

2019 年上半年软件设计师上午答案及解析

1.A 单击此链接查看真题解析视频 <https://ke.qq.com/course/263366?tuin=7d818187>

解析:

- 运算器和控制器组成中央处理器(CPU)。
- 控制器负责依次访问程序指令, 进行指令译码, 并协调其他设备, 通常由程序计数器(PC)、指令寄存器、指令译码器、状态/条件寄存器、时序发生器、微操作信号发生器组成。
- 运算器负责完成算术、逻辑运算功能, 通常由ALU (算术逻辑单元)、寄存器、多路转换器、数据总线组成。

高级项目经理 任铄

向上人生路!

2.C

解析:

DMA 控制器是一种在系统内部转移数据的独特外设, 可以将其视为一种能够通过一组专用总线将内部和外部存储器与每个具有 DMA 能力的外设连接起来的控制器。

3.C

解析:

局部性原理是指计算机在执行某个程序时, 倾向于使用最近使用的数据。局部性原理有两种表现形式:

- 时间局部性是指被引用过的存储器位置很可能会被再次引用, 例如: 重复的引用一个变量时则表现出较好的时间局部性
- 空间局部性是指被引用过的存储器位置附近的数据很可能将被引用; 例如: 遍历二维数组时按行序访问数据元素具有较好的空间局部性

4.C

解析:

可靠性计算

1. 串联系统

各个子系统的可靠性分别用 R_1, R_2, \dots, R_n 表示

- 系统的可靠性为:

$$R = R_1 \times R_2 \times \dots \times R_n$$

- 系统的失效率为:

$$\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$$

高级项目经理 任铄

向上人生路!

2. 并系统

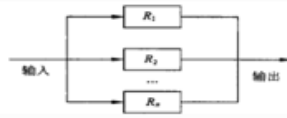
假如一个系统由2个子系统组成，只要有一个子系统能够正常工作，系统就能正常工作，设系统各个子系统的可靠性用 R_1, R_2, \dots, R_n 表示

- 则系统的可靠性为：

$$R = 1 - (1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)$$

- 则系统的失效率为：

$$\mu = \frac{1}{\lambda \sum_{j=1}^n \frac{1}{\lambda_j}}$$



向上人生路!

5.D

6.B

解析：

四、RISC与CISC

- 为提高操作系统的效率，人们最初选择向指令系统中添加更多、更复杂的指令来实现，导致指令集越来越大。这种类型的计算机称为复杂指令集计算机(CISC)。
- 对指令数目和寻址方式做精简，指令的指令周期相同，采用流水线技术，指令并行执行程度更好，这就是精简指令集计算机(RISC)。

高级项目经理 任钰

向上人生路!

7.B

解析：

应用级网关可以工作在 OSI 七层模型的任一层上，能够检查进出的数据包，通过网关复制传递数据，防止在受信任服务器和客户机与不受信任的主机间直接建立联系。应用级网关能够理解应用层上的协议，能够做复杂一些的访问控制，起到防火墙的作用，称为应用级网关。

8. C

解析：

MIME (Multipurpose Internet Mail Extensions) 多用途互联网邮件扩展类型，是描述消息内容类型的因特网标准。MIME 消息能包含文本、图像、音频、视频以及其他应用程序专用的数据。

9. 10. A D

解析：

一、数字证书

数字证书就是互联网通讯中标志通讯各方身份信息的一串数字，是一种在Internet上验证通信实体身份的方式。

作用类似于我们的身份证。它由一个由权威机构CA(证书授权中心)发行的，人们可以在网上用它来识别对方的身份。

数字证书是一个经CA数字签名的，包含公开密钥拥有者信息以及公开密钥的文件。

高级项目经理 任铎

向上人生路！

二、数字签名

数字签名是指通过一个单向函数对要传送的报文进行处理，得到用以认证报文来源并核实报文是否发生变化的一个字母数字串。它与数据加密技术一起，构建起了安全的商业加密体系。

- 传统的数据加密是保护数据的最基本方法，它只能防止第三者获得真实的数据(数据的机密性)，而数字签名则可以解决否认、伪造、篡改和冒充的问题(数据的完整性和不可抵赖性)。
- 数字签名使用的是公钥算法（非对称密钥技术）。

高级项目经理 任铎

向上人生路！

11. D

解析：

震网（Stuxnet），指一种蠕虫病毒。于 2010 年 6 月首次被检测出来，是第一个专门定向攻击真实世界中基础（能源）设施的“蠕虫”病毒，比如核电站，水坝，国家电网。

12. B

13. 14. A B

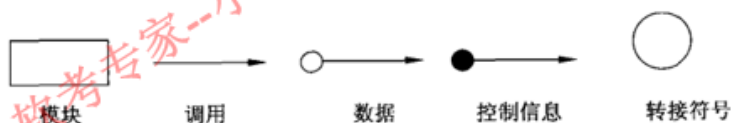
15. B

数据流图的基本原则：从基本系统模型出发，自顶向下、从抽象到具体分层次地画。

16. ?

解析：P201

模块结构图是结构化设计中描述系统结构的图形工具。作为一种文档，它必须严格地定义模块的名字、功能和接口，同时还应当在模块结构图上反映出结构化设计的思想。模块结构图由模块、调用、数据、控制和转接 5 种基本符号组成，如图 4-11 所示，现说明如下。



17. D

解析：

沟通渠道公式如下： $M=n*(n-1)/2$ M 表示沟通渠道数，n 表示项目中的成员数。

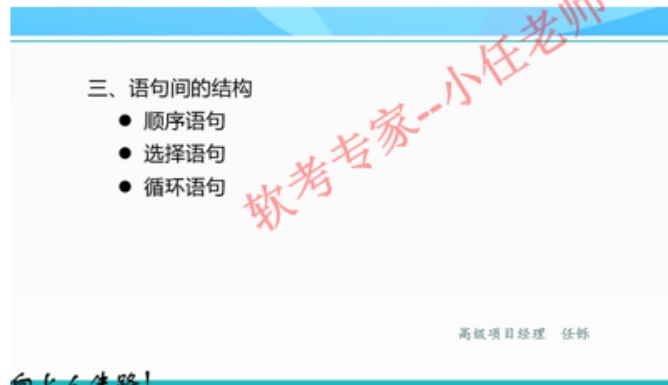
18.D

关键路径为 A1--A3--A9--A11--A12. 总工期为 $8+15+15+7+10=55$

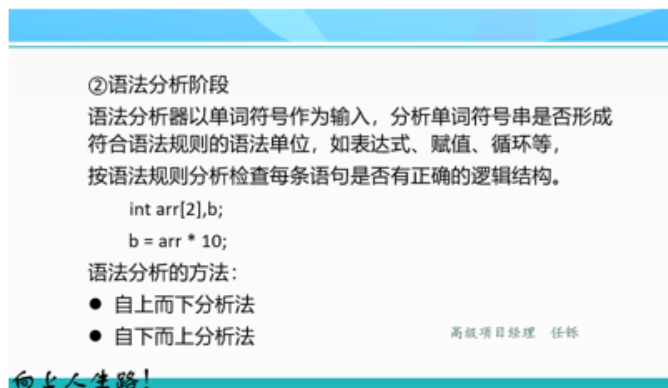
19.A

20.A

解析：



21.C



22.B

解析：

符号表是一种用于语言翻译器(例如编译器和解释器)中的数据结构。在符号表中，程序源代码中的每个标识符都和它的声明或使用信息绑定在一起，比如其数据类型、作用域以及内存地址。编译程序对高级语言源程序进行编译的过程中，要不断搜集、记录和使用源程序中一些相关符号的类型和特征等信息，并将其存入符号表中。

23.24.AC

25. C

解析：

位示图是利用二进制的一位来表示磁盘中的一个磁盘块的使用情况。当其值为“0”时，表示对应的盘块空闲；为“1”时，表示已经分配使用。

$1024*1024/4/64=4096$

26.C

解析:

绝对路径:是从根目录开始的路径,以”\”代表根目录。

相对路径:是从当前路径开始的路径。

27.B

解析:

PV 操作由 P 操作原语和 V 操作原语组成(原语是不可中断的过程),PV 操作来实现进程的同步和互斥。PV 操作属于进程的低级通信。

28.A

解析:

嵌入式操作系统的特点

- 1.微型化:从性能和成本角度考虑,希望占用的资源和系统代码量少。
- 2.可定制:从减少成本和缩短研发周期考虑,要求嵌入式操作系统能运行在不同的微处理器平台上,能针对硬件变化进行结构与功能上的配置,以满足不同应用的需求。
- 3.实时性:嵌入式操作系统主要应用于过程控制、数据采集、传输通信、多媒体信息及关键要害领域需要迅速响应的场合,所以对实时性要求较高。
- 4.可靠性:系统构件、模块和体系结构必须达到应有的可靠性,对关键要害应用还要提供容错和防故障措施。
- 5.易移植性:为了提高系统的易移植性,通常采用硬件抽象层和板级支持包的底层设计技术。

29.C

30.B

解析:

极限编程(XP)的 12 个最佳实践:

- 1.现场客户 (On-site Customer): 要求至少有一名实际的客户代表在整个项目开发周期在现场负责确定需求、回答团队问题以及编写功能验收测试。
- 2.代码规范 (Code Standards): 强调通过指定严格的代码规范来进行沟通,尽可能减少不必要的文档。
- 3.每周 40 小时工作制 (40-hour Week): 要求项目团队人员每周工作时间不能超过 40 小时,加班不得连续超过两周,否则反而会会影响生产率。
- 4.计划博弈 (Planning Game): 要求结合项目进展和技术情况,确定下一阶段要开发与发布的系统范围。
- 5.系统隐喻 (System Metaphor): 通过隐喻来描述系统如何运作、新的功能以何种方式加入到系统。它通常包含了一些可以参照和比较的类和设计模式。XP 不需要事先进行详细的架构设计。
- 6.简单设计 (Simple Design): 认为代码的设计应该尽可能的简单,只要满足当前功能的要求,不多也不少。
- 7.测试驱动 (Test-driven): 强调“测试先行”。在编码开始之前,首先将测试写好,而后再进行编码,直至所有的测试都得以通过。
- 8.代码重构 (Refactoring): 强调代码重构在其中的作用,认为开发人员应该经常进行重构,

通常有两个关键点应该进行重构：对于一个功能实现和实现后。

9.成对编程 (Pair Programming)：认为在项目中采用成对编程比独自编程更加有效。成对编程是由两个开发人员在同一台电脑上共同编写解决同一问题的代码，通常一个人负责写代码，而另一个负责保证代码的正确性与可读性。

10. 集体代码所有制 (Collective Ownership)：认为开发小组的每个成员都有更改代码的权利，所有的人对于全部代码负责。

11.持续集成 (Continuous Integration)：提倡在一天中集成系统多次，而且随着需求的变化，要不断的进行回归测试。因为，这样可以使得团队保持一个较高的开发速度，同时避免了一次系统集成的恶梦。

12.小型发布 (Small Release)：强调在非常短的周期内以递增的方式发布新版本，从而可以很容易地估计每个迭代周期的进度，便于控制工作量和风险；同时，也可以及时处理用户的反馈。

31.A

解析：

高级项目经理 任标

二、软件质量特性

ISO/IEC9126软件质量模型是一种评价软件质量的通用模型，包括3个层次：

1、质量特性

2、质量子特性

3、度量指标

质量特性	功能性	可靠性	易用性	效率	维护性	可移植性
质量子特性	适合性	成熟性	易理解性	时间特性	易分析性	适应性
	准确性	容错性	易学性	资源利用性	易改变性	易安装性
	互操作性	易恢复性	易操作性		稳定性	共存性
	保密安全性		吸引性		易测试性	易替换性
	依从性	依从性	依从性	依从性	依从性	依从性

向上人生路！

32.D

解析：

(1)非直接耦合：两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的。这种模块的耦合度最低、模块独立性最强。

(2)数据耦合：指两个模块之间有调用关系，传递的是简单的数据值，相当于高级语言的值传递。

(3)标记耦合：指两个模块之间传递的是数据结构，如高级语言中的数组名、记录名、文件名等这些名字即标记，其实传递的是这个数据结构的地址。

高级项目经理 任标

向上人生路！

33.B

解析:

设计时必须遵从三个黄金法则。

- 置用户于控制之下
- 减少用户的记忆负担
- 保持界面的一致

34.D

解析:

常用的白盒测试用例设计方法有:

语句覆盖、判定覆盖、条件覆盖、条件判定覆盖、条件组合覆盖、路径覆盖等,发现错误的能力呈由弱至强的变化。

- 语句覆盖每条语句至少执行一次。
- 判定覆盖每个判定的每个分支至少执行一次。
- 条件覆盖每个判定的每个条件应取到各种可能的值。
- 判定/条件覆盖同时满足判定覆盖条件覆盖。
- 条件组合覆盖每个判定中各条件的每一种组合至少出现一次。
- 路径覆盖使程序中每一条可能的路径至少执行一次。

向上人生路!

35.C

解析:

设计测试用例时,一个好的无效等价类,应该只从一个角度违反规则。C有2个维度的错误,出现错误后,不能直接定位错误的原因。

36.C

解析:

(3)完善性维护

是为扩充功能和改善性能而进行的修改,主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。

(4)预防性维护

为了改进应用软件的可靠性和可维护性,为了适应未来的软硬件环境的变化,应主动增加预防性的新的功能,以使应用系统适应各类变化而不被淘汰。

高级项目经理 任铎

向上人生路!

37.C

解析:

方法重载是指在一个类中定义多个同名的方法,但要求每个方法具有不同的参数的类型或参

数的个数。方法重载通常用于创建完成一组任务相似但参数的类型或参数的个数不同的方法。

38.B

解析：

b. 组合关系

也是关联关系的一种特例，这种关系比聚合更强，也称为强聚合，他同样体现整体与部分间的关系，但此时整体与部分是不可分的，整体的生命周期结束也就意味着部分的生命周期结束。

在UML中，使用带有实心菱形的实线——表示组合关系。

Head

- mouth : Mouth
- + Head()

Mouth

高级项目经理 任铎
QQ: 1530841586

向上人生路!

39.A

解析：

面向对象设计（含设计模式）的原则：

- (1)单一职责原则。
- (2)开放--封闭原则。
- (3)李氏(Liskov)替换原则。
- (4)依赖倒置原则。
- (5)接口隔离原则。
- (6)组合重用原则。
- (7)迪米特(Demeter)原则。

高级项目经理 任铎

向上人生路!

40.C

41.D

解析：

八、部署图 (deployment diagram)

- 用来显示系统中软件和硬件的物理架构。
- 从部署图中，可以了解到软件和硬件组件之间的物理关系以及处理节点的组件分布情况。
- 是物理方面进行建模的两种图之一

高级项目经理 任铎

向上人生路!

42.43.CB

解析:

五、顺序图 (sequence diagram)

是一种交互图 (interaction diagram)，交互图展现了一种交互，它由一组对象或角色以及它们之间可能发送的消息构成。交互图专注于系统的动态视图。顺序图是强调消息的时间次序的交互图。



向上人生路!

44~46. A D C

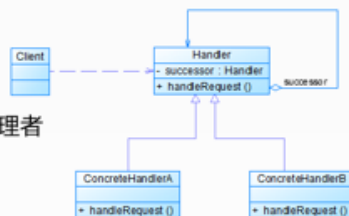
解析:

15. 责任链 (Chain of Responsibility) 高级项目经理 任务

使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。

职责链模式包含如下角色:

- Handler: 抽象处理者
- ConcreteHandler: 具体处理者
- Client: 客户类



向上人生路!

17. 迭代器模式 (Iterator)

高级项目经理 任务

提供一种方法顺序访问一个聚合对象中各个元素，而又不需暴露该对象的内部表示。

怎样遍历一个聚合对象，又不需要了解聚合对象的内部结构，还能够提供多种不同的遍历方式，这就是迭代器模式所要解决的问题。



向上人生路!

二、设计模式分为三大类：

- **创建型模式**主要用于创建对象。共五种：工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式。
- **结构型模式**主要用于处理类或对象的组合。共七种：适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式。
- **行为型模式**主要用于描述类或对象怎样交互和怎样分配职责。共十一种：策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。

向上人生路！

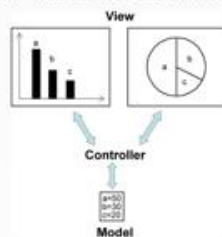
47.D

解析：

20.观察者模式 (Observer)

高级项目经理 任铎

定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。



向上人生路！

48.A

①词法分析阶段

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

一个程序语言的基本语法符号分为五类：关键字、标识符、常量、运算符、界符等

词法分析器所输出单词符号常常表示成如下的二元式：

(单词种别，单词符号的属性值)

描述词法规则通常用：正规式和有限自动机

高级项目经理 任铎
QQ: 1530841586

向上人生路！

49.C

50.B

解析：

$y = f(2)$ $x=2$; $la=x+1=3$; 引用调用 $g(int x) x=3$; $3*3+1=10$; 由于是引用调用，结果会影响 la 的值， $la=10$; $10*2=20$

$y = f(2)$ $x=2$; $la=x+1=3$; 传值调用 $g(\text{int } x) \ x=3$; $3*3+1=10$; 由于是传值调用, 结果不影响 la 的值, $la=3$; $3*2=6$

51.B

解析:

3.自然连接 (Natural join) 高级项目经理 任铎

自然连接是一种特殊的等值连接

- 两个关系中进行比较的分量必须是相同的属性组
- 在结果中把重复的属性列去掉

自然连接的含义

R和S具有相同的属性组B

$$R \bowtie S \equiv \{t_r, t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B]\}$$

向上人生路!

52.53.CA

54.C

解析:

授权命令 GRANT 语法格式):

GRANT <权限> ON 表名[(列名)] TO 用户 [WITH GRANT OPTION]

WITH GRANT OPTION 这个选项表示该用户可以将自己拥有的权限授权给别人。

55.D

解析:

(1)排他型封锁(X封锁)

如果事务T对数据A(可以是数据项、记录、数据集以至整个数据库)实现了X封锁,那么只允许事务T读取和修改数据A,其他事务要等事务T解除X封锁以后,才能对数据A实现任何类型的封锁。可见X封锁只允许一个事务独锁某个数据,具有排他性。

(2)共享型封锁(S封锁)

S封锁允许并发读,但不允许修改,也就是说,如果事务T对数据A实现了S封锁,那么允许事务T读取数据A,但不能修改数据A,在所有S封锁解除之前决不允许任何事务对数据A实现X封锁。

向上人生路!

56.C

57.D

58.C

解析: 含 n 个节点的二叉树有 $f(n)$ 种形态:(注: $n!$ 表示 n 的阶乘。如 $4! = 4*3*2*1=24$)

$$f(n) = \frac{(2n)!}{n!(n+1)!}$$

59.A

解析：

按 a,b,c,d 的顺序进入 A/B 端后，在 d 出队后，应该是 b 先出，a 才能出来。

60.B

解析：

将元素的值按散列函数进行哈希计算得到哈希地址，再将元素值存储到该地址。如果该地址已有元素，称之为存在“冲突”，当冲突后，将计算得到哈希地址+1，再判断该地址是否有元素，如果没有则存入该地址。这种冲突处理方式称为“线性探查法”。

此题， $10\%11$ 余数为 10，所以存储在 10 格内。而 $65\%11$ 余数也为 10，但 10 格内存储了数据，所以加 1，即存储到 0 格内。

61.C

解析：按下面的规则，每 1 次和 26 比较。如果查找数小于 26，则第 2 次和 15 比较。如果查找数大于 26，则第 2 次和 40 比较。所以选 C。

2、折半查找（二分查找）

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

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

向上人生路！

62~65.B C A D

66.C

67.A

解析：

短连接：客户端和服务端每进行一次 HTTP 操作，就建立一次连接，任务结束就中断连接。

长连接：当一个网页打开完成后，客户端和服务端之间用于传输 HTTP 数据的 TCP 连接不会关闭，客户端再次访问这个服务器时，会继续使用这一条已经建立的连接。

68. D

解析：

TCP 和 UDP 有各自的端口号相互独立，均使用 16 位端口号。

69.B

70.C