该采购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。

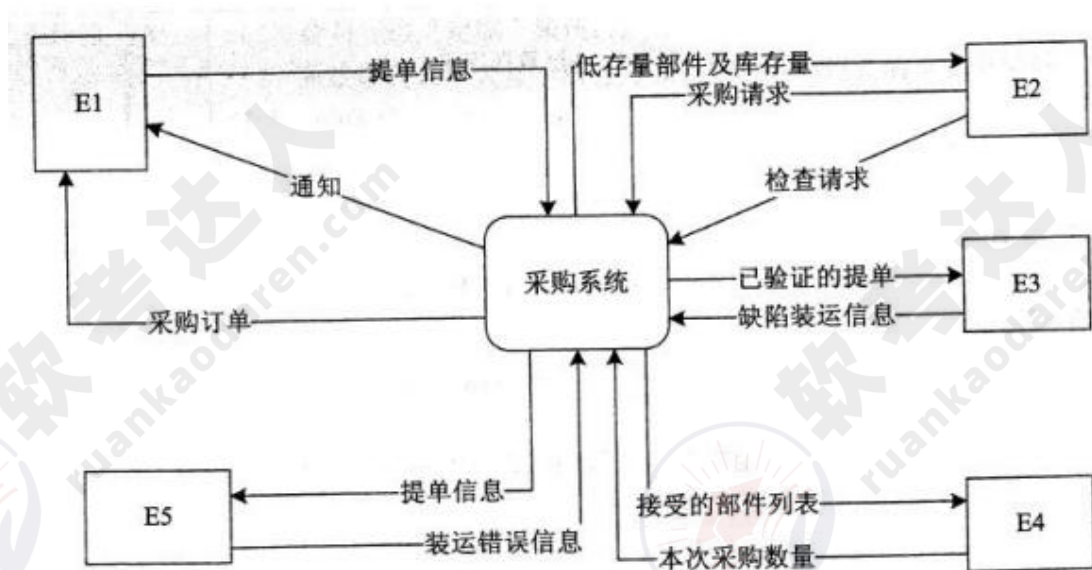


图 1-1 上下文数据流图

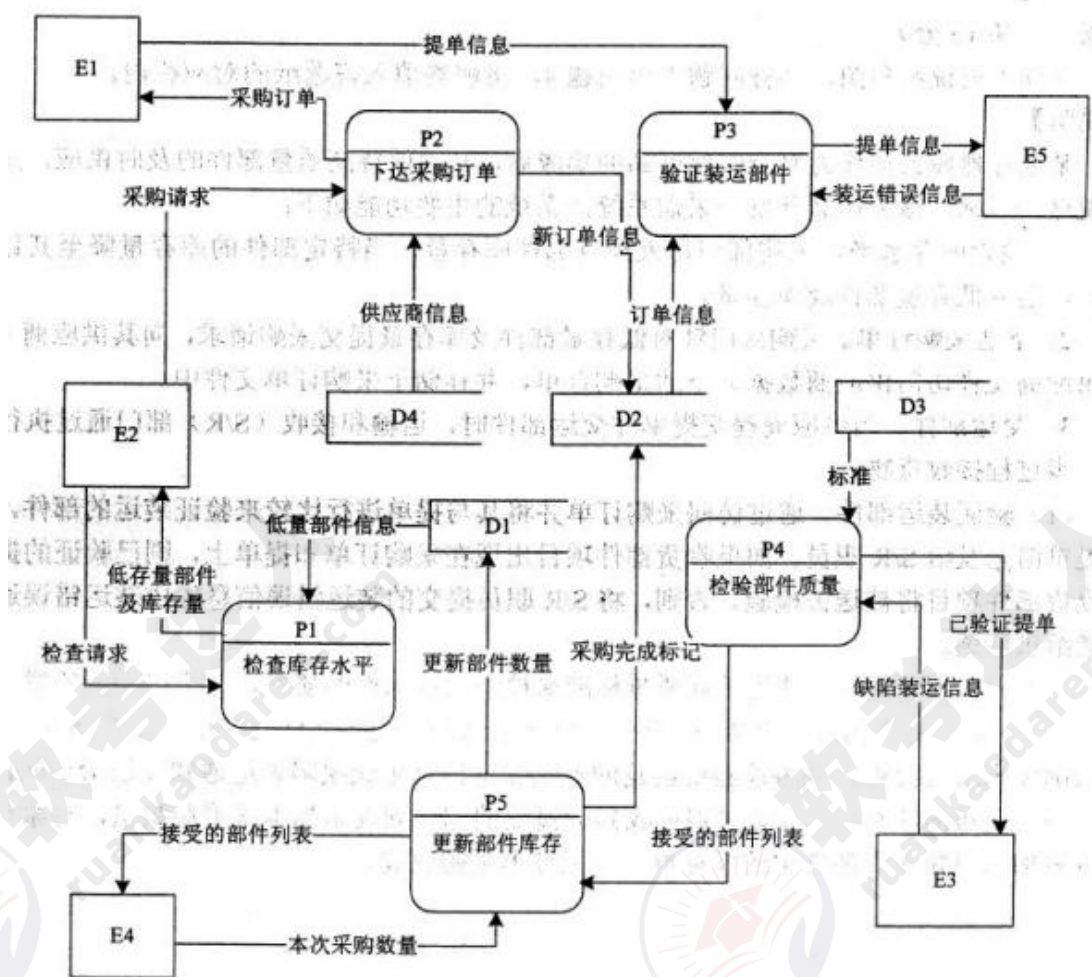


图 1-2 0 层数据流图

【问题 1】(5 分)

使用说明中的词语，给出图 1-1 中的实体 E1 ~E5

E1 供应商

E2 采购部门

E3 检验员

E4 库管员

E5 S/R 职员

【问题 2】(4 分)

使用说明中的词语，给出图 1-2 中的数据存储 D1~D4 的名称。

D1 部件库存表

D2 采购订单文件

D3 质量标准文件

D4 供应商文件

【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。

检查库存信息：P1（检查库存水平）-----D1（部件库存表）

产品送检：P3（验证装运部件）-----P4（校验部件质量）

装运错误通知：P3（验证装运部件）-----E1（供应商）

缺陷装运通知：P4（校验部件质量）-----E1（供应商）

【问题 4】(2 分)

用 200 字以内文字，说明建模图 1-1 和图 1-2 时如何保持数据流图平衡。

父图中某个加工的输入输出数据流必须与其子图的输入输出数据流在数量上和内容上保持一致

即数据不会凭空产生，也不能凭空消失。父图的一个输入（或输出）数据流应对应子图中几个输入

（或输出）数据流，而子图中组成的这些数据流的数据项全体正好是父图中的这个数据流。

试题二(共 15 分)

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】

某房屋租赁公司拟开发一个管理系统用于管理其持有的房屋、租客及员工信息。请根据下述需求描述完成系统的数据库设计。

【需求描述】

1. 公司拥有多幢公寓楼，每幢公寓楼有唯一的楼编号和地址。每幢公寓楼中有多套公寓，每套公寓在楼内有唯一的编号(不同公寓楼内的公寓号可相同)。系统需记录每套公寓的卧室数和卫生间数。
2. 员工和租客在系统中有唯一的编号(员工编号和租客编号)。
3. 对于每个租客，系统需记录姓名、多个联系电话、一个银行账号(方便自动扣房租)、一个紧急联系人的姓名及联系电话。
4. 系统需记录每个员工的姓名、一个联系电话和月工资。员工类别可以是经理或维修工，也可兼任。每个经理可以管理多幢公寓楼。每幢公寓楼必须由一个经理管理。系统需记录每个维修工的业务技能，如：水暖维修、电工、木工等。
5. 租客租赁公寓必须和公司签订租赁合同。一份租赁合同通常由一个或多个租客(合租)与该公寓楼的经理签订，一个租客也可租赁多套公寓。合同内容应包含签订日期、开始时间、租期、押金和月租金。

【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(不完整)如图 2-1 所示。

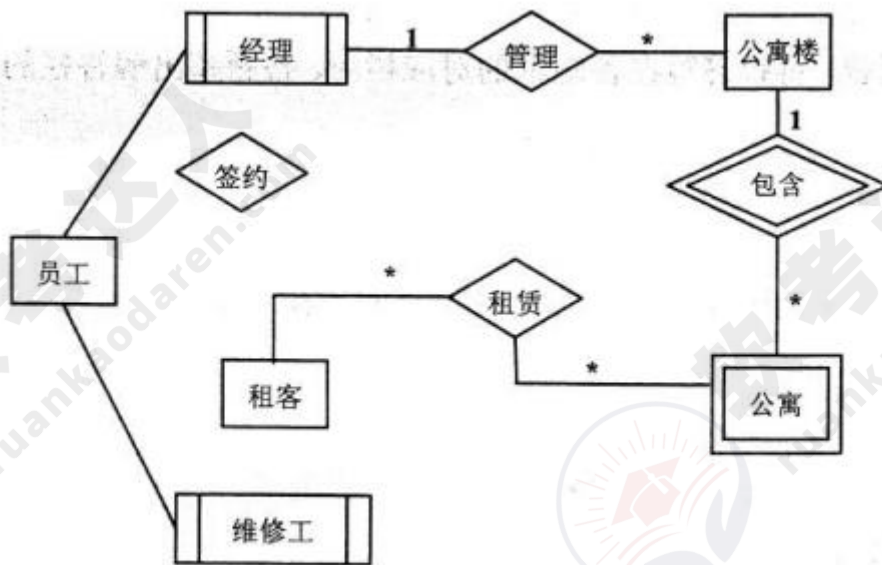


图 2-1 实体联系图

【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(不完整)：

联系电话(电话号码，租客编号)

租客(租客编号，姓名，银行账号，联系人姓名，联系人电话)

员工(员工编号，姓名，联系电话，类别，月工资， (a))

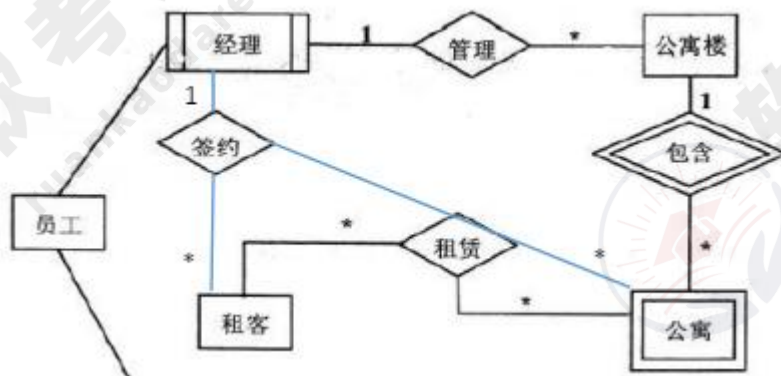
公寓楼((b) ，地址，经理编号)

公寓(楼编号，公寓号，卧室数，卫生间数)

合同(合同编号，租客编号，楼编号，公寓号，经理编号，签订日期，
起始日期，租期， (c) ，押金)

【问题 1】(4.5 分)

补充图 2-1 中的“签约”联系所关联的实体及联系类型。



维修工

【问题 2】(4.5 分)

补充逻辑结构设计中的 (a)、(b)、(c) 三处空缺。

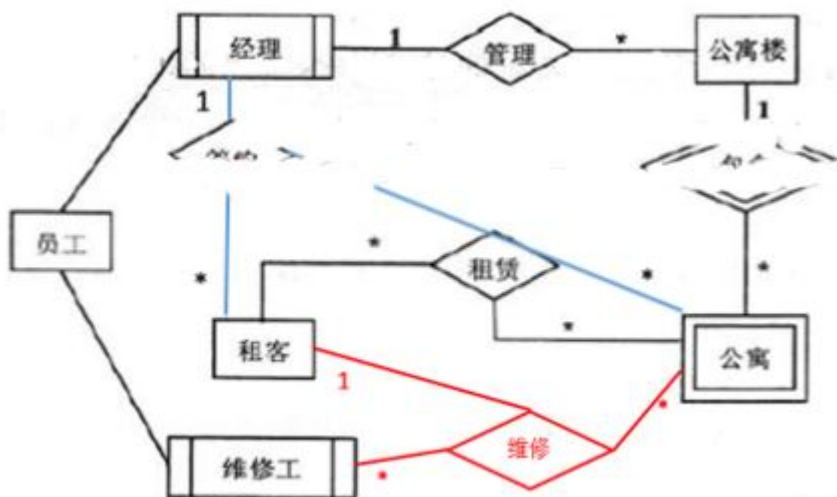
(a) 业务技能

(b) 楼编号

(c) 月租金

【问题 3】(6 分)

在租期内，公寓内设施如出现问题，租客可在系统中进行故障登记，填写故障描述，每项故障由系统自动生成唯一的故障编号，由公司派维修工进行故障维修，系统需记录每次维修的维修日期和维修内容。请根据此需求，对图 2-1 进行补充，并将所补充的 ER 图内容转换为一个关系模式，请给出该关系模式。



试题三（共 15 分）

阅读下列系统设计说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某玩具公司正在开发一套电动玩具在线销售系统，用于向注册会员提供端对端的玩具定制和销售服务。在系统设计阶段，“创建新订单（New Order）”的设计用例详细描述如表 3-1 所示，候选设计类分类如表 3-2 所示，并根据该用例设计出部分类图如圈子 3-1 所示。

表 3-1 创建新订单（NewOrder）设计用例

| | |
|-------|--|
| 用例名称 | 创建新订单 New Order |
| 用例编号 | ETM-R002 |
| 参与者 | 会员 |
| 前提条件 | 会员已经注册并成功登录系统 |
| 典型事件流 | <div>1. 会员（C1）点击“新的订单”按钮；</div> <div>2. 系统列出所有正在销售的电动玩具清单及价格（C2）</div> <div>3. 会员点击复选框选择所需电动玩具并输入对应数量，点击“结算”按钮；</div> <div>4. 系统自动计算总价（C3），显示销售清单和会员预先设置个人资料的收货地址和支付方式（C4）；</div> <div>5. 会员点击“确认支付”按钮；</div> <div>6. 系统自动调用支付系统（C5）接口支付该账单；</div> <div>7. 若支付系统返回成功标识，系统生成完整订单信息持久存储到数据库订单表（C6）中；</div> <div>8. 系统将以表格形式显示完整订单信息（C7），同时自动发送完整订单信息（C8）至会员预先配置的邮箱地址</div> |

| | | |
|-------|-------|--|
| | (C9)。 | |
| 候选事件流 | 3a | <p>(1) 会员点击“定制”按钮；</p> <p>(2) 系统以列表形式显示所有可以定制的电动玩具清单和定制属性（如尺寸、颜色等）（C10）；</p> <p>(3) 会员点击单选按钮选择所需要定制的电动玩具并填写所需要定制的属性要求，点击“结算”按钮；</p> <p>(4) 回到步骤 4.</p> |
| | 7a | <p>(1) 若支付系统返回失败标识，系统显示会员当前默认支付方式（C11）让会员确认；</p> <p>(2) 若会员点击“修改付款”按钮，调用“修改付款”用例，可以新增并存储为默认支付方式（C12），回到步骤 4；</p> <p>(3) 若会员点击“取消订单”，则该用例终止执行。</p> |

表 3-2 候选设计类分类

| | |
|------------------------------|-----|
| 接口类（Interface, 负责系统与用户之间的交互） | (a) |
| 控制类（Control, 负责业务逻辑的处理） | (b) |
| 实体类（Entity, 负责持久化数据的存储） | (c) |

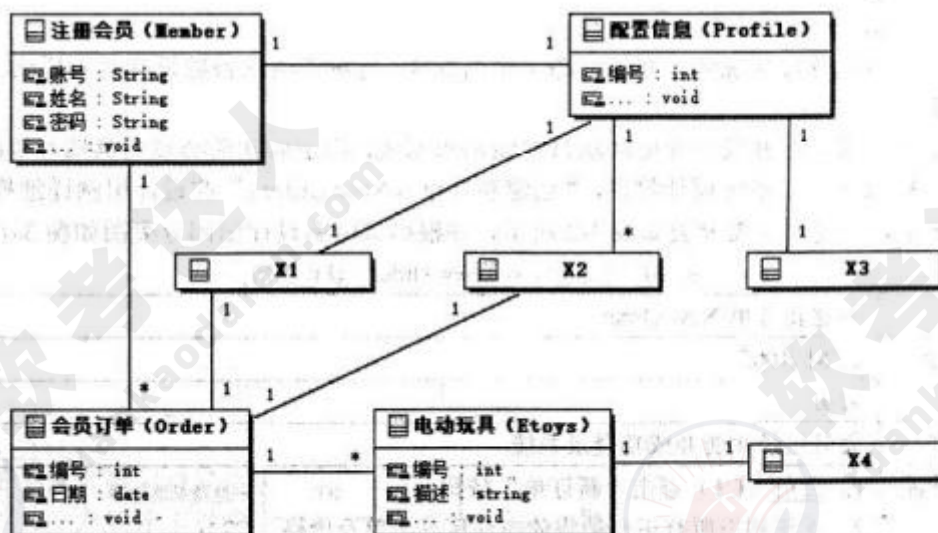


图 3-1 部分类图

在订单处理的过程中，会员可以点击“取消订单”取消该订单。如果支付失败，该订单将被标记为挂起状态，可后续重新支付，如果挂起超时 30 分钟未支付，系统将自动取消该订单。订单支付成功后，系统判断订单类型：(1)对于常规订单，标记为备货状态，订单信息发送到货运部，完成打包后交付快递发货；(2)对于定制订单，会自动进入定制状态，定制完成后交付快递发货。会员在系统中点击”收货”按钮变为收货状态，结束整个订单的处理流程。根据订单处理过程所设计的状态图如图 3-2 所示。

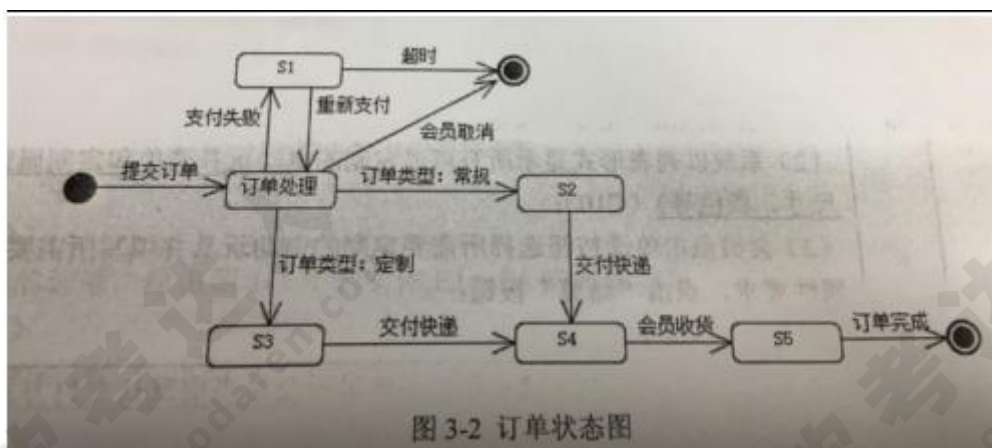


图 3-2 订单状态图

【问题 1】(6 分)

根据表 3-1 中所标记的候选设计类，请按照其类别将编号 C1~C12 分别填入表 3-2 中的 (a)、(b) 和 (c) 处。

(a): C4、C5、C7、C10、C11

(b): C3、C8

(c): C1、C2、C6、C9、C12

【问题 2】（4 分）

根据创建新订单的用例描述，请给出图 3-1 中 X1~X4 处对应类的名称。

X1: 收货地址

X2: 支付方式

X3: 邮箱地址

X4: 定制

【问题 3】（5 分）

根据订单处理过程的描述，在图 3-2 中 S1~S5 处分别填入对应的状态名称。

S1: 订单挂起

S2: 订单备货

S3: 订单定制

S4: 订单发货

S5: 订单收货

试题四

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】

假币问题：有 n 枚硬币，其中有一枚是假币，已知假币的重量较轻。现只有一个天平，要求用尽量少的比较次数找出这枚假币。

【分析问题】

将 n 枚硬币分成相等的两部分：

(1) 当 n 为偶数时，将前后两部分，即 $1 \dots n/2$ 和 $n/2+1 \dots 0$ ，放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币：

(2) 当 n 为奇数时，将前后两部分，即 $1 \dots (n-1)/2$ 和 $(n+1)/2+1 \dots 0$ ，放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币：若两端重量相等，则中间的硬币，即第 $(n+1)/2$ 枚硬币是假币。

【C 代码】

下面是算法的 C 语言实现，其中：

coins[]：硬币数组

first, last：当前考虑的硬币数组中的第一个和最后一个下标

```
#include <stdio.h>

int getCounterfeitCoin(int coins[], int first, int last)
{
    int firstSum = 0, lastSum = 0;
    int i;
    If(first==last-1) { /*只剩两枚硬币*/
        if(coins[first] < coins[last])
            return first;
        return last;
    }
    if((last - first + 1) % 2 == 0) { /*偶数枚硬币*/
        for(i = first; i < (last - 1); i++) {
            firstSum += coins[i];
        }
        for(i = last; i > (last + 1); i--) {
            lastSum += coins[i];
        }
        if(firstSum < lastSum)
            return first;
        else
            return last;
    }
    if(firstSum < lastSum)
        return first;
    else
        return last;
}
```



```

    }

    for(i=first + (last-first) / 2 + 1;i < last +1;i++){

        lastSum += coins[i];

    }

    if( 2 ){

        Return  getCounterfeitCoin(coins, first, first+(last-first)/2;)

    }else{

        Return  getCounterfeitCoin(coins, first+(last-first)/2+1, last;)

    }

    } else    { /*奇数枚硬币*/

        For(i=first;i<first+(last-first)/2;i++){

            firstSum+=coins[i];

        }

        For(i=first+(last-first)/2+1;i<last+1;i++){

            lastSum+=coins[i];

        }

        If(firstSum<lastSum){

            Return  getCounterfeitCoin(coins, first, first+(last-first)/2-1);

        }else if(firstSum>lastSum){

            Return  getCounterfeitCoin(coins, first+(last-first)/2-1, last);

        }else{

            Return( 3 )

        }

    }

}

```

【问题一】

根据题干说明，填充 C 代码中的空（1）-（3）

- (1) $\text{first} + (\text{last} - \text{first}) / 2$ 或 $(\text{first} + \text{last}) / 2$
- (2) $\text{firstSum} < \text{lastSum}$
- (3) $\text{first} + (\text{last} - \text{first}) / 2$ 或 $(\text{first} + \text{last}) / 2$

【问题二】

根据题干说明和 C 代码，算法采用了（ ）设计策略。函数 `getCounterfeitCoin` 的时间复杂度为（ ）（用 O 表示）。

- (4) 分治法
- (5) $O(n \log n)$

【问题三】

若输入的硬币数为 30，则最少的比较次数为（ ），最多的比较次数为（ ）。

- (6) 2
- (7) 4

试题五(共 15 分)

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】

某快餐厅主要制作并出售儿童套餐，一般包括主餐(各类比萨)、饮料和玩具，其餐品种类可能不同，但其制作过程相同。前台服务员(Waiter)调度厨师制作套餐。现采用生成器(Builder)模式实现制作过程，得到如图 5-1 所示的类图。

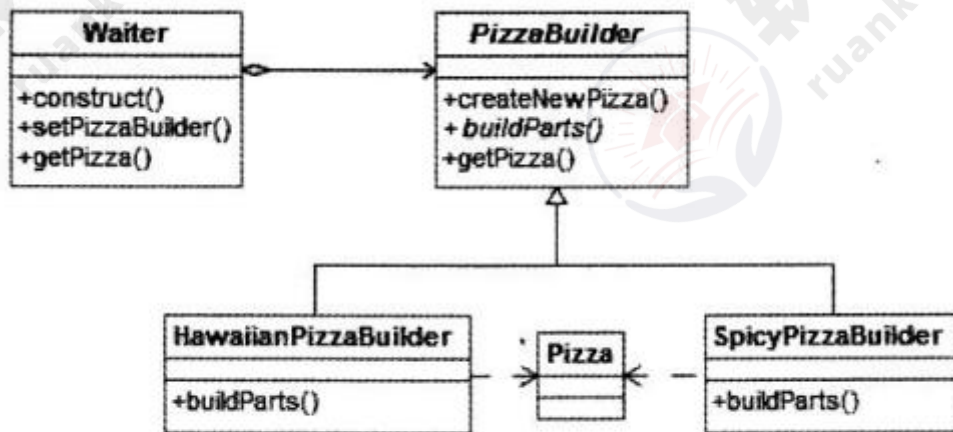


图 5-1 类图

【C++代码】

```

#include<iostream>

#include <string>

using namespace std;

class Pizza {
private: string parts;
public:
void setParts(string parts) { this->parts=parts; }
string getParts() { return parts; }
};

class PizzaBuilder {
protected:Pizza* pizza;

```

```

public:

Pizza* getPizza() { return pizza; }

void createNewPizza() { pizza = new Pizza(); }

( 1 );
}

class HawaiianPizzaBuilder :public PizzaBuilder {
public:
void buildParts() { pizza->setParts("cross +mild +ham&pineapple"); }
class SpicyPizzaBuider: public PizzaBuilder {

public:

void buildParts() { pizza->setParts("pan baked +hot +ham&pineapple"); }

}

Class Waiter{

Private:

PizzaBuilder* pizzaBuilder;

public:

void setPizzaBuilder(PizzaBuilder* pizzaBuilder) { /*设
置构建器*/
( 2 )
}

Pizza* getPizza() { return pizzaBuilder->getPizza(); }

void construct() { /*构建*/
pizzaBuilder->createNewPizza();
( 3 )
}

};

int main() {

```

```
Waiter*waiter=new Waiter();

PizzaBuilder*hawaiian pizzabuilder=new
HawaiianPizzaBuilder()
( 4 );
( 5 );

cout<< "pizza: "<< waiter->getPizza()->getParts()<< endl;
}
```

程序的输出结果为：

pizza: cross + mild + ham&pineapple

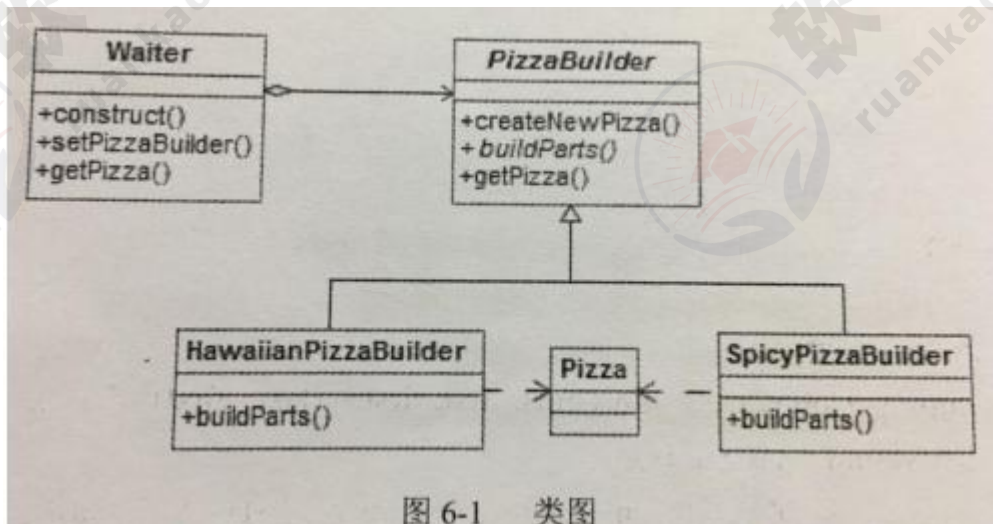
- (1) virtual void buildParts()
- (2) this->pizzaBuilder=pizzaBuilder
- (3) pizzaBuilder->buildParts()
- (4) waiter->setPizzaBuilder(hawaiian_pizzabuilder)
- (5) waiter->construct()

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】

某快餐厅主要制作并出售儿童套餐，一般包括主餐(各类比萨)、饮料和玩具，其餐品种类可能不同，但其制作过程相同。前台服务员(Waiter) 调度厨师制作套餐。现采用生成器(Builder) 模式实现制作过程，得到如图 6-1 所示的类图。



【Java 代码】

```

class Pizza {

    private String parts;

    public void setParts(String parts) { this.parts = parts;}

    public String toString() { return this.parts; }

}
  
```

```

abstract class PizzaBuilder {

    protected Pizza pizza;

    public Pizza getPizza() { return pizza; }

    public void createNewPizza() { pizza = new Pizza(); }

    public (1) ;

}
  
```

```

class HawaiianPizzaBuilder extends PizzaBuilder {
  
```

```
public void buildParts() { pizza.setParts("cross + mild +ham&pineapple" );  
}
```

```
class SpicyPizzaBuilder extends PizzaBuilder {  
    public void buildParts() { pizza.setParts("pan baked + hot  
+pepperoni&salami"); }  
}
```

```
class Waiter {  
    private PizzaBuilder pizzaBuilder;  
    public void setPizzaBuilder(PizzaBuilder pizzaBuilder) { /*  
        设置构建器*/  
        (2) ;  
    }  
    public Pizza getPizza() { return pizzaBuilder.getPizza(); }  
    public void construct() { /*构建*/  
        pizzaBuilder.createNewPizza();  
        (3) ;  
    }  
}
```

```
Class FastFoodOrdering {  
    public static void main(String[] args) {  
        Waiter waiter = new Waiter();  
        PizzaBuilder hawaiian_pizzabuilder = new  
        HawaiianPizzaBuilder();  
        (4) ;  
        (5) ;  
        System.out.println("pizza: " + waiter.getPizza());  
    }  
}
```

```
}
```

程序的输出结果为：

Pizza:cross + mild + ham&pineapple

- (1) abstract void buildParts();
- (2) this.pizzaBuilder=pizzaBuilder
- (3) pizzaBuilder.buildParts()
- (4) waiter.setPizzaBuilder(hawaiian_pizzabuilder)
- (5) waiter.construct()