

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](http://www.educity.cn)

希赛网软件水平考试网: [www.educity.cn/rk](http://www.educity.cn/rk)

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

2015 年评测上午真题答案与解析: <http://www.educity.cn/tiku/tp19304.html>

## 2015 年下半年软件评测师考试上午真题

● CPU 响应 DMA 请求是在\_\_(1)\_\_结束时。

- (1) A. 一条指令执行  
B. 一段程序  
C. 一个时钟周期  
D. 一个总线周期

● 虚拟存储体系是由\_\_(2)\_\_两线存储器构成。

- (2) A. 主存, 辅存  
B. 寄存器, Cache  
C. 寄存器, 主体  
D. Cache, 主存

● 浮点数能够表示的数的范围是由其\_\_(3)\_\_的位数决定的。

- (3) A. 尾数  
B. 阶码  
C. 数符  
D. 阶符

● 在机器指令的地址段中, 直接指出操作数本身的寻址方式称为\_\_(4)\_\_。

(4) A. 隐含寻址

B. 寄存器寻址

C. 立即寻址

D. 直接寻址

● 内存按字节编址从 B3000H 到 DABFFH 的区域其存储容量为\_\_ (5) \_\_。

(5) A. 123KB

B. 159KB

C. 163KB

D. 194KB

● 编译器和解释器是两种基本上的高级语言处理程序。编译器对高级语言源程序的处理过程可以划分为词法分析，语法分析，语义分析，中间代码生成，代码优化，目标代码生成等阶段，其中，\_\_ (6) \_\_并不是每个编译器都必需的。

(6) A. 词法分析各语法分析

B. 语义分析和中间代码生成

C. 中间代码生成和代码优化

D. 代码优化和目标代码生成

● 表达式采用逆波兰式表示时，利用\_\_ (7) \_\_进行求值。

(7) A. 栈

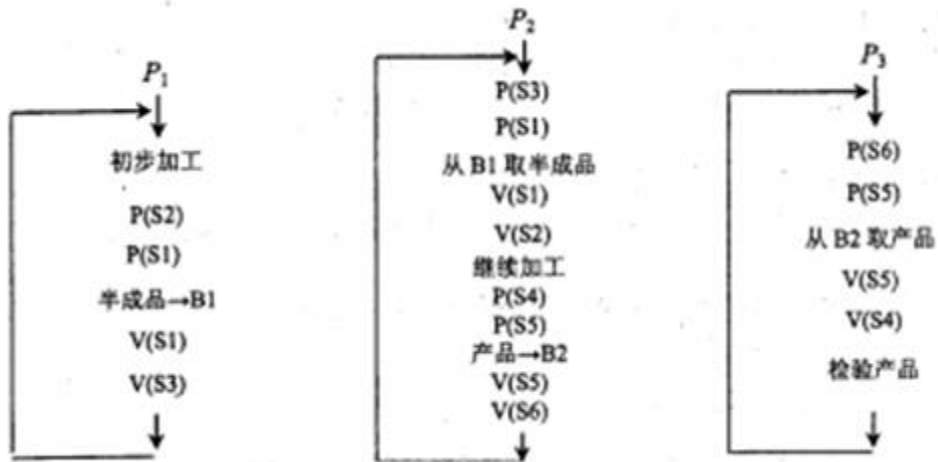
B. 队列

C. 符号表

D. 散列表

● 某企业的生产流水线上有 2 名工人 P1 和 P2，1 名检验员 P3。P1 将初步加工的半成品放入半成品箱 B1，P2 从半成品箱 B1 取出继续加工，加工好的产品放入成品箱 B2，P3 从成品箱 B2，P3 从成品箱 B2 取出产品检验。假设 B1 可存放 N 件半成品 B2 可存放 M 件产品，并设置 6 个信号量 S1、S2、S3、S4、S5 和 S6，且 S3 和 S6 的初值都为 0，采用 PV 操作实现 P1 P2

和 P3 的同步模型如下图所示, 则信号量 S1 和 S5\_\_ (8) \_\_ S2、S4 的初值分别为 ( )。



- (8) A. 分别为同步信号量和互斥信号量, 初值分别为 0 和 1  
 B. 都是同步信号量, 其初值分别为 0 和 0  
 C. 都是互斥信号量, 其初值分别为 1 和 1  
 D. 都是互斥信号量, 其初值分别为 0 和 1

- (9) A. n,o  
 B. m,o  
 C. m,n  
 D. n,m

● 在支付多线程的操作系统中, 假设进程 P 创建了若干个线程, 那么\_\_ (10) \_\_是不能被这些线程共享的。

- (10) A. 该进程中打开的文件  
 B. 该进程的代码段  
 C. 该进程中某线程的栈指针  
 D. 该进程的全局变量

● 软件设计师王某在其公司的某一综合楼信息管理系统软件开发工作中承担了大部分程序设计工作, 该系统交付用户后, 投入试运行后, 王某离职离开公司, 并带走了该综合信息管理系统源程序, 拒不交还公司, 王某认为, 综合信息管理系统源程序是他独立完成的, 他是综合信息管理系统源程序的软件著作权人, 王某的行为\_\_ (11) \_\_。

- (11) A. 侵犯了公司的软件著作权  
 B. 未侵犯了公司的软件著作权

- C. 侵犯了公司的商业秘密权
- D. 不涉及侵犯了公司的软件著作权

● 颜色深度是表达单个像素的颜色或灰度所占的位数 (bit)，若每个像素具有 8 位的颜色深度，则可表示\_\_(12)\_\_种不同的颜色。

- (12) A. 8  
B. 64  
C. 256  
D. 512

● 视觉上的颜色可用亮度，色调和饱和度三个特征来描述，其中饱和度是指颜色的\_\_(13)\_\_。

- (13) A. 种数  
B. 纯度  
C. 感觉  
D. 存储量

● \_\_(14)\_\_不属于主动攻击。

- (14) A. 流量分析  
B. 重放  
C. IP 地址欺骗  
D. 拒绝服务

● 防火墙不具备\_\_(15)\_\_功能。

- (15) A. 包过滤  
B. 查毒  
C. 记录访问过程  
D. 代理

- 如下图所示，从输出的信息中可以确定的是信息是\_\_(16)\_\_。

```
C:\>netstat -n
```

Active Connections

Proto	Local Address	Foreign Address	State
TCP	192.168.0.200:2011	202.100.112.12:443	ESTABLISHED
TCP	192.168.0.200:2038	100.29.200.110:110	TIME_WAIT
TCP	192.168.0.200:2052	128.105.129.30:80	ESTABLISHED

- (16) A. 本地主机正在使用的端口的公共端口号  
 B. 192.168.0.200 正在与 128.105.129.30 建立连接  
 C. 本地主机与 202.100.112.12 建立安全连接  
 D. 本地主正在与 100.29.200.110 建立连接

- 数据库系统通常采用三级模式结构:外模式，模式和内模式，这三级模式分别对应的数据库的\_\_(17)\_\_。

- (17) A. 基本表，存储文件和视图  
 B. 视图，基本表和存储文件  
 C. 基本表，视图和存储文件  
 D. 视图，存储文件和基本表

- 在数据库逻辑设计阶段，若实体中存在多值属性，那么将 E-R 图转为关系模式时\_\_(18)\_\_得到的关系模式属于 4NF。

- (18) A. 将所有多值属性组成一个关系模式  
 B. 使多值属性不在关系模式中出现  
 C. 将实体的码分别和每个多值属性独立构成一个关系模式  
 D. 将多值属性各其它属性一起构与该实体对应的关系模式

- 在分布式数据库中有分片透明，复制透明，位置透明和逻辑透明等基本概念，其中，\_\_(19)\_\_是指局部数据模型透明，即用户或应用程序无需知道局部使用的是哪种数据模型，\_\_(20)\_\_是指用户或应用程序不需要知道逻辑上访问的表是怎么分块存储的。

- (19) A. 分片透明  
 B. 复制透明

- C. 位置透明
  - D. 逻辑透明
- (20) A. 分片透明
- B. 复制透明
  - C. 位置透明
  - D. 逻辑透明

● 设有关系模式  $R(A_1, A_2, A_3, A_4, A_5, A_6)$ ，其中：函数依赖集  $F = \{A_1 \rightarrow A_2, A_1 A_3 \rightarrow A_4, A_5 A_6 \rightarrow A_1, A_2 A_5 \rightarrow A_6, A_3 A_5 \rightarrow A_6\}$ ，则\_\_(21)\_\_关系模式  $R$  的一个主键， $R$  规范化程度最高达到\_\_(22)\_\_。

- (21) A.  $A_1 A_4$
- B.  $A_2 A_4$
  - C.  $A_3 A_5$
  - D.  $A_4 A_5$
- (22) A. 1NF
- B. 2NF
  - C. 3NF
  - D. BCNF

● POP3 协议采用\_\_(23)\_\_模式，客户端代理与 POP3 服务器通过建立\_\_(24)\_\_连接来传送数据。

- (23) A. Browser/Server
- B. Client/Server
  - C. Peer to Peer
  - D. Peer to Server
- (24) A. TCP
- B. UDP
  - C. P2P
  - D. IP

● 如果在查找路由表时发现有多项匹配，那么应该根据\_\_(25)\_\_原则进行选择，假设路由表有 4 全表项如下所示，那么与地址 139.17.179.92 匹配的表项是\_\_(26)\_\_。

- (25) A. 包含匹配
- B. 恰当匹配

C. 最长匹配

D. 最短匹配

(26) A. 139.17.145.32

B. 139.17.145.64

C. 139.17.147.64

D. 139.17.177.64

● 在层次化局域网模型中，以下关于核心层的描述，正确的是\_\_(27)\_\_。

(27) A. 为了保障安全性，对分组要进行有效性检查

B. 将分组从一个区域高速地转发到另一个区域

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

● 算术表达式  $a+b-c*d$  的后缀式是\_\_(28)\_\_ (-、+、\*表示算术的减，加，乘运算，运算符的优先级和结合性遵循惯例)。

(28) A.  $ab+cd*-$

B.  $abc+-d*$

C.  $abcd+-*$

D.  $ab+c-d*$

● 函数  $f$ \_\_(29)\_\_、 $g$ \_\_(30)\_\_ 的定义如下所示，已知调用  $f$  时传递给其形参  $x$  的值是 10，若以传值方式调用  $g$ ，则函数  $f$  的返回值为\_\_(31)\_\_。

**f(int x)**

```
int y = 0;
y = g(x);
return x+y;
```

**g(int x)**

```
x = x+5;
return x;
```

(29) A. 10

B. 15

C. 25

D. 30

- 当用户需求不清晰, 需求经常发生变化, 系统规模不太大时, 最适宜采用软件开发方法是\_\_(30)\_\_\_。

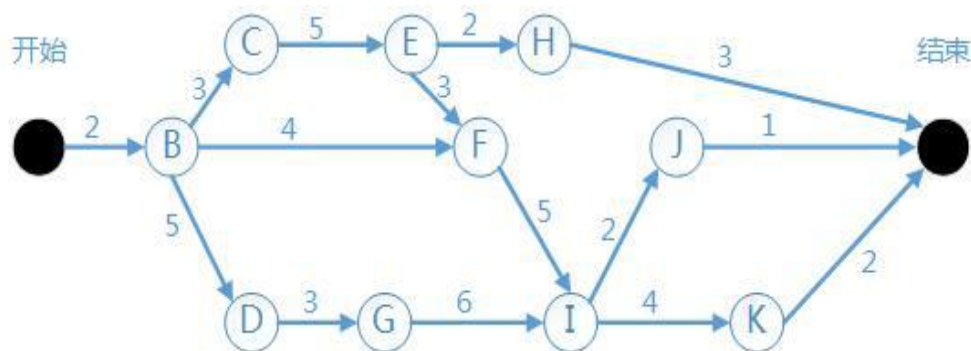
- (30) A. 结构化  
 B. 原型  
 C. 面向对象  
 D. 敏捷

- 在结构化分析方法中, 利用分层数据流图对系统功能建模, 以下关于分层数据流图的叙述中, 不正确的是\_\_(31)\_\_\_。采用数据字典为数据流图中的每个数据流、文件、加工以及组成数据流或文件的数据项进行说明, 其条目不包括\_\_(32)\_\_\_。

- (31) A. 顶层的数据流图只有一个加工, 即要开的软件系统  
 B. 在整套分层数据流图中, 每个数据存储应该有加工对其进行读操作, 有加工对其进行写操作  
 C. 一个加工的输入数据流和输出数据流可以同名  
 D. 每个加工至少有一个输入数据流和一个输出数据流

- (32) A. 数据流  
 B. 外部实体  
 C. 数据项  
 D. 基本加工

- 下图是一个软件项目的活动图, 其中顶点表示项目的里程碑, 连接顶点的边表示包含的活动, 则完成该项目的最少时间为\_\_(33)\_\_\_天, 活动 BD 最多可以晚开始\_\_(34)\_\_\_天而不会影响整个项目的进度。





(33) A. 15

B. 21

C. 22

D. 24

(34) A. 2

B. 3

C. 5

D. 9

- 开发过程中以用户需求为动力, 以对象作为驱动, \_\_ (35) \_\_ 适合于面向对象的开发方法。

(35) A. 瀑布

B. 原型

C. 螺旋

D. 喷泉

- 以下关于极限编程 XP 的叙述中, 不正确的是 \_\_ (36) \_\_。

(36) A. 由价值观, 原则, 实践和行为四个部分组成

B. 每个不同的项目都需要一套不同的策略, 约定和方法论

C. 有四个价值观, 即沟通, 简单性, 反馈和勇气

D. 有五大原则, 即快速反馈, 简单性假设, 逐步修改, 提倡更改和优质工作

- 以下关于分层体系结构的叙述中不正确有的是 \_\_ (37) \_\_。

(37) A. 可以很好的表示软件系统的不同抽象层次

B. 对每个层的修改通常只影响其相邻的两层

C. 将需求定义到多层上很容易

D. 有利于开发任务的分工

- 以下关于模块耦合关系的叙述中, 耦合程度最低的是 \_\_ (38) \_\_, 其耦合类型为 \_\_ (39) \_\_ 耦合。

(38) A. 模块 M2 根据模块 M1 传递如标记量的控制信息来确定 M2 执行哪部分语名

B. 模块 M2 直接访问块 M1 内部

C. 模块 M1 和模块 M2 用公共的数据结构

D. 模块 M1 和模块 M2 有部分代码是重叠的

(39) A. 数据

B. 标记

C. 控制

D. 内容

● 堆是一种数据结构，分为大顶堆和小顶堆两种类型，大（小）顶堆要求父元素大于等于（小于等于）其左右孩子元素。则\_\_(40)\_\_是一个大顶堆结构，该堆结构用二叉树表示，其高度（或层数）为\_\_(41)\_\_。

(40) A. 94,31,53,23,16,27

B. 94,53,31,72,16,23

C. 16,53,23,94,31,72

D. 16,31,23,94,53,72

(41) A. 2

B. 3

C. 4

D. 5

● 在 ISO/IEC 软件质量模型中，功能性是与一组功能及其指定的性质的存在有关的一组属性，其子特性不包括\_\_(42)\_\_。

A. 适应性

B. 准确性

C. 安全性

D. 成熟性

● 程序质量评审通常是从开发者的角度进行评审，其内容不包括\_\_(43)\_\_。

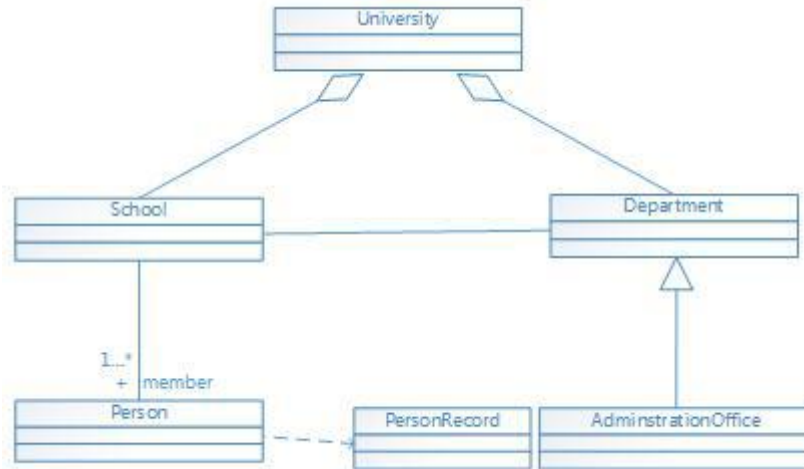
A. 功能结构

B. 功能的通用性

C. 模块层次

D. 与硬件的接口

- 在面向对象分析和设计中，用类图给出的静态设计视图，其应用场合不包括\_\_(44)\_\_\_。下图是一个 UML 类图，其中类 University 和类 School 之间是\_\_(45)\_\_\_关系，类 Person 和类 PersonRecord 之间是\_\_(46)\_\_\_关系，表示 Person 与 PersonRecord\_\_(47)\_\_\_。



- (44) A. 对系统的词汇建模  
B. 对简单的协作建模  
C. 对逻辑数据库模式建模  
D. 对系统的需求建模
- (45) A. 依赖  
B. 关联  
C. 聚集  
D. 泛化
- (46) A. 依赖  
B. 关联  
C. 聚集  
D. 泛化
- (47) A. 这间的语义关系，其中 PersonRecord 发生变化会影响 Person 的语义  
B. 之间的一种结构关系，描述了一组链，即对象之间的连接  
C. 是整体和部分的关系  
D. 是一般和特殊的关系

- 软件复杂性是指理解和处理软件的难易程度。其度量参数不包括\_\_(48)\_\_\_。

- (48) A. 规模

- B. 类型
- C. 结构
- D. 难度

● 对现有软件系统中一些数据处理的算法进行改进, 以提高效率, 从而更快地响应用户服务要求。这种行为属于\_\_(49)\_\_维护。

- (49) A. 正确性
- B. 适应性
  - C. 完善性
  - D. 预防性

● 软件测试的对象包括\_\_(50)\_\_。

- ①需求规格说明
- ②概要设计文档
- ③软件测试报告
- ④软件代码
- ⑤用户手册
- ⑥软件开发人员

- (50) A. ①②③④⑤⑥
- B. ①②③④⑤
  - C. ①②④
  - D. ①②③④

● 以下不属于系统测试的是\_\_(51)\_\_。

- ①单元测试
- ②集成测试
- ③安全性测试
- ④可靠性测试
- ⑤确认测试
- ⑥验证测试

- (51) A. ①②③④⑤⑥
- B. ①②③④
  - C. ①②⑤⑥
  - D. ①②④⑤⑥

- 以下关于软件测试原则叙述中，不正确的是\_\_(52)\_\_。

- (52) A. 测试阶段在实现阶段之后，因此实现完成后再开始进行测试  
B. 测试用例需要完善和修订  
C. 发现错误越来的地方应该进行更多的测试  
D. 测试用例本身也需要测试

- 一条 BUG 记录应该包括\_\_(53)\_\_。

- ①编号
- ②bug 描述
- ③bug 级别
- ④bug 所属模块
- ⑤发现人

- (53) A. ①②  
B. ①②③  
C. ①②③④  
D. ①②③④⑤

- \_\_(54)\_\_ 不属于使用软件测试工具的目的。

- (54) A. 帮助测试寻找问题  
B. 协助问题的诊断  
C. 节省测试时间  
D. 替代手工测试

- 以下关于验收测试的叙述中，不正确的是\_\_(55)\_\_。

- (55) A. 验收测试由开发方主导，用户参与  
B. 验收测试也需要制定测试计划  
C. 验收测试之前需要先明确验收方法  
D. 验收测试需要给出验收通过或者不通过结论

- 以下关于黑盒测试的测试方法选择的叙述中，不正确的是\_\_(56)\_\_。

- (56) A. 在任何情况下都要采用边边界值分析法  
B. 必要时有等价类划分法补充测试用例

- C. 可以用错误推测法追加测试用例
- D. 如果输入条件之前不存在组合情况, 则采用因果图法

● 以下关于等价划分法的叙述中不正确的是\_\_(57)\_\_\_。

- (57) A. 如果规定输入值 string1 必须是'\0'结束, 那么得到两个等价类, 即有效等价类 {string1|string1 以'\0'结束}, 无效等价类 {string1|string1 不以'\0'结束}
- B. 如果规定输入值 int1 取值为 1、-1 两个数之一, 那么得到 3 个等价类, 即有效等价类 {int1|int1=1}、{int1|int1=-1}, 无效等价类 {int1|int1≠1 并且 int1≠-1}
- C. 如果规定输入值 int2 取值范围为-10~9, 那么得到两个等价类, 即有效等价类 {int2|-10<=int2<=9}, 无效等价类 {int2|int2<-10 或者 int2>9}
- D. 如果规定输入值 int3 为质数, 那么得到两个等价类, 即有效等价类 {int3|int3 是质数}, 无效等价类 {int3|int3 不是质数}

● 以下关于白盒测试的叙述中, 不正确的是\_\_(58)\_\_\_。

- (58) A. 满足判定覆盖一定满足语句覆盖
- B. 满足条件覆盖一定满足判定覆盖
- C. 满足判定条件覆盖一定满足条件覆盖
- D. 满足条件组合覆盖一定满足判定条件覆盖

● 对于逻辑表达式((a||(b&c))||(c&&d)), 需要\_\_(59)\_\_\_个测试用例才能完成条件组合覆盖。

- (59) A. 4
- B. 8
- C. 16
- D. 32

● 为了解系统在何种服务级别下会崩溃, 应进行\_\_(60)\_\_\_。

- (60) A. 负载测试
- B. 压力测试
- C. 大数据量测试
- D. 疲劳测试

● 兼容性测试的测试范围包括\_\_(61)\_\_。

- ①硬件兼容性测试
- ②软件兼容性测试
- ③数据兼容性测试
- ④平台兼容性测试

- (61) A. ①②③④  
B. ①②③  
C. ①②  
D. ①

● 以下不能作为测试结束标准的是\_\_(62)\_\_。

- (62) A. 测试超过了预定时间  
B. 执行完了所有测试用例但没有发现新的故障  
C. 单位时间内查出的故障数目低于预定值  
D. 测试人员或者其它资源不足

● 以下属于静态测试方法的是\_\_(63)\_\_。

- (63) A. 代码审查  
B. 判定覆盖  
C. 路径覆盖  
D. 语句覆盖

● 单元测试的测试内容包括\_\_(64)\_\_。

- ①模块接口
- ②局部数据库结构
- ③模块内路径
- ④边界条件
- ⑤错误处理
- ⑥系统性能

- (64) A. ①②③④⑤⑥  
B. ①②③④⑤  
C. ①②③④  
D. ①②③

● 一个 Web 信息系统所需要的进行的测试包括\_\_(65)\_\_。

- ①功能测试
- ②性能测试
- ③可用性测试
- ④客户端兼容性测试
- ⑤安全性测试

- (65) A. ①②  
B. ①②③  
C. ①②③④  
D. ①②③④⑤

● 以下不属于网络测试的测试指标的是\_\_(66)\_\_。

- (66) A. 吞吐量  
B. 延时  
C. 并发用户数  
D. 丢包率

● 对于其于用户口令的用户认证机制来说,\_\_(67)\_\_不属于增强系统安全性应使用的防范措施。

- (67) A. 对本地存储的口令进行加密  
B. 在用户输入的非法口令达到规定的次数之后, 禁用相应账户  
C. 建议用户使用英文单词或姓名等容易记忆的口令  
D. 对于关键领域或安全性要求较高的系统, 应当保证用过的用户删除或停用后, 保留该用户记录, 且新用户不能与该用户名

● 对于防病毒系统的测试是系统安全测试的重要内容, 下列不属于防病毒系统安全测试基本测试点的是\_\_(68)\_\_。

- (68) A. 能否提供对病毒特征与检测引擎的定期在线更新服务  
B. 能否在不更新特征库的前提下对新的未知病毒进行有效查杀  
C. 能否支持多种平台的病毒防范  
D. 能否支持对电子邮件附件的病毒防治



● 1976 年 Diffie 与 Hellman 首次公开提出\_\_(69)\_\_\_的概念与结构, 采用两个从此独立的密钥对数据分别行加密或解密, 且加密过程基于数学函数, 从而带来了加密领域的革命性进步。

- (69) A. 公钥加密  
B. 对称加密  
C. 单向 Hash 函数  
D. RSA 加密

● 集线器与网桥的区别是\_\_(70)\_\_\_。

- (70) A. 集线器不能检测发生冲突, 而网桥可以检测冲突  
B. 集线器是物理层设备, 而网桥是数据链路层设备  
C. 网桥只有两个端口, 而集线器是一种多端口网桥  
D. 网桥是物理设备, 而集线器是数据链路层设备

● In a world where it seems we already have too much to do, and too many things to think about, it seems the last thing we need is something new that we have to learn.

But use cases do solve a problem with requirements: with \_\_(71)\_\_\_ declarative requirements it's hard to describe steps and sequences of events.

Use cases, stated simply, allow description of sequences of events that, taken together, lead to a system doing something useful. As simple as this sounds, this is important. When confronted only with a pile of requirements, it's often \_\_(72)\_\_\_ to make sense of what the authors of the requirements really wanted the system to do. In the preceding example, use cases reduce the ambiguity of the requirements by specifying exactly when and under what conditions certain behavior occurs; as such, the sequence of the behaviors can be regarded as a requirement. Use cases are particularly well suited to capture approaches. Although this may sound simple, the fact is that \_\_(73)\_\_\_ requirement capture approaches, with their emphasis on declarative requirements and "shall" statements, completely fail to capture fail to capture the \_\_(74)\_\_\_ of the system's behavior. Use cases are a simple yet powerful way to express the behavior of the system in way that all stakeholders can easily understand.

But, like anything, use cases come with their own problems, and as useful as they are, they can be \_\_(75)\_\_\_ . The result is something that is as bad, if not worse, than the original problem. Therein it's important to utilize use cases effectively without creating a greater problem than the one you started with.

- (71) A. plenty  
B. loose  
C. extra

- D. strict
- (72) A. impossible
  - B. possible
  - C. sensible
  - D. practical
- (73) A. modern
  - B. conventional
  - C. different
  - D. formal
- (74) A. statics
  - B. nature
  - C. dynamics
  - D. originals
- (75) A. misapplied
  - B. applied
  - C. used
  - D. powerful

希赛网在线题库