
全国计算机技术与软件专业技术资格（水平）考试

2023 年下半年 软件设计师

线性表采用链表存储结构的特点中不包括 (1)。

- (1) A. 所需空间大小与表长成正比
B. 可随机访问表中的任意元素
C. 插入和删除操作不需要移动元素
D. 无需事先估计存储空间大小

- A. 所需空间大小与表长成正比: 这是链表的一个特点。链表中的每个元素包含数据部分和指向下一个元素的指针部分, 因此链表的总空间需求直接与元素个数 (即表长) 成正比。
- B. 可随机访问表中的任一元素 (错误): 链表不支持随机访问。要访问链表中的某个元素, 通常需要从头节点开始, 沿着指针逐个遍历, 直到找到目标元素, 这称为顺序访问。相比数组或顺序表的随机访问 (通过索引直接定位), 链表的访问效率较低。
- C. 插入和删除操作不需要移动元素: 这是链表相较于数组等顺序存储结构的优势之一。在链表中插入或删除元素, 只需要修改相邻节点的指针即可, 不需要移动元素的实际位置。
- D. 无需事先估计存储空间大小: 这也是链表的一个优势。链表可以在运行时动态地分配和释放内存, 不像数组那样需要预先知道确切的存储需求。

参考答案 (1) B

在双核处理器中，双核是指(2)。

- (2) A. 执行程序时有两条指令流水线并行工作
B. 在一个 CPU 中集成两个运算核心以提高运算能力
C. 利用超线程技术实现的多任务并行处理
D. 在主板上设置两个独立的 CPU 以提高处理能力

A. 执行程序时有两条指令流水线并行工作：这个描述可能让人联想到超标量处理器，它能够同时发送多条指令到不同执行单元，但并不直接定义“双核”。双核处理器确实可能包含多个指令流水线，但这不是“双核”的本质特征。

B. 在一个 CPU 中集成两个运算核心以提高运算能力（正确）：这是对双核处理器最直接和准确的描述。双核意味着在一个物理芯片上集成了两个独立的计算单元（核心），它们可以并行处理任务，从而显著提升处理能力和效率。

C. 利用超线程技术实现的多任务并行处理：超线程（Hyper-Threading）是一种允许单个物理核心模拟出两个或更多逻辑核心的技术，以更高效地利用 CPU 资源。虽然它可以增强并行处理能力，但它并不是“双核”的定义。双核处理器可以结合超线程技术，但二者概念不同。

D. 在主板上设置两个独立的 CPU 以提高处理能力：这是对双 CPU 系统的描述，而不是双核处理器。双 CPU 系统确实能提供更高的处理能力，但每个 CPU 都是独立的实体，不同于双核处理器中两个核心集成在同一芯片上。

参考答案 (2) B

采用贪心策略求解(3)问题，一定可以得到最优解。

- (3) A. 分数背包 B. 0-1 背包 C. 旅行商 D. 最长公共子序列

A. 分数背包（正确）：分数背包问题（Fractional Knapsack Problem）是指在有一个容量限制的背包中，选择放入一些物品以最大化总价值，且物品可以分割。对于分数背包问题，贪心策略是有效的，即按照单位重量的价值比（价值/重量）从高到低对物品排序，然后尽可能多地选取这些物品。这种方式可以确保获得最大总价值，因为它总是优先考虑性价比最高的物品。

B. 0-1 背包：0-1 背包问题与分数背包不同，因为在这个问题中，物品不能分割，只能选择放或不放（0 或 1）。贪心策略在这里不适用，因为最优解可能需要跳过某些高价值但高重量的物品，而选择低价值但轻的物品来达到背包容量限制下的最大价值。解决 0-1 背包通常需要使用动态规划或回溯等方法。

C. 旅行商（Traveling Salesman Problem, TSP）：旅行商问题是寻找最短可能路线访问每个城市一次并返回起点的问题。贪心策略，比如总是选择最近未访问的城市作为下一个目的地，通常不能保证得到最优解。TSP 是一个 NP 完全问题，通常需要更复杂的算法如动态规划、近似算法或遗传算法等来求解。

D. 最长公共子序列（Longest Common Subsequence, LCS）：LCS 问题是要找到两个序列的最长子序列，该子序列不需要在原序列中连续出现，但必须保持原序列中的相对顺序。贪心策略不能直接有效地解决 LCS 问题，这个问题通常通过动态规划来解决。

参考答案（3）A

统一过程模型的四个阶段中，在 (4) 阶段进行需求分析和架构演进。

- (4) A. 移交 B. 精化 C. 构建 D. 起始

254 软件设计师教程 (第 5 版)

个“袖珍项目”都包含正常软件项目的所有元素：计划、分析和设计、构造、集成和测试，以及内部和外部发布。

统一过程定义了 4 个技术阶段及其制品。

1) 起始阶段 (Inception Phase)

起始阶段专注于项目的初创活动，产生的主要工作产品有构想文档 (Vision Document)、初始用例模型、初始项目术语表、初始业务用例、初始风险评估、项目计划 (阶段及迭代)、业务模型以及一个或多个原型 (需要时)。

2) 精化阶段 (Elaboration Phase)

精化阶段在理解了最初领域范围之后进行需求分析和架构演进，产生的主要工作产品有用例模型、补充需求 (包括非功能需求)、分析模型、软件体系结构描述、可执行的软件体系结构原型、初步的设计模型、修订的风险列表、项目计划 (包括迭代计划、调整的工作流、里程碑和技术工作产品) 以及初始用户手册。

3) 构建阶段 (Construction Phase)

构建阶段关注系统的构建，产生实现模型，产生的主要工作产品有设计模型、软件构件、集成的软件增量、测试计划及步骤、测试用例以及支持文档 (用户手册、安装手册和对于并发增量的描述)。

4) 移交阶段 (Transition Phase)

移交阶段关注于软件提交方面的工作，产生软件增量，产生的主要工作产品有提交的软件增量、 β 测试报告和综合用户反馈。

官方教程 P254 页

参考答案 (4) B

下列协议中，不属于安全协议的是 (5)。

- (5) A. IPsec B. SNMP C. SFTP D. HTTPS

A. IPsec: Internet Protocol Security, 提供了 IP 层的数据加密和身份验证服务, 是常用的网络安全协议。

B. SNMP (简单网络管理协议): 主要用于网络设备的管理与监控, 传统上 SNMP 缺乏内置的安全机制, 虽然有 SNMPv3 引入了安全性改进, 但原始 SNMP 版本因安全性问题而不被视为典型的安全协议。

C. SFTP: Secure File Transfer Protocol, 安全文件传输协议, 是基于 SSH 协议的文件传输协议, 提供了数据加密和认证功能, 确保文件传输的安全性。

D. HTTPS: Hypertext Transfer Protocol Secure, 超文本传输安全协议, 是 HTTP 协议的加密版本, 通过 TLS/SSL 提供安全数据传输。

参考答案 (5) B

以下关于汇编语言程序的叙述中，错误的是(6)。

(6) A. 汇编程序的功能是将汇编语言源程序翻译为相应的目标程序

B. 用汇编语言编写的程序可以直接被计算机硬件执行

C. 汇编语言是低级程序设计语言

D. 汇编语言与计算机硬件体系结构密切相关

A. 汇编程序的功能是将汇编语言源程序翻译为相应的目标程序（正确）：汇编程序是一个翻译工具，它将程序员用汇编语言编写的源代码转换为计算机可以直接执行的机器语言代码（目标程序）。

B. 用汇编语言编写的程序可以直接被计算机硬件执行（错误）：汇编语言代码必须首先通过汇编程序转换成机器语言（目标代码），然后才能被执行。直接编写在硬件上的指令是机器语言，而非汇编语言。

C. 汇编语言是低级程序设计语言（正确）：汇编语言是一种接近机器语言的编程语言，它使用助记符来代表机器指令的操作码，通常与特定的计算机体系结构紧密相关。

D. 汇编语言与计算机硬件体系结构密切相关（正确）：由于汇编语言直接映射到硬件的机器指令，因此不同的计算机体系结构会有不同的汇编语言，程序的可移植性较差。

参考答案 (6) B

以下关于方法重载（Overload）和方法覆盖（Override）与多态的关系的叙述中，不正确的是(7)。

- (7) A. 覆盖通过动态绑定机制实现多态
- B. 重载通过动态绑定机制实现多态**
- C. 重载属于编译时多态，在一个类中定义多个名称相同而参数不同的方法
- D. 覆盖属于运行时多态，子类重新定义父类中已定义的方法

A. 覆盖通过动态绑定机制实现多态（正确）：方法覆盖是子类对父类中同名方法的重新实现，这种行为在运行时根据实际对象的类型决定调用哪个方法，体现了运行时多态性，确实依赖于动态绑定机制。

B. 重载通过动态绑定机制实现多态（错误）：方法重载是同一类内存在多个同名方法，这些方法的参数列表不同（包括参数数量、类型或顺序不同）。重载是在编译时期由编译器根据调用时提供的参数类型和数量来决定调用哪个方法，属于编译时多态，不涉及动态绑定。

C. 重载属于编译时多态，在一个类中定义多个名称相同而参数不同的方法（正确）：这准确描述了方法重载的概念，编译器根据方法签名在编译时期就能确定调用哪个方法版本。

D. 覆盖属于运行时多态，子类重新定义父类中已定义的方法（正确）：当子类对象调用一个被覆盖的方法时，实际执行的是子类中的实现，这是在运行时决定的，展示了多态性。

参考答案 (7) B

以下关于白盒测试原则的叙述中，不正确的是(8)。

- (8) A. 在所有的逻辑判断中，取“真”和取“假”的两种情况至少都能执行一次
B. 程序模块中的所有独立路径至少执行一次
C. 每个循环都应在边界条件和一般条件下各执行一次
D. 在输入条件规定的取值范围的情况下，合理的输入和不合理的输入至少都能执行一次

A. 在所有的逻辑判断中，取“真”和取“假”的两种情况至少都能执行一次（正确）：这是逻辑覆盖原则的一部分，特别是涉及到条件覆盖，确保了代码中每个逻辑条件的分支都被测试到。

B. 程序模块中的所有独立路径至少执行一次（正确）：这是达到路径覆盖的原则，目的是测试程序中的每一条可能的执行路径至少一次。

C. 每个循环都应在边界条件和一般条件下各执行一次（正确）：这是边界值分析的一种体现，确保循环不仅在常规情况下运行，还要特别测试边界条件，因为边界往往是错误的多发地带。

D. 在输入条件规定的取值范围的情况下，合理的输入和不合理的输入至少都能执行一次（错误）：此叙述虽然体现了测试输入的全面性，但是它更偏向于黑盒测试的关注点，尤其是提到了“不合理的输入”，这超出了白盒测试的直接范畴。白盒测试主要关注代码结构和逻辑，而不是输入的有效性；合理与不合理输入的测试是黑盒测试或灰盒测试中对功能和边界测试的考虑。

276 软件设计师教程（第5版）

白盒测试的原则如下。

- (1) 程序模块中的所有独立路径至少执行一次。
- (2) 在所有的逻辑判断中，取“真”和取“假”的两种情况至少都能执行一次。
- (3) 每个循环都应在边界条件和一般条件下各执行一次。
- (4) 测试程序内部数据结构的有效性等。

官方教程 P276

参考答案 (8) D

对采用面向对象方法开发的系统进行测试时，通常从不同层次进行测试。对类中定义的每个方法进行测试属于 (9) 层。

- (9) A. 系统 B. 算法 C. 类 D. 模板

一般来说，对面向对象软件的测试可分为下列 4 个层次进行。

(1) 算法层。测试类中定义的每个方法，基本上相当于传统软件测试中的单元测试。

(2) 类层。测试封装在同一个类中的所有方法与属性之间的相互作用。在面向对象软件中类是基本模块，因此可以认为这是面向对象测试中所特有的模块测试。

(3) 模板层。测试一组协同工作的类之间的相互作用，大体上相当于传统软件测试中的集成测试，但是也有面向对象软件的特点（例如，对象之间通过发送消息相互作用）。

(4) 系统层。把各个子系统组装成完整的面向对象软件系统，在组装过程中同时进行测试。

软件工程中传统的测试用例设计技术，如逻辑覆盖、等价类划分和边界值分析等方法，仍然可以作为测试类中每个方法的主要技术。面向对象测试的主要目标也是用尽可能低的测试成本和尽可能少的测试用例，发现尽可能多的错误。但是，面向对象程序中特有的封装、继承和多态等机制，也给面向对象测试带来一些新特点，增加了测试和调试的难度。

官方教程 P363 页

参考答案 (9) B

下列算法中，不属于公开密钥加密算法的是 (10)。

- (10) A. DSA B. ECC C. DES D. RSA

A. DSA (数字签名算法): 是一种公开密钥加密标准，用于数字签名，属于非对称加密算法。

B. ECC (椭圆曲线密码学): 是一种基于椭圆曲线数学的公开密钥加密技术，广泛应用于安全领域。

C. DES (数据加密标准): 是一种对称密钥加密算法，即加密和解密使用相同的密钥，不属于公开密钥加密算法。

D. RSA (Rivest-Shamir-Adleman): 是最著名的非对称加密算法之一，广泛应用于公开密钥加密和数字签名。

公开密钥是非对称密钥（公钥 + 私钥），非公开密钥是对称密钥（只有一把密钥）

参考答案 (10) C

POP3 服务默认的 TCP 端口号是 (11)。

(11) A. 110

B. 25

C. 20

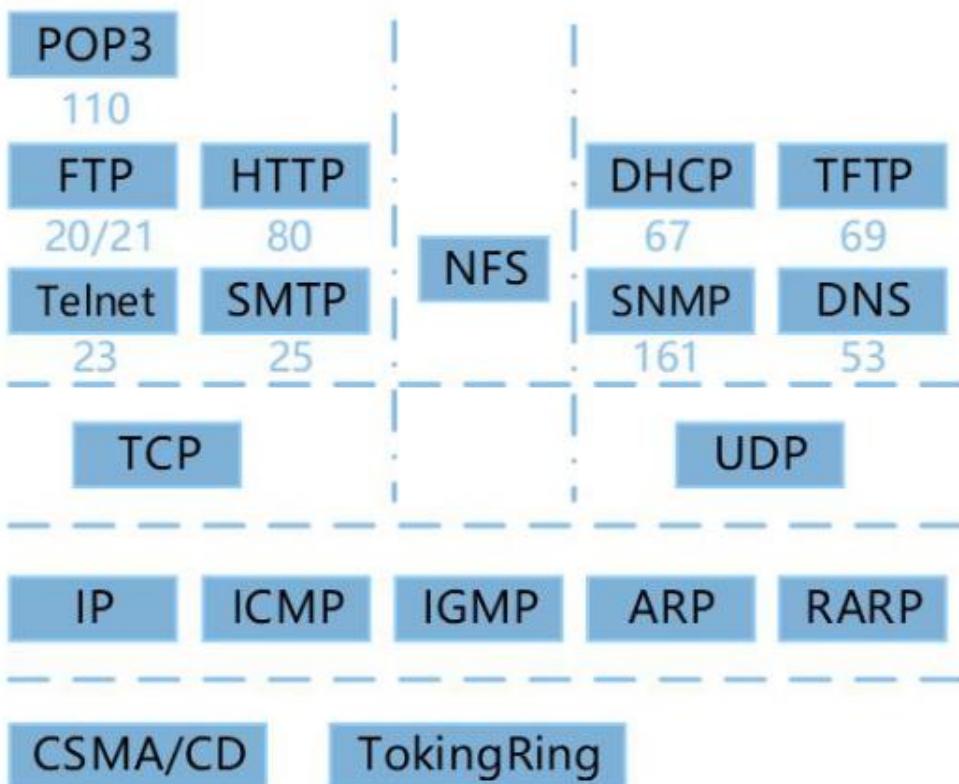
D. 80

A. 110 (正确): 这是 POP3 协议的标准端口, 用于接收邮件。客户端应用程序通常会连接到这个端口以下载或检索存储在邮件服务器上的邮件。

B. 25 (错误): 端口 25 通常与 SMTP (简单邮件传输协议) 关联, 用于发送邮件, 而不是接收。

C. 20 (错误): 端口 20 通常用于 FTP (文件传输协议) 的数据传输, 是 FTP 被动模式下服务器响应并建立数据连接的端口, 与邮件服务无关。

D. 80 (错误): 端口 80 通常与 HTTP (超文本传输协议) 关联, 用于网页浏览, 不是用于邮件服务的端口。



参考答案 (11) A

以下关于折半查找的叙述中，不正确的是 (12)。采用折半查找等概率查找某个包含 8 个元素的有序表，查找成功的平均查找长度为 (13)。

(12) A. 是一个分治算法

B. 只能应用于有序表

C. 查找成功和不成功的平均查找长度是一样的

D. 若表长为 n ，时间复杂度为 $O(\log_2 n)$

(13) A. $9/8$

B. $1/8$

C. $20/8$

D. $21/8$

A. 折半查找确实是一个分治算法，因为它将问题分成两个子问题（一半），每次比较后排除一半的元素，故 A 选项正确。

B. 折半查找只能应用于有序表，因为其核心在于根据中间元素来决定搜索区间，故 B 选项正确。

C. 查找成功和不成功的平均查找长度并不是一样的。查找成功时，平均查找长度与元素在表中的位置有关，而查找不成功时，平均查找长度通常接近于 $O(\log_2 n)$ 。因此，C 选项是不正确的。

D. 折半查找的时间复杂度确实是 $O(\log_2 n)$ ，因为它每次都将搜索区间减半，故 D 选项正确。

对于含有 8 个元素的有序表，进行等概率查找时，查找成功的平均查找长度计算如下：

成功查找到第一个元素需要比较 3 次。成功查找到第二个元素需要比较 2 次。

成功查找到第三个元素需要比较 3 次。成功查找到第四个元素需要比较 1 次。

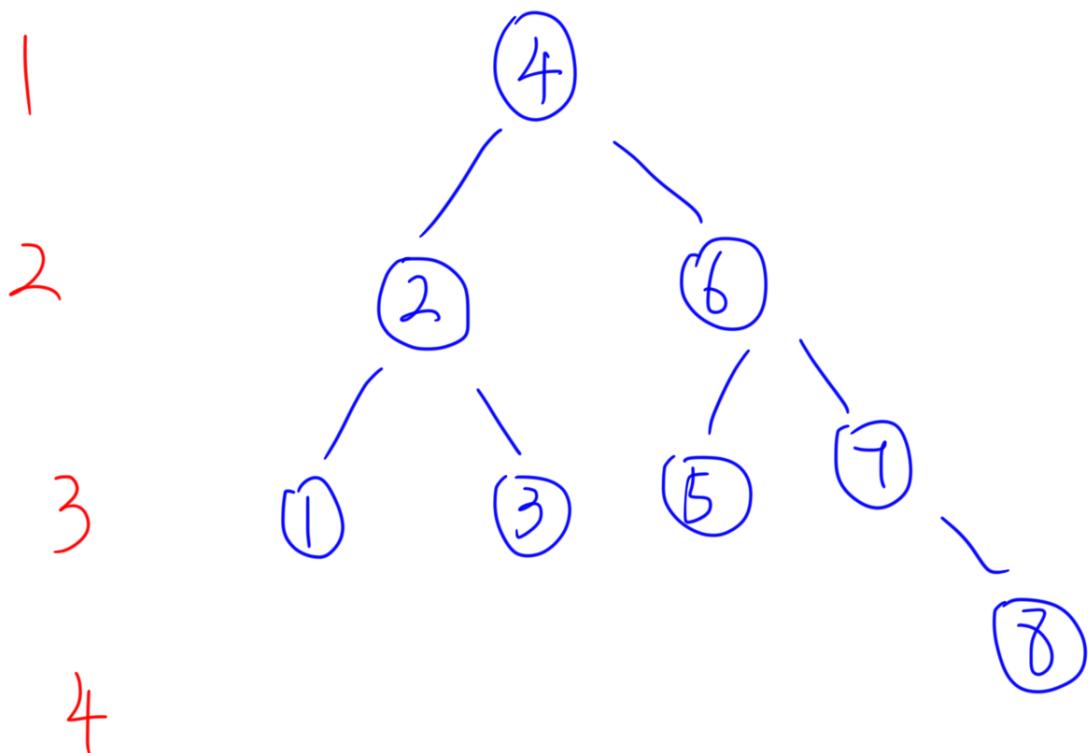
成功查找到第五个元素需要比较 3 次。成功查找到第六个元素需要比较 2 次。

成功查找到第七个元素需要比较 3 次。成功查找到最后一个元素需要比较 4 次。

$$ASL_{\text{成功}} = \frac{1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 1}{8}$$

$$ASL_{\text{成功}} = \frac{21}{8}$$

参考答案 (12) C (13) D



$$\begin{aligned}
 ASL_{\text{成功}} &= \frac{1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 1}{8} \\
 &= \frac{21}{8}
 \end{aligned}$$

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。若计算机系统的字长为 128 位，磁盘的容量为 1024GB，物理块的大小为 8MB，那么该位示图的大小为(14)个字。

(14) nA. 4096

B. 1024

C. 2048

D. 4098

首先，我们需要计算磁盘总共包含多少个物理块。磁盘的容量为 1024GB，每个物理块的大小为 8MB，所以磁盘总块数为：

$$\text{总块数} = \frac{\text{磁盘容量}}{\text{块大小}} = \frac{1024\text{GB}}{8\text{MB}} = \frac{1024 \times 1024\text{MB}}{8\text{MB}} = 131072 \text{ 块}$$

接下来，考虑到位示图中每个位代表一个物理块的使用状态(使用或未使用)，我们需要确定位示图所需的位数。由上述计算得知需要 131072 位。(用一位表示一个物理块的使用情况)

计算机系统的字长为 128 位，这意味着每个字可以表示 128 个物理块的使用情况。为了得到位示图所需的字数，我们将总位数除以字长：

$$\text{所需字数} = \frac{\text{总块数}}{\text{字长}} = \frac{131072}{128} = 1024$$

因此，位示图的大小为 1024 个字，选项 B 正确。

参考答案 (14) B

在项目开发过程中，（15）不属于项目估算的主要因素。

- （15） A. 规模 B. 类型 C. 成本 D. 工作量

在项目开发过程中，项目估算的主要因素通常包括以下几个方面：

- A. 规模：项目的规模，比如代码行数、功能点数量或用户故事点数，是估算工作量、时间安排和成本的基础，因此属于项目估算的主要因素。
- B. 虽然项目类型（如软件开发、建筑施工、科研项目等）会影响项目的管理和实施方式，但它本身并不直接作为项目估算的主要因素。不同类型项目的估算方法可能有所不同，但项目估算的核心仍然是围绕规模、成本和工作量进行的。因此，相较于其他三个选项，类型不直接决定项目估算的具体数值，故答案是 B。
- C. 成本：成本是项目估算中不可或缺的一部分，它涵盖了人力成本、材料成本、设备成本等各个方面。准确的成本估算确保项目在预算范围内完成的基础，也是决定项目是否可行的重要依据。
- D. 工作量：工作量直接关系到完成项目所需的人力资源和时间，是项目估算的核心内容之一，用以预测项目持续时间和成本。

参考答案（15）B

执行以下 Python 语句之后，列表 X 为 (16)。

```
x = [1, 2, 3]  
x.append([4, 5])
```

- (16) A. [1, 2, 3, 4, 5] B. [1, 2, 3] C. [4, 5] D. [1, 2, 3, [4,5]]

A. [1, 2, 3, 4, 5]: 这个选项认为列表 x 的末尾添加了 4 和 5 两个单独的元素，但根据 Python 的 append()方法行为，这并不正确。append()方法是在列表末尾添加一个元素，而不是展开或合并列表。可以使用 `x += [4,5]` 或 `x.extend([4, 5])` 来达到选项结果

B. [1, 2, 3]: 这个选项表示列表 x 没有变化，显然与执行了 `x.append([4, 5])` 的事实不符。

C. [4, 5]: 这个选项只显示了被添加的部分，忽略了原有的列表内容，因此也是错误的。

D. [1, 2, 3, [4,5]]: 这是正确的答案。`x.append([4, 5])` 操作将整个列表 [4, 5] 作为一个单一元素添加到了列表 x 的末尾，因此 x 变为 [1, 2, 3, [4,5]]。这里，[4, 5] 作为一个元素整体存在于列表 x 中，保持了其内部结构不变。

参考答案 (16) D

防火墙不具备 (17) 功能。

- (17) A. 病毒防治 B. 状态检测 C. 代理 D. 包过滤

A. 病毒防治：这是防火墙不具备的功能。防火墙主要用于监控和过滤进出网络的数据包，基于预定义的安全规则来允许或阻止流量。它不涉及文件内容的深度检查，无法直接检测或清除病毒。病毒防治通常是通过专门的反病毒软件来实现的，这类软件能够扫描文件、邮件等，识别并清除恶意代码。

B. 状态检测：这是现代防火墙的一个重要功能。状态检测防火墙不仅检查单个数据包，还会跟踪数据包之间的关联性及网络连接的状态，以此来决定是否允许数据包通过，提高了安全性同时减少了对性能的影响。

C. 代理：防火墙可以提供代理服务，作为内外网通信的中介。应用级网关（也称为代理防火墙）能够代表内部网络用户发起对外部网络的请求，并将响应传回，这样可以隐藏内部网络的细节，增加安全性。

D. 包过滤：这是防火墙最基本的功能之一，通过检查数据包的源地址、目的地址、端口号和协议类型等信息，依据预设的规则决定是否允许数据包通过，实现基本的网络安全控制。

参考答案 (17) A

一棵哈夫曼树共有 127 个结点，对其进行哈夫曼编码，共能得到 (18) 个字符的编码。

(18) A. 64

B. 127

C. 63

D. 126

哈夫曼树 (Huffman Tree) 是一种带权路径长度最短的二叉树，用于构建哈夫曼编码。在构造哈夫曼树的过程中，除了最后一个结点（假设为叶子结点）外，每合并两个结点就会生成一个新的内部结点，这意味着每增加一个结点到树中，实际上代表了一个字符的编码过程。最终的哈夫曼树除了叶结点外的所有结点都是由叶结点合并而成。

如果有 n 个叶结点（即要编码的字符数），那么会构建 $n - 1$ 个内部结点。因为每次合并减少一个叶结点，直至最后只剩下一个结点（根结点）。所以总结点数 $N = n + (n - 1) = 2n - 1$ 。

给定题目中的哈夫曼树有 127 个结点，我们可以通过公式反推出叶结点的数量 n :

$$127 = 2n - 1$$

解这个方程得到:

$$128 = 2n$$

$$64 = n$$

这表示有 64 个叶结点，即有 64 个字符需要编码。但题目要求的是能得到多少个字符的编码，实际上每个叶结点都对应一个字符的编码，因此答案应该是叶结点的数量，即 64。

参考答案 (18) A

浮点加（减）法运算过程中需要以下操作要素：

- ① 零操作数检查
- ② 规格化及舍入处理
- ③ 尾数加（减）法运算
- ④ 对阶操作

正确的加（减）法运算操作流程是 (19)。

- (19) A. ①③④② B. ①④③② C. ②①④③ D. ④③②①

- A. ①③④②：错误，因为先做尾数加减再对阶会导致无法正确对齐操作数的值。
- C. ②①④③：错误，规格化应在尾数加减之后进行，且零操作数检查应是最开始的步骤。
- D. ④③②①：错误，零操作数应最先检查，且对阶应在尾数加减之前完成。

正确的步骤是：

① 零操作数检查：首先检查参与运算的操作数是否为零，因为零参与运算可以直接得出结果，无需后续步骤，这样可以简化计算流程。如果任一操作数为零，根据加法或减法直接得出结果（加零得原数，减零也得原数；减法中被减数为零时需考虑特殊处理）。

④ 对阶操作：浮点数的表示包括阶码和尾数部分。为了能够执行加减运算，需要确保两个操作数的小数点位置相同，即阶码对齐。这意味着较高阶的操作数需要将其尾数右移（相当于乘以2的较小幂次），较低阶的操作数则相反，以此来调整它们的指数部分相等。

③ 尾数加（减）法运算：在完成对阶后，两个操作数的尾数可以直接相加或相减。这是实际的数值运算部分。

② 规格化及舍入处理：尾数相加（减）后，结果可能不是规格化的形式（即尾数的最高位不是1，或者超出了尾数的固定精度）。这时需要进行规格化，即通过左移尾数和调整阶码保持最高位为1，同时可能需要进行舍入处理以符合浮点数的精度要求。此外，还需检查是否溢出或下溢等特殊情况。

参考答案 (19) B

某队列允许在其两端进行入队操作，但仅允许在一端进行出队操作。若元素 a、b、c、d 依次全部入队列，之后进行出队列操作，则不能得到的出队序列是 (20)。

- (20) A. dbac B. cabd C. acdb D. bacd

A. dbac: 可能。如果 a、b 先从一端入队，然后 c、d 从另一端入队，则可以先出 d，再按 b、a、c 的顺序出队。

B. cabd: 可能。a、b 先入队，c、d 从另一端入队，先出 c，接着 b、a、d 顺序出队。

C. acdb: 不可能。要使 d 最后出队，d 必须是最晚入队的。但若 c 先于 d 出队，说明 c 比 d 早入队，而 a、c 从一端入队，b、d 从另一端入队的话，无论如何安排入队顺序，都不可能先出 c 再出 d，因为一旦开始出队，d 作为最后入队的元素，在 c 之前出队是不可能的，违反了队列的基本规则。

D. bacd: 可能。类似地，若 a、b 先入队，然后 c、d 从另一端入队，可以先出 b，接着是 a、c，最后 d 出队。

参考答案 (20) C

以下关于基于构件的开发模型的叙述中，不正确的是 (21)。

- (21) A. 本质上是演进模型，以迭代方式构建软件
B. 必须采用面向对象开发技术
C. 采用预先打包的软件构件构造软件
D. 构件可以是组织内部开发的，也可以是商品化成品软件构件

A. "本质上是演进模型，以迭代方式构建软件": 这个描述并非不正确。基于构件的开发模型确实支持迭代开发，可以在开发过程中逐步集成和测试软件构件，但这并不是所有基于构件开发模型的本质特征，有些模型也可以是计划驱动的。不过，这句话本身不构成直接的错误。

B. "必须采用面向对象开发技术": 这是不正确的。虽然面向对象技术为构件复用提供了便利，基于构件的开发模型并不限定必须使用面向对象开发技术。构件可以基于函数库、服务、模块等形式，只要它们能够作为独立的、可组合的单元使用即可。因此，这项叙述是不正确的。

C. "采用预先打包的软件构件构造软件": 这是基于构件开发模型的核心思想之一，即通过使用已有的、预先设计好的软件构件来构建应用程序，以提高开发效率和质量。因此，这个选项是正确的。

D. "构件可以是组织内部开发的，也可以是商品化成品软件构件": 这也是正确的。基于构件的开发鼓励利用内部已有资源和外部成熟产品，既包括企业内部开发的定制构件，也包括市场上可购买的商品化构件。

252 软件设计师教程（第5版）

5.2.5 基于构件的开发模型 (Component-based Development Model)

基于构件的开发是指利用预先包装的构件来构造应用系统。构件可以是组织内部开发的构件，也可以是商品化成品 (Commercial Off-The-Shelf, COTS) 软件构件。基于构件的开发模型具有许多螺旋模型的特点，它本质上是演化模型，需要以迭代方式构建软件。其不同之处在于，基于构件的开发模型采用预先打包的软件构件开发应用系统。

官方教程 P252

参考答案 (21) B

在 C/C++ 程序中，对于函数中定义的非静态局部变量，其存储空间在 (22) 分配。

- (22) A. 栈区 B. 静态数据区 C. 文本区 D. 自由堆区

A. 栈区：这是正确的选项。在 C/C++ 程序中，函数中定义的非静态局部变量存储在栈区。当函数被调用时，编译器会自动为这些局部变量分配内存，并在函数退出时自动释放这些内存。栈区的特点是自动分配和自动释放，适用于存储生存周期与函数调用周期一致的变量。

B. 静态数据区：静态数据区用于存储全局变量、静态变量（包括全局静态变量和局部静态变量）。这些变量在整个程序生命周期中都存在，与非静态局部变量的生命周期不同，因此非静态局部变量不由静态数据区分配空间。

C. 文本区：也称为代码区，存放程序的机器指令，不用于存储变量。

D. 自由堆区：通过 `malloc`、`calloc`、`realloc`、`new` 等动态内存分配函数所得到的内存空间位于堆区，需要程序员显式释放。非静态局部变量不由程序员手动分配和释放，因此不会存储在堆区。

参考答案 (22) A

以下关于测试原则的叙述中，不正确的是(23)。

- (23) A. 充分注意测试中的群集现象
- B. 设计测试用例时，应包括合理的输入条件和不合理的输入条件
- C. 应该由程序员测试自己编写的程序
- D. 严格执行测试计划，避免测试的随意性

A. 充分注意测试中的群集现象：这是正确的。测试中的群集现象指的是发现的错误往往集中出现在某些模块中，这意味着应该对这些模块投入更多测试资源。忽视这一点可能导致测试资源分配不当，从而错过大量潜在错误。

B. 设计测试用例时，应包括合理的输入条件和不合理的输入条件：这也是正确的。全面的测试策略应涵盖所有可能的输入情况，包括边界值、正常值以及异常或不合逻辑的输入，以确保软件的健壮性。

C. 应该由程序员测试自己编写的程序：这是不正确的。一般推荐遵循“程序员避免测试自己的程序”的原则，因为程序员可能受其编写代码时的思维定势影响，难以发现自己的错误。独立的测试团队或个人能以不同的视角审查代码，提高测试的有效性。

D. 严格执行测试计划，避免测试的随意性：正确。测试应该遵循明确的计划和流程，确保测试活动系统性和完整性，避免随机和无序的测试，这样才能有效地发现和追踪问题。

参考答案 (23) C

在微型计算机中，管理键盘最适合采用的 I/O 控制方式是 (24) 方式。

- (24) A. DMA B. 无条件传送 C. 程序查询 D. 中断

A. DMA (Direct Memory Access, 直接存储器访问): 这种方式适用于高速外设与内存之间大批量数据的传输，如硬盘、网卡等。键盘输入数据量小且不定时，不适合使用 DMA 方式。

B. 无条件传送: 这是一种简单的 I/O 控制方式，CPU 直接读取或写入 I/O 设备，不考虑设备是否准备好，对于键盘这类交互性很强的设备不适用，因为它无法处理设备未准备好或数据未就绪的情况。

C. 程序查询: 即轮询方式，CPU 不断循环检查外设状态，直到外设准备好进行数据传输。这种方式会占用大量的 CPU 时间，对于键盘这类低速且不可预测的输入设备来说，会造成 CPU 资源的浪费。

D. 中断: 这是管理键盘最适合的 I/O 控制方式。当用户按下键盘按键时，键盘控制器向 CPU 发送中断请求，CPU 响应后暂停当前任务，转而去处理键盘中断，读取键值，然后返回原先的任务。这种方式可以有效利用 CPU 时间，及时响应用户的输入操作，非常适合于键盘这类交互频繁且数据传输速率低的设备。

参考答案 (24) D

当一棵非空二叉树的 (25) 时，对该二叉树进行中序遍历和后序遍历所得到的序列相同。

- (25) A. 每个非叶子结点都只有左子树 B. 每个非叶子结点都只有右子树
C. 每个非叶子结点的度都为 1 D. 每个非叶子结点的度都为 2

A. 每个非叶子结点都只有左子树：在这种情况下，中序遍历会先访问左子树（即第一个孩子），然后访问根节点，最后访问空的右子树（因为没有右子树，所以实际上不访问任何东西），而后序遍历的顺序是左子树、右子树、根节点。由于右子树为空，后序遍历会在访问完左子树后直接访问根节点，因此中序遍历和后序遍历的顺序不相同。

B. 每个非叶子结点都只有右子树：这种情况下，中序遍历会先访问根节点（因为左子树不存在），然后访问右子树（即唯一的孩子），而后序遍历的顺序是左子树（空）、右子树、根节点。由于左子树总是空的，中序和后序遍历都会先访问根节点，然后再访问右子树，因此在这种特定条件下，中序遍历和后序遍历序列相同。

C. 每个非叶子结点的度都为 1：这意味着每个非叶子结点要么只有左子树，要么只有右子树，但由于这个选项同时包含两种情况，所以不能直接确定遍历序列是否相同。不过，结合前面的分析，只有当所有非叶子结点都只有右子树时，中序和后序遍历才相同。

D. 每个非叶子结点的度都为 2：这意味着每个非叶子结点都有左右两个子树。在这种情况下，中序遍历的顺序是左子树、根节点、右子树，而后序遍历的顺序是左子树、右子树、根节点。两者顺序显然不同。

参考答案 (25) A

以下关于甘特图的叙述中，不正确的是 (26)。

A. 一种进度管理的工具：这是正确的。甘特图是一种经典的项目管理工具，广泛用于展示项目进度计划，包括任务的起始时间、持续时间和完成时间。

B. 易于看出每个子任务的持续时间：这也是正确的。甘特图中，每个任务通常由一根横条表示，横条的长度直观展示了该任务的持续时间。

C. 易于看出目前项目的实际进度情况：正确。通过在甘特图中加入实际进度线，可以直观对比计划进度与实际完成情况，便于管理者了解项目的实时进展。

D. 易于初看子任务之间的衔接关系：这是不正确的。甘特图的一个主要局限在于，它不直观地展示任务之间的逻辑依赖关系。虽然可以通过横条的排列顺序和注释来一定程度上暗示任务之间的顺序，但不如网络图（如 PERT 图或 CPM 图）那样直接清晰地表现出哪些任务完成后才能开始其他任务。因此，甘特图在显示复杂项目中的任务依赖关系方面并不直观。

第5章 软件工程基础知识

291

1) Gantt 图

Gantt图能清晰地描述每个任务从何时开始，到何时结束，任务的进展情况以及各个任务之间的并行性。但是它不能清晰地反映出各任务之间的依赖关系，难以确定整个项目的关键所在，也不能反映计划中有潜力的部分。

官方教程 P291 页

参考答案 (26) D

以下关于 PERT 图的叙述中，不正确的是 (27)。

- (27) A. 易于看出每个子任务的持续时间 B. 易于看出目前项目实际进度情况
C. 易于看出子任务之间的衔接关系 D. 易于识别出关键的子任务

A. 易于看出每个子任务的持续时间：这是正确的。在 PERT 图中，每个任务通常表示为一个带有时间刻度的条形图，可以直接反映出每个任务的预期持续时间。

B. 易于看出目前项目实际进度情况：这是不正确的。虽然 PERT 图能够展示任务间的依赖关系和时间安排，但它本身并不直接显示项目的实时进度信息。要了解实际进度，通常需要结合 Gantt 图或者在 PERT 图中额外标注当前完成情况。

C. 易于看出子任务之间的衔接关系：这是正确的。PERT 图通过箭头连接任务，明确表示了任务之间的依赖关系和执行顺序，有助于理解哪些任务完成后才能开始后续任务。

D. 易于识别出关键的子任务：这也是正确的。通过计算各个路径的总时间，可以识别出项目的关键路径，即那些决定项目最短完成时间的子任务序列，这对于优先级设置和风险管理至关重要。

292 软件设计师教程（第 5 版）

2) PERT 图

PERT 图不仅给出了每个任务的开始时间、结束时间和完成该任务所需的时间，还给出了任务之间的关系，即哪些任务完成后才能开始另外一些任务，以及如期完成整个工程的关键路径。图中的松弛时间则反映了完成某些任务时可以推迟其开始时间或延长其所需完成的时间。但是，PERT 图不能反映任务之间的并行关系。

官方教程 P292 页

参考答案 (27) B

以下关于软件工程标准化的叙述中，不正确的是(28)。

- (28) A. 可以提高开发人员之间的沟通效率 B. 有助于提高管理水平
C. 有助于提高软件产品质量 D. 可以提高每一位开发人员的开发技能

A. 可以提高开发人员之间的沟通效率：这是正确的。标准化的术语、文档模板、编程规范等可以帮助团队成员更容易理解彼此的工作，从而提高沟通效率。

B. 有助于提高管理水平：正确。软件工程标准化提供了统一的流程、标准和度量方法，使得项目管理更加规范化，有利于监控项目进度、成本和质量，从而提高管理水平。

C. 有助于提高软件产品质量：这也是正确的。标准化过程确保遵循最佳实践，比如代码审查、测试规范等，这些都有助于发现并修复错误，减少缺陷，最终提升软件产品的质量。

D. 可以提高每一位开发人员的开发技能：这是不正确的。虽然标准化实践能够促进开发人员遵循良好的编程习惯和设计原则，但它并不直接提升个人的技术能力或专业知识。开发技能的提高更多依赖于专业培训、经验积累和个人学习。

参考答案 (28) D

数据库的基本表、存储文件和视图的结构分别对应 (29)。

- (29) A. 用户视图、内部视图和概念视图
- B. 用户视图、概念视图和内部视图
- C. 概念视图、用户视图和内部视图
- D. 概念视图、内部视图和用户视图

数据库的三级模式结构包括外模式 (User View, 用户视图)、模式 (Conceptual View, 概念视图) 和内模式 (Internal View, 内部视图)。

基本表对应的是模式 (Conceptual View)，也就是数据库中表的定义，它是全体用户 (或应用程序) 所看到的数据视图的总体描述，涉及到表的结构、字段、数据类型等，是数据库中表的真实存储结构的抽象描述。

存储文件对应的是内部视图 (Internal View)，指的是数据库在物理存储方面的实现细节，包括数据的实际存储方式、存储位置、索引结构等，是数据库最低层的视图，体现了数据在磁盘上的具体存储方式。

视图对应的是外模式 (User View)，它是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是基表或其他视图的特定选择与投影。视图是用户与数据库系统的接口，为用户提供了一种定制化的数据视角，隐藏了数据的复杂性，简化了数据访问。

因此，正确的选项是 D，基本表对应概念视图，存储文件对应内部视图，视图对应用户视图。其他选项的视图与结构对应关系均不正确。

参考答案 (29) D

(30) 模式可以给对象动态地添加一些额外的职责，而不改变该对象的结构。

- (30) A. 装饰 (Decorator) B. 外观 (Facade)
C. 组合 (Composite) D. 享元 (Flyweight)

A. 装饰 (Decorator) 模式：这是正确的选项。装饰模式可以在不修改原始类（对象）的基础上，通过创建一个装饰类给对象动态地添加额外的职责。装饰类包装原有的对象，并提供额外的功能，这种方式允许行为或责任被逐层添加到对象上，符合题目要求。

B. 外观 (Facade) 模式：错误。外观模式提供了一个统一的接口，用来访问子系统中的一群接口，它定义了一个高层接口，让子系统更加容易使用。它不是为了动态添加职责，而是为了简化接口。

C. 组合 (Composite) 模式：错误。组合模式是一种结构型设计模式，它将对象组合成树形结构以表示“部分-整体”的层次结构，使得用户对单个对象和组合对象的使用具有一致性。它关注的是对象的组合结构，而非动态添加职责。

D. 享元 (Flyweight) 模式：错误。享元模式是一种用于性能优化的设计模式，它通过共享技术有效支持大量细粒度的对象。其目的是减少内存中对象的数量，而不是动态添加职责。

参考答案 (30) A

在 SQL 中,结束事务通常可以使用 COMMIT 和 ROLLBACK 语句。若某事务 T 执行了 (31)。

- (31) A. ROLLBACK 语句, 则可将 T 对数据库的更新撤销
B. ROLLBACK 语句, 则可将 T 对数据库的更新写入数据库
C. COMMIT 语句, 则 T 对数据库的影响可用 ROLLBACK 语句来撤销
D. ROLLBACK 语句, 则表示 T 已正确的执行完毕

A. ROLLBACK 语句, 则可将 T 对数据库的更新撤销: 这是正确的。当执行 ROLLBACK 语句时, 会撤销当前事务 T 中对数据库所做的所有更改, 将数据库状态恢复到事务开始之前的状态, 或者最近的保存点状态。

B. ROLLBACK 语句, 则可将 T 对数据库的更新写入数据库: 这是错误的。ROLLBACK 的作用是撤销更改, 而不是提交更改到数据库。

C. COMMIT 语句, 则 T 对数据库的影响可用 ROLLBACK 语句来撤销: 这也是错误的。一旦事务 T 执行了 COMMIT 语句, 它对数据库所做的更改就会被永久保存, 无法再通过 ROLLBACK 语句撤销。

D. ROLLBACK 语句, 则表示 T 已正确的执行完毕: 这是错误的理解。ROLLBACK 语句通常是在事务未能正确执行或需要取消更改时使用, 而不是表示事务已经成功完成。

参考答案 (31) A

利用报文摘要算法生成报文摘要的目的是 (32)。

- (32) A. 防止发送的报文被篡改
B. 对传输数据进行加密，防止数据被窃听
C. 验证通信对方的身份，防止假冒
D. 防止发送方否认发送过的数据

A. 防止发送的报文被篡改：这是报文摘要算法的主要目的之一。报文摘要算法（如 MD5、SHA-1、SHA-256 等）通过对原始报文计算一个固定长度的摘要值，接收方收到报文后重新计算摘要并与发送方提供的摘要比较，如果两者一致，则说明报文在传输过程中未被篡改。

B. 对传输数据进行加密，防止数据被窃听：这是错误的。虽然报文摘要有时与加密技术结合使用以增强安全性，但报文摘要本身并不提供加密功能，其主要作用是验证数据完整性而非保密性。

C. 验证通信对方的身份，防止假冒：这通常是通过数字签名和公钥基础设施（PKI）来实现的，而不是报文摘要算法直接提供的功能。虽然报文摘要可以作为数字签名过程的一部分，但单独的报文摘要不能验证发送者的身份。

D. 防止发送方否认发送过的数据：这个目的可以通过数字签名配合报文摘要实现，因为数字签名可以提供不可否认性。但是，仅凭报文摘要本身并不能防止发送方否认，必须有发送方的私钥签名的摘要才能达到这一目的。

参考答案 (32) A

采用冒泡排序算法对序列 (49, 38, 65, 97, 76, 13, 27, 49) 进行非降序排序，两趟后的序列为 (33)。

- (33) A. (49, 38, 65, 13, 27, 49, 76, 97)
- B. (38, 49, 65, 76, 13, 27, 49, 97)
- C. (38, 49, 65, 13, 27, 49, 76, 97)
- D. (49, 38, 65, 97, 76, 13, 27, 49)

冒泡排序的基本思想是通过重复遍历待排序的数列，比较相邻元素的值，如果顺序错误就把它们交换过来。每一轮遍历后，最大的元素会被“冒泡”到数列的末尾。对于非降序排序，即从小到大排序，较大的数值会逐渐移动到序列的右侧。

第一趟排序过程：

比较 49 和 38, 49 比 38 大，进行交换，此时序列 (38, 49, 65, 97, 76, 13, 27, 49)

比较 49 和 65, 不交换；比较 65 和 97, 不交换；

比较 97 和 76, 97 比 76 大，进行交换，此时序列 (38, 49, 65, 76, 97, 13, 27, 49)

比较 97 和 13, 97 比 13 大，进行交换，此时序列 (38, 49, 65, 76, 13, 97, 27, 49)

比较 97 和 27, 97 比 27 大，进行交换，此时序列 (38, 49, 65, 76, 13, 27, 97, 49)

比较 97 和 49, 97 比 49 大，进行交换，此时序列 (38, 49, 65, 76, 13, 27, 49, 97)

第二趟排序过程：

比较 38 和 49, 不交换；比较 49 和 65, 不交换；比较 65 和 76, 不交换；

比较 76 和 13, 76 比 13 大，进行交换，此时序列 (38, 49, 65, 13, 76, 27, 49, 97)

比较 76 和 27, 76 比 27 大，进行交换，此时序列 (38, 49, 65, 13, 27, 76, 49, 97)

比较 76 和 49, 76 比 49 大，进行交换，此时序列 (38, 49, 65, 13, 27, 49, 76, 97)

比较 76 和 97, 不交换；

两趟排序后的序列为：(38, 49, 65, 13, 27, 49, 76, 97)

参考答案 (33) C

采用简单选择排序算法对序列 (34, 12, 49, 28, 31, 52, 51, 49) 进行非降序排序，两趟后的序列为 (34)。

- (34) A. (12, 28, 49, 34, 31, 52, 51, 49)
B. (12, 28, 34, 49, 31, 52, 51, 49)
C. (12, 28, 31, 49, 34, 52, 51, 49)
D. (34, 12, 49, 28, 31, 49, 51, 52)

简单选择排序算法的基本思想是遍历数组，每次从未排序的部分找到最小(或最大)的元素，将其放到已排序序列的末尾。对于非降序排序，即寻找当前未排序部分中的最小值。

第一趟排序过程：第一趟排序将最小的元素归位到第一个位置

比较 34 和 12，12 比 34 小，此时 12 是最小元素。之后 12 均小于后面的所有元素；
则 12 为整个未排序序列最小元素，并和第一个位置的元素交换。也就是 34 和 12 交换位置。
则第一趟排序后的序列为：(12, 34, 49, 28, 31, 52, 51, 49)

同理，从第二个元素开始查找第二小的元素。

比较 34 和 49，34 小于 49；

比较 34 和 28，28 小于 34，更新未排序序列中最小元素为 28。

之后 28 均小于后面的所有元素，则 28 为整个未排序序列最小元素，并和第二个位置的元素交换。也就是 34 和 28 交换位置。

则第二趟排序后的序列为：(12, 28, 49, 34, 31, 52, 51, 49)

两趟排序后的序列为：(12, 28, 49, 34, 31, 52, 51, 49)

参考答案 (34) A

在设计模块 M 和模块 N 时, _(35)_ 是最佳的设计。

- (35) A. M 和 N 通过通信模块传送数据
- B. M 和 N 通过公共数据域传送数据
- C. M 和 N 通过简单数据参数交换信息**
- D. M 直接访问 N 的数据

A. M 和 N 通过通信模块传送数据: 虽然使用通信模块可以实现模块间的数据传递, 但这通常适用于分布式系统或需要复杂通信协议的情况, 对于大多数常规的软件模块设计而言, 这可能引入不必要的复杂度。(外部耦合)

B. M 和 N 通过公共数据域传送数据: 这种做法通常被认为是不良设计, 因为它增加了模块间的耦合度。当多个模块直接访问同一块数据时, 可能会导致数据同步问题、数据不一致性, 以及维护困难等问题。(公共耦合)

C. M 和 N 通过简单数据参数交换信息: 这是推荐的最佳设计方式。通过函数或方法调用传递必要的数据作为参数, 可以保持模块间的独立性, 降低耦合度, 使得每个模块更易于理解、测试和维护。(数据耦合)

D. M 直接访问 N 的数据: 这同样违反了模块间的封装原则, 直接访问另一个模块的数据会导致模块间的紧密耦合, 影响模块的复用性和系统的稳定性。(内部耦合)

因此, 选项 C 是最佳设计, 因为它遵循了模块设计中的低耦合、高内聚原则, 有利于提高软件的可维护性和可扩展性。

参考答案 (35) C

程序员甲将其编写完成的软件程序发给同事乙并进行讨论，之后由于甲对该程序极不满意，因此甲决定放弃该程序，后来乙将该程序稍加修改并署自己名在某技术论坛发布。下列说法中，正确是 (36)。

- (36) A. 乙对该程序进行了修改，因此乙享有该程序的软件著作权
B. 乙的行为没有侵犯甲的软件著作权，因为甲已放弃程序
C. 乙的行为未侵权，因其发布的场合是以交流学习为目的的技术论文
D. 乙的行为侵犯了甲对该程序享有的软件著作权

A. 乙对该程序进行了修改，因此乙享有该程序的软件著作权：这个选项是错误的。尽管乙对程序进行了修改，但原程序的著作权仍属于甲。修改他人作品仅在特定条件下（如遵循开源许可协议）可能不构成侵权，但并不直接赋予修改者原始著作权。

B. 乙的行为没有侵犯甲的软件著作权，因为甲已放弃程序：这个选项也是错误的。即便甲对程序不满意并打算放弃，著作权的转让或放弃需要有明确的法律行为或协议，简单地表达不满或放弃意图并不等于法律意义上的放弃著作权。

C. 乙的行为未侵权，因其发布的场合是以交流学习为目的的技术论文：这个选项错误。发布场合的性质（即便是交流学习）并不能改变未经许可使用他人作品的侵权本质，除非有适用的版权例外条款，如合理使用，但即便是合理使用也有严格限制条件，通常不包括署名发表。

D. 乙的行为侵犯了甲对该程序享有的软件著作权：这是正确的。甲作为程序的原创作者，拥有该软件的著作权。乙未获得甲的授权就擅自修改并以自己的名义发布，侵犯了甲的署名权、修改权及可能的发行权等著作权权利。

参考答案 (36) D

软件文档在软件生存期中起着重要的作用，其作用不包括(37)。

- (37) A. 提高软件运行效率
B. 作为开发过程的阶段工作成果和结束标记
C. 提高开发过程的能见度
D. 提高开发效率

A. 提高软件运行效率：这是不正确的。软件文档主要关注于记录、描述和规定软件的开发过程、设计、功能、使用方法等内容，并不直接影响软件的运行效率。软件的运行效率主要由程序代码的质量、算法设计、资源管理等因素决定。

B. 作为开发过程的阶段工作成果和结束标记：这是正确的。软件文档在软件开发的各个阶段（需求分析、设计、编码、测试等）都是重要产出物，标志着各阶段工作的完成，并作为后续工作或阶段的输入。

C. 提高开发过程的能见度：这也是正确的。良好的文档记录可以帮助项目管理者和团队成员了解项目的进度、设计决策、问题解决方案等，增加开发过程的透明度，便于沟通和协作。

D. 提高开发效率：正确。虽然直接效果不那么明显，但长期来看，高质量的文档可以减少沟通成本，帮助开发者更快地理解代码结构和设计思路，避免重复劳动，从而提高开发效率。

参考答案 (37) A

数据库概念结构设计阶段的工作步骤包括①~④，其正确的顺序为(38)。

- ① 设计局部视图
- ② 抽象数据
- ③ 修改重构消除冗余
- ④ 合并取消冲突

- (38) A. ①→②→④→③ B. ①→②→③→④
C. ②→①→③→④ D. ②→①→④→③

数据库概念结构设计阶段的工作步骤大致如下：

② 抽象数据：这是概念设计的起点，涉及从需求分析中得到的用户需求进行综合、归纳与抽象，形成对现实世界的概念化表示，通常是通过实体、属性和关系来描述。

① 设计局部视图：在抽象出基本的概念实体和关系后，设计者可能会为不同的用户群体或系统功能设计局部的概念视图，每个视图反映了特定视角下的数据需求。

④ 合并取消冲突：接下来，需要将这些局部视图整合成一个全局的概念模型，此过程涉及解决不同视图间可能存在的冗余、不一致或冲突，确保全局模型的逻辑一致性。

③ 修改重构消除冗余：在得到初步的全局模型后，进一步分析模型，识别并消除不必要的冗余，优化模型结构，确保模型既简洁又能满足所有已识别的需求。

因此，正确的顺序是先进行数据抽象（②），然后设计局部视图（①），接着合并并解决冲突（④），最后修改重构以消除冗余（③），选项 D 正确。

其他选项的错误在于它们没有正确反映这一设计流程的逻辑顺序。例如，A、B、C 选项中，局部视图的设计被置于数据抽象之前或同时进行，这与通常的设计流程不符，因为设计局部视图应在对数据有了一定抽象理解之后进行。

参考答案 (38) D

试题一至试题四为必答题

试题一（共 15 分）

阅读下列说明和图，回答问题 1 至问题 4，将解答填入对应栏内。

【说明】

随着深度学习的广泛应用，现代聊天机器人系统需要大规模的训练数据集才能达到其最佳性能，而手动收集如此庞大的数据集需要耗费巨大的人力和时间成本。现欲开发一众包信息系统来辅助收集训练数据集，其主要功能是：

- (1) 用户管理：众包工作者提供角色和标识，并存储在用户表中。
- (2) 添加问题：在不同情况下接收来自众包工作者和管理员输入的问题；众包工作者输入问题建议，管理员负责添加初始问题。将问题和问题类别分别进行存储。问题类别说明问题是众包工作者还是管理员提供的。
- (3) 答复问题：众包工作者回答或拒绝系统随机展示的 5 个问题。答复流程是：如果回答问题则提供答案；如果拒绝问题则提供拒绝原因；如果回答问题数不足 5 个，继续展示问题，否则众包工作者提供问题建议。无论是回答还是拒绝，数据都存储在带有不同状态标记的答复表中。
- (4) 数据服务：根据其它训练平台的请求，为其提供问题、问题类别、回复的数据集。

现采用结构化方法对众包信息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。

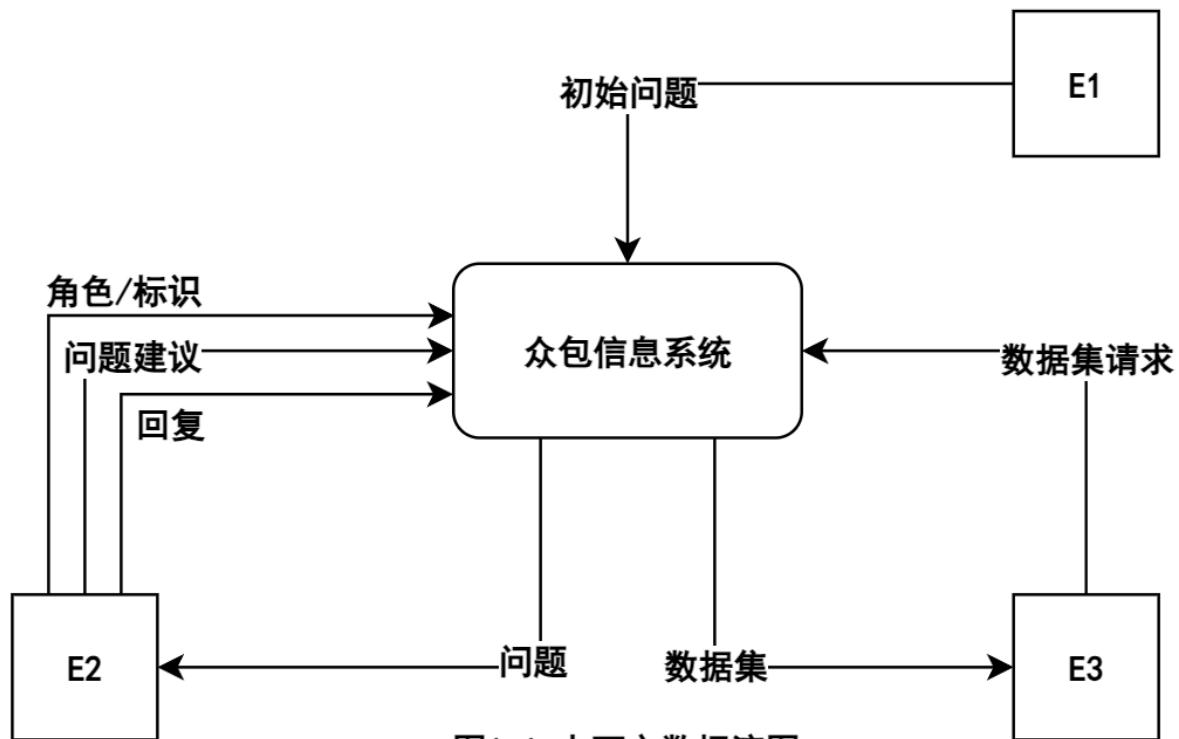


图1-1 上下文数据流图

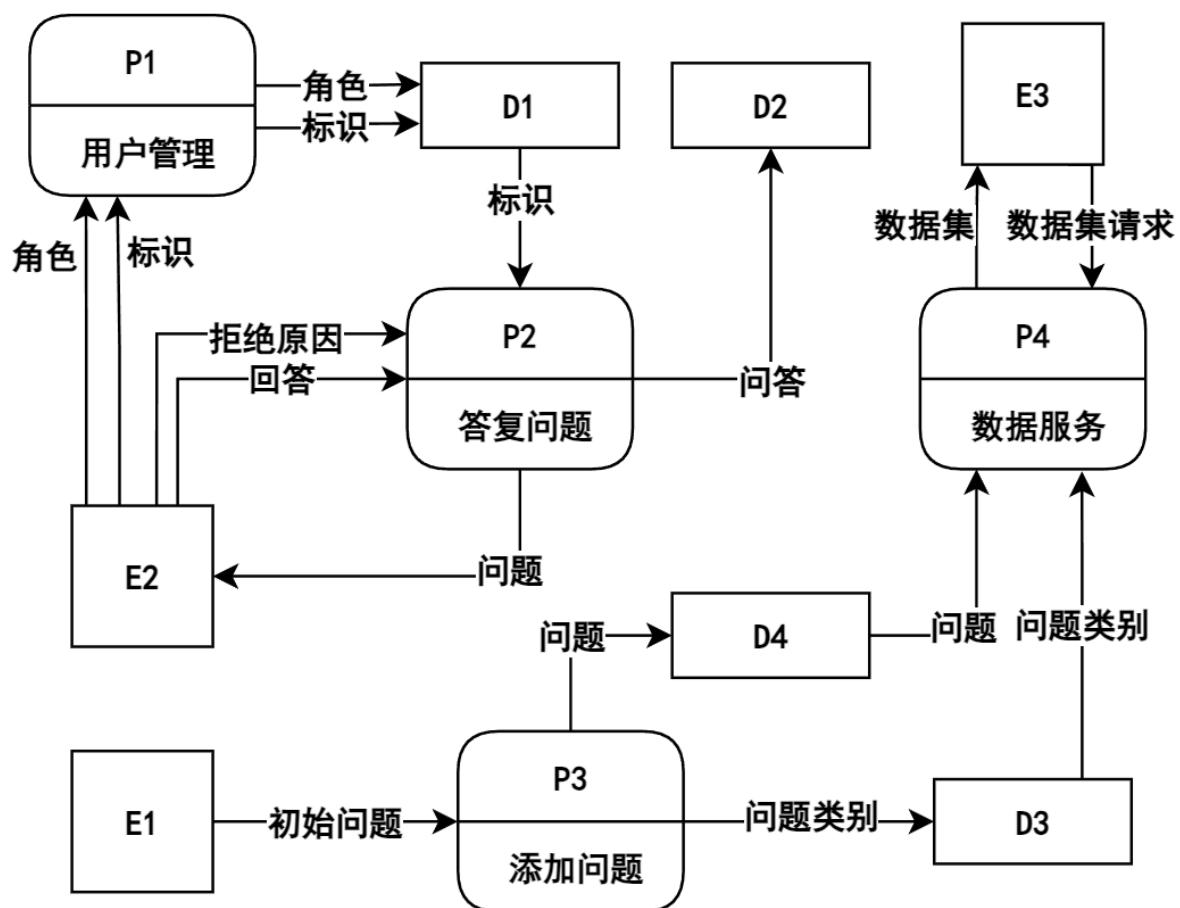


图1-2 0层数据流图

【问题 1】(3 分)

使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

E1: 管理员

E2: 众包工作者

E3: 其它训练平台

【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。

自拟的答案用表、信息表、记录、文件等等关键字均可

D1: 用户表

D2: 答复表

D3: 问题类别表

D4: 问题表

【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。

首先考试应该只给了 4 行作答，也就是缺失 4 条数据流。

如果你在考试的时候也找到了超过表格行数的数据流，那么就选择概率最高的填入。

首选答案：

数据流名称	起点	终点
问题建议	E2	P2
问题建议	E2	P3
回复	D2	P4
问题	D4	P2

次选答案：

数据流名称：标识

起点：E2

终点：P2

【问题 4】(4 分)

请说明什么是分层数据流图中父图与子图的平衡？如何保持。

该问题与（2012年上半年下午题试题一问题四）相似，以下是相似问题的官方解析和答案。

问题**【问题 4】(2 分)**

说明问题 3 中绘制 1 层数据流图时要注意的问题。

解析**【问题 4】**

本问题考查在绘制下层数据流图时需要注意的问题。问题 3 明确给出是对复杂处理进行进一步精化，绘制下层数据流图，因此需要注意的问题是绘制下层数据流图时要保持父图与子图平衡。父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同。如果父图的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

答案**【问题 4】**

保持父图与子图平衡。父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同。如果父图的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

父图子图平衡：

父图的数据流要和子图的数据流

在数量、名称和方向上保持一致。

如何保持：

首先要保证基本的数量、名称和方向上一致，

其次就是父图中的一条数据流可以对应子图中

若干条数据流，只要子图中的这些数据流全体是

父图中的这一个数据流，那么也可以保持平衡。

试题二（共 15 分）

阅读下列说明，回答问题 1 至问题 3，将解答填入对应栏内。

【说明】

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。

【需求分析结果】

(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个快递员，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。

【概念结构设计】

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。

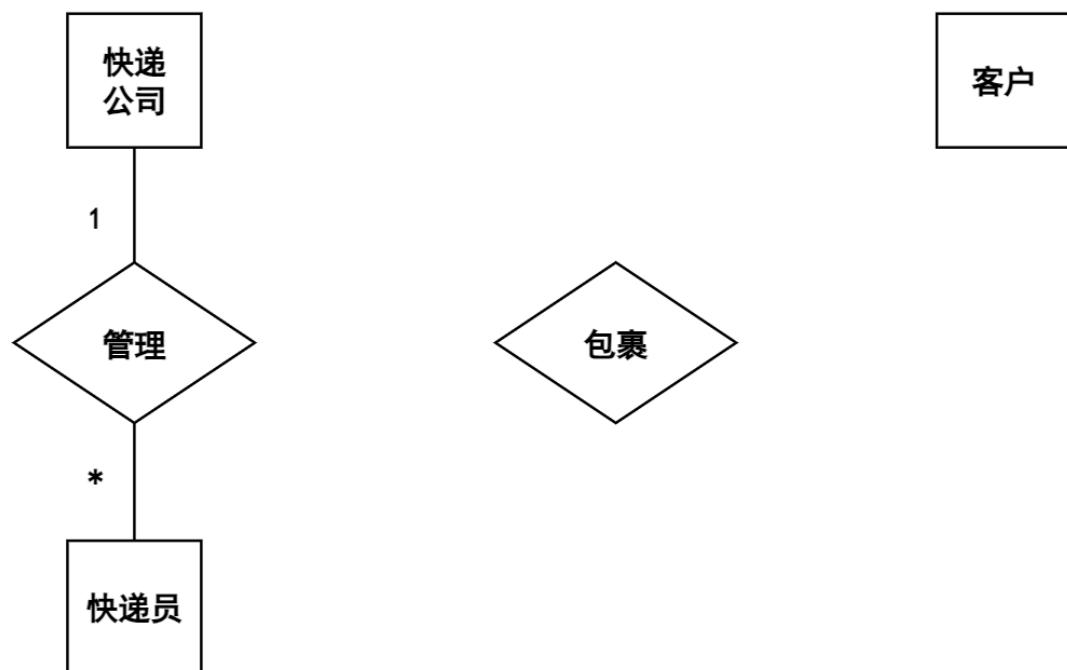


图2-1 实体联系图

【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）：

快递公司（公司名称，地址，电话）

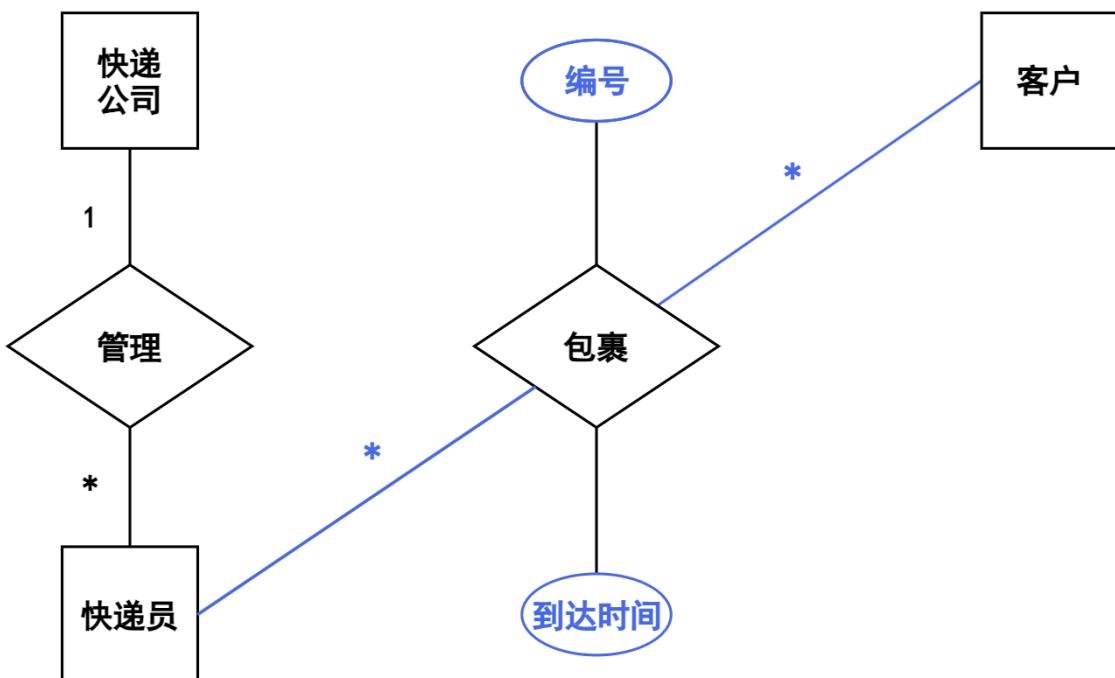
快递员（姓名，快递员手机号码，(a)）

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）

【问题1】（6分）

根据问题描述，补充图2-1的实体联系图。



问题一 补充后的 图2-1 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。

(a): 公司名称;

(b): 客户手机号码

快递员关系外键：公司名称；

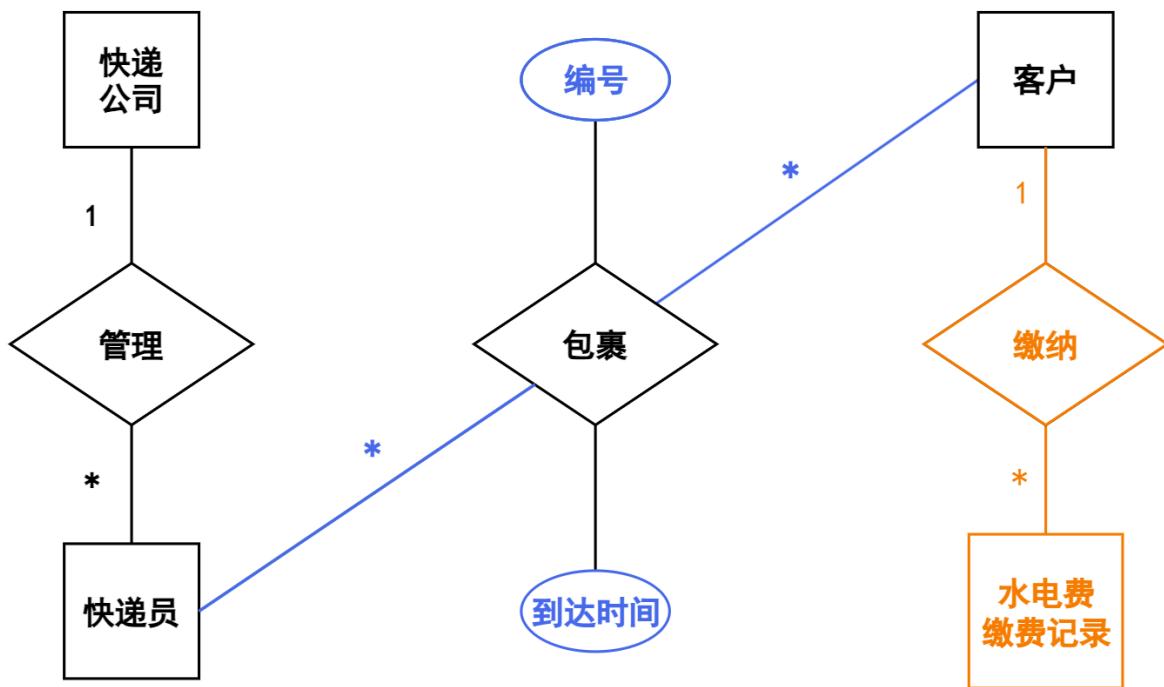
包裹关系外键：客户手机号码

用下划线和虚线分别标记主键和外键或者直接写出来都可以，机考应该是给文本输入框的。

【问题3】 (5分)

若快递驿站还兼有缴水电费业务，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 2-1 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。



问题三 补充后的 图2-1 实体联系图
(问题一的内容画和不画都可以)

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: 客户手机号码

用下划线和虚线分别标记主键和外键或者直接写出来都可以，机考应该是给文本输入框的。

试题三（共 15 分）

题目缺失

试题四（共 15 分）

题目缺失

从下列的 2 道试题（试题五至试题六）中任选 1 道解答。

试题五（共 15 分）

题目缺失

试题六（共 15 分）

题目缺失