

希赛网, 专注于软考、PMP、通信考试的专业 IT 知识库和在线教育平台, 希赛网在线题库, 提供历年真题、模拟试题、章节练习、知识点练习、错题本练习等在线做题服务, 更有能力评估报告, 让你告别盲目做题, 针对性地攻破自己的薄弱点, 备考更高效。

希赛网官网: www.educity.cn

希赛网软件水平考试网: www.educity.cn/rk

希赛网在线题库: <http://www.educity.cn/tiku/>

2013 年架构综合知识真题答案与解析: <http://www.educity.cn/tiku/tp18933.html>

2013 年系统架构设计师考试真题 (综合)

某操作系统采用分页存储管理方式, 下图给出了进程 A 和进程 B 的页表结构。如果物理页的大小为 512 字节, 那么进程 A 逻辑地址为 1111 (十进制) 的变量存放在 (1) 号物理内存页中。假设进程 A 的逻辑页 4 与进程 B 的逻辑页 5 要共享物理页 8, 那么应该在进程 A 页表的逻辑页 4 和进程 B 页表的逻辑页 5 对应的物理页处分别填 (2)。

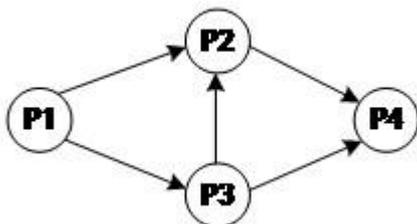
| 进程 A 页表 | |
|---------|-----|
| 逻辑页 | 物理页 |
| 0 | 9 |
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | |
| 5 | |

| 进程 B 页表 | |
|---------|-----|
| 逻辑页 | 物理页 |
| 0 | 1 |
| 1 | 3 |
| 2 | 5 |
| 3 | 7 |
| 4 | 2 |
| 5 | |

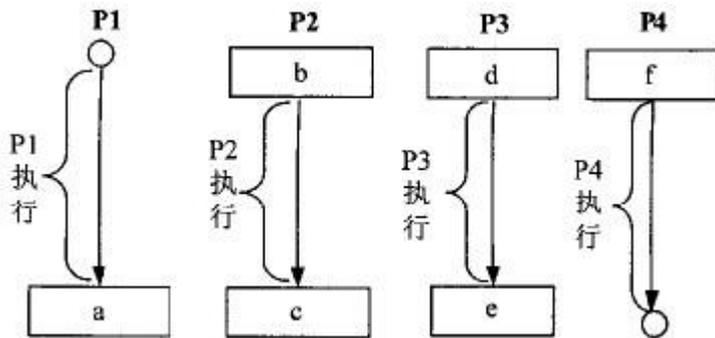
| 物理页 |
|-----|
| 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |

- (1) A. 9
 B. 2
 C. 4
 D. 6
- (2) A. 4、5
 B. 5、4
 C. 5、8
 D. 8、8

进程 P1、P2、P3 和 P4 的前趋图如下所示:



若用 PV 操作控制进程 P1~P4 并发执行的过程, 则需要设置 5 个信号量 S1、S2、S3、S4 和 S5, 且信号量 S4-S5 的初值都等于 0。下图中 a、b 和 c 处应分别填写 (3); d、e 和 f 处应分别填写 (4)。



- (3) A. V(S1)V(S2)、P(S1)V(S3) 和 V(S4)
 B. P(S1)V(S2)、P(S1)P(S2) 和 V(S1)
 C. V(S1)V(S2)、P(S1)P(S3) 和 V(S4)
 D. P(S1)P(S2)、V(S1)P(S3) 和 V(S2)
- (4) A. P(S2)、V(S3)V(S5) 和 P(S4)P(S5)
 B. V(S2)、P(S3)V(S5) 和 V(S4)P(S5)
 C. P(S2)、V(S3)P(S5) 和 P(S4)V(S5)
 D. V(S2)、V(S3)P(S5) 和 P(S4)V(S5)

假设关系模式 R(U, F), 属性集 U={A, B, C}, 函数依赖集 F={A→B, B→C}。若将其分解为 ρ={R1(U1, F1), R2(U2, F2)}, 其中 U1={A, B}, U2={A, C}。那么, 关系模式 R、R1、R2 分别达到了 (5); 分解 ρ (6)。

- (5) A. 1NF、2NF、3NF
 B. 1NF、3NF、3NF
 C. 2NF、2NF、3NF
 D. 2NF、3NF、3NF
- (6) A. 有损连接但保持函数依赖
 B. 既无损连接又保持函数依赖
 C. 有损连接且不保持函数依赖
 D. 无损连接但不保持函数依赖

给定员工关系 EMP (EmpID, Ename, sex, age, tel, DepID), 其属性含义分别为: 员工号、姓名、性别、年龄、电话、部门号; 部门关系 DEP (DepID, Dname, Dtel, DEmpID), 其属性含义分别为: 部门号、部门名、电话, 负责人号。若要求 DepID 参照部门关系 DEP 的主码 DepID, 则可以在定义 EMP 时用 (7) 进行约束。若要查询开发部的负责人姓名、年龄, 则正确的关系代数表达式为 (8)。

- (7) A. Primary Key (DepID) On DEP (DepID)
 B. Primary Key (DepID) On EMP (DepID)
 C. Foreign Key (DepID) References DEP (DepID)
 D. Foreign Key (DepID) References EMP (DepID)
- (8) A. $\pi_{2,4}(\sigma_{8='开发部'}(EMP \times DEP))$
 B. $\pi_{2,4}(\sigma_{1=9}(EMP \bowtie \sigma_{2='开发部'}(DEP)))$

- C. $\pi_{2,3}(\text{EMP} \times \sigma_{2='开发部'}(\text{DEP}))$
- D. $\pi_{2,3}(\pi_{1,2,4,6}(\text{EMP}) \bowtie \sigma_{2='开发部'}(\text{DEP}))$

在实时操作系统中，两个任务并发执行，一个任务要等待另一个任务发来消息，或建立某个条件后再向前执行，这种制约性合作关系被称为任务的（9）。

- A. 同步
- B. 互斥
- C. 调度
- D. 执行

在嵌入式系统设计中，用来进行 CPU 调试的常用接口是（10）。

- A. PCI 接口
- B. USB 接口
- C. 网络接口
- D. JTAG 接口

看门狗(Watch Dog)是嵌入式系统中一种常用的保证系统可靠性的技术，（11）会产生看门狗中断。

- A. 软件喂狗
- B. 处理器温度过高
- C. 外部中断
- D. 看门狗定时器超时

以下关于实时操作系统（RTOS）任务调度器的叙述中，正确的是（12）。

- A. 任务之间的公平性是最重要的调度目标
- B. 大多数 RTOS 调度算法都是抢占方式（可剥夺方式）
- C. RTOS 调度器都采用了基于时间片轮转的调度算法
- D. 大多数 RTOS 调度算法只采用一种静态优先级调度算法

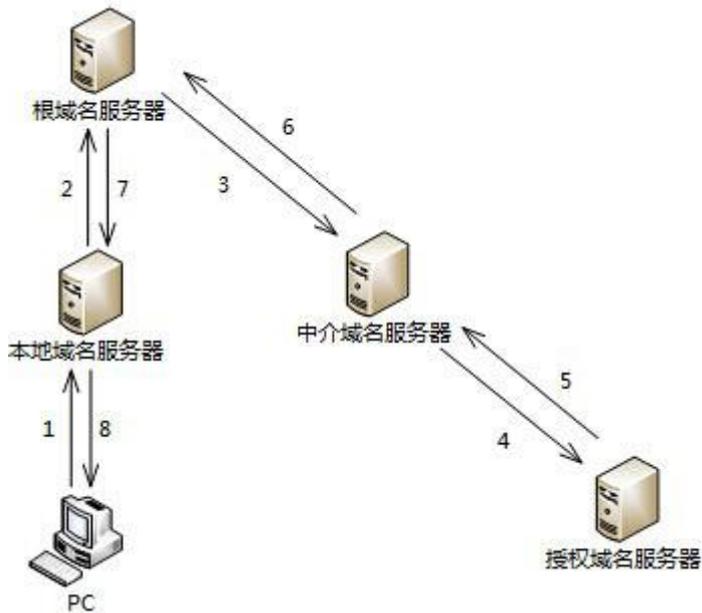
以下关于层次化网络设计原则的叙述中，错误的是（13）。

- A. 一般将网络划分为核心层、汇聚层、接入层三个层次
- B. 应当首先设计核心层，再根据必要的分析完成其他层次设计
- C. 为了保证网络的层次性，不能在设计中随意加入额外连接
- D. 除去接入层，其他层次应尽量采用模块化方式，模块间边界应非常清晰

网络需求分析包括网络总体需求分析、综合布线需求分析、网络可用性与可靠性分析、网络安全需求分析，此外还需要进行（14）。

- A. 工程造价估算
- B. 工程进度安排
- C. 硬件设备选型
- D. IP 地址分配分析

主机 PC 对某个域名进行查询，最终由该域名的授权域名服务器解析并返回结果，查询过程如下图所示。这种查询方式中不合理的是（15）。



- A. 根域名服务器采用递归查询, 影响了性能
- B. 根域名服务器采用迭代查询, 影响了性能
- C. 中介域名服务器采用迭代查询, 加重了根域名服务器负担
- D. 中介域名服务器采用递归查询, 加重了根域名服务器负担

把应用程序中应用最频繁的那部分核心程序作为评价计算机性能的标准程序, 称为 (16) 程序。(17) 不是对 Web 服务器进行性能评估的主要指标。

- (16) A. 仿真测试
 B. 核心测试
 C. 基准测试
 D. 标准测试
- (17) A. 丢包率
 B. 最大并发连接数
 C. 响应延迟
 D. 吞吐量

与电子政务相关的行为主体主要有三个, 即 (18), 政府的业务活动也主要围绕着这三个行为主体展开。

- A. 政府、数据及电子政务系统
- B. 政府、企(事)业单位及中介
- C. 政府、服务机构及企事业单位
- D. 政府、企(事)业单位及公民

企业信息化涉及到对企业管理理念的创新, 按照市场发展的要求, 对企业现有的管理流程重新整合, 管理核心从对 (19) 的管理, 转向对 (20) 的管理, 并延伸到对企业技术创新、工艺设计、产品设计、生产制造过程的管理, 进而还要扩展到对 (21) 的管理乃至发展到电子商务。

- (19) A. 人力资源和物资

- B. 信息技术和知识
- C. 财务和物料
- D. 业务流程和数据

(20) A. 业务流程和数据

- B. 企业信息系统和技术
- C. 业务流程、数据和接口
- D. 技术、物资和人力资源

(21) A. 客户关系和供应链

- B. 信息技术和知识
- C. 生产技术和信息技术
- D. 信息采集、存储和共享

企业信息集成按照组织范围分为企业内部的信息集成和外部的信息集成。在企业内部的信息集成中，(22) 实现了不同系统之间的互操作，使得不同系统之间能够实现数据和方法的共享；(23) 实现了不同应用系统之间的连接、协调运作和信息共享。

(22) A. 技术平台集成

- B. 数据集成
- C. 应用系统集成
- D. 业务过程集成

(23) A. 技术平台集成

- B. 数据集成
- C. 应用系统集成
- D. 业务过程集成

数据挖掘是从数据库的大量数据中揭示出隐含的、先前未知的并有潜在价值的信息的非平凡过程，主要任务有(24)。

- A. 聚类分析、联机分析、信息检索等
- B. 信息检索、聚类分析、分类分析等
- C. 聚类分析、分类分析、关联规则挖掘等
- D. 分类分析、联机分析、关联规则挖掘等

详细的项目范围说明书是项目成功的关键，(25) 不属于项目范围定义的输入。

- A. 项目章程
- B. 项目范围管理计划
- C. 批准的变更申请
- D. 项目文档管理方法

活动定义是项目时间管理中的过程之一，(26) 是进行活动定义时通常使用的一种工具。

- A. Gantt 图
- B. 活动图
- C. 工作分解结构 (WBS)
- D. PERT 图

以下叙述中，(27) 不属于可行性分析的范畴。

- A. 对系统开发的各种候选方案进行成本/效益分析
- B. 分析现有系统存在的运行问题
- C. 评价该项目实施后可能取得的无形收益
- D. 评估现有技术能力和信息技术是否足以支持系统目标的实现

遗留系统的演化可以采用淘汰、继承、改造和集成四种策略。若企业中的遗留系统技术含量较高,业务价值较低,在局部领域中工作良好,形成了一个信息孤岛时,适合于采用(28)演化策略。

- A. 淘汰
- B. 继承
- C. 改造
- D. 集成

逆向工程导出的信息可以分为实现级、结构级、功能级和领域级四个抽象层次。程序的抽象语法树属于(29);反映程序分量之间相互依赖关系的信息属于(30)。

- (29) A. 实现级
- B. 结构级
 - C. 功能级
 - D. 领域级

- (30) A. 实现级
- B. 结构级
 - C. 功能级
 - D. 领域级

在面向对象设计中,(31)可以实现界面控制、外部接口和环境隔离。(32)作为完成用例业务的责任承担者,协调、控制其他类共同完成用例规定的功能或行为。

- (31) A. 实体类
- B. 控制类
 - C. 边界类
 - D. 交互类

- (32) A. 实体类
- B. 控制类
 - C. 边界类
 - D. 交互类

基于RUP的软件过程是一个迭代过程。一个开发周期包括初始、细化、构建和移交四个阶段,每次通过这四个阶段就会产生一代软件,其中建立完善的架构是(33)阶段的任务。采用迭代式开发,(34)。

- (33) A. 初始
- B. 细化
 - C. 构建
 - D. 移交

- (34) A. 在每一轮迭代中都要进行测试与集成
- B. 每一轮迭代的重点是对特定的用例进行部分实现

- C. 在后续迭代中强调用户的主动参与
- D. 通常以功能分解为基础

某系统中的文本显示类 (TextView) 和图片显示类 (PictureView) 都继承了组件类 (Component), 分别显示文本和图片内容, 现需要构造带有滚动条或者带有黑色边框, 或者既有滚动条又有黑色边框的文本显示控件和图片显示控件, 但希望最多只增加 3 个类。那么采用设计模式 (35) 可实现该需求, 其优点是 (36)。

- (35) A. 外观
B. 单体
C. 装饰
D. 模板方法
- (36) A. 比静态继承具有更大的灵活性
B. 提高已有功能的重复使用性
C. 可以将接口与实现相分离
D. 为复杂系统提供了简单接口

以下关于自顶向下开发方法的叙述中, 正确的是 (37)。

- A. 自顶向下过程因为单元测试而比较耗费时间
- B. 自顶向下过程可以更快地发现系统性能方面的问题
- C. 相对于自底向上方法, 自顶向下方法可以更快地得到系统的演示原型
- D. 在自顶向下的设计中, 如发现了一个错误, 通常是因为底层模块没有满足其规格说明 (因为高层模块已经被测试过了)

以下关于白盒测试方法的叙述中, 错误的是 (38)。

- A. 语句覆盖要求设计足够多的测试用例, 使程序中每条语句至少被执行一次
- B. 与判定覆盖相比, 条件覆盖增加对符合判定情况的测试, 增加了测试路径
- C. 判定/条件覆盖准则的缺点是未考虑条件的组合情况
- D. 组合覆盖要求设计足够多的测试用例, 使得每个判定中条件结果的所有可能组合最多出现一次

以下关于面向对象软件测试的叙述中, 正确的是 (39)。

- A. 在测试一个类时, 只要对该类的每个成员方法都进行充分的测试就完成了对该类充分的测试
- B. 存在多态的情况下, 为了达到较高的测试充分性, 应对所有可能的绑定都进行测试
- C. 假设类 B 是类 A 的子类, 如果类 A 已经进行了充分的测试, 那么在测试类 B 时不必测试任何类 B 继承自类 A 的成员方法
- D. 对于一棵继承树上的多个类, 只有处于叶子节点的类需要测试

软件系统架构是关于软件系统的结构、(40) 和属性的高级抽象。在描述阶段, 主要描述直接构成系统的抽象组件以及各个组件之间的连接规则, 特别是相对细致地描述组件的 (41)。在实现阶段, 这些抽象组件被细化为实际的组件, 比如具体类或者对象。软件系统架构不仅指定了软件系统的组织和 (42) 结构, 而且显示了系统需求和组件之间的对应关系, 包括设计决策的基本方法和基本原理。

- (40) A. 行为

- B. 组织
- C. 性能
- D. 功能

(41) A. 交互关系

- B. 实现关系
- C. 数据依赖
- D. 功能依赖

(42) A. 进程

- B. 拓扑
- C. 处理
- D. 数据

软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义了一类架构所共有的特征，主要包括架构定义、架构词汇表和架构（43）。

- A. 描述
- B. 组织
- C. 约束
- D. 接口

以下叙述，（44）不是软件架构的主要作用。

- A. 在设计变更相对容易的阶段，考虑系统结构的可选方案
- B. 便于技术人员与非技术人员就软件设计进行交互
- C. 展现软件的结构、属性与内部交互关系
- D. 表达系统是否满足用户的功能性需求

特定领域软件架构 (Domain Specific Software Architecture, DSSA) 是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。DSSA 通常是一个具有三个层次的系统模型，包括（45）环境、领域特定应用开发环境和应用执行环境，其中（46）主要在领域特定应用开发环境中工作。

(45) A. 领域需求

- B. 领域开发
- C. 领域执行
- D. 领域应用

(46) A. 操作员

- B. 领域架构师
- C. 应用工程师
- D. 程序员

“编译器”是一种非常重要的基础软件，其核心功能是对源代码形态的单个或一组源程序依次进行预处理、词法分析、语法分析、语义分析、代码生成、代码优化等处理，最终生成目标机器的可执行代码。考虑以下与编译器相关的软件架构设计场景：传统的编译器设计中，上述处理过程都以独立功能模块的形式存在，程序源代码作为一个整体，依次在不同模块中进行传递，最终完成编译过程。针对这种设计思路，传统的编译器采用（47）架构风格比较合适。

随着编译、链接、调试、执行等开发过程的一体化趋势发展, 集成开发环境 (IDE) 随之出现。IDE 集成了编译器、连接器、调试器等多种工具, 支持代码的增量修改与处理, 能够实现不同工具之间的信息交互, 覆盖整个软件开发生命周期。针对这种需求, IDE 采用 (48) 架构风格比较合适。IDE 强调交互式编程, 用户在修改程序代码后, 会同时触发语法高亮显示、语法错误提示、程序结构更新等多种功能的调用与结果呈现, 针对这种需求, 通常采用 (49) 架构风格比较合适。

某公司已经开发了一款针对某种嵌入式操作系统专用编程语言的 IDE, 随着一种新的嵌入式操作系统上市并迅速占领市场, 公司决定对 IDE 进行适应性改造, 支持采用现有编程语言进行编程, 生成符合新操作系统要求的运行代码, 并能够在现有操作系统上模拟出新操作系统的运行环境, 以支持代码调试工作。针对上述要求, 为了使 IDE 能够生成符合新操作系统要求的运行代码, 采用基于 (50) 的架构设计策略比较合适; 为了模拟新操作系统的运行环境, 通常采用 (51) 架构风格比较合适。

(47) A. 管道-过滤器

B. 顺序批处理

C. 过程控制

D. 独立进程

(48) A. 规则引擎

B. 解释器

C. 数据共享

D. 黑板

(49) A. 隐式调用

B. 显式调用

C. 主程序-子程序

D. 层次结构

(50) A. 代理

B. 适配

C. 包装

D. 模拟

(51) A. 隐式调用

B. 仓库结构

C. 基于规则

D. 虚拟机

某公司采用基于架构的软件设计 (Architecture-Based Software Design, ABSD) 方法进行软件设计与开发。ABSD 方法有三个基础, 分别是对系统进行功能分解、采用 (52) 实现质量属性与商业需求、采用软件模板设计软件结构。

ABSD 方法主要包括架构需求等 6 个主要活动, 其中 (53) 活动的目标是标识潜在的风险, 及早发现架构设计中的缺陷和错误; (54) 活动针对用户的需求变化, 修改应用架构, 满足新的需求。

小王是该公司的一位新任架构师, 在某项目中主要负责架构文档化方面的工作。小王 (55) 的做法不符合架构文档化的原则。架构文档化的主要输出结果是架构规格说明书和 (56)。

(52) A. 架构风格

B. 设计模式

C. 架构策略

D. 架构描述

(53) A. 架构设计

B. 架构实现

C. 架构复审

D. 架构演化

(54) A. 架构设计

B. 架构实现

C. 架构复审

D. 架构演化

(55) A. 从使用者的角度书写文档

B. 随时保证文档都是最新的

C. 将文档分发给相关人员

D. 针对不同背景的人员书写文档的方式不同

(56) A. 架构需求说明书

B. 架构实现说明书

C. 架构质量说明书

D. 架构评审说明书

架构权衡分析方法 (Architecture Tradeoff Analysis Method, ATAM) 是一种系统架构评估方法, 主要在系统开发之前, 针对性能、(57)、安全性和可修改性等质量属性进行评价和折中。ATAM 可以分为 4 个主要的活动阶段, 包括需求收集、(58) 描述、属性模型构造和分析、架构决策与折中, 整个评估过程强调以 (59) 作为架构评估的核心概念。

某软件公司采用 ATAM 进行软件架构评估, 在评估过程中识别出了多个关于质量属性的描述。其中, “系统在进行文件保存操作时, 应该与 Windows 系统的操作方式保持一致, 主要与 (60) 质量属性相关: “系统应该提供一个开放的 API 接口, 支持远程对系统的行为进行控制与调试, 主要与 (61) 质量属性相关。在识别出上述描述后, 通常采用 (62) 对质量属性的描述进行刻画与排序。在评估过程中, (63) 是一个会影响多个质量属性的架构设计决策。

(57) A. 可测试性

B. 可移植性

C. 可用性

D. 易用性

(58) A. 架构视图

B. 架构排序

C. 架构风格

D. 架构策略

(59) A. 用例

B. 视图

C. 属性

D. 模型

(60) A. 可测试性

B. 互操作性

C. 可移植性

D. 易用性

(61) A. 可测试性

- B. 互操作性
- C. 可移植性
- D. 易用性

(62) A. 期望管理矩阵

- B. 决策表
- C. 优先队列
- D. 效用树

(63) A. 风险点

- B. 决策点
- C. 权衡点
- D. 敏感点

以下关于第三方认证服务的叙述中, 正确的是 (64)。

- A. Kerberos 认证服务中保存数字证书的服务器叫 CA
- B. 第三方认证服务的两种体制分别是 Kerberos 和 PKI
- C. PKI 体制中保存数字证书的服务器叫 KDC
- D. Kerberos 的中文全称是“公钥基础设施”

采用 Kerberos 系统进行认证时, 可以在报文中加入 (65) 来防止重放攻击。

- A. 会话密钥
- B. 时间戳
- C. 用户 ID
- D. 私有密钥

以下关于为撰写学术论文引用他人资料的叙述中, 错误的是 (66)。

- A. 既可引用发表的作品, 也可引用未发表的作品
- B. 只能限于介绍、评论或为了说明某个问题引用作品
- C. 只要不构成自己作品的主要部分, 可引用资料的部分或全部
- D. 不必征得著作权人的同意, 不向原作者支付合理的报酬

以下作品中, 不适用或不受著作权法保护的作品是 (67)。

- A. 国务院颁布的《计算机软件保护条例》
- B. 某作家的作品《绿化树》
- C. 最高人民法院组织编写的《行政诉讼案例选编》
- D. 某人在公共场所的即兴演说

以下著作权权利中, (68) 的保护期受时间限制。

- A. 署名权
- B. 发表权
- C. 修改权
- D. 保护作品完整权

某企业拟生产甲、乙、丙、丁四个产品。每个产品必须依次由设计部门、制造部门和检验部门进行设计、制造和检验, 每个部门生产产品的顺序是相同的。各产品各工序所需的时间如

下表所示:

| 项目 | 设计(天) | 制造(天) | 检验(天) |
|----|-------|-------|-------|
| 甲 | 13 | 15 | 20 |
| 乙 | 10 | 20 | 18 |
| 丙 | 20 | 16 | 10 |
| 丁 | 8 | 10 | 15 |

只要适当安排好项目实施顺序, 企业最快可以在(69)天全部完成这四个项目。

- A. 84
- B. 86
- C. 91
- D. 93

1路和2路公交车都将在10分钟内均匀随机地到达同一车站, 则它们相隔4分钟内到达该站的概率为(70)。

- A. 0.36
- B. 0.48
- C. 0.64
- D. 0.76

A system's architecture is a representation of a system in which there is a mapping of (71) onto hardware and software components, a mapping of the (72) onto the hardware architecture, and a concern for the human interaction with these components. That is, system architecture is concerned with a total system, including hardware, software, and humans.

Software architectural structures can be divided into three major categories, depending on the broad nature of the elements they show. 1) (73) embody decisions as a set of code or data units that have to be constructed or procured. 2) (74) embody decisions as to how the system is to be structured as set of elements that have runtime behavior and interactions. 3) (75) embody decisions as to how the system will relate to nonsoftware structures in its environment (such as CPUs, file systems, networks, development teams, etc.).

- (71) A. attributes
 B. constraint
 C. functionality
 D. requirements

- (72) A. physical components
 B. network architecture
 C. software architecture
 D. interface architecture

- (73) A. Service structures
 B. Module structures

- C. Deployment structures
- D. Work assignment structures
- (74) A. Decomposition structures
- B. Layer structures
- C. Implementation structures
- D. Component-and-connector structures
- (75) A. Allocation structures
- B. Class structures
- C. Concurrency structures
- D. Uses structures

希赛网在线题库