

1. 在采用面向对象技术构建软件系统时，很多敏捷方法都建议的一种重要的设计活动是(C)，它是一种重新组织的技术，可以简化构件的设计而无需改变其功能或行为。

A. 精化 B. 设计类 C. 重构 D. 抽象

2. 一个软件开发过程描述了“谁做”、“做什么”、“怎么做”和“什么时候做”，RUP用(A)来表述“谁做”。

A. 角色 B. 活动 C. 制品 D. workflow

RUP应用了角色、活动、制品和工作流4种重要的模型元素，其中角色表述“谁做”，制品表述“做什么”，活动表述“怎么做”，工作流表述“什么时候做”。

3. 软件能力成熟度模型(CMM)将软件能力成熟度自低到高依次划分为5级；目前，达到CMM第3级(已定义级)是许多组织努力的目标，该级的核心是(B)。

A. 建立基本的项目管理和实践来跟踪项目费用、进度和功能特性  
B. 使用标准开发过程(或方法论)构建(或集成)系统  
C. 管理层寻求更主动地应对系统的开发问题  
D. 连续地监督和改进标准化的系统开发过程

本题考查软件成熟度模型(CMM)的基本概念：

建立基本的项目管理和实践来跟踪项目费用、进度和功能特性为可重复级的核心；

使用标准开发过程(或方法论)构建(或集成)系统为已定义级的核心；

管理层寻求更主动地应对系统的开发问题为已管理级的核心；

连续地监督和改进标准化的系统开发过程为优化级的核心；

4. RUP在每个阶段都有主要目标，并在结束时产生一些制品；在(C)结束时产生“在适当的平台上集成的软件产品”。

A. 初启阶段 B. 精化阶段 C. 构建阶段 D. 移交阶段

本题考查RUP中每个阶段产生的制品：

初启阶段结束时产生一个构想文档、一个有关用例模型的调查、一个初始的业务用例、一个早期的风险评估和一个可以显示阶段和迭代的项目计划等制品；

精化阶段结束时产生一个补充需求分析、一个软件架构描述和一个可执行的架构原型等制品；

构建阶段结束时的成果是一个准备交到最终用户手中的产品，包括具有最初运作能力的在适当的平台上集成的软件产品、用户手册和对当前版本的描述；

移交阶段结束时产生移交给用户产品发布版本；

5. 根据ISO/IEC9126软件质量度量模型定义，一个软件的时间和资源质量特性属于(B)质量特性。

A. 功能性 B. 效率 C. 可靠性 D. 易使用性

本题考查ISO/IEC9126软件质量度量模型中的质量特性，效率质量特性包括时间特性和资源特性两个质量特性。

6. 在开发信息系统时，用于系统开发人员与项目管理人员沟通的主要文档是()。

A. 系统开发合同 B. 系统设计说明书  
C. 系统开发计划 D. 系统测试报告

本题考查开发文档的作用，系统开发人员与项目管理人员在项目期内进行沟通的文档主要有系统开发计划、系统开发月报以及系统开发总结报告等项目管理文件。

7. 软件工程每一个阶段结束前，应该着重对可维护性进行复审；在系统设计阶段的复审期间，应该从(B)出发，评价软件的结构和过程。

- A. 指出可移植性问题以及可能影响软件维护的系统界面
- B. 容易修改、模块化和功能独立的目的
- C. 强调编码风格和内部说明文档
- D. 可测试性

本题考查软件复审基本概念：

(可维护性是所有软件都应具有的基本特点，必须在开发阶段保证软件具有可维护的特点，可测试性是可维护性的一个评价指标)。

- 在系统分析阶段的复审过程中，应该指出软件的可移植性问题以及可能影响软件维护的系统界面；
- 在系统设计阶段的复审期间，应该从容易修改、模块化和功能独立的目的出发，评价软件的结构和过程；
- 在系统实施阶段的复审期间，代码复审应该强调编码风格和内部说明文档这两个影响可维护性的因素；

8. 在采用结构化方法进行系统分析时，根据分解与抽象的原则，按照系统中数据处理的流程，用(B)来建立系统的逻辑模型，从而完成分析工作。

- A. E-R图 B. 数据流图
- C. 程序流程图 D. 软件体系结构

本题考查结构化分析方法中图形工具的作用：数据流图摆脱系统的物理内容，在逻辑上描述系统的功能、输入、输出和数据存储等，是系统逻辑模型的重要组成部分。

9. 面向对象开发方法的基本思想是尽可能按照人类认识客观世界的方法来分析和解决问题，(D)方法不属于面向对象方法。

- A. Booch B. Coad C. OMT D. Jackson

本题考查面向对象开发方法：

面向对象开发方法有Booch方法、Coad方法和OMT方法；

Jackson方法是一种面向数据结构的开发方法；

10. 确定构建软件系统所需要的人数时，无需考虑(A)

- A. 系统的市场前景 B. 系统的规模
- C. 系统的技术复杂性 D. 项目计划

11. 一个项目为了修正一个错误而进行了变更；但这个错误被修正后，却引起以前可以正确运行的代码出错；(C)最可能发现这一问题。

- A. 单元测试 B. 接受测试
- C. 回归测试 D. 安装测试

本题考查软件测试知识：

回归测试是在软件发生变更之后进行的测试，以发现在变更时可能引起的其他错误；

12. 风险预测从两个方面评估风险，即风险发生的可能性以及(D)

- A. 风险产生的原因      B. 风险监控技术
- C. 风险能否消除      D. 风险发生所产生的后果

13. 软件能力成熟度模型(CMM)的第4级(已管理级)的核心是(C).

- A. 建立基本的项目管理和实践来跟踪项目费用、进度和功能特性
- B. 组织具有标准软件过程
- C. 对软件过程和产品都有定量的理解和控制
- D. 先进的新思想和新技术促进过程不断改进

14. 软件系统设计的主要目的是为系统制定蓝图, (D)并不是软件设计模型所关注的.

- A. 系统总体结构    B. 数据结构    C. 界面模型    D. 项目范围

15. ISO/IEC9126软件质量模型中, 可靠性质量特性包括多个子特性;一软件在故障发生后, 要求在90秒内恢复其性能和受影响的数据, 与达到此目的有关的软件属性为(C)子特性。

- A. 容错性    B. 成熟性    C. 易恢复性    D. 易操作性

16. 系统开发计划用于系统开发人员与项目管理人员在项目期内进行沟通, 它包括(A)和预算分配表等.

- A. PERT图    B. 总体规划    C. 测试计划    D. 开发合同

用于系统开发人员与项目管理人员在项目期内进行沟通的文档主要有系统开发计划, 包括工作任务分解表、PERT图、甘特图和预算分配表等; 总体规划和开发合同用于与系统分析人员在系统规划和系统分析阶段的沟通; 测试计划用于系统测试人员与系统开发人员之间的沟通.

17. 改正在软件系统开发阶段已经发生而系统测试阶段还没有发现的错误, 属于(A)维护。

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

软件维护的内容一般包括准确性维护、适应性维护、完善性维护和预防性维护:

正确性维护是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误;

适应性维护是指使应用软件适应信息技术变化和管理需求变化而进行的修改;

完善性维护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;

预防性维护是指为了适应未来的软硬件环境的变化, 主动增加预防性的新功能, 以使应用系统适应各类变化而不被淘汰;

18. 某系统重用了第三方组件(但无法获得其源代码), 则应采用(D)对组件进行测试。

- A. 基本路径覆盖    B. 分支覆盖    C. 环路覆盖    D. 黑盒测试

软件测试方法分为静态测试和动态测试:

- 静态测试是被测试程序不在机器上运行而采用人工检测和计算机辅助静态分析手段对程序进行检测;
- 动态测试是通过运行程序发现错误。对软件产品进行动态测试时可以采用黑盒测试和白盒测试法:
  - 黑盒测试法在完全不考虑软件的内部结构和特性的情况下, 测试软件的外部特性;
  - 对于无法获得源代码的第三方组件, 可采用黑盒对组件进行测试;

基本路径覆盖、分支覆盖和环路覆盖需要根据程序的内部结构和逻辑进行测试。

19. 极限编程(XP)由价值观、原则、实践和行为四个部分组成，其中价值观包括沟通、简单性、(C)。

- A. 好的计划 B. 不断的发布 C. 反馈和勇气 D. 持续集成

极限编程(XP)是敏捷开发的典型方法之一，是一种轻量级(敏捷)、高效、低风险、柔性、可预测的、科学的软件开发方法，它由价值观、原则、实践和行为4个部分组成；其中4大价值观为沟通、简单性、反馈和勇气。

20. 以下关于面向对象分析的叙述中，错误的是(B)。

- A. 面向对象分析着重分析问题域和系统责任  
B. 面向对象分析需要考虑系统的测试问题  
C. 面向对象分析忽略与系统实现有关的问题  
D. 面向对象分析建立独立于实现的系统分析模型

21. 以下关于面向对象设计的叙述中，错误的是(D)。

- A. 高层模块不应该依赖于底层模块  
B. 抽象不应该依赖于细节  
C. 细节可以依赖于抽象  
D. 高层模块无法不依赖于底层模块

22. 基于构件的软件开发，强调使用可复用的软件“构件”来设计和构建软件系统，对所需的构件进行合格性检验、(C)，并将它们集成到新系统中。

- A. 规模度量 B. 数据验证 C. 适应性修改 D. 正确性测试

23. 采用面向对象方法开发软件的过程中，抽取和整理用户需求并建立问题域精确模型的过程叫(D)。

- A. 面向对象测试 B. 面向对象实现  
C. 面向对象设计 D. 面向对象分析

采用面向对象的软件开发，通常有面向对象分析、面向对象设计、面向对象实现：

面向对象分析是为了获得对应用问题的理解，其主要任务是抽取和整理用户需求并建立问题域精确模型；

面向对象设计是采用协作的对象、对象的属性和方法说明软件解决方案的一种方式，强调的是定义软件对象和这些软件对象如何协作来满足需求，延续了面向对象分析；

面向对象实现主要强调采用面向对象程序设计语言实现系统；

面向对象测试是根据规范说明来验证系统设计的正确性；

24. 使用白盒测试方法时，应根据(A)和指定的覆盖标准确定测试数据。

- A. 程序的内部逻辑 B. 程序结构的复杂性  
C. 使用说明书 D. 程序的功能

白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的执行路径和过程进行测试，检查是否满足设计的需要；白盒测试常用的技术涉及不同覆盖标准，在测试时需根据指定的覆盖标准确定测试数据。

25. 对于一个大型软件来说，不加控制地变更很快就会引起混乱；为有效地实现变更控制，需借助于配置数据库和基线的概念；(C)不属于配置数据库。

- A. 开发库 B. 受控库 C. 信息库 D. 产品库

软件变更控制是变更管理的重要内容，要有效进行变更控制，需要借助配置数据库和基线的概念；配置数据库一般包括开发库、受控库和产品库。

26. 软件设计时需要遵循抽象、模块化、信息隐蔽和模块独立原则；在划分软件系统模块时，应尽量做到()

- A. 高内聚高耦合    B. 高内聚低耦合  
C. 低内聚高耦合    D. 低内聚低耦合

27. 能力成熟度集成模型CMMI是CMM模型的最新版本，它有连续式和阶段式两种表示方式；基于连续式表示的CMMI共有6个(0~5)能力等级，每个能力等级对应到一个一般目标以及一组一般执行方法和特定方法，其中能力等级(C)主要关注过程的组织标准化和部署。

- A. 1    B. 2    C. 3    D. 4

本题考查软件能力成熟度集成模型的基础知识。

能力成熟度集成模型CMMI是CMM模型的最新版本，基于连续式表述的CMMI共有6个(0~5)能力等级，对应于未完成级、已执行级、已管理级、已定义级、量化管理级、优化级；每个能力等级对应到一个一般目标，以及一组一般执行方法和特定方法。

- 能力等级0指未执行过程，表明过程域的一个或多个特定目标没有被满足；
- 能力等级1指过程通过转化可识别的输入工作产品，产生可识别的输出工作产品，关注于过程域的特定目标的完成；
- 能力等级2指过程作为已管理的过程制度化，针对单个过程实例的能力；
- 能力等级3指过程作为已定义的过程制度化，关注过程的组织级标准化和部署；
- 能力等级4指过程作为定量管理的过程制度化；
- 能力等级5指过程作为优化的过程制度化，表明过程得到很好地执行且持续得到改进；

28. 统一过程(UP)定义了初启阶段、精化阶段、构建阶段、移交阶段和产生阶段，每个阶段以达到某个里程碑时结束，其中(B)的里程碑是生命周期架构。

- A. 初启阶段    B. 精化阶段    C. 构建阶段    D. 移交阶段

统一过程(UP)定义了初启阶段、精化阶段、构建阶段、移交阶段和产生阶段，每个阶段达到某个里程碑时结束：

其中初启阶段的里程碑是生命周期目标，精化阶段的里程碑是生命周期架构，构建阶段的里程碑是初始运作功能，移交阶段的里程碑是产品发布；

29. (C)不属于软件配置管理的活动。

- A. 变更标识    B. 变更控制    C. 质量控制    D. 版本控制

30. 确定系统边界和关系规范化分别在数据库设计的(A)阶段进行。

- A. 需求分析和逻辑设计    B. 需求分析和概念设计  
C. 需求分析和物理设计    D. 逻辑设计和概念设计

31. 某项目组拟开发一个大规模系统，且具备了相关领域及类似规模系统的开发经验；下列过程模型中，(B)最适合开发此项目。

- A. 原型模型    B. 瀑布模型    C. V模型    D. 螺旋模型

常见的软件生存周期模型有瀑布模型、演化模型、螺旋模型、喷泉模型等。

瀑布模型是将软件生存周期各个活动规定为依线性顺序连接的若干阶段的模型，适合于软件需求很明确的软件项目；

V模型是瀑布模型的一种演变模型，将测试和分析与设计关联进行，加强分析与设计的验证；

原型模型是一种演化模型，通过快速构建可运行的原型系统，然后根据运行过程中获取的用户反馈进行改进；



演化模型特别适用于对软件需求缺乏准确认识的情况；

螺旋模型将瀑布模型和演化模型结合起来，加入了两种模型均忽略的风险分析。

32. 敏捷开发方法XP是一种轻量级、高效、低风险、柔性、可预测的、科学的软件开发方法，其特性包含在12个最佳实践中；系统的设计要能够尽可能早交付，属于(C)最佳实践。

A. 隐喻 B. 重构 C. 小型发布 D. 持续集成

敏捷开发方法XP是一种轻量级、高效、低风险、柔性、可预测的、科学的软件开发方法，其特性包含在12个最佳实践中：

(1)计划游戏：快速制定计划、随着细节的不断变化而完善；

(2)小型发布：系统的设计要能够尽可能早地交付；

(3)隐喻：找到合适的比喻传达信息；

(4)简单设计：只处理当前的需求使设计保持简单；

(5)测试先行：先写测试代码再编写程序；

(6)重构：重新审视需求和设计，重新明确地描述它们，以符合新的和现有的需求；

(7)结对编程；

(8)集体代码所有制；

(9)持续集成：可以按日甚至按小时为客户提供可运行的版本；

(10)每周工作40个小时；

(11)现场客户；

(12)编码标准。

33. 在软件开发过程中进行风险分析时，(D)活动的目的是辅助项目组建立处理风险的策略，有效的策略应考虑风险避免、风险监控、风险管理及意外事件计划。

A. 风险识别 B. 风险预测 C. 风险评估 D. 风险控制

风险分析实际上是4个不同的活动：风险识别、风险预测、风险评估和风险控制。

风险识别是试图系统化地确定对项目计划(估算、进度、资源分配)的威胁；

风险预测又称为风险估算，它从两个方面评估一个风险：风险发生的可能性或概率；以及如果风险发生时所产生的后果；

风险评估根据风险及其发生的概率和产生的影响预测是否影响参考水平值；

风险控制的目的是辅助项目组建立处理风险的策略，有效的策略应考虑风险避免、风险监控、风险管理及意外事件计划；

34. 以下关于过程改进的叙述中，错误的是(B)。

A. 过程能力成熟度模型基于这样的理念：改进过程将改进产品，尤其是软件产品

B. 软件过程改进框架包括评估、计划、改进和监控4个部分

C. 软件过程改进不是一次性的，需要反复进行

D. 在评估后要把发现的问题转化为软件过程改进计划

软件成熟度模型CMM是对软件组织进化阶段的描述，该模型在解决软件过程存在问题方面取得了很大的成功，因此在软件界产生了巨大影响，促使软件界重视并认真对待过程改进工作。

过程能力成熟度模型基于这样的理念：改进过程将改进产品，尤其是软件产品。

软件组织为提高自身的过程能力，把不够成熟的过程提升到较成熟的过程涉及4个方面，这4个方面构成了软件过程改进的框架，即过程改进基础设施、过程改进线路图、软件过程评估方法和软件过程改进计划。

在进行评估后需要把发现的问题转化为软件过程改进计划，而过程改进通常不可能是一次性的，需要反复进行；每一次改进要经历4个步骤：评估、计划、改进和监控。

35. 软件复杂度量度的参数不包括(B)

- A. 软件的规模    B. 开发小组的规模
- C. 软件的难度    D. 软件的结构

软件复杂度量度是软件度量的一个重要分支。

软件复杂度量度的参数有很多，主要包括：

- (1)规模，即指令数或者源程序行数；
- (2)难度，通常由程序中出现的操作数所决定的量来表示；
- (3)结构，通常用与程序结构有关的度量来表示；
- (4)智能度，即算法的难易程度；

36. 软件系统的可维护性评价指标不包括(C).

- A. 可理解性    B. 可测试性    C. 可扩展性    D. 可修改性

37. 以下关于软件系统文档的叙述中，错误的是(C).

- A. 软件系统文档既包括有一定格式要求的规范文档，又包括系统建设过程中的各种来往文件、会议纪要、会计单据等资料形成的不规范文档
- B. 软件系统文档可以提高软件开发的可见度
- C. 软件系统文档不能提高软件开发效率
- D. 软件系统文档便于用户理解软件的功能、性能等各项指标

38. 以下关于软件测试的叙述中，正确的是(A).

- A. 软件测试不仅能表明软件中存在错误，也能说明软件中不存在错误
- B. 软件测试活动应从编码阶段开始
- C. 一个成功的测试能发现至今未发现的错误
- D. 在一个被测程序段中，若已发现的错误越多，则残存的错误数越少

39. 不属于黑盒测试技术的是(B).

- A. 错误猜测    B. 逻辑覆盖    C. 边界值分析    D. 等价类划分

黑盒测试也称为功能测试，在完全不考虑软件的内部结构和特性的情况下来测试软件的外部特性；常用的黑盒测试技术包括等价类划分、边界值分析、错误猜测和因果图的报告。

白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的执行路径和过程进行测试，检查是否满足设计的需要；常用的白盒测试技术包括逻辑覆盖和基本路径测试。

40. 耦合是模块之间的相对独立性(互相连接的紧密程度)的度量，耦合程度不取决于(D)。

- A. 调用模块的方式
- B. 各个模块之间接口的复杂程度
- C. 通过接口的信息类型
- D. 模块提供的功能数

41. 为了有效地捕获系统需求，应采用(C)

- A. 瀑布模型
- B. V模型
- C. 原型模型
- D. 螺旋模型

42. 关于过程改进，以下叙述中不正确的是(D)。

- A. 软件质量依赖于软件开发过程的质量，其中个人因素占主导作用
- B. 要使过程改进有效，需要制定过程改进目标
- C. 要使过程改进有效，需要进行培训
- D. CMMI成熟度模型是一种过程改进模型，仅支持阶段性过程改进而不支持连续性过程改进

43. 软件产品的可靠性并不取决于(D)。

- A. 潜在错误的数量
- B. 潜在错误的位置
- C. 软件产品的使用方式
- D. 软件产品的开发方式

44. 软件(A)是指一个系统在给定时间间隔内和给定条件下无失效运行的概率。

- A. 可靠性
- B. 可用性
- C. 可维护性
- D. 可伸缩性

45. 高质量的文档所应具有的特性中，不包括(D)。

- A. 针对性，文档编制应考虑读者对象群
- B. 精确性，文档的行文应该十分确切，不能出现多义性的描述
- C. 完整性，任何文档都应当是完整的、独立的，应该自成体系
- D. 无重复性，同一软件系统的几个文档之间应该没有相同的内容，若确实存在相同内容，则可以用“见XX文档XX节”的方式引用。

46. 在软件维护阶段，为软件的运行增加监控设施属于(C)维护。

- A. 改正性
- B. 适应性
- C. 完善性
- D. 预防性

在软件开发完成交付用户使用后，就进入软件运行/维护阶段，软件维护活动根据其内容可以分为4种类型：

改正性维护，为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应进行的诊断和改正错误的过程；

适应性维护，由于信息技术飞速发展，软件运行的外部环境或数据环境可能会发生变化，为了使软件适应这种变化，而修改软件的过程；



完善性维护，在软件使用过程中，用户往往会对软件提出新的功能与性能要求，为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性而进行的维护活动；

预防性维护是为了提高软件的可维护性和可靠性等，为以后进一步改进软件打下良好基础而进行的维护工作。

47. 若全面采用新技术开发一个大学记账系统，以替换原有的系统，则宜选择采用(A)进行开发。

A. 瀑布模型    B. 演化模型    C. 螺旋模型    D. 原型模型

48. 将每个用户的数据和其他用户的数据隔离开，是考虑了软件的(A)质量特性。

A. 功能性    B. 可靠性    C. 可维护性    D. 易使用性

49. 确定软件的模块划分及模块之间的调用关系是(B)阶段的任务。

A. 需求分析    B. 概要设计    C. 详细设计    D. 编码

需求分析确定软件要完成的功能及非功能性要求；概要设计将需求转化为软件的模块划分，确定模块之间的调用关系；详细设计将模块进行细化，得到详细的数据结构和算法；编码根据详细设计进行代码的编写，得到可以运行的软件，并进行单元测试。

50. 利用结构化分析模型进行接口设计时，应以(A)为依据。

A. 数据流图    B. 实体一关系图    C. 数据字典    D. 状态一迁移图

51. 数据流图(DFD)对系统的功能和功能之间的数据流进行建模，其中顶层数据流图描述了系统的(B)。

A. 处理过程    B. 输入与输出    C. 数据存储    D. 数据实体

52. 模块A执行几个逻辑上相似的功能，通过参数确定该模块完成哪一个功能，则该模块具有(C)内聚。

A. 顺序    B. 过程    C. 逻辑    D. 功能

53. (C)最不适于采用无主程序员组的开发人员组织形式。

A. 开发人数少(如3~4人)的项目    B. 采用新技术的项目  
C. 大规模项目    D. 确定性较小的项目

54. 若软件项目组对风险采用主动的控制方法，则(A)是最好的风险控制策略。

A. 风险避免    B. 风险监控  
C. 风险消除    D. 风险管理及意外事件计划

55. 软件开发的增量模型(B)

- A. 最适用于需求被清晰定义的情况
- B. 是一种能够快速构造可运行产品的好方法
- C. 最适合于大规模团队开发的项目
- D. 是一种不适用于商业产品的创新模型

56. 以下关于数据流图的叙述中，不正确的是(C)

- A. 每条数据流的起点或终点必须是加工
- B. 必须保持父图与子图平衡
- C. 每个加工必须有输入数据流，但可以没有输出数据流
- D. 应保持数据守恒

57. 在软件设计阶段，划分模块的原则是：一个模块的(A).

- A. 作用范围应该在其控制范围之内      B. 控制范围应该在其作用范围之内
- C. 作用范围与控制范围互不包含      D. 作用范围与控制范围不受任何限制

58. 定义风险参照水准是(C)活动常用的技术.

- A. 风险识别      B. 风险预测      C. 风险评估      D. 风险控制

59. 以下关于文档的叙述中，不正确的是(A).

- A. 文档仅仅描述和规定了软件的使用范围及相关的操作命令
- B. 文档是软件产品的一部分，没有文档的软件不能称之为软件产品
- C. 软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量
- D. 高质量文档对于发挥软件产品的效益有着重要的意义

60. 如果要表示待开发软件系统中软件组件和硬件之间的物理关系，通常采用UML中的(B).

- A. 组件图      B. 部署图      C. 类图      D. 网络图

61. 在设计测试用例时，应遵循(C)原则.

- A. 仅确定测试用例的输入数据，无须考虑输出结果
- B. 只需检验程序是否执行应有的功能，不需要考虑程序是否做了多余的功能
- C. 不仅要设计有效合理的输入，也要包含不合理、失效的输入
- D. 测试用例应设计得尽可能复杂

62. 单元测试中，检验模块接口时，不需要考虑(D).

- A. 测试模块的输入参数和形式参数在个数、属性、单位上是否一致
- B. 全局变量在各模块中的定义和用法是否一致
- C. 输入是否改变了形式参数
- D. 输入参数是否使用了尚未赋值或者尚未初始化的变量

63. 以下关于UML部署图的叙述中，正确的是(B).

- A. 因为一条消息总是有某种响应，部署组件之间的依赖是双向的
- B. 部署组件之间的依赖关系类似于包图
- C. 部署图不用于描述代码的物理模块
- D. 部署图不用于描述系统在不同计算机系统的物理分布