

【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

计算机系统中硬件层之上的软件通常按照三层来划分，如下图所示，图中①②③分别表示（1）。

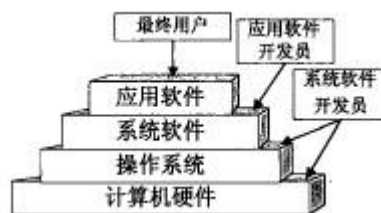


- (1) A. 操作系统、应用软件和其他系统软件 B. 操作系统、其他系统软件和应用软件
C. 其他系统软件、操作系统和应用软件 D. 应用软件、其他系统软件和操作系统

【答案】B

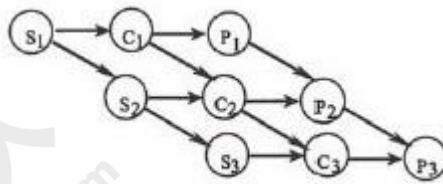
【解析】本题考查计算机系统中软件方面的基本知识。

操作系统（Operating System）的目的是为了填补人与机器之间的鸿沟，即建立用户与计算机之间的接口，而为裸机配置的一种系统软件，如下图所示。



从上图可以看出，操作系统是裸机上的第一层软件，是对硬件系统功能的首次扩充。它在计算机系统中占据重要而特殊的地位，其他系统软件属于第二层，如编辑程序、汇编程序、编译程序和数据库管理系统等系统软件；大量的应用软件属于第三层，例如银行账务查询、股市行情和机票预定系统等。其他系统软件和应用软件都是建立在操作系统基础之上的，并得到它的支持和取得它的服务。从用户角度看，当计算机配置了操作系统后，用户不再直接使用计算机系统硬件，而是利用操作系统所提供的命令和服务去操纵计算机，操作系统已成为现代计算机系统中必不可少的最重要的系统软件，因此把操作系统看作是用户与计算机之间的接口。

某计算机系统有一个 CPU、一台扫描仪和一台打印机。现有三个图像处理任务，每个任务有三个程序段：扫描 S_i ，图像处理 Q 和打印 P_i ($i=1, 2, 3$)。下图为三个任务各程序段并发执行的前驱图，其中，(2)可并行执行，(3)的直接制约，(4)的间接制约。



- (2) A. “C1S2”, “P1C2S3”, “P2C3” B. “C1S1”, “S2C2P2”, “C3P3”
 C. “S1C1P1”, “S2C2P2”, “S3C3P3” D. “S1S2S3”, “C1C2C3”, “P1P2P3”
- (3) A. S1 受到 S2 和 S3、C1 受到 C2 和 C3、P1 受到 P2 和 P3
 B. S2 和 S3 受到 S1、C2 和 C3 受到 C1、P2 和 P3 受到 P1
 C. C1 和 P1 受到 S1、C2 和 P2 受到 S2、C3 和 P3 受到 S3
 D. C1 和 S1 受到 P1、C2 和 S2 受到 P2、C3 和 S3 受到 P3
- (4) A. S1 受到 S2 和 S3、C1 受到 C2 和 C3、P1 受到 P2 和 P3
 B. S2 和 S3 受到 S1、C2 和 C3 受到 C1、P2 和 P3 受到 P1
 C. C1 和 P1 受到 S1、C2 和 P2 受到 S2、C3 和 P3 受到 S3
 D. C1 和 S1 受到 P1、C2 和 S2 受到 P2、C3 和 S3 受到 P3

【答案】A C B

【解析】本题考查操作系统多道程序设计中的基础知识。

(2) 前趋图是一个有向无循环图，图由结点和结点间的有向边组成，结点代表各程序段的操作，而结点间的有向边表示两程序段操作之间存在的前趋关系（“→”）。两程序段 P_i 和 P_j 的前趋关系表示成 $P_i \rightarrow P_j$ ，其中 P_i 是 P_j 的前趋， P_j 是 P_i 的后继，其含义是 P_i 执行完毕才能由 P_j 执行。可见， S_1 执行完毕后，计算 C_1 与扫描 S_2 可并行执行； C_1 与 S_2 执行完毕后，打印 P_1 、计算 C_2 与扫描 S_3 可并行执行； P_1 、 C_2 与 S_3 执行完毕后，打印 P_2 与计算 C_3 可并行执行。

(3) 根据题意，系统中有三个任务，每个任务有三个程序段，从前趋图中可以看出，系统要先进行扫描 S_i ，然后再进行图像处理 C_i ，最后进行打印 P_i ，所以 C_1 和 P_1 受到 S_1 的直接制约、 C_2 和 P_2 受到 S_2 的直接制约、 C_3 和 P_3 受到 S_3 的直接制约。

(4) 根据题意，系统中有一台扫描仪，因此 S_2 和 S_3 不能运行是受到了 S_1 的间接制约，如果系统中有三台扫描仪，那么 S_2 和 S_3 能运行；同理， C_2 和 C_3 受到 C_1 的直接制约、 P_2 和 P_3 受到 P_1 的间接制约。

在数据库设计的需求分析阶段应完成包括 (5) 在内的文档。

- (5) A. E-R 图 B. 关系模式 C. 数据字典和数据流图 D. 任务书和设计方案

【答案】C

【解析】本题考查数据库设计方面的相关知识。

需求分析阶段的任务是对现实世界要处理的对象（组织、部门和企业等）进行详细调查，在了解现行系统的概况，确定新系统功能的过程中收集支持系统目标的基础数据及处理方法。需求分析是在用户调查的基础上，通过分析，逐步明确用户对系统的需求，包括数据需求和围绕这些数据的业务处理需求，以及对数据安全性和完整性方面的要求。在需求分析阶段应完成的文档是数据字典和数据流图。

设有职务工资关系 P（职务，最低工资，最高工资），员工关系 EMP（员工号，职务，工资），要求任何一名员工，其工资值必须在其职务对应的工资范围之内，实现该需求的方法是（6）。

- (6) A. 建立“EMP. 职务”向“P. 职务”的参照完整性约束
 B. 建立“P. 职务”向“EMP. 职务”的参照完整性约束
 C. 建立 EMP 上的触发器程序审定该需求
 D. 建立 P 上的触发器程序审定该需求

【答案】C

【解析】本题考查对数据完整性约束方面基础知识的掌握。

完整性约束分为实体完整性约束、参照完整性约束和用户自定义完整性约束三类。其中实体完整性约束可以通过 Primary Key 指定，参照完整性约束通过 Foreign Key 指定，某些简单的约束可以通过 Check、Assertion 等实现。针对复杂的约束，系统提供了触发器机制，通过用户编程实现。本题中的约束条件只能通过编写职工表上的触发器，在对工资进行修改或插入新记录时触发，将新工资值与工资范围表中职工职务对应的工资范围比对，只有在范围内才提交，否则回滚。

设关系模式 R(U, F)，其中 R 上的属性集 $U = \{A, B, C, D, E\}$ ，R 上的函数依赖集 $F = \{A \rightarrow B, DE \rightarrow B, CB \rightarrow E, E \rightarrow A, B \rightarrow D\}$ 。（7）为关系 R 的候选关键字。分解（8）是无损连接，并保持函数依赖的。

- (7) A. AB B. DE C. CE D. DB

- (8) A. $\rho = \{R_1(AC), R_2(ED), R_3(B)\}$ B. $\rho = \{R_1(AC), R_2(E), R_3(DB)\}$

C. $p = \{R_1(AC), R_2(ED), R_3(AB)\}$

D. $p = \{R_1(ABC), R_2(ED), R_3(ACE)\}$

【答案】C D

【解析】本题考查如何求解候选关键字和对模式分解知识的掌握。

给定一个关系模式 $R(U, F)$, $U = \{A_1, A_2, \dots, A_n\}$, F 是 R 的函数依赖集, 若 $X_F^+ = U$, 则 X 必为 R 的唯一候选关键字。对于试题 (7), A 选项 $(AB)_F^+ = ABD \neq U$, 所以 AB 非候选关键字; $(DE)_F^+ = ABDE \neq U$ 所以 DE 非候选关键字; C 选项 $(DB)_F^+ = BD \neq U$, 所以 CE 为候选关键字; D 选项所以 DB 非候选关键字。

根据无损连接的判定算法, 对于选项 A 的构造初始的判定表如下:

分解的关系模式	A	B	C	D	E
$R_1(AC)$	a_1	b_{12}	a_3	b_{14}	b_{15}
$R_2(ED)$	b_{21}	b_{22}	b_{23}	a_4	a_5
$R_3(B)$	b_{31}	a_2	b_{33}	b_{34}	b_{35}

由于 $A \rightarrow B$, $DE \rightarrow B$, $CB \rightarrow E$, $E \rightarrow A$, $B \rightarrow D$ 的决定因素中没有两行是相同的, 因此选项 A 是有损连接的。

对选项 B 构造初始的判定表如下:

分解的关系模式	A	B	C	D	E
$R_1(AC)$	a_1	b_{12}	a_3	b_{14}	b_{15}
$R_2(E)$	b_{21}	b_{22}	b_{23}	b_{24}	a_5
$R_3(DB)$	b_{31}	a_2	b_{33}	a_4	b_{35}

由于 $A \rightarrow B$, $DE \rightarrow B$, $CB \rightarrow E$, $E \rightarrow A$, $B \rightarrow D$ 的决定因素中没有两行是相同的, 因此选项 B 是有损连接的。

对选项 C 构造初始的判定表如下:

分解的关系模式	A	B	C	D	E
$R_1(AC)$	a_1	b_{12}	a_3	b_{14}	b_{15}
$R_2(ED)$	b_{21}	b_{22}	b_{23}	a_4	a_5
$R_3(AB)$	a_1	a_2	b_{33}	b_{34}	b_{35}

由于 $A \rightarrow B$, 属性 A 的第 1 行和第 3 行相同, 可以将第 1 行 b_{12} 改为 a_2 ; 又由于 $B \rightarrow D$, 属性 B 的第 1 行和第 3 行相同, 而属性 D 第 1 行 b_{14} 和第 3 行 b_{34} 没有一行为 a_4 , 因此改为同一符号, 即取行号值最小的 b_{14} 。修改后的判定表如下:

分解的关系模式	A	B	C	D	E
R ₁ (AC)	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
R ₂ (ED)	b ₂₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (AB)	a ₁	a ₂	b ₃₃	b ₁₄	b ₃₅

反复检查函数依赖集 F，无法修改上表，所以选项 C 是有损连接的。对选项 D 构造初始的判定表如下：

分解的关系模式	A	B	C	D	E
R ₁ (ABC)	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
R ₂ (ED)	b ₂₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (ACE)	a ₁	b ₃₂	a ₃	b ₃₄	a ₅

由于 A→B，属性 A 的第 1 行和第 3 行相同，可以将第 3 行 b₃₂ 改为 a₂；E→A，属性 E 的第 2 行和第 3 行相同，可以将属性 A 第 2 行 b₂₁ 改为 a₁；AC→E，属性 E 的第 2 行和第 3 行相同，可以将属性 E 第 1 行 b₁₅ 改为 a₅；B→D，属性 B 的第 1 行和第 3 行相同，属性 D 第 1 行 b₁₄ 和第 3 行 b₃₄ 没有一行为 a₄，因此改为同一符号，即取行号值最小的 b₁₄。修改后的判定表如下：

分解的关系模式	A	B	C	D	E
R ₁ (ABC)	a ₁	a ₂	a ₃	b ₁₄	a ₅
R ₂ (ED)	a ₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (ACE)	a ₁	a ₂	a ₃	b ₃₄	a ₅

由于 E→D，属性 E 的第 1~3 行相同。可以将属性 D 第 1 行 b₁₄ 和第 3 行 b₃₄ 改为 a₄，修改后的判定表如下：

分解的关系模式	A	B	C	D	E
R ₁ (ABC)	a ₁	a ₂	a ₃	a ₄	a ₅
R ₂ (ED)	a ₁	b ₂₂	b ₂₃	a ₄	a ₅
R ₃ (ACE)	a ₁	a ₂	a ₃	a ₄	a ₅

由于上表第一行全为 a，故分解无损。

现在分析该分解是否保持函数依赖。若分解保持函数依赖，那么分解的子模式的函数依

赖集 $F^+ = \left(\bigcup_{R_i \in \rho} F_{R_i}^+ \right)^+$ 。 $F_{R_1} = A \rightarrow B, CB \rightarrow A$ ， $F_{R_2} = E \rightarrow D$ （根据 Armstrong 公理，系统传递依赖，E→A，A→B，B→D，所以 E→D）， $F_{R_3} = E \rightarrow A$ 。可以求证 F^+ 与 $(F_{R_1} + F_{R_2} + F_{R_3})^+$ 等价，即 $F^+ = (F_{R_1} + F_{R_2} + F_{R_3})^+ = (A \rightarrow B, CB \rightarrow A, E \rightarrow D, E \rightarrow A)^+$ ，所以该分解保持函数依赖。

嵌入式系统中采用中断方式实现输入输出的主要原因是 (9)。在中断时，CPU 断点信息一般保存到 (10) 中。

- (9) A. 速度最快 B. CPU 不参与操作
C. 实现起来比较容易 D. 能对突发事件做出快速响应
- (10) A. 通用寄存器 B. 堆 C. 栈 D. I/O 接口

【答案】D C

【解析】本题主要考查嵌入式系统中断的基础知识。

嵌入式系统中采用中断方式实现输入输出的主要原因是能对突发事件做出快速响应。在中断时，CPU 断点信息一般保存到栈中。

在嵌入式系统设计时，下面几种存储结构中对程序员是透明的是 (11)。

- (11) A. 高速缓存 B. 磁盘存储器 C. 内存 D. flash 存储器

【答案】A

【解析】本题主要考查嵌入式系统程序设计中存储结构的操作。

对照 4 个选项，可以立即看出高速缓存 (Cache) 对于程序员来说是透明的。

系统间进行异步串行通信时，数据的串/并和并/串转换一般是通过 (12) 实现的。

- (12) A. I/O 指令 B. 专用的数据传送指令
C. CPU 中有移位功能的数据寄存器 D. 接口中的移位寄存器

【答案】D

【解析】本题主要考查嵌入式系统间进行异步串行通信时数据的串/并和并/串转换方式。

一般来说，嵌入式系统通常采用接口中的移位寄存器来实现数据的串/并和并/串转换操作。

以下关于网络核心层的叙述中，正确的是 (13)。

- (13) A. 为了保障安全性，应该对分组进行尽可能多的处理
B. 在区域间高速地转发数据分组
C. 由多台二、三层交换机组成
D. 提供多条路径来缓解通信瓶颈

【答案】B

【解析】

三层模型主要将网络划分为核心层、汇聚层和接入层，每一层都有着特定的作用：核心层提供不同区域或者下层的高速连接和最优传送路径；汇聚层将网络业务连接到接入层，并且实施与安全、流量负载和路由相关的策略；接入层为局域网接入广域网或者终端用户访问网络提供接入。其中核心层是互连网络的高速骨干，由于其重要性，因此在设计中应该采用冗余组件设计，使其具备高可靠性，能快速适应变化。

在设计核心层设备的功能时，应尽量避免使用数据包过滤、策略路由等降低数据包转发处理的特性，以优化核心层获得低延迟和良好的可管理性。

核心层应具有有限的和一致的范围，如果核心层覆盖的范围过大，连接的设备过多，必然引起网络的复杂度加大，导致网络管理性降低；同时，如果核心层覆盖的范围不一致，必然导致大量处理不一致情况的功能都在核心层网络设备中实现，会降低核心网络设备的性能。对于那些需要连接因特网和外部网络的网络工程来说，核心层应包括一条或多条连接到外部网络的连接，这样可以实现外部连接的可管理性和高效性。

网络开发过程中，物理网络设计阶段的任务是(14)。

- (14)A. 依据逻辑网络设计的功能要求，确定设备的具体物理分布和运行环境
- B. 分析现有网络和新网络各类资源分布，掌握网络所处状态
- C. 根据需求规范和通信规范，实施资源分配和安全规划
- D. 理解网络应该具有的功能和性能，最终设计出符合用户需求的网络

【答案】A

【解析】

网络的生命周期至少包括网络系统的构思计划、分析设计、实时运行和维护的过程。对于大多数网络系统来说，由于应用的不断发展，这些网络系统需要不断重复设计、实施、维护的过程。

网络逻辑结构设计是体现网络设计核心思想的关键阶段，在这一阶段根据需求规范和通信规范，选择一种比较适宜的网络逻辑结构，并基于该逻辑结构实施后续的资源分配规划、安全规划等内容。

物理网络设计是对逻辑网络设计的物理实现，通过对设备的具体物理分布、运行环境等的确定，确保网络的物理连接符合逻辑连接的要求。在这一阶段，网络设计者需要确定具体的软硬件、连接设备、布线和服务。

现有网络体系分析的工作目的是描述资源分布，以便于在升级时尽量保护已有投资，通过该工作可以使网络设计者掌握网络现在所处的状态和情况。

需求分析阶段有助于设计者更好地理解网络应该具有什么功能和性能，最终设计出符合用户需求的网络，它为网络设计提供依据。

某公司欲构建一个网络化的开放式数据存储系统，要求采用专用网络连接并管理存储设备和存储管理子系统。针对这种应用，采用 (15) 存储方式最为合适。

- (15) A. 内置式存储 B. DAS C. SAND D. NAS

【答案】C

【解析】

开放系统的直连式存储 (Direct-Attached Storage, DAS) 在服务器上外挂了一组大容量硬盘，存储设备与服务器主机之间采用 SCSI 通道连接，带宽为 10MB/s、20MB/S、40MB/S 和 80MB/S 等。直连式存储直接将存储设备连接到服务器上，这种方法难以扩展存储容量，而且不支持数据容错功能，当服务器出现异常时会造成数据丢失。

网络接入存储 (Network Attached Storage, NAS) 是将存储设备连接到现有的网络上，提供数据存储和文件访问服务的设备。NAS 服务器是在专用主机上安装简化了的瘦操作系统（只具有访问权限控制、数据保护和恢复等功能）的文件服务器。NAS 服务器 内置了与网络连接所需要的协议，可以直接联网，具有权限的用户都可以通过网络访问 NAS 服务器中的文件。

存储区域网络 (Storage Area Network, SAN) 是一种连接存储设备和存储管理子系统的专用网络，专门提供数据存储和管理功能。SAN 可以被看作是负责数据传输的后端网络，而前端网络（或称为数据网络）则负责正常的 TCP/IP 传输。也可以把 SAN 看作是通过特定的互连方式连接的若干台存储服务器组成的单独的数据网络，提供企业级的数据存储服务。

以下关于基准测试的叙述中，正确的是 (16)。

- (16) A. 运行某些诊断程序，加大负载，检查哪个设备会发生故障
B. 验证程序模块之间的接口是否正常起作用
C. 运行一个标准程序对多种计算机系统进行检查，以比较和评价它们的性能
D. 根据程序的内部结构和内部逻辑，测试该程序是否正确

【答案】C

【解析】

各种类型的计算机都具有自己的性能指标，计算机厂商当然希望自己研制的计算机有较高的性能。同样的计算机，如果采用不同的评价方法，所获得的性能指标也会不同。因此，用户希望能有一些公正的机构采用公认的评价方法来测试计算机的性能。这样的测试称为基准测试，基准测试采用的测试程序称为基准程序(Benchmark)基准程序就是公认的标准程序，用它来测试多种计算机系统，比较和评价它们的性能，定期公布测试结果，供用户选购计算机时参考。

对计算机进行负载测试就是运行某种诊断程序，加大负载，检查哪个设备会发生故障。在程序模块测试后进行的集成测试，主要测试各模块之间的接口是否正常起作用。白盒测试就是根据程序内部结构和内部逻辑，测试其功能是否正确。

以下关于计算机性能改进的叙述中，正确的是 (17)。

- (17) A. 如果某计算机系统的 CPU 利用率已经接近 100%，则该系统不可能再进行性能改进
- B. 使用虚存的计算机系统如果主存太小，则页面交换的频率将增加，CPU 的使用效率就会降低，因此应当增加更多的内存
- C. 如果磁盘存取速度低，引起排队，此时应安装更快的 CPU，以提高性能
- D. 多处理机的性能正比于 CPU 的数目，增加 CPU 是改进性能的主要途径

【答案】B

【解析】

计算机运行一段时间后，经常由于应用业务的扩展，发现计算机的性能需要改进。计算机性能改进应针对出现的问题，找出问题的瓶颈，再寻求适当的解决方法。

计算机的性能包括的面很广，不单是 CPU 的利用率。即使 CPU 的利用率已经接近 100%，这只说明目前计算机正在运行大型计算任务。其他方面的任务可能被外设阻塞着，而改进外设成为当前必须解决的瓶颈问题。

如果磁盘存取速度低，则应增加新的磁盘或更换使用更先进的磁盘。安装更快的 CPU 不能解决磁盘存取速度问题。

多处理机的性能并不能正比于 CPU 的数目，因为各个 CPU 之间需要协调，需要花费一定的开销。

使用虚存的计算机系统如果主存太小，则主存与磁盘之间交换页面的频率将增加，业务

处理效率就会降低，此时应当增加更多的内存。这就是说，除 CPU 主频外，内存大小对计算机实际运行的处理速度也密切相关。

商业智能是指利用数据挖掘、知识发现等技术分析和挖掘结构化的、面向特定领域的存储与数据仓库的信息。它可以帮助用户认清发展趋势、获取决策支持并得出结论。以下(18)活动,并不属于商业智能范畴。

- (18)A. 某大型企业通过对产品销售数据进行挖掘，分析客户购买偏好
B. 某大型企业查询数据仓库中某种产品的总体销售数量
C. 某大型购物网站通过分析用户的购买历史记录，为客户进行商品推荐
D. 某银行通过分析大量股票交易的历史数据，做出投资决策

【答案】B

【解析】本题主要考查商业智能的基本概念。

商业智能是指利用数据挖掘技术、知识发现等技术分析和挖掘结构化的、面向特定领域的存储与数据仓库的信息，它可以帮助用户认清发展趋势、识别数据模式、获取职能决策支持并得出结论。商务智能技术主要体现在“智能”上，即通过对大量数据的分析，得到趋势变化等重要知识，并为决策提供支持。选项 A、C、D 都是对数据进行分析，获得知识的过程；选项 B 仅仅是获取数据，并没有对数据进行分析，因此不属于商业智能范畴。

企业应用集成通过采用多种集成模式构建统一标准的基础平台，将具有不同功能和目的且独立运行的企业信息系统联合起来。其中，面向(19)的集成模式强调处理不同应用系统之间的交互逻辑，与核心业务逻辑相分离，并通过不同应用系统之间的协作共同完成某项业务功能。

- (19)A. 数据 B. 接口 C. 过程 D. 界面

【答案】C

【解析】本题考查企业应用集成的方式和特点。

企业应用集成通过采用多种集成模式，构建统一标准的基础平台，将具有不同功能和目的而又独立运行的企业信息系统联合起来。目前市场上主流的集成模式有三种，分别是面向信息的集成、面向过程的集成和面向服务的集成。其中面向过程的集成模式强调处理不同应用系统之间的交互逻辑，与核心业务逻辑相分离，并通过不同应用系统之间的协作共同完成某项业务功能。

电子数据交换（EDI）是电子商务活动中采用的一种重要的技术手段。以下关于 EDI 的叙述中，错误的是（20）。

(20) A. EDI 的实施需要一个公认的标准和协议，将商务活动中涉及的文件标准化和格式化

B. EDI 的实施在技术上比较成熟，成本也较低

C. EDI 通过计算机网络，在贸易伙伴之间进行数据交换和自动处理

D. EDI 主要应用于企业与企业、企业与批发商之间的批发业务

【答案】B

【解析】本题主要考查电子数据交换（EDI）的基本概念和特点。

电子数据交换是电子商务活动中采用的一种重要的技术手段。EDI 的实施需要一个公认的标准和协议，将商务活动中涉及的文件标准化和格式化；EDI 通过计算机网络，在贸易伙伴之间进行数据交换和自动处理；EDI 主要应用于企业与企业、企业与批发商之间的批发业务；EDI 的实施在技术上比较成熟，但是实施 EDI 需要统一数据格式，成本与代价较大。

用户文档主要描述所交付系统的功能和使用方法。下列文档中，（21）属于用户文档。

(21) A. 需求说明书

B. 系统设计

C. 安装文档

D. 系统测试计划

【答案】C

【解析】

用户文档主要描述所交付系统的功能和使用方法，并不关心这些功能是怎样实现的。用户文档是了解系统的第一步，它可以让用户获得对系统准确的初步印象。

用户文档至少应该包括下述 5 方面的内容。

①功能描述：说明系统能做什么。

②安装文档：说明怎样安装这个系统以及怎样使系统适应特定的硬件配置。

③使用手册：简要说明如何着手使用这个系统（通过丰富的例子说明，怎样使用常用的系统功能，并说明用户操作错误是怎样恢复和重新启动的）。

④参考手册：详尽描述用户可以使用的系统设施以及它们的使用方法，并解释系统可能产生的各种出错信息的含义（对参考手册最主要的要求是完整，因此通常使用形式化的描述技术）。

⑤操作员指南（如果有系统操作员的话）：说明操作员应如何处理使用中出现的各

种情况。

系统文档是从问题定义、需求说明到验收测试计划这样一系列和系统实现有关的文档。

描述系统设计、实现和测试的文档对于理解程序和维护程序来说是非常重要的。

配置项是构成产品配置的主要元素，其中(22)不属于配置项。

- (22) A. 设备清单 B. 项目质量报告 C. 源代码 D. 测试用例

【答案】A

【解析】

配置项是构成产品配置的主要元素，配置项主要有以下两大类：

- (1) 属于产品组成部分的工作成果：如需求文档、设计文档、源代码和测试用例等；
- (2) 属于项目管理和机构支撑过程域产生的文档：如工作计划、项目质量报告和项目跟踪报告等。

这些文档虽然不是产品的组成部分，但是值得保存。所以设备清单不属于配置项。

一个大型软件系统的需求通常是会发生变化的。以下关于需求变更策略的叙述中错误的是(23)

- (23) A. 所有需求变更必须遵循变更控制过程
- B. 对于未获得核准的变更，不应该做变更实现工作
- C. 完成了对某个需求的变更之后，就可以删除或者修改变更请求的原始文档
- D. 每一个集成的需求变更必须能追溯到一个经核准的变更请求

【答案】C

【解析】

一个大型软件系统的需求通常是会发生变化的。在进行需求变更时，可以参考以下的需求变更策略：

- (1) 所有需求变更必须遵循变更控制过程；
- (2) 对于未获得批准的变更，不应该做设计和实现工作；
- (3) 变更应该由项目变更控制委员会决定实现哪些变更；
- (4) 项目风险承担者应该能够了解变更数据库的内容；
- (5) 决不能从数据库中删除或者修改变更请求的原始文档；
- (6) 每一个集成的需求变更必须能跟踪到一个经核准的变更请求。

以下关于需求管理的叙述中，正确的是(24)。

(24) A. 需求管理是一个对系统需求及其变更进行了解和控制的过程

B. 为了获得项目，开发人员可以先向客户做出某些承诺

C. 需求管理的重点在于收集和分析项目需求

D. 软件开发过程是独立于需求管理的活动

【答案】A

【解析】

需求管理是一个对系统需求变更、了解和控制的过程。需求管理过程与需求开发过程相互关联，当初始需求导出的同时就启动了需求管理计划，一旦形成了需求文档的初稿，需求管理活动就开始了。

关于需求管理过程域内的原则和策略，可以参考：

①需求管理的关键过程领域不涉及收集和分析项目需求，而是假定已收集了软件需求，或者已由更高一级的系统给定了需求。

②开发人员在向客户以及有关部门承诺某些需求之前，应该确认需求和约束条件、风险、偶然因素、假定条件等。

③关键处理领域同样建议通过版本控制和变更控制来管理需求文档。

(25)方法以原型开发思想为基础，采用迭代增量式开发，发行版本小型化，比较适合需求变化较大或者开发前期对需求不是很清晰的项目。

(25) A. 信息工程

B. 结构化

C. 面向对象

D. 敏捷

【答案】D

【解析】

敏捷方法以原型开发思想为基础，采用迭代增量式开发，发行版本小型化，比较适合需求变化较大或者开发前期对需求不是很清晰的项目。

项目管理工具用来辅助项目经理实施软件开发过程中的项目管理活动，它不能 (26)。

(27)就是一种典型的项目管理工具。

(26) A. 覆盖整个软件生存周期

B. 确定关键路径、松弛时间、超前时间和滞后时间

C. 生成固定格式的报表和裁剪项目报告

D. 指导软件设计人员按软件生存周期各个阶段的适用技术进行设计工作

(27) A. 需求分析工具 B. 成本估算工具 C. 软件评价工具 D. 文档分析工具

【答案】D B

【解析】

项目管理工具用来辅助软件的项目管理活动。通常项目管理活动包括项目的计划、调度、通信、成本估算、资源分配及质量控制等。一个项目管理工具通常把重点放在某一个或某几个特定的管理环节上，而不提供对管理活动包罗万象的支持。

项目管理工具具有以下特征：

- (1) 覆盖整个软件生存周期；
- (2) 为项目调度提供多种有效手段；
- (3) 利用估算模型对软件费用和工作量进行估算；
- (4) 支持多个项目和子项目的管理；
- (5) 确定关键路径，松弛时间，超前时间和滞后时间；
- (6) 对项目组成员和项目任务之间的通信给予辅助；
- (7) 自动进行资源平衡；
- (8) 跟踪资源的使用；
- (9) 生成固定格式的报表和剪裁项目报告。

成本估算工具就是一种典型的项目管理工具。

逆向工程导出的信息可以分为 4 个抽象层次，其中(28)可以抽象出程序的抽象语法树、符号表等信息；(29)可以抽象出反应程序段功能及程序段之间关系的信息。

(28) A. 实现级 B. 结构级 C. 功能级 D. 领域级

(29) A. 实现级 B. 结构级 C. 功能级 D. 领域级

【答案】A C

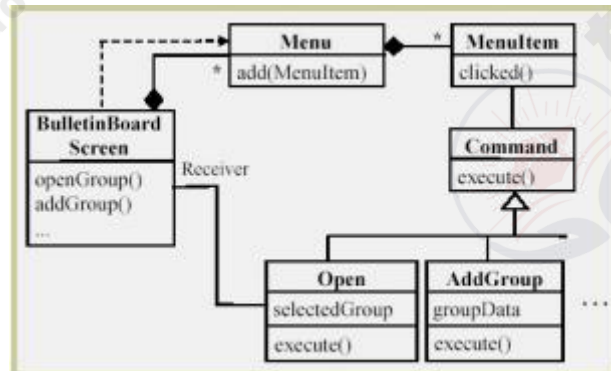
【解析】

逆向工程导出的信息可分为如下 4 个抽象层次。

- ①实现级：包括程序的抽象语法树、符号表等信息。
- ②结构级：包括反映程序分量之间相互依赖关系的信息，例如调用图、结构图等。
- ③功能级：包括反映程序段功能及程序段之间关系的信息。

④领域级：包括反映程序分量或程序与应用领域概念之间对应关系的信息。

某软件公司欲开发一个 Windows 平台上的公告板系统。在明确用户需求后，该公司的架构师决定采用 Command 模式实现该系统的界面显示部分，并设计 UML 类图如下图所示。图中与 Command 模式中的“Invoker”角色相对应的类是 (30)，与“ConcreteCommand”角色相对应的类是 (31)。



(30) A. Command B. MenuItem C. Open D. BulletinBoardScreen

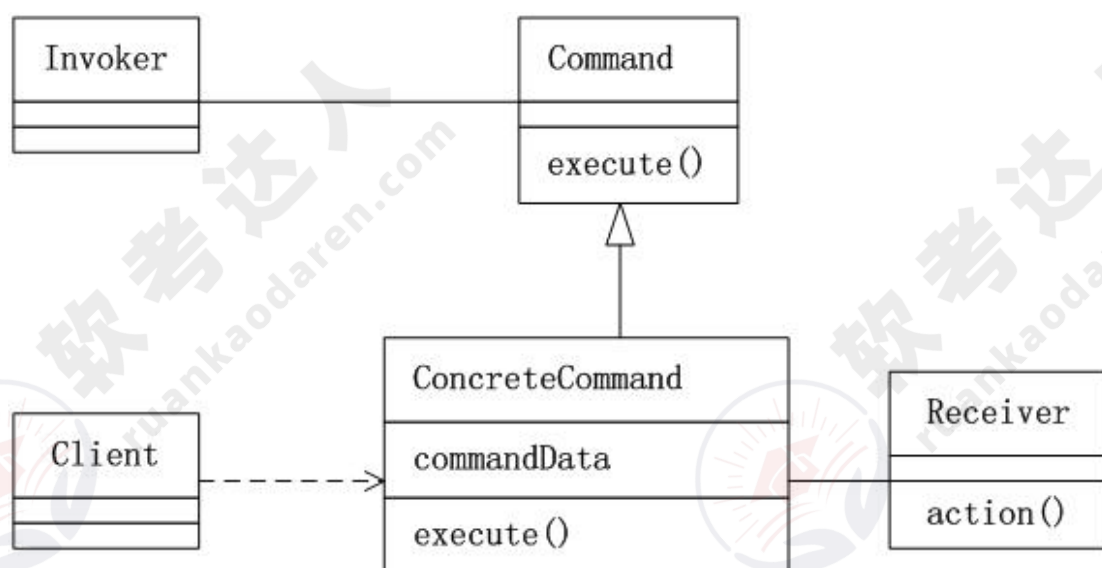
(31) A. Command B. MenuItem C. Open D. BulletinBoardScreen

【答案】B C

【解析】

Command（命令）模式是设计模式中行为模式的一种，它将“请求”封装成对象，以便使用不同的请求、队列或者日志来参数化其他对象。Command 模式也支持可撤销的操作。

Command 模式的类图也如图所示：



对于题目所给出的图，与“Invoker”角色相对应的类是 MenuItem，与“Concrete Command”

角色相对应的类是 Open。

用例 (usecase) 用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个“订单输入子系统”中，创建新订单和更新订单都需要核查用户账号是否正确。用例“创建新订单”、“更新订单”与用例“核查户账号”之间是 (32) 关系。

- (32) A. 包含 (include) B. 扩展 (extend)
C. 分类 (classification) D. 聚集 (aggregation)

【答案】A

【解析】

用例是在系统中执行的一系列动作，这些动作将生成特定参与者可见的价值结果。它确定了一个和系统参与者进行交互，并可由系统执行的动作序列。用例模型描述的是外部执行者 (Actor) 所理解的系统功能。用例模型用于需求分析阶段，它的建立是系统开发者和用户反复讨论的结果，表明了开发者和用户对需求规格达成的共识。

两个用例之间的关系主要有两种情况：一种是用于重用的包含关系，用构造型 include 表示；另一种是用于分离出不同行为的扩展，用构造型 extend 表示。

①包含关系：当可以从两个或两个以上的原始用例中提取公共行为，或者发现能够使用一个构件来实现某一个用例的部分功能是很重要的事时，应该使用包含关系来表示它们。

②扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种事情，可以断定将这个用例分为一个主用例和一个或多个辅用例描述可能更加清晰。

面向对象的设计模型包含以 (33) 表示的软件体系结构图，以 (34) 表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

- (33) A. 部署图 B. 包图 C. 协同图 D. 交互图
(34) A. 部署图 B. 包图 C. 协同图 D. 交互图

【答案】B D

【解析】

面向对象的设计模型包含以包图表示的软件体系结构图，以交互图表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

基于构件的开发模型包括软件的需求分析定义、(35)、(36)、(37)，以及测试和发布 5

个顺序执行的阶段。

(35) A. 构件接口设计 B. 体系结构设计 C. 元数据设计 D. 集成环境设计

(36) A. 数据库建模 B. 业务过程建模 C. 对象建模 D. 构件库建立

(37) A. 应用软件构建 B. 构件配置管理 C. 构件单元测试 D. 构件编码实现

【答案】B D A

【解析】本题考查基于构件的软件开发模型的基础知识。

基于构件的开发模型利用模块化方法将整个系统模块化，并在一定构件模型的支持下复用构件库中的一个或多个软件构件，通过组合手段高效率、高质量地构造应用软件系统的过程。基于构件的开发模型融合了螺旋模型的许多特征，本质上是演化形的，开发过程是迭代的。基于构件的开发模型由软件的需求分析定义、体系结构设计、构件库建立、应用软件构建以及测试和发布 5 个阶段组成。

以下关于软件构件及其接口的叙述，错误的是 (38)。

(38) A. 构件是软件系统中相对独立且具有一定意义的构成成分

B. 构件在容器中进行管理并获取其属性或者服务

C. 构件不允许外部对所支持的接口进行动态发现或调用

D. 构件可以基于对象实现，也可以不基于对象实现

【答案】C

【解析】本题考查软件构件的基本概念。

软件构件是软件系统中具有一定意义的、相对独立的可重用单元。与对象相比，构件可以基于对象实现，也可以不作为对象实现。构件需要在容器中管理并获取容器提供的服务；客户程序可以在运行状态下利用接口动态确定构件所支持的功能并调用。

在一个典型的基于 MVC (Model View Controller) 的 J2EE 应用中，分发客户请求、有效组织其他构件为客户端提供服务的控制器由 (39) 实现。

(39) A. Entity Bean B. Session Bean C. Servlet D. JSP

【答案】C

【解析】本题考查 J2EE 应用架构的基本知识。

在一个典型的基于 MVC (Model View Controller) 的 J2EE 应用中，系统的界面由 JSP 构件实现，分发客户请求、有效组织其他构件为客户端提供服务的控制器由 Servlet 构件

实现，数据库相关操作由 Entity Bean 构件实现，系统核心业务逻辑由 Session Bean 构件实现。

以下关于 RDBMS 数据分布的叙述中，错误的是 (40)。

- (40) A. 数据垂直分割是将不同表的数据存储到不同的服务器上
- B. 数据水平分割是将不同行的数据存储到不同的服务器上
- C. 数据复制是将数据的多个副本存储到不同的服务器上
- D. 数据复制中由 RDBMS 维护数据的一致性

【答案】A

【解析】本题考查数据分布方面的基本知识。

数据分割和数据复制是数据分布的两种重要方式。数据分割有垂直分割和水平分割两种模式，前者是将表中不同字段的数据存储到不同的服务器上；后者是将表中不同行的数据存储到不同的服务器上。数据复制是为了提升数据访问效率而采用的一种增加数据冗余的方法，它将数据的多个副本存储到不同的服务器上，由 RDBMS 负责维护数据的一致性。

系统应用架构设计中，网络架构数据流图的主要作用是将处理器和设备分配到网络中。

(41) 不属于网络架构数据流图的内容。

- (41) A. 服务器、客户端及其物理位置
- B. 处理器说明信息
- C. 单位时间的数据流大小
- D. 传输协议

【答案】C

【解析】本题考查网络规划与设计的基本知识。

应用架构建模中要绘制的第一个物理数据流图 (PDFD) 是网络架构 DFD，它们不显示单位时间的数据流量，需要显示的信息包括服务器及其物理位置；客户端及其物理位置；处理器说明；传输协议。

系统输入设计中应尽可能考虑人的因素，以下关于输入设计的一般原理中，错误的是

(42)。

- (42) A. 只让用户输入变化的数据
- B. 使用创新的模式吸引用户的眼球
- C. 表格中各个数据项应有提示信息

D. 尽可能使用选择而不是键盘输入的方式获取数据试

【答案】B

【解析】本题考查应用系统输入设计的基本知识。

人的因素在系统输入设计中扮演了很重要的角色。输入应该尽可能地简单，以降低错误发生的可能性，如对于范围可控的数据，使用选择的方式替代用户输入；只输入变化的数据等。输入应该尽可能使用已有含义明确的设计，需要采用模仿的方式而非创新。为了避免用户理解的二义性，应该对表格中输入的数据给出提示信息。

系统测试将软件、硬件、网络等其他因素结合，对整个软件进行测试。(43)不是系统测试的内容。

(43) A. 路径测试 B. 可靠性测试 C. 安装测试 D. 安全测试

【答案】A

【解析】

系统测试是将已经确认的软件、计算机硬件、外设和网络等其他因素结合在一起，进行信息系统的各种集成测试和确认测试，其目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方。系统测试是根据系统方案说明书来设计测试用例，常见的系统测试主要有恢复测试、安全性测试、压力测试、性能测试、可靠性测试、可用性测试、可维护性测试和安装测试。

软件测试是为了发现错误而执行程序的过程。黑盒测试法主要根据(44)来设计测试用例。

(44) A. 程序内部逻辑 B. 程序外部功能 C. 程序数据结构 D. 程序流程图

【答案】B

【解析】

软件测试是为了发现错误而执行程序的过程。黑盒测试也称为功能测试，是根据规格说明所规定的功能来设计测试用例，它不考虑程序的内部结构和处理过程。常用的黑盒测试技术有等价类划分、边值分析、错误猜测和因果图等。

软件架构贯穿于软件的整个生命周期，但在不同阶段对软件架构的关注力度并不相同，在(45)阶段，对软件架构的关注最多。

(45) A. 需求分析与设计 B. 设计与实现 C. 实现与测试 D. 部署与变更

【答案】B

【解析】本题主要考查软件架构对软件开发的影响和在生命周期中的关注力度。

软件架构贯穿于软件的整个生命周期，但在不同的阶段对软件架构的关注力度并不相同。其中需求分析阶段主要关注问题域；设计阶段主要将需求转换为软件架构模型；软件实现阶段主要关注将架构设计转换为实际的代码；软件部署阶段主要通过组装软件组件提高系统的实现效率。其中设计与实现阶段在软件架构上的工作最多，也最重要，因此关注力度最大。

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键活动。以下关于软件架构重要性的叙述中，错误的是 (46)。

- (46) A. 架构设计能够满足系统的性能、可维护性等品质
- B. 良好的架构设计能够更好地捕获并了解用户需求
- C. 架构设计能够使得不同的利益相关人 (stakeholders) 达成一致的目标
- D. 杂构设计能够支持项目计划和项目管理等活动

【答案】B

【解析】

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键因素。架构设计能够满足系统的性能、可维护性等品质；能够使得不同的利益相关人 (stakeholders) 达成一致的目标；能够支持项目计划和项目管理等活动；能够有效地管理复杂性；等等。然而系统架构的给出必须建立在需求明确的基础上。

软件架构需求是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。以下活动中，不属于软件架构需求过程范畴的是 (47)。

- (47) A. 设计构件 B. 需求获取 C. 标识构件 D. 架构需求评审

【答案】A

【解析】

软件架构需求是指用户对目标软件系统在功能、行为、性能和设计约束等方面的期望。需求过程主要是获取用户需求，标识系统中所要用到的构件，并进行架构需求评审。其中标识构件又详细分为生成类图、对类图进行分组和将类打包成构件三步。软件架构需求并不应该包括设计构件的过程。

基于架构的软件设计（ABSD）强调由商业、质量和功能需求的组合驱动软件架构设计。

以下关于 ABSD 的叙述中，错误的是（48）。

(48) A. 使用 ABSD 方法，设计活动可以从项目总体功能框架明确就开始

B. ABSD 方法是一个自顶向下，递归细化的过程

C. ABSD 方法有三个基础：功能分解、选择架构风格实现质量和商业需求以及软件模板的使用

D. 使用 ABSD 方法，设计活动的开始意味着需求抽取和分析活动可以终止

【答案】D

【解析】

基于架构的软件设计（ABSD）强调由商业、质量和功能需求的组合驱动软件架构设计。使用 ABSD 方法，设计活动可以从项目总体功能框架明确就开始，并且设计活动的开始并不意味着需求抽取和分析活动可以终止，而是应该与设计活动并行。ABSD 方法有三个基础：第一个基础是功能分解，在功能分解中使用已有的基于模块的内聚和耦合技术。第二个基础是通过选择体系结构风格来实现质量和商业需求。第三个基础是软件模板的使用。ABSD 方法是一个自顶向下，递归细化的过程，软件系统的架构通过该方法得到细化，直到能产生软件构件的类。

软件架构文档是对软件架构的正式描述，能够帮助与系统有关的开发人员更好地理解软件架构。软件架构文档的写作应该遵循一定的原则。以下关于软件架构文档写作原则的叙述中，错误的是（49）。

(49) A. 架构文档应该从架构设计者的角度进行编写

B. 应该保持架构文档的即时更新，但更新不要过于频繁

C. 架构文档中的描述应该尽量避免不必要的重复

D. 每次架构文档修改，都应该记录修改的原则

【答案】A

【解析】

软件架构文档是对软件架构的一种描述，帮助程序员使用特定的程序设计语言实现软件架构。软件架构文档的写作应该遵循一定的原则，这些原则包括：文档要从使用者的角度进行编写；必须分发给所有与系统有关的开发人员；应该保持架构文档的即时更新，但更新不要过于频繁；架构文档中描述应该尽量避免不必要的重复；每次架构文档修改都应该记录

进行修改的原则。

架构复审是基于架构开发中一个重要的环节。以下关于架构复审的叙述中，错误的是 (50)。

- (50) A. 架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误
- B. 架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试
- C. 架构复审人员由系统设计与开发人员组成
- D. 架构设计、文档化和复审是一个迭代的过程

【答案】C

【解析】

架构复审是基于架构开发中一个重要的环节。架构设计、文档化和复审是一个迭代的过程。从这个方面来说，在一个主版本的软件架构分析之后，要安排一次由外部人员（用户代表和领域专家）参加的复审。架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试。架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误。

Windows 操作系统在图形用户界面处理方面采用的核心架构风格是 (51) 风格。Java 语言宣传的“一次编写，到处运行”的特性，从架构风格上看符合 (52) 风格的特点。

- (51) A. 虚拟机 B. 管道-过滤器 C. 事件驱动 D. 微内核-扩展
- (52) A. 虚拟机 B. 管道-过滤器 C. 事件驱动 D. 微内核-扩展

【答案】C A

【解析】

Windows 操作系统在图形用户界面处理方面采用的是典型的“事件驱动”的架构风格，首先注册事件处理的是回调函数，当某个界面事件发生时（例如键盘敲击、鼠标移动等），系统会查找并选择合适的回调函数处理该事件。Java 语言是一种解释型语言，在 Java 虚拟机上运行，这从架构风格上看是典型的“虚拟机”风格，即通过虚拟机架构屏蔽不同的硬件环境。

某软件开发公司负责开发一个 Web 服务器服务端处理软件，其核心部分是对客户端请求消息的解析与处理，包括 HTTP 报头分离、SOAP 报文解析等功能。该公司的架构师决定采用成熟的架构风格指导整个软件的设计，以下 (53) 架构风格，最适合该服务端处理软件。

(53) A. 虚拟机 B. 管道-过滤器 C. 黑板结构 D. 分层结构

【答案】B

【解析】

根据题干描述，Web 服务器服务端的核心功能是数据处理，由于 Web 服务在数据传输方面具有协议分层的特征，即底层协议会包装上层协议（HTTP 协议体中包含整个 SOAP 消息内容），因此需要数据内容的逐步分解与分阶段处理。比较选项中的架构风格，由于管道-过滤器的架构风格支持分阶段数据处理，因此特别适合该服务端处理软件的要求。

某公司欲开发一个基于图形用户界面的集成调试器。该调试器的编辑器和变量监视器可以设置调试断点。当调试器在断点处暂停运行时，编辑程序可以自动卷屏到断点，变量监视器刷新变量数值。针对这样的功能描述，采用 (54) 的架构风格最为合适。

(54) A. 数据共享 B. 虚拟机 C. 隐式调用 D. 显式调用

【答案】C

【解析】

根据题干描述，调试器在设置端点时，其本质是在断点处设置一个事件监听函数，当程序执行到断点位置时，会触发并调用该事件监听函数，监听函数负责进行自动卷屏、刷新变量数值等动作。这是一个典型的回调机制，属于隐式调用的架构风格。

某公司欲开发一种工业机器人，用来进行汽车零件的装配。公司的架构师经过分析与讨论，给出了该机器人控制软件的两候选架构方案：闭环控制和分层结构。以下对于这两种候选架构的选择理由，错误的是 (55)。

(55) A. 应该采用闭环控制架构，因为闭环结构给出了将软件分解成几个协作构件的方法，这对于复杂任务特别适合

B. 应该采用闭环控制结构，因为闭环控制架构中机器人的主要构件（监控器、传感器、发动机等）是彼此分开的，并能够独立替换

C. 应该采用分层结构，因为分层结构很好地组织了用来协调机器人操作的构件，系统结构更加清晰

D. 应该采用分层结构，因为抽象层的存在，满足了处理不确定性的需要：在较低层次不确定的实现细节在较高层次会变得确定

【答案】A

【解析】

采用闭环结构的软件通常由几个协作构件共同构成，且其中的主要构件彼此分开，能够进行替换与重用，但闭环结构通常适用于处理简单任务（如机器装配等），并不适用于复杂任务。分层结构的特点是通过引入抽象层，在较低层次不确定的实现细节在较高层次会变得确定，并能够组织层间构件的协作，系统结构更加清晰。

一个软件的架构设计是随着技术的不断进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。以下关于编译器架构的叙述中，错误的是 (56)。

(56) A. 早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码

B. 早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表并不是程序数据的一部分

C. 现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示

D. 现代的编译器采用以数据共享为中心的架构风格，但由于分析树是在语法分析阶段结束后才产生作为语义分析的输入，因此分析树不是数据中心的共享数据

【答案】D

【解析】

一个软件的架构设计是随着技术的不断进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码。早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表并不是程序数据的一部分。现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示。现代的编译器采用以数据共享为中心的架构风格，分析树是在语法分析阶段结束后才产生作为语义分析的输入，分析树是数据中心的共享数据，为后续的语义分析提供了帮助。

(57) 的选择是开发一个软件系统时的基本设计决策；(58) 是最低层的模式，关注软件系统的设计与实现，描述了如何实现构件及构件之间的关系。引用-计数是 C++ 管理动态资源

时常用的一种 (59)。

(57) A. 架构模式 B. 惯用法 C. 设计模式 D. 分析模式

(58) A. 架构模式 B. 惯用法 C. 设计模式 D. 分析模式

(59) A. 架构模式 B. 惯用法 C. 设计模式 D. 分析模式

【答案】A B B

【解析】本题考查软件设计中使用的架构模式、设计模式和惯用法的基本概念。

架构模式是软件设计中的高层决策，例如 C/S 结构就属于架构模式，架构模式反映了开发软件系统过程中所作的基本设计决策；设计模式主要关注软件系统的设计，与具体的实现语言无关；惯用法则是实现时通过某种特定的程序设计语言来描述构件与构件之间的关系，例如引用-计数就是 C++ 语言中的一种惯用法。

某软件公司基于面向对象技术开发了一套图形界面显示构件库 VisualComponent。在使用该库构建某图形界面时，用户要求为界面定制一些特效显示效果，如带滚动条、能够显示艺术字体的透明窗体等。针对这种需求，公司采用 (60) 最为灵活。

(60) A. 桥接模式 B. 命令模式 C. 组合模式 D. 装饰模式

【答案】D

【解析】

根据题干描述，可以看出其基础是一个图形界面，并要求为图形界面提供一些定制的特 效，例如带滚动条的图形界面，能够显示艺术字体且透明的图形界面等。这要求能够动态地 对一个对象进行功能上的扩展，也可以对其子类进行功能上的扩展。对照选项中的 4 种设计 模式，装饰模式最符合这一要求。

某软件公司承接了为某 workflow 语言开发解释器的工作。该 workflow 语言由多种活动节点构 成，具有类 XML 的语法结构。用户要求解释器工作时，对每个活动节点进行一系列的处理， 包括执行活动、日志记录、调用外部应用程序等，并且要求处理过程具有可扩展能力。针对 这种需求，公司采用 (61) 最为恰当。

(61) A. 适配器模式 B. 迭代器模式 C. 访问者模式 D. 观察者模式

【答案】C

【解析】

根据题干描述，可以看出本题的核心在于对某个具有固定结构的 活动节点需要多种处理

能力，且处理能力可扩展，也就是说要求在不改变原来类结构（活动节点）的基础上增加新功能。对照 4 个选项，发现访问者模式最符合要求。

Architecture Tradeoff Analysis Method (ATAM) 是一种软件架构的评估方法，以下关于该方法的叙述中，正确的是 (62)。

- (62) A. ATAM 是一种代码评估方法 B. ATAM 需要评估软件的需求是否准确
C. ATAM 需要对软件系统进行测试 D. ATAM 不是一种精确的评估工具试题

【答案】D

【解析】 本题考查软件体系结构中的评估方法。

ATAM 是软件体系结构评估中的一种方法，主要对软件体系结构的设计结果进行评估。评估是软件系统详细设计、实现和测试之前的阶段工作，因此评估不涉及系统的实现代码和测试，因为评估是考查软件体系结构是否能够合适地解决软件系统的需求，并不对软件需求自身是否准确进行核实，而软件需求是否准确是需求评审阶段的工作。ATAM 并不是一种精确的评估方法，该方法表现的主要形式是评审会议。

识别风险点、非风险点、敏感点和权衡点是 ATAM 方法中的关键步骤。已知针对某系统所做的架构设计中，提高其加密子系统的加密级别将对系统的安全性和性能都产生非常大的影响，则该子系统一定属于 (63)。

- (63) A. 风险点和敏感点 B. 权衡点和风险点
C. 权衡点和敏感点 D. 风险点和非风险点

【答案】C

【解析】 本题考查软件体系结构中的评估方法。

加密子系统的加密级别会对安全性和性能产生影响，一般而言，加密程度越高，安全性越好，但是其性能会降低；而加密程度越低，安全性越差，但性能一般会提高。因此该子系统将在安全性和性能两个方面产生冲突，所以该子系统一定属于权衡点和敏感点。

信息安全策略应该全面地保护信息系统整体的安全，网络安全体系设计是网络逻辑设计工作的重要内容之一，可从物理线路安全、网络安全、系统安全、应用安全等方面来进行安全体系的设计与规划。其中，数据库的容灾属于 (64) 的内容。

- (64) A. 物理线路安全与网络安全 B. 网络安全与系统安全
C. 物理线路安全与系统安全 D. 网络安全与应用安全

【答案】D

【解析】

网络安全体系设计是逻辑设计工作的重要内容之一，数据库容灾属于网络安全和应用安全考虑范畴。

公司总部与分部之间需要传输大量数据，在保障数据安全的同时又要兼顾密钥算法效率，最合适的加密算法是 (65)。

- (65) A. RC-5 B. RSA C. ECC D. MD5

【答案】A

【解析】

公司总部与分部之间通过 Internet 传输数据，需要采用加密方式保障数据安全。加密算法中，对称加密比非对称加密效率要高。RSA 和 ECC 属于非对称加密算法，MD5 为摘要算法，故选择 RC-5。

我国的《著作权法》对一般文字作品的保护期是作者有生之年和去世后 50 年，德国的《版权法》对一般文字作品的保护期是作者有生之年和去世后 70 年。假如某德国作者已去世 60 年，以下说法中正确的是 (66)。

(66) A. 我国 M 出版社拟在我国翻译出版该作品，需要征得德国作者继承人的许可，方可在我国出版发行

B. 我国 M 出版社拟在我国翻译出版该作品，不需要征得德国作者继承人的许可，就可在我国出版发行

C. 我国 M 出版社未征得德国作者继承人的许可，将该翻译作品销售到德国，不构成侵权

D. 我国 M 出版社未征得德国作者继承人的许可，将该翻译作品在我国销售，构成侵权

【答案】B

【解析】本题考查知识产权方面的基础知识。

按照《伯尔尼公约》的规定，一个成员国给予 其他成员国作品的版权保护期，应按照

该成员国版权法的规定。依据我国著作权法的规定，该德国作者的作品已经超过法定版权保护期，不再受到版权保护。因此，出版社不需要征得德国作者继承人的许可，即可在我国出版发行该德国作者的作品。如果将该翻译出版作品未征得德国作者继承人的许可销售到德国，已构成侵权。这是因为德国的《版权法》规定作品的售到德国，则构成侵权。

我国的《著作权法》对一般文字作品的保护期是作者有生之年和去世后 50 年，该作者已去世 60 年，超过了我国《著作权法》对一般文字作品的保护期，在我国也不再受著作权保护。所以我国 M 出版社不需要征得德国作者继承人的许可，即可在我国出版发行该德国作者的作品。

(67) 不属于我国著作权法所保护的内容。

- (67) A. 为保护其软件著作权而采取的技术措施 B. 软件权利电子信息
C. 通过信息网络传播的软件 D. 采用反编译技术获得的软件

【答案】D

【解析】本题考查知识产权方面的基础知识。

我国著作权法采取列举方法，规定了侵权行为的表现形式。其中包括未经著作权人许可，复制、发行、表演、放映、广播、汇编、通过信息网络向公众传播其作品的行为；未经著作权人或者与著作权有关的权利人许可，故意避开或者破坏权利人为其作品、录音录像制品等采取的保护著作权或者与著作权有关的权利的技术措施的行为；未经著作权人或者与著作权有关的权利人许可，故意删除或者改变作品、录音录像制品等的权利管理电子信息的行为。这三种表现形式分别涵盖了试题中 A、B、C 选项。虽然利用反向编译技术、净室技术和反向工程技术等获得他人软件技术构思、技术方案并直接用于其软件产品中的行为是一种“复制”软件技术构思、技术方案的行为，但是对于这些行为在法律上不会受到制止。目前，我国对软件实施反编译是否合法还没有相应的法律规定。

由于著作权不保护思想，软件开发设计人员对体现在软件中的创造性的构思和技术方案不能得到保护。任何人都可以利用反向编译技术、净室技术和反向工程技术等获得他人软件所使用的思路、原理、结构、算法、处理过程和运行方法等设计要素，直接用于自己的软件产品中，这在著作权法上并不构成侵权。

王某原是 X 公司的项目经理，在 X 公司任职期间主持开发了某软件，但未与 X 公司签定劳动合同及相应的保密协议。X 公司对该软件进行了软件著作权登记并获准。王某随后离职

并将其在 X 公司任职期间掌握的该软件技术信息、客户需求及部分源程序等秘密信息提供给另一软件公司。王某的行为 (68)

- (68) A. 既侵犯了科技公司的商业秘密权，又侵犯了科技公司的软件著作权
B. 既未侵犯科技公司的商业秘密权，又未侵犯科技公司的软件著作权
C. 侵犯了科技公司的商业秘密权
D. 侵犯了科技公司的软件著作权

【答案】D

【解析】本题考查知识产权方面的基础知识，涉及著作权和商业秘密权的相关概念。

王某作为公司的职员，在任职期间主持开发的软件为职务软件，公司对该软件享有软件著作权。王某将该软件源程序擅自提供给其他公司的行为已构成对公司软件著作权的侵犯。王某的行为将使得另一软件公司很快就会开发出类似的产品，在市场上与科技公司竞争，这样无疑会损害科技公司的利益。软件商业秘密包括软件技术秘密，如源程序、设计方法、技术方案、功能规划、开发情况和测试结果等；软件经营秘密，如经营方法、产销策略、客户情报（客户名单、客户需求）和软件市场分析等。商业秘密受到法律保护的依据是必须具备构成商业秘密的三个条件，即不为公众所知悉、具有实用性、采取了保密措施，缺少三个条件之一都会造成商业秘密丧失法律保护。公司未与王某签定劳动合同及相应的保密协议，可以认为科技公司主观上没有保守商业秘密的意愿，客观上没有采取相应的保密措施，那么公司的软件技术秘密和软件经营秘密就不具有保密性。所以，不认为王某侵犯了公司的商业秘密权。

对实际应用问题建立了数学模型后，一般还需要对该模型进行检验。通过检验尽可能找出模型中的问题，以利于改进模型，有时还可能会否定该模型。检验模型的做法有多种，但一般不会 (69)。

- (69) A. 利用实际案例数据对模型进行检验
B. 进行逻辑检验，分析该模型是否会出现矛盾
C. 用计算机模拟实际问题来检验模型
D. 检验该模型所采用的技术能否被企业主管理解

【答案】D

【解析】

针对实际问题建立的数学模型往往是近似的，往往忽略了许多复杂因素。这种模型能

否解决实际问题还需要检验。检验的方法有多种。

利用实际案例数据对模型进行检验是很常见的。将模型作为一个黑盒，通过案例数据的输入，检查其输出是否合理。这是应用人员常用的方法。

有时可以请专家来分析模型是否合理。经验丰富的专家一般会根据模型自身的逻辑，再结合实际情况，分析是否会出现矛盾或问题。

有时很难用实际案例或聘请专家来检验模型，例如，试验或实验的代价太大，难以取得实际案例，有的项目技术比较新，缺乏有经验的专家。这时，如果能利用计算机来模拟实际问题，再在计算机上检验该数学模型，这往往是一种有效的办法。例如，对某种核辐射防护建立的数学模型，采用计算机模拟方法来检验就十分有效。

企业负责人需要提供一切必要的支持来解决实际问题。至于解决过程中采用的技术问题，则需要由技术人员研究决定。企业负责人只需要听取汇报，从宏观上认可就可以，不需要理解其中的技术细节。

某类产品《种品牌在某地区的市场占有率常用概率向量 $u=(u_1, u_2, \dots, u_n)$ 表示(各分量分别表示各品牌的市场占有率，值非负，且总和为 1)。市场占有率每隔一定时间的变化常用转移矩阵 $P_{n \times n}$ 表示。如果在相当长时期内，该转移矩阵的元素均是常数，又设初始时刻的市场占有率为向量 u ，则下一时刻的市场占有率就是 uP ，再下一时刻的市场占有率就是 uP^2, \dots ，而且，市场占有率会逐步稳定到某个概率向量 Z ，即出现 $ZP=Z$ 。这种稳定的市场占有率体现了转移矩阵的特征，与初始时刻的市场占有率无关。

假设占领某地区市场的冰箱品牌 A 与 B，每月市场占有率的变化可用如下常数转移矩阵来描述：

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}$$

则冰箱品牌 A 与 B 在该地区最终将逐步稳定到市场占有率 (70)。

(70) A. (1/4, 3/4) B. (1/3, 2/3) C. (1/2, 1/2) D. (2/3, 1/3)

【答案】D

【解析】

根据题意，该地区冰箱品牌 A 与 B 每月占有率的变化描述为常数转移矩阵 P 。不管初始时刻这两种品牌的市场占有率（以概率向量来描述）如何，最终将稳定到概率向量 Z ，而且有关系式 $ZP=Z$ 。这表明， Z 的下一时刻仍然是 Z 。

设 $Z = (Z_1, Z_2)$ ，其中 $Z_1 \geq 0, Z_2 \geq 0, Z_1 + Z_2 = 1$ ，从 $ZP = Z$ 可以列出方程：

$$0.8Z_1 + 0.4Z_2 = Z_1$$

$$0.2Z_1 + 0.6Z_2 = Z_2$$

根据上述条件，求解该方程，得到 $Z_1 = 2/3, Z_2 = 1/3$ 。

因此，冰箱品牌 A 与 B 在该地区最终将逐步稳定到市场占有率 $(2/3, 1/3)$ 。品牌 A 将占有 $2/3$ 的市场，品牌 B 将占有 $1/3$ 的市场。

An architectural style defines as a family of such systems in terms of a (71) of structural organization. More specifically, an architectural style defines a vocabulary of (72) and connector types, and a set of (73) on how they can be combined. For many styles there may also exist one or more (74) that specify how to determine a system's overall properties from the properties of its parts. Many of architectural styles have been developed over the years. The best-known examples of (75) architectures are programs written in the Unix shell.

- | | |
|-------------------------|--------------------|
| (71) A. pattern | B. data flow |
| C. business process | D. position level |
| (72) A. metadata | B. components |
| C. models | D. entities |
| (73) A. functions | B. code segments |
| C. interfaces | D. constraints |
| (74) A. semantic models | B. weak entities |
| C. data schemas | D. business models |
| (75) A. event-based | B. object-oriented |
| C. pipe-and-filter | D. layered |

【答案】A B D A C

【解析】

一种架构风格以一种结构化组织模式定义一组这样的系统。具体来说，一种架构风格定义了一个构件及连接器类型的词汇表，以及一组关于它们如何能够被关联的约束。对于许多风格来说，可能也存在一个或多个语义模型，从系统部件的特性来确定系统的整体特性。许多架构风格已经发展了很多年，众所周知的管道-过滤器架构的例子就是用 UNIX shell 编写的程序。

试题一

某软件开发公司欲为某电子商务企业开发一个在线交易平台，支持客户完成网上购物活动中的在线交易。在系统开发之初，企业对该平台提出了如下要求：

- (1) 在线交易平台必须在 1s 内完成客户的交易请求。
- (2) 该平台必须保证客户个人信息和交易信息的安全。
- (3) 当发生故障时，该平台的平均故障恢复时间必须小于 10s。
- (4) 由于企业业务发展较快，需要经常为该平台添加新功能或进行硬件升级。添加新功能或进行硬件升级必须在 6 小时内完成。

针对这些要求，该软件开发公司决定采用基于架构的软件开发方法，以架构为核心进行在线交易平台的设计与实现。

【问题 1】(9 分)

软件质量属性是影响软件架构设计的重要因素。请用 200 字以内的文字列举六种不同的软件质量属性名称并解释其含义。

常见的软件质量属性有多种，例如性能 (Performance)、可用性 (Availability)、可靠性 (Reliability)、健壮性 (Robustness)、安全性 (Security)、可修改性 (Modification)、可变性 (Changeability)、易用性 (Usability)、可测试性 (Testability)、功能性 (Functionality) 和互操作性 (Inter-operation) 等。

这些质量属性的具体含义是：

- (1) 性能是指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理事件的个数。
- (2) 可用性是系统能够正常运行的时间比例。
- (3) 可靠性是指软件系统在应用或错误面前，在意外或错误使用的情况下维持软件系统功能特性的基本能力。
- (4) 健壮性是指在处理或环境中，系统能够承受压力或变更的能力。
- (5) 安全性是指系统向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。
- (6) 可修改性是指能够快速地对系统性能价格比进行变更的能力。
- (7) 可变性是指体系结构经扩充或变更成为新体系结构的能力。
- (8) 易用性是衡量用户使用一个软件产品完成指定任务的难易程度。

(9) 可测试性是指软件发现故障并隔离、定位其故障的能力特性，以及在一定的时间和成本前提下，进行测试设计、测试执行的能力。

(10) 功能性是系统所能完成所期望工作的能力。

(11) 互操作性是指系统与外界或系统与系统之间的相互作用能力。

【解析】

本题考查考生对于质量属性及质量属性实现策略的掌握情况。

【问题 2】(16 分)

请对该在线交易平台的 4 个要求进行分析，用 300 字以内的文字指出每个要求对应何种软件质量属性；并针对每种软件质量属性，各给出 2 种实现该质量属性的架构设计策略。

(1) 在线交易平台必须在 1s 内完成客户的交易请求。该要求主要对应性能，可以采用的架构设计策略有增加计算资源、改善资源需求（减少计算复杂度等）、资源管理（并发、数据复制等）和资源调度（先进先出队列、优先级队列等）。

(2) 该平台必须严格保证客户个人信息和交易信息的保密性和安全性。该要求主要对应安全性，可以采用的架构设计策略有抵御攻击（授权、认证和限制访问等）、攻击检测（入侵检测等）、从攻击中恢复（部分可用性策略）和信息审计等。

(3) 当发生故障时，该平台的平均故障恢复时间必须小于 10s。该要求主要对应可用性，可以采用的架构设计策略有 Ping/Edio、心跳、异常和主动冗余等。

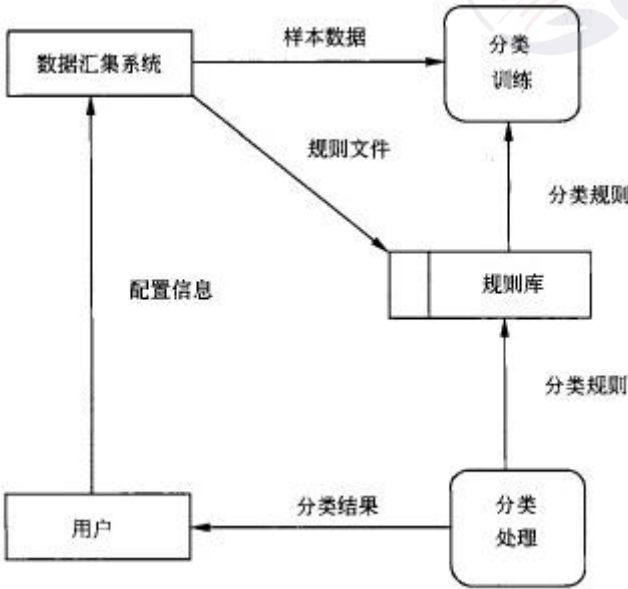
(4) 由于企业业务发展较快，需要经常为该平台添加新功能或进行硬件升级。添加新功能或进行平台升级必须在 6 小时内完成。该要求主要对应可修改性，可以采用的架构设计策略有软件模块泛化、限制模块之间通信、使用中介和延迟绑定等。

试题二

某公司拟开发一个商业情报处理系统,使公司能够及时针对市场环境的变化及时调整发展战略,以获取最大的商业利益。项目组经过讨论,决定采用结构化分析和设计方法。在系统分析阶段,为了更好地对情报数据处理流程及其与外部角色的关联进行建模,项目组成员分别给出了自己的设计思路:

(1) 小张提出先构建系统流程图(SystemFlowcharts),以便更精确地反映系统的业务处理过程及数据的输入和输出。

(2) 小李提出先构建系统数据流图(Data Flow Diagrams),来展现系统的处理过程和定义业务功能边界,并给出了情报分类子系统的0层和1层数据流图,后者如下图所示。



项目组经讨论确定以数据流图作为本阶段的建模手段。工程师老王详细说明了流程图和数据流图之间的区别与联系,并指出了上图所示数据流图中存在的错误。

【问题1】(11分)

流程图和数据流图是软件系统分析设计中常用的两种手段,请用300字以内文字简要说明流程图与数据流图的含义及其区别,并说明项目组为何确定采用数据流图作为建模手段。

数据流图作为一种图形化工具,用来说明业务处理过程、系统边界内所包含的功能和系统中的数据流。

流程图以图形化的方式展示应用程序从数据输入开始到获得输出为止的逻辑过程,描述处理过程的控制流。

两者的区别主要包括:

- (1) 数据流图中的处理过程可并行；流程图在某个时间点只能处于一个处理过程。
- (2) 数据流图展现系统的数据流；流程图展现系统的控制流。
- (3) 数据流图展现全局的处理过程，过程之间遵循不同的计时标准；流程图中处理过程遵循一致的计时标准。
- (4) 数据流图适用于系统分析中的逻辑建模阶段；流程图适用于系统设计中的物理建模阶段。

【解析】 本题考查考生对于结构化系统建模方法的掌握情况。

数据流图和流程图是结构化建模中使用的重要工具，能够帮助开发人员更好地分析和设计系统，增强系统开发人员之间交流的准确性和有效性。数据流图作为一种图形化工具，用来说明业务处理过程、系统边界内所包含的功能和系统中的数据流，适用于系统分析中的逻辑建模阶段。流程图以图形化的方式展示应用程序从数据输入开始到获得输出为止的逻辑过程，描述处理过程的控制流，往往涉及到具体的技术和环境，适用于系统设计中的物理建模阶段。

本题项目组为了更好地对情报数据处理流程及其与外部角色的关联进行建模，符合数据流图的应用场景要求，因此，首先构建数据流图以反映数据流向和系统边界。

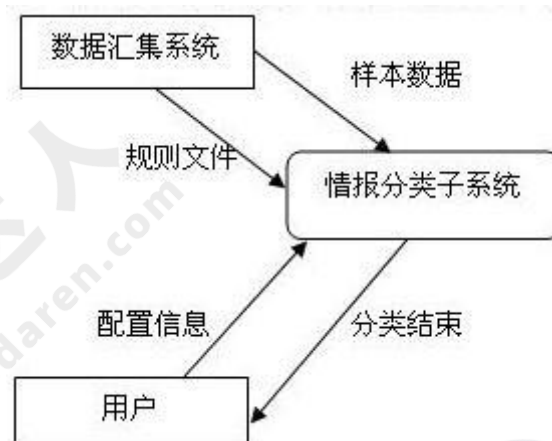
数据流图和流程图是为了达到不同的目的而产生的，其所采用的标准符号集合也不相同。在实际应用中，区别主要包括是否可以描述处理过程的并发性；描述内容是数据流还是控制流等几个方面。

【问题 2】（6 分）

请分析指出上图所示的数据流图中存在的错误及其原因，并针对 1 层数据流图绘制出情报分类子系统的 0 层数据流图。

如图所示的数据流图中存在的错误有以下 4 种：

- (1) “分类训练”加工：只有输入没有输出，产生数据黑洞；
- (2) “分类处理”加工：只有输出没有输入，无中生有；
- (3) “规则文件”数据流：外部实体没有经过加工处理，直接到数据存储；
- (4) “配置信息”数据流：外部实体之间没有加工处理，存在直接数据流。



【解析】

数据流图中常见的错误分为两种类型：一类是语法错误，包括外部实体之间、数据存储之间或外部实体与数据存储之间不经过加工而存在直接数据流；另一类是逻辑错误，包括数据黑洞（只有输入没有产生输出）、灰洞（输入不足以产生输出）和无输入。

- “分类训练”加工属于数据黑洞错误；
- “分类处理”加工属于无输入错误；
- “规则文件”和“配置信息”数据流属于语法错误。

对于复杂系统，一般采用分层数据流图逐步细化系统的内部处理逻辑。0层数据流图是1层数据流图更高级别的抽象。“分类训练”和“分类处理”加工属于内部加工，“分类规则”数据流属于内部数据流，抽象为“情报分类子系统”加工。其中，“样本数据”、“规则文件”和“配置信息”为输入数据流；“分类结果”为输出数据流。

【问题3】（8分）

高质量的数据流图是可读的、内部一致的并能够准确表示系统需求。请用300字以内文字说明在设计高质量的数据流图时应考虑的三个原则。

高质量数据流图设计时应考虑的三个原则：

(1) 复杂性最小化原则。DFD 分层结构就是把信息划分为小的且相对独立的一大批子集例子，这样就可以单独考查每一个 DFD。如果要了解某个过程更加详细的信息，可以跳转到该过程的下一层；如果要知道一个 DFD 如何与其他 DFD 相关联，可以跳转到上一层的 DFD 进行考查。

(2) 接口最小化原则。接口最小化是复杂性最小化的一种具体规则，在设计模型时，应使得模型中各个元素之间的接口数或连接数最小化。

(3) 数据流一致性原则。一个过程和它的过程分解在数据流内容中是否有差别？是否存在有数据流出但没有相应的数据流入的加工？是否存在有数据流入但没有相应的数据流出的加工？

【解析】

高质量的数据流图首先要求满足数据流一致性要求，即数据流的输入输出应该匹配，输入数据的信息量足以产生输出数据，所有的输出数据都能够从输入数据中找到数据的来源。

对于复杂性系统，可以采用 DFD 分层结构将系统的复杂性逐步分解降低，以提高数据流图的有效性。高质量的数据流图应该简化复杂的接口，最小化接口或连接的数目。

试题三

某公司承担了一项宇航嵌入式设备的研制任务。本项目除对硬件设备环境有很高的要求外，还要求支持以下功能：

- (1) 设备由多个处理机模块组成，需要时外场可快速更换（即 LRM 结构）；
- (2) 应用软件应与硬件无关，便于软硬件的升级；
- (3) 由于宇航嵌入式设备中要支持不同功能，系统应支持完成不同功能任务间的数据隔离；
- (4) 宇航设备可靠性要求高，系统要有故障处理能力。

公司在接到此项任务后，进行了反复论证，提出三层栈（TLS）软件总体架构，如下图所示，并将软件设计工作交给了李工，要求他在三周内完成软件总体设计工作，给出总体设计方案



【问题1】（8分）

用 150 字以内的文字，说明公司制定的 TLS 软件架构的层次特点，并针对上述功能需求（1）-（4），说明架构中各层内涵。

TLS 结构框架的主要特点：

- (1) 应用软件仅与操作系统服务相关，不直接操作硬件。
- (2) 操作系统通过模块支持原软件访问硬件，可与具体硬件无关。
- (3) 模块支持层将硬件抽象成标准操作。
- (4) 通过三层栈的划分可实现硬件的快速更改与升级，应用软件的升级不会引起硬件的变更。

TLS 结构框架的各层内涵是：

- (1) 应用层主要完成宇航设备的具体工作，由多个功能任务组成，各功能任务间的隔离由操作系统层实现。
- (2) 操作系统层实现应用软件与硬件的隔离，为应用软件提供更加丰富的计算机资源服

务。操作系统为应用软件提供标准的 API 接口（如 POSIX），确保了应用软件的可升级性。

(3) 模块支持层为操作系统管理硬件资源提供统一管理方法，用一种抽象的标准接口实现软件与硬件的无关性，达到硬件的升级要求，便于硬件的外场快速更换。

【试题解析】

本题考查嵌入式系统的概念，主要包括嵌入式软件体系结构的框架设计，提高宇航设备可靠性的设计方法和掌握根据用户需求开展软件需求分析的手段。

此类题目要求考生认真阅读题目对用户需求的描述，在分析、论证和概念设计的基础上，根据实例要求完善问题的解析。本题重点考查考生归纳问题、分析问题和解决问题的能力。

本题首先给出用户针对宇航设备的 4 项功能要求：

- (1) 设备由多个处理机模块组成，需要时外场可快速更换（即 LRM 结构）；
- (2) 应用软件应与硬件无关，便于软硬件的升级；
- (3) 由于宇航嵌入式设备中要支持不同功能，系统应支持完成不同功能任务间的数据隔离；
- (4) 宇航设备可靠性要求高，系统要有故障处理能力。

从 4 项要求分析看，第一项要求是考查考生的基础硬件知识，宇航设备应该由多个硬件模块组成，各个模块承担着不同功能，外场可快速更换是指在实验现场（如机场、发射基地）能够用备份模块替换掉故障硬件模块，从而节省维修时间。此项要求是本题设计软件体系总体架构的基础，也是采用三层栈（TLS）的首要条件。第二项要求是考查考生从软 / 硬件升级方面理解三层栈（TLS）架构的优势，因为硬件要方便升级的基础是不会引起软件的更改（或极少更改），要想达到应用软件与硬件的无关性，必须采用操作系统实现硬件隔离。第三项要求是考查考生根据用户软件的技术要求，如何考虑操作系统的分析方法。要支持任务间的数据隔离，具有进程管理的操作系统最为合适，这里考查了考生必须深入了解什么是操作系统的进程概念。第四项要求是考查考生掌握设计高可靠性设备通常采用的技术。

如果考生理解了上述 4 个用户要求，就能容易地回答本题的三个问题。

【解析】

本题给出的 TLS 结构框架主要通过硬件抽象，形成模块支持层软件，将操作系统对硬件资源的管理抽象成对一种逻辑上相关的抽象资源的管理，这样就将操作系统软件与硬件进行了分离。而应用软件访问硬件资源靠操作系统提供的服务进行，这样，应用软件仅仅和具体功能实现有关，而不需关心硬件配置。因此，TLS 结构框架的主要特点可概述为以下 4 点：

- (1) 应用软件仅与操作系统服务相关，不直接操作硬件。

(2) 操作系统通过模块支持原软件访问硬件，可与具体硬件无关。

(3) 模块支持层将硬件抽象成标准操作。

(4) 通过三层栈的划分可实现硬件的快速更改与升级，应用软件的升级不会引起硬件的变更。

基于这种思想，TLS 结构框架实现了宇航设备中软件功能服务，这样考生就可以完整地叙述各层软件的设计内涵，即：

(1) 应用层主要完成宇航设备的具体工作，由多个功能任务组成，各功能任务间的隔离由操作系统层实现。

(2) 操作系统层实现应用软件与硬件的隔离，为应用软件提供更加丰富的计算机资源服务。操作系统为应用软件提供标准的 API 接口（如 POSIX），确保了应用软件的可升级性。

(3) 模块支持层为操作系统管理硬件资源提供统一管理方法，用一种抽象的标准接口实现软件与硬件的无关性，达到硬件的升级要求，便于硬件的外场快速更换。

【问题 2】（10 分）

在 TLS 软件架构的基础上，关于选择哪种类型的嵌入式操作系统问题，李工与总工程师发生了严重分歧。李工认为，宇航系统是实时系统，操作系统的处理时间越快越好，隔离意味着以时间作代价，没有必要，建议选择类似于 VxWorks5.5 的操作系统；总工程师认为，应用软件间隔离是宇航系统安全性要求，宇航系统在选择操作系统时必须考虑这一点，建议选择类似于 Linux 的操作系统。

请说明两种操作系统的主要差异，完成下表中的空白部分，并针对本任务要求，用 200 字以内的文字说明你选择操作系统的类型和理由。

两种操作系统的主要差异

比较类型	VxWorks5.5	Linux
工作方式	操作系统与应用程序处于同一存储空间	①
多任务支持	支持多任务（线程）操作	②
实时性	③	实时系统
安全性	④	⑤
标准API	支持	支持

两种操作系统的差异见下表。

比较类型	VxWorks5.5	Linux
工作方式		①操作系统与应用程序处于不同存储空间
多任务支持		②支持多进程、多线程操作
实时性	③硬实时系统	
安全性	④任务间无隔离保护	⑤支持进程间隔离保护
标准API		

选择类似于 Linux 的嵌入式操作系统。理由如下：

(1) Linux 操作系统是一种安全性较强的操作系统。内核工作在系统态，应用软件工作在用户态，可以有效防止应用软件对操作系统的破坏。

(2) Linux 操作系统调度的最小单位是线程，线程归属于进程，进程具有自己独立的资源。进程通过存储器管理部件（MMU）实现多功能应用间隔离。

(3) 嵌入式 Linux 操作系统支持硬件抽象，可有效实现 TLS 结构，并将硬件抽象与操作系统分离，可方便实现硬件的外场快速更换。

【解析】

选择操作系统的依据是本题第三项的用户要求，主要考查考生对目前常用的嵌入式操作系统的了解程度。第一小问的表格给出的 VxWorks 和 Linux 两个操作系统是嵌入式系统常用的软件，两个软件在资源管理上存在本质区别，考生必须明确差异后，方能回答第二小问。两种操作系统的差异如下表所示。

两种操作系统的主要差异

比较类型	VxWorks5.5	Linux
工作方式	操作系统与应用程序处于同一存储空间	①操作系统与应用程序处于不同存储空间
多任务支持	支持多任务（线程）操作	②支持多进程、多线程操作
实时性	③硬实时系统	实时系统
安全性	④任务间无隔离保护	⑤支持进程间隔离保护
标准 API	支持	支持

通过比较，显然选用类似于 Linux 的嵌入式操作系统适用本题要求。理由包括三点：

(1) Linux 操作系统是一种安全性较强的操作系统。内核工作在系统态，应用软件工作在用户态（这点是系统安全性要求），可以有效防止应用软件对操作系统的破坏。

(2) Linux 操作系统调度的最小单位是线程，线程归属于进程，进程具有自己独立的资源。进程通过存储器管理部件（MMU）实现多功能应用间隔离。

(3) 嵌入式 Linux 操作系统支持硬件抽象，可有效实现 TLS 结构，并将硬件抽象与操作系统分离，可方便实现硬件的外场快速更换。

【问题 3】（7 分）

故障处理是宇航系统软件设计中极为重要的组成部分。故障处理主要包括故障监视、故障定位、故障隔离和系统容错（重组）。用 150 字以内的文字说明嵌入式系统中故障主要分哪几类？并分别给出两种常用的故障滤波算法和容错算法。

(1) 嵌入式系统中故障主要分为：

- ①硬件故障：如 CPU、存储器和定时器等；
- ②应用软件故障：如数值越界、异常和超时等；
- ③操作系统故障：如越权访问、死锁和资源枯竭等。

(2) 滤波算法：

- ①门限算法
- ②递减算法
- ③递增算法
- ④周期滤波算法

(3) 容错算法：

- ①N+1 备份
- ②冷备
- ③温备
- ④热备

【解析】

作为宇航系统的嵌入式设备，硬 / 软件故障是宇航系统最为关注的内容，根据宇航系统的特点和系统组成，故障一般分为三类，即硬件故障、应用软件故障和操作系统故障，在设计中，应考虑这三类故障的处理方法。硬件故障一般包括 CPU 运算错误、存储器访问 / 越界错误、MMU 配置错误、定时器计数错误和内总线错误等；应用软件故障一般包括计算越界、除 0、溢出和超时等各种异常情况；操作系统故障一般包括越权访问、死锁、资源枯竭、调度超时、配置越界和操作系统异常等。

一般情况下，宇航系统的故障分为瞬态故障和永久故障，瞬态故障是指偶然发生的错误，而永久故障是指发生后不可消失的错误。在容错系统中，故障一般是由瞬态故障向永久故障转变，将鉴别瞬态故障和永久故障的算法称为滤波算法。在嵌入式实时系统中，常用的滤波算法包括门限算法、递减算法、递增算法和周期滤波算法等。

故障的鉴别目的是实现宇航系统的容错与重构。常用的容错算法是 N+1 备份、冷备、温备和热备。N+1 备份是指 N 个通用模块之一的任何一个模块发生故障后，将故障模块的任务迁移到备份模块运行；冷备、温备和热备是三种备份方式，可根据宇航系统的总体备份时间或重要程度安排不同的备份算法。

试题四

某公司欲开发一个车辆定速巡航控制系统，以确保车辆在不断变化的地形中以固定的速度行驶。该系统的简化示意图如下图所示。各种系统输入的含义见下表。

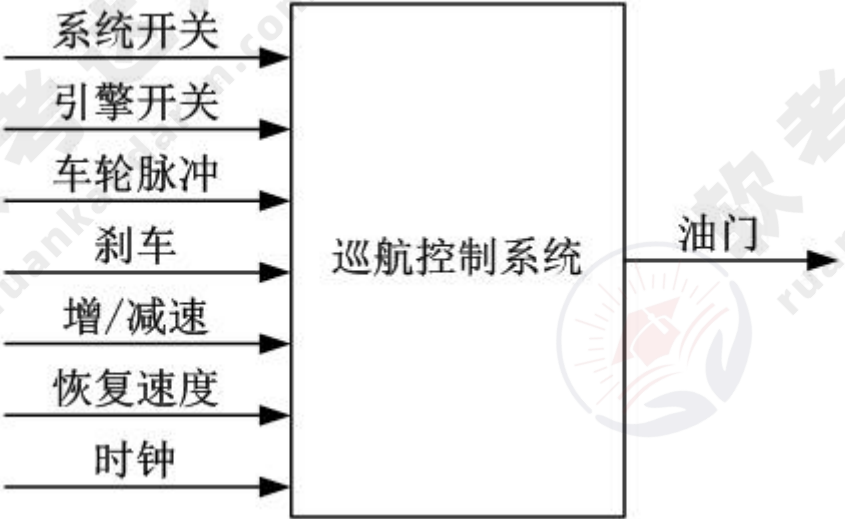


图 4-1 定速巡航控制系统的简化示意图

表 4-1 定速巡航控制系统输入说明

输入名称	作用
系统开关	开启/关闭巡航控制系统
引擎开关	开启/关闭洗车引擎（引擎开启时，巡航控制系统处于就绪状态）
车轮脉冲	车轮每转一次，相应地发出一次脉冲
刹车	当刹车被踩下时，定速巡航控制系统会临时恢复到人工控制
增/减速	增加或减慢当前车速（仅在定速巡航控制系统处于开启的状态下可用）
恢复速度	恢复原来保持的车速（仅在定速巡航控制系统处于开启的状态下可用）
时钟	每毫秒定时脉冲

公司的领域专家对需求进行深入分析后，将系统需求认定为：任何时刻，只要定速巡航控制系统处于工作状态，就要有确定的期望速度，并通过调整引擎油门的设定值来维持期望速度。

在对车辆定速巡航控制系统的架构进行设计时，公司的架构师王工提出采用面向对象的架构风格，而李工则主张采用控制环路的架构风格。在架构评估会议上，专家对这两种方案进行综合评价，最终采用了面向对象和控制环路相结合的混合架构风格。

【问题 1】(5 分)

在实际的软件项目开发中，采用成熟的架构风格是项目成功的保证。请用 200 字以内的文字说明：什么是软件架构风格；面向对象和控制环路两种架构风格各自的特点。

软件架构风格是描述某一类特定应用领域中软件系统组织方式和惯用方式。

面向对象架构风格的特征是将数据表示和基本操作封装在对象中。这种模式的构件是对象，对象维护自身表示的完整性，对象之间通过消息机制进行通信，对象交互时需要知道彼此的标识，通过对象之间的协作完成计算过程。

控制环路架构风格是将过程输出的指定属性维护在一个特定的参考值（设定点）。控制环路风格包括过程变量、被控变量、输入变量、操纵变量和设定点等构件，通过收集实际和理想的过程状态信息，并能调整过程变量使得实际状态趋于理想状态。

【问题 2】(12 分)

用户需求没有明确给出该系统如何根据输入集合计算输出。请用 300 字以内的文字针对该系统的增减速功能，分别给出两种架构风格中的主要构件，并详细描述计算过程。

对于系统的增减速功能，来用面向对象风格的巡航控制系统首先会定义司机、油门、时钟、速度计和车轮等构件 a 整个计算的主要过程是：

- (1) 司机进行增/减速操作设置期望速度，该期望速度以消息的形式传递给速度计；
- (2) 速度计通过向车轮和时钟发送消息获取车轮转速和时钟值，得到当前速度；
- (3) 速度计计算前速度和期望速度的速度差值；
- (4) 该差值以消息的形式发送给油门，油门通过速度差值调节自身状态；
- (5) 整个过程在时钟的控制下定期向速度计发送消息，重复执行 (2) ~ (4)。

控制环路的架构风格以控制器为核心，期望速度、车轮脉冲、时钟和油门等作为

构件。

具体的计算过程是：

- (1) 司机进行增/减速操作设置期望速度值；
- (2) 将设定值置为期望速度值；
- (3) 控制器采集车轮脉冲和时钟值，计算出当前速度；
- (4) 比较期望速度和当前速度，计算速度差值，控制油门动作；
- (5) 反复执行(3)和(4)。

【问题3】(8分)

实际的软件系统架构通常是多种架构风格的混合，不同的架构风格都有其适合的应用场景。以该系统为例，针对面向对象架构风格和控制环路架构风格，各给出两个适合的应用场景，并简要说明理由。

适合面向对象架构风格的应用场景：

(1) 用户刹车，立即退出巡航控制系统。理由：这是一个典型的事件驱动的场景，适合于面向对象风格。

(2) 系统对突发事件的处理，如某些部件失灵等。理由：当发生突发事件时，系统会同时产生数据和事件，这种情况用对象建模较为恰当。

适合面向控制环路架构风格的应用场景：

(1) 在达到期望速度后，系统维持恒定速度行驶。理由：这是一个典型的闭环控制的情景，系统需要在外界情况不断发生变化的情况下进行调整，使得系统状态尽可能接近期望状态。

(2) 用户改变期望速度后，系统不断进行调整，直至到达恒定速度。理由：这是一个闭环控制情景，当用户设定期望速度值后，系统需要在不断获取当前速度和外界条件的情况下对系统状态持续调整，使得系统状态尽可能接近这个新的期望状态。

试题五

某企业根据业务扩张的要求，需要将原有的业务系统扩展到互联网上，建立自己的 B2C 业务系统，此时系统的安全性成为一个非常重要的设计需求。为此，该企业向软件开发商提出如下要求：

- (1) 合法用户可以安全地使用该系统完成业务；
- (2) 灵活的用户权限管理；
- (3) 保护系统数据的安全，不会发生信息泄漏和数据损坏；
- (4) 防止来自于互联网上各种恶意攻击；
- (5) 业务系统涉及到各种订单和资金的管理，需要防止授权侵犯；
- (6) 业务系统直接面向最终用户，需要在系统中保留用户使用痕迹，以应对可能的商业诉讼。

该软件开发商接受任务后，成立方案设计小组，提出的设计方案是：在原有业务系统的基础上，保留了原业务系统中的认证和访问控制模块；为了防止来自互联网的威胁，增加了防火墙和入侵检测系统。

企业和软件开发商共同组成方案评审会，对该方案进行了评审，各位专家对该方案提出了多点不同意见。李工认为，原业务系统只针对企业内部员工，采用了用户名/密码方式是可以的，但扩展为基于互联网的 B2C 业务系统后，认证方式过于简单，很可能造成用户身份被盗取；王工认为，防止授权侵犯和保留用户痕迹的要求在方案中没有体现。而刘工则认为，即使是在原有业务系统上的扩展与改造，也必须全面考虑信息系统面临的各种威胁，设计完整的系统安全架构，而不是修修补补。

【问题 1】（9 分）

信息系统面临的安全威胁多种多样，来自多个方面。请指出信息系统面临哪些方面的安全威胁并分别予以简要描述。

信息系统面临的安全威胁来自于物理环境、通信链路、网络系统、操作系统、应用系统以及管理等多个方面。

物理安全威胁是指对系统所用设备的威胁，如自然灾害、电源故障、数据库故障和设备被盗等造成数据丢失或信息泄漏。

通信链路安全威胁是指在传输线路上安装窃听装置或对通信链路进行干扰。

网络安全威胁当前主要是指由于因特网的开放性、国际性与无安全管理性，对内部网络形成的严重安全威胁。

操作系统安全威胁指的是操作系统本身的后门或安全缺陷，如“木马”和“陷阱门”等。

应用系统安全威胁是指对于网络服务或用户业务系统安全的威胁，包括应用系统自身漏洞，也受到“木马”的威胁。

管理系统安全威胁指的是人员管理和各种安全管理制度。

【问题2】（8分）

认证是安全系统中不可缺少的环节，请简要描述主要的认证方式，并说明该企业应采用哪种认证方式。

目前主要的认证方式有三类：

(1) 用户名和口令认证：主要是通过一个客户端与服务器知的口令（或与口令相关的数据）进行验证。根据处理形式的不同，分为验证数据的明文传送、利用单向散列函数处理验证数据、利用单向散列函数和随机数处理验证数据。

(2) 使用令牌认证：该方式中，进行验证的密钥存储于令牌中，目前的令牌包括安全证书和智能卡等方式。

(3) 生物识别认证：主要是根据认证者的图像、指纹、气味和声音等作为认证数据。根据该企业的业务特征，采用令牌认证较为合适。

【问题3】（8分）

请解释授权侵犯的具体含义；针对王工的意见给出相应的解决方案，说明该解决方案的名称、内容和目标。

授权侵犯指的是被授权以某一目的使用某一系统或资源的某个人，将此权限用于其他非授权的目的，也称作“内部攻击”。

针对王工的建议，从系统安全架构设计的角度需要提供抗抵赖框架。

抗抵赖服务包括证据的生成、验证和记录，以及在解决纠纷时随即进行的证据恢复和再次验证。

框架中抗抵赖服务的目的是提供有关特定事件或行为的证据。例如，必须确认数据原发者和接收者的身份和数据完整性，在某些情况下，可能需要涉及上下文关系（如日期、时间、原发者/接收者的地点等）的证据，等等。

【解析】 本题考查信息系统的安全威胁以及采用的常用方案。

信息系统面临的安全主要包括信息系统所依赖环境的安全、信息系统自身安全和使用信息系统的人员管理和相关规章制度。

信息系统所依赖的环境带来的安全威胁有物理环境、通信链路和操作系统。物理安全威胁是指对系统所用设备的威胁，如自然灾害、电源故障、数据库故障和设备被盗等造成数据丢失或信息泄漏。通信链路安全威胁是指在传输线路上安装窃听装置或对通信链路进行干扰。网络安全威胁当前主要是指由于因特网的开放性、国际性与无安全管理性，对内部网络形成的严重安全威胁。操作系统安全威胁指的是操作系统本身的后门或安全缺陷，如“木马”和“陷阱门”等。

应用系统安全威胁是指对于网络服务或用户业务系统安全的威胁，包括应用系统自身漏洞，也受到“木马”的威胁。

管理系统安全威胁指的是人员管理和各种安全管理制度。所谓的安全措施，是指“三分技术，七分管理”。

在信息系统安全中，认证是必不可少的环节。常见的认证方式有三种：（1）用户名和口令认证：主要是通过一个客户端与服务器共知的口令（或与口令相关的数据）进行验证。根据处理形式的不同，分为验证数据的明文传送、利用单向散列函数处理验证数据、利用单向散列函数和随机数处理验证数据。（2）使用令牌认证：该方式中，进行验证的密钥存储于令牌中，目前的令牌包括安全证书和智能卡等方式。（3）生物识别认证：主要是根据认证者的图像、指纹、气味和声音等作为认证数据。根据该企业原有信息系统中的认证方式，并考虑到接入因特网后的用户特征，采用令牌认证的方式较为适合。

目前威胁信息系统安全中，有相当一部分是由于内部人员犯罪所引起的，即授权侵犯。授权侵犯指的是被授权以某一目的使用某一系统或资源的某个人，将此权限用于其他非授权的目的，也称作“内部攻击”。防止授权侵犯的主要手段是提供类似于审计的功能，从系统安全架构设计的角度来讲，即提供抗抵赖框架。抗抵赖服务包括证据的生成、验证和记录，以及在解决纠纷时随即进行的证据恢复和再次验证。框架中抗抵赖服务的目的是提供有关特定事件或行为的证据。例如，必须确认数据原发者和接收者的身份和数据完整性，在某些情况下，可能需要涉及上下文关系（如日期、时间、原发者 / 接收者的地点等）的证据，等等。

试题一 论基于 DSSA 的软件架构设计与应用

软件架构设计的一个重要课题是如何解决软件重用问题。特定领域软件架构 (Domain Specific Software Architecture, DSSA) 是一种有效实现特定领域软件重用的手段。按照 Tracz 的说法, DSSA 就是一个特定的问题领域中由领域模型、参考需求、参考架构等组成的开发基础架构, 其目标就是支持一个特定领域中多个应用的生成。

DSSA 的基本活动包括领域分析、领域设计和领域实现。领域分析的主要目的是获得领域模型, 领域模型描述领域中系统之间共同的需求, 即领域需求; 领域设计的主要目标是获得 DSSA, DSSA 描述领域模型中表示需求的解决方案; 领域实现的主要目标是依据领域模型和 DSSA 开发和组织可重用信息。

请围绕“基于 DSSA 的软件架构设计与应用”论, 依次从以下三个方面进行论述。

1. 概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。
2. 就你所熟悉的领域, 请给出针对该特定领域, 在基于 DSSA 的软件设计开发中所涉及的领域模型、参考需求和参考架构以及相应的支持环境或设施。
3. 具体阐述你参与管理和开发的项目中使用 DSSA 的情况, 包括领域分析、领域设计和领域实现等活动是如何具体实施的, 最终实际效果如何。

一、简要叙述所参与管理和开发的软件项目, 需要明确指出在其中承担的主要任务和开展的主要工作。

二、应结合自己所熟悉的领域, 定义领域范围, 确定领域应用需要满足的用户需求; 定义领域特定的元素、领域字典和领域术语; 定义领域特定的设计和实现需求约束; 在此基础上, 定义领域模型, 产生该领域的参考架构, 并说明构件的语法和语义; 最后, 产生、搜集可重用的产品单元, 为 DSSA 增加构件, 为问题域实现新应用提供支持。这个 DSSA 的建立过程是并发、递归和反复进行的。

所给出的 DSSA 应该具备以下 4 个方面的特征:

- (1) 一个严格定义的问题域和/或解决域;
- (2) 具有普遍性, 使其可以用于领域中某个特定应用的开发;
- (3) 对整个领域能有合适程度的抽象;
- (4) 具备该领域固定的、典型的在开发过程中的可重用元素。

三、需要结合项目实际, 指出在架构设计时使用 DSSA 的情况, 包括领域分析、领域设计和领域实现等活动是如何具体实施的, 要给出实际的效果并进行分析。

试题二 论信息系统建模方法

系统模型在软件开发中扮演着重要的角色。可为已有的系统创建模型，以便更好地理解这些系统；也可以针对待开发的系统创建模型，作为记录业务需求或技术设计的方法。模型是建立信息系统的基础。恰当地运用信息系统建模方法，是成功地进行软件开发的一个关键环节。

请围绕“论信息系统建模方法”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与管理和开发的信息系统项目以及你在其中所承担的主要工作。
2. 论述常见的信息系统建模方法的主要内容（包括每种建模方法的核心思想以及所创建的模型）。
3. 具体阐述你参与管理和开发的项目中选择使用的信息系统建模方法以及选择该方法的原因，给出具体的实施过程和实施效果。

一、应结合自己参与的信息系统项目，说明在其中所承担的工作。

二、需要较为详细地说明目前各种常见的信息系统建模方法的核心思想，并对每种方法所创建的模型进行简要描述。

(1) 结构化建模方法。

结构化建模方法是以过程为中心的技术，可用于分析一个现有的系统以及定义新系统的业务需求。结构化建模方法所绘制的模型称为数据流图（DFD）。

(2) 信息工程建模方法（或数据库建模方法）。

信息工程建模方法是一种以数据为中心，但过程敏感的技术，它强调在分析和研究过程需求之前，首先研究和分析数据需求。信息工程建模方法所创建的模型被称为实体联系图（ERD）。

(3) 面向对象建模方法。

面向对象建模方法将“数据”和“过程”集成到被称为“对象”的结构中，消除了数据和过程的人为分离现象。面向对象建模方法所创建的模型被称为对象模型。随着面向对象技术的不断发展和应用，形成了面向对象的建模标准，即 UML(统一建模语言)。UML 定义了几种不同类型的模型图，这些模型图以对象的形式共建一个信息系统或应用系统。

三、论文中需要结合项目实际工作，详细论述在项目是如何使用所选定的信息系统建模方法创建系统的逻辑模型和物理模型，并具体说明这些模型对项目开发所产生的影响。

试题三 论基于 REST 服务的 Web 应用系统设计

REST (REpresentational State Transfer)是指从几种基于网络的架构风格衍生出来的一种混合架构风格,它是目前互联网的核心架构风格。基于 REST 服务(RESTful Service) 的 Web 应用系统设计任务主要包括:识别并设计 REST 风格的服务,采用面向服务的思想进行 REST 服务集成。采用这种方法设计的 Web 应用系统能够结合 REST 风格和面向服务思想优点,近年来受到了广泛的关注。

请围绕“基于 REST 服务的 Web 应用系统设计”论题,依次从以下三个方面进行论述。

1. 概要叙述你参与实施的 Web 应用系统开发项目以及你所承担的主要工作。
2. 简要叙述与传统的 Web 服务相比,采用 REST 服务构建的 Web 应用具有哪些优势和不足。
3. 阐述你在设计基于 REST 服务的 Web 应用系统时遇到了哪些问题,如何解决。

一、论文中要具体介绍项目的总体需求(特别是质量属性需求)、Web 应用系统的逻辑与物理拓扑结构、采用的技术等内容和承担的实际工作。

二、REST (REpresentational State Transfer)是指从几种基于网络的架构风格衍生出来的一种混合架构风格,目前 Web 的体系结构正是基于 REST 风格的。REST 风格中的特点是客户端/服务器、无状态、缓存、统一接口、分层系统和按需代码。REST 组件通过以一种数据格式转移资源的表述进行通信,可以基于接收者的能力和期待的内容,以及资源的性质动态地选择不同的表述。

与传统的 Web 服务相比,REST 服务主要有以下优势:

- (1) REST 服务基于 W3C/IETF 的标准与规范(包括 HTTP、XML、URI 和 MIME 等),其实现技术简单、成熟。
- (2) REST 服务基于 URI 和超链接技术,不需要通过集中式的服务信息仓库即可发现服务资源。
- (3) REST 服务支持缓存,具有无状态的特性,这些使得 REST 服务能够支持大量客户端,构建的应用系统具有较强的伸缩性。
- (4) REST 服务基于轻量级的 Web 框架,仅仅需要基本的开发工具支持,构建过程简单且成本较低。
- (5) REST 服务的测试相对简单,采用浏览器即可完成服务功能测试。

与传统的 Web 服务相比,REST 服务主要存在如下不足:

(1) REST 服务倡导的 REST 风格与实际实现尚存在一定差距。例如高层 REST 服务倡导使用 GET、PUT、POST 和 DELETE 所有 4 个统一接口，在 REST 实现部分通常只能采用 GET 和 POST 接口，因为大多数的代理和防火墙会屏蔽其他接口；并且 XHTML 表单中只能使用 GET 和 POST 接口。

(2) REST 服务要求所有的输入参数都必须在 URI 中传递，这样会产生对参数容量大小的限制（目前的大小是 4KB）。如果超出该数量，会导致 HTTP 协议错误（错误代码 414: Request-URI too long）。

(3) 在 URI 中表达复杂类型的参数比较困难，且目前对 URI 中的参数不存在一种公认的编组（marshalling）和解编（un-marshalling）方法。

三、进行基于 REST 服务的 Web 应用系统的设计时可能遇到的问题包括：如何识别并设计 REST 风格服务；构建 REST 服务的运行时环境，包括 HTTP 服务器应用服务器选型等；富客户端表现方式及编程语言的选择；系统逻辑与物理拓扑结构的分析与设计等。

试题四 论软件可靠性设计与应用

目前在企业中，以软件为核心的产品得到了广泛的应用。随着系统中软件部分比例的不断增加，使得系统对软件的依赖性越来越强，对软件的可靠性要求也越来越高。软件可靠性与其他质量属性一样，是衡量软件架构的重要指标。

软件工程中已有很多比较成熟的设计技术，如结构化设计、模块化设计、自顶向下设计等，这些技术为保障软件的整体质量发挥了重要作用。在此基础上，为了进一步提高软件的可靠性，通常会采用一些特殊的设计技术，即软件可靠性设计技术。

在软件可靠性工程体系中，包含有可靠性模型与预测、可靠性设计和可靠性测试方法等。实践证明，保障软件可靠性最有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制。

请围绕“软件可靠性设计与应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与实施的软件开发项目以及你所承担的主要工作。
2. 简要叙述影响软件可靠性的因素有哪些。
3. 阐述常用的软件可靠性设计技术以及你如何应用到实际项目中，效果如何。

一、论文中要具体介绍项目的总体需求（特别是可靠性需求）、采用的技术等内容和承担的实际工作。

二、影响软件可靠性的主要因素有：运行环境（软件可靠性的定义是相对于运行环境的）；软件规模；软件内部结构（内部结构越复杂，包含的缺陷数就可能越多）；软件的开发方法和开发环境；软件的可靠性投入等。

三、可靠性设计是在常规的软件设计中，应用各种方法和技术使程序设计在兼顾用户功能和性能需求的同时，全面满足软件的可靠性要求。软件可靠性设计技术就是以提高和保障软件的可靠性为目的，在软件设计阶段运用的一种特殊的设计技术。

主要的软件可靠性设计技术包括：

(1)容错设计技术。对于软件失效后果特别严重的场合，例如宇航器控制系统、空中交通控制和核反应堆控制系统等，可采用容错设计方法。常用的软件容错技术主要有恢复块设计、N 版本程序设计和冗余设计。恢复块设计就是选择一组操作作为容错设计单元，从而把普通的程序块变为恢复块。一个恢复块中包含有若干功能相同、设计差异的程序块，每一时刻有一个程序块处于运行状态，一旦某程序块出现故障，则用备份程序块予以替换。N 版本程序设计的核心是通过设计出多个模块或不同版本，对于相同初始条件和相同输入的操作结

果进行多数表决（防止因其中某一软件模块/版本的故障而提供了错误的服务，以实现软件容错>。冗余设计的思路来源于硬件系统，但有所不同。软件冗余设计技术是采用多种不同路径、不同算法或不同实现方法的模块或系统作为备份，在出现故障时进行替换，维持系统的正常运行。

(2) 检测技术。在无须在线容错或不能采用冗余设计技术的部分，俾又有较高的可靠性要求时，一般采用检测性设计，在软件出现故障后能及时发现并报警。但其明显的缺点是不能自动解决故障，如果没有人工干预，最终将导致系统不能正常运行。

(3) 降低复杂度设计。软件的复杂性与软件可靠性有密切关系。软件复杂性是产生软件缺陷的重要根源。降低复杂度设计的思想就是在保证实现软件功能基础上，简化软件结构。