窗体顶端

# 软件工程

第 6 章：软件工程作者：[希赛软考学院](http://www.educity.cn/jiaocheng/a%cf%a3%c8%fc%c8%ed%bf%bc%d1%a7%d4%ba.html" \o "希赛软考学院" \t "http://www.educity.cn/jiaocheng/_blank)    来源：希赛软考学院    2017年11月21日

# **试题1(2017年下半年试题19)**

某企业财务系统的需求中，属于功能需求的是（  ）。  
（19）A．每个月特定的时间发放员工工资  
B.系统的响应时间不超过 3 秒  
C.系统的计算精度符合财务规则的要求  
D.系统可以允许100个用户同时查询自己的工资

**试题分析**

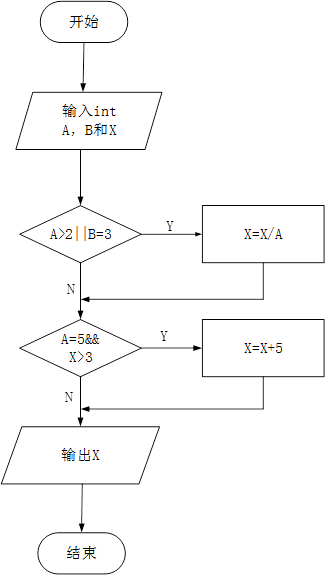
BCD为非功能需求。

**试题答案**

（19）A

# **试题2(2017年下半年试题29-30)**

用自盒测试方法对如下图所示的流程图进行测试。若要满足分支覆盖，则至少需要（  ）个测试用例，正确的测试用例对是（  ）（测试用例的格式为（A，B，X；X））。



（29）A．1  
B.2  
C.3  
D.4  
  
（30）A．（1,3,3;3）和（5,2,15;3）  
B.（1,1,5;5）和（5,2,20;9）  
C.（2,3,10;5）和（5,2,18;3）  
D.（5,2,16;3）和（5,2,21;9）

**试题分析**

1、两个测试用例，一个走真分支，一个走假分支即可。  
2、A（5，2，15，8） B（1，1，5，10）C(2,3,10,10)。

**试题答案**

（29）B（30）D

# **试题3(2017年下半年试题32)**

极限编程（XP）的十二个最佳实践不包括（  ）。  
（32）A．小的发布  
B.结对编程  
C.持续集成  
D.精心设计

**试题分析**

D应为简单设计。

**试题答案**

（32）D

# **试题4(2017年下半年试题33)**

以下关于管道过滤器体系结构的有点的叙述中，不正确的是（  ）。  
（33）A．软件构件具有良好的高内聚、低耦合的特点  
B.支持重用  
C.支持并行执行  
D.提高性能

**试题分析**

管道过滤器不支持批处理和并发操作。

**试题答案**

（33）C

# **试题5(2017年下半年试题34)**

模块A将学生信息，即学生姓名、学号、手机号等放到一个结构体中，传递给模块B。模块A和B之间的耦合类型为（  ）耦合。  
（34）A．数据  
B.标记  
C.控制  
D.内容

**试题分析**

题干说明数据是参数传递的，所以是数据耦合。

**试题答案**

（34）A

# **试题6(2017年下半年试题35)**

某模块内涉及多个功能，这些功能必须以特定的次序执行，则该模块的内聚类型为（  ）内聚。  
（35）A．时间  
B.过程  
C.信息  
D.功能

**试题分析**

要求功能是以特定的次序执行，所以是过程内聚。

**试题答案**

（35）B

# **试题7(2017年下半年试题36)**

系统交付用户使用后，为了改进系统的图形输出而对系统进行修改的维护行为属于（  ）维护。  
（36）A．改正性  
B.适应性  
C.改善性  
D.预防性

**试题分析**

改善系统的功能和性能

**试题答案**

（36）C

# **试题8(2017年下半年试题56)**

 在基于Web 的电子商务应用中，访问存储于数据库中的业务对象的常用方式之一是（  ）。  
（56）A．JDBC  
B.XML  
C.CGI  
D.COM

**试题分析**

数据库访问是jDBC（Java DataBase Connectivity,java数据库连接）

**试题答案**

（56）A

# **试题9(2017年下半年试题70)**

测试网络连通性通常采用的命令是（  ）。  
（70）A．Netstat  
B.Ping  
C.Msconfig  
D.Cmd

**试题分析**

Ping是Windows、Unix和Linux系统下的一个命令。ping也属于一个通信协议，是TCP/IP协议的一部分。利用“ping”命令可以检查网络是否连通，可以很好地帮助我们分析和判定网络故障。

**试题答案**

（70）B

# **试题10(2017年上半年试题29)**

以下关于螺旋模型的叙述中，不正确的是（  ）。  
  
（29）A．它是风险驱动的，要求开发人员必须具有丰富的风险评估知识和经验  
B.它可以降低过多测试或测试不足带来的风险  
C.它包含维护周期，因此维护和开发之间没有本质区别  
D.它不适用于大型软件开发

**试题分析**

螺旋模型是一种演化软件开发过程模型，它兼顾了快速原型的迭代的特征以及瀑布模型的系统化与严格监控。螺旋模型最大的特点在于引入了其他模型不具备的风险分析，使软件在无法排除重大风险时有机会停止，以减小损失。同时，在每个迭代阶段构建原型是螺旋模型用以减小风险的途径。螺旋模型更适合大型的昂贵的系统级的软件应用。

**试题答案**

（29）D

# **试题11(2017年上半年试题30)**

以下关于极限编程(XP) 中结对编程的叙述中，不正确的是（  ）。  
  
（30）A． 支持共同代码拥有和共同对系统负责  
B. 承担了非正式的代码审查过程  
C. 代码质量更高  
D. 编码速度更快

**试题分析**

极限编程是一个轻量级的、灵巧的软件开发方法；同时它也是一个非常严谨和周密的方法。它的基础和价值观是交流、朴素、反馈和勇气；即，任何一个软件项目都可以从四个方面入手进行改善：加强交流；从简单做起；寻求反馈；勇于实事求是。XP是一种近螺旋式的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。XP就提倡结对编程（PairProgramming），而且代码所有权是归于整个开发队伍。其中的结对编程就是一种对代码的审查过程，XP主要解决代码质量低的问题，编码速度不能改变。

**试题答案**

（30）D

# **试题12(2017年上半年试题32)**

在设计软件的模块结构时，（  ）不能改进设计质量。  
  
（32）A．尽量减少高扇出结构  
B.模块的大小适中  
C.将具有相似功能的模块合并  
D.完善模块的功能

**试题分析**

在结构化设计中，系统由多个逻辑上相对独立的模块组成，在模块划分时需要遵循如下原则：  
（1）模块的大小要适中。系统分解时需要考虑模块的规模，过大的模块可能导致系统分解不充分，其内部可能包括不同类型的功能，需要进一步划分，尽量使得各个模块的功能单一；过小的模块将导致系统的复杂度增加，模块之间的调用过于频繁，反而降低了模块的独立性。一般来说，一个模块的大小使其实现代码在1～2页纸之内，或者其实现代码行数在50～200行之间，这种规模的模块易于实现和维护。  
（2）模块的扇入和扇出要合理。一个模块的扇出是指该模块直接调用的下级模块的个数；扇出大表示模块的复杂度高，需要控制和协调过多的下级模块。扇出过大一般是因为缺乏中间层次，应该适当增加中间层次的控制模块；扇出太小时可以把下级模块进一步分解成若干个子功能模块，或者合并到它的上级模块中去。一个模块的扇入是指直接调用该模块的上级模块的个数；扇入大表示模块的复用程度高。设计良好的软件结构通常顶层扇出比较大，中间扇出较少，底层模块则有大扇入。一般来说，系统的平均扇入和扇出系数为3或4，不应该超过7，否则会增大出错的概率。  
（3）深度和宽度适当。深度表示软件结构中模块的层数，如果层数过多，则应考虑是否有些模块设计过于简单，看能否适当合并。宽度是软件结构中同一个层次上的模块总数的最大值，一般说来，宽度越大系统越复杂，对宽度影响最大的因素是模块的扇出。在系统设计时，需要权衡系统的深度和宽度，尽量降低系统的复杂性，减少实施过程的难度，提高开发和维护的效率。

**试题答案**

（32）D

# **试题13(2017年上半年试题33-34)**

模块A、B和 C有相同的程序块，块内的语句之间没有任何联系，现把改程序块取出来，形成新的模块D，则模块D的内聚类型为（  ）内聚。以下关于该内聚类型的叙述中，不正确的是（  ）。  
  
A．巧合  
B.逻辑  
C.时间  
D.过程  
  
（34）A．具有最低的内聚性  
B.不易修改和维护  
C.不易理解  
D.不影响模块间的耦合关系

**试题分析**

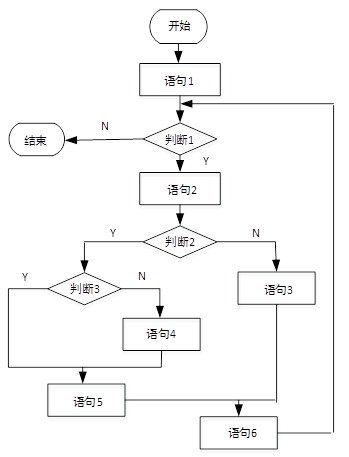
功能内聚：完成一个单一功能，各个部分协同工作，缺一不可。  
顺序内聚：处理元素相关，而且必须顺序执行。  
通信内聚：所有处理元素集中在一个数据结构的区域上。  
过程内聚：处理元素相关，而且必须按特定的次序执行。  
瞬时内聚：所包含的任务必须在同一时间间隔内执行（如初始化模块）。  
逻辑内聚：完成逻辑上相关的一组任务。  
偶然内聚：完成一组没有关系或松散关系的任务。  
巧合内聚就是偶然内聚。偶然内聚由于内容都是不相关的，所以必然导致它与外界多个模块有关联，这也使得模块间的耦合度增加。

**试题答案**

（33）A（34）D

# **试题14(2017年上半年试题35-36)**

对下图所示的程序流程图进行语句覆盖测试和路径覆盖测试，至少需要(  )个测试用例。采用McCabe 度量法计算其环路复杂度为（  ）。



（35）A．2和3  
B.2和4  
C.2和5  
D.2和6  
  
（36）A．1  
B.2  
C.3  
D.4

**试题分析**

要满足语句覆盖的要求，只需要覆盖两条路径就能达到，所以语句覆盖2个用例即可。路径覆盖需要把程序中的4条路径均覆盖一遍，需要4个用例。  
整个程序流程图转化为结点图之后，一共11个结点，13条边，根据环路复杂度公式有：13－11+2＝4。

**试题答案**

（35）B（36）D

# **试题15(2016年下半年试题15)**

结构化开发方法中，（  ）主要包含对数据结构和算法的设计。  
  
（15）A．体系结构设计  
B.数据设计  
C.接口设计  
D.过程设计

**试题分析**

数据结构跟算法是系统的基础，是过程设计确定的任务。

**试题答案**

（15）D

# **试题16(2016年下半年试题16)**

在敏捷过程的开发方法中，（  ）使用了迭代的方法，其中，把每段时间（30天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。  
  
（16）A．极限编程XP  
B.水晶法  
C.并列争球法  
D.自适应软件开发

**试题分析**

并列争球法使用了迭代的方法，其中，把每段时间（30天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。

**试题答案**

（16）C

# **试题17(2016年下半年试题29-30)**

某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不适宜采用（  ）模型，主要是因为这种模型（  ）。  
（29）A．瀑布  
B.原型  
C.增量  
D.喷泉   
  
（30）A．不能解决风险  
B.不能快速提交软件  
C.难以适应变化的需求  
D.不能理解用户的需求

**试题分析**

对于较大型软件系统的需求往往难以在前期确定，所以瀑布模型最不适合。

**试题答案**

（29）A（30）C

# **试题18(2016年下半年试题31)**

（  ）不属于软件质量特性中的可移植性。  
  
（31）A．适应性  
B.易安装性  
C.易替换性  
D.易理解性

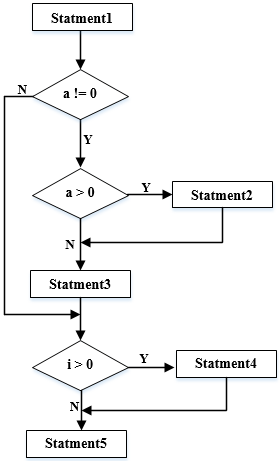
**试题分析**

可移植性包含：适应性、易安装性、共存性和易替换性四个特性。

**试题答案**

（31）D

# **试题19(2016年下半年试题32-33)**

对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要（  ）个测试用例。采用McCabe度量法计算该程序的环路复杂性为（  ）。  
  
（32）A．3  
B.4  
C.6  
D.8   
  
（33）A．1  
B.2  
C.3  
D.4

**试题分析**

环形复杂度V(G)=E-N+2，其中，E是流图中边的条数，N是结点数。  
V(G)=E-N+2=10-8+2=4。

**试题答案**

（32）C（33）D

# **试题20(2016年下半年试题35)**

以下关于软件测试的叙述中，不正确的是（  ）。  
  
（35）A．在设计测试用例时应考虑输入数据和预期输出结果  
B.软件测试的目的是证明软件的正确性  
C.在设计测试用例时，应该包括合理的输入条件  
D.在设计测试用例时，应该包括不合理的输入条件

**试题分析**

软件测试的目的是为了发现尽可能多的缺陷。

**试题答案**

（35）B

# **试题21(2016年下半年试题36)**

某模块中有两个处理A和B，分别对数据结构X写数据和读数据，则该模块的内聚类型为（  ）内聚。  
（36）A．逻辑  
B.过程  
C.通信  
D.内容

**试题分析**

如果一个模块的所有成分都操作同一数据集或生成同一数据集，则称为通信内聚。

**试题答案**

（36）C

# **试题22(2016年上半年试题17-18)**

在结构化分析中，用数据流图描述（  ）。当采用数据流图对一个图书馆管理系统进行分析时，（  ）是一个外部实体。  
（17）A．数据对象之间的关系，用于对数据建模   
B.数据在系统中如何被传送或变换，以及如何对数据流进行变换的功能或子功能，用于对功能建模  
C.系统对外部事件如何响应，如何动作，用于对行为建模   
D.数据流图中的各个组成部分   
  
（18）A．读者   
B.图书   
C.借书证   
D.借阅

**试题分析**

数据流图用来记录系统中的数据和数据在特定的过程中的流动，即数据如何被采集、处理、保存和使用的（围绕信息系统的功能）。外部实体指系统之外又与系统有联系的人或事物。它表达了该系统数据的外部来源和去处。

**试题答案**

（17）B（18）A

# **试题23(2016年上半年试题19)**

软件开发过程中，需求分析阶段的输出不包括（  ）。  
（19）A．数据流图   
B.实体联系图   
C.数据字典   
D.软件体系结构图

**试题分析**

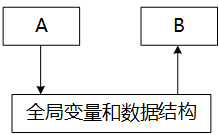
其他三项均为需求分析阶段确定。

**试题答案**

（19）D

# **试题24(2016年上半年试题29)**

如下图所示，模块A和模块B都访问相同的全局变量和数据结构，则这两个模块之间的耦合类型为（  ）耦合。



（29）A．公共   
B.控制   
C.标记   
D.数据

**试题分析**

公共耦合指通过一个公共数据环境相互作用的那些模块间的耦合。

**试题答案**

（29）A

# **试题25(2016年上半年试题30)**

以下关于增量开发模型的叙述中，不正确的是（  ）。  
（30）A．不必等到整个系统开发完成就可以使用   
B.可以使用较早的增量构件作为原型，从而获得稍后的增量构件需求   
C.优先级最高的服务先交付，这样最重要的服务接受最多的测试   
D.有利于进行好的模块划分

**试题分析**

在利用增量模型进行开发时，如何进行模块的划分往往是难点所在，而不是这种模型的优点。

**试题答案**

（30）D

# **试题26(2016年上半年试题31)**

在设计软件的模块结构时，（  ）不能改进设计质量。  
（31）A．模块的作用范围应在其控制范围之内   
B.模块的大小适中   
C.避免或减少使用病态连接（从中部进入或访问一个模块）   
D.模块的功能越单纯越好

**试题分析**

高内聚是使模块的功能要相对独立和单一，这个功能单纯意思有出入

**试题答案**

（31）D

# **试题27(2016年上半年试题32-33)**

软件体系结构的各种风格中，仓库风格包含一个数据仓库和若干个其他构件。数据仓库位于该体系结构的中心，其他构件访问该数据仓库并对其中的数据进行增、删、改等操作。以下关于该风格的叙述中，不正确的是（  ）。（  ）不属于仓库风格。  
（32）A．支持可更改性和可维护性   
B.具有可复用的知识源   
C.支持容错性和健壮性   
D.测试简单   
  
（33）A．数据库系统   
B.超文本系统   
C.黑板系统   
D.编译器

**试题分析**

仓库风格优点包括：

1、解决问题的多方法性

2、具胡可更改性和可维护性

3、有可重用的知识源

4、支持容错性和健壮性

缺点：

1、测试困难：由于黑板模式的系统有中央数据构件来描述系统的体现系统的状态，所以系统的执行没有确定的顺序，其结果的可再现性比较差，难于测试，

2、不能保证有好的求解方案

3、效率低

4、开发成本高

5、缺少对并行机的支持

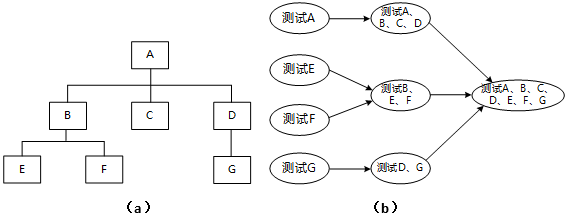
仓库风格包括：数据库系统、黑板系统、超文本系统。  
编译器可用多种架构风格实现。

**试题答案**

（32）D（33）D

# **试题28(2016年上半年试题34-35)**

下图（a）所示为一个模块层次结构的例子，图（b）所示为对其进行集成测试的顺序，则此测试采用了（  ）测试策略。该测试策略的优点不包括（  ）。



（34）A．自底向上   
B.自顶向下   
C.三明治   
D.一次性   
  
（35）A．较早地验证了主要的控制和判断点   
B.较早地验证了底层模块   
C.测试的并行程度较高   
D.较少的驱动模块和桩模块的编写工作量

**试题分析**

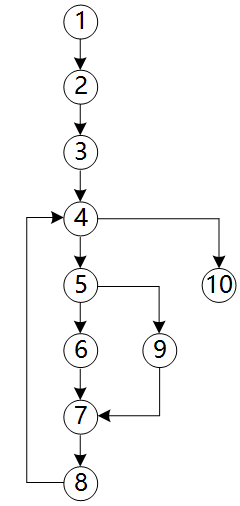
从先测试A，再测试A、B、C、D可以看出集成测试时用到了自顶向下的方式。  
而从先测试E、F，再测试B、E、F可以看出集成测试时用到了自底向上的方式，两者结合即为三明治方式。  
这种策略的优点是自顶向下与自底向上两种方式优点的综合。所以可以较早地验证了主要的控制和判断点且较早地验证了底层模块。同时由于可以两端向中间发展，所以效率也是比较高的，且运用一定的技巧，能够减少了桩模块和驱动模块的开发。所以本题的问题本身存在不严谨的现象。在选择答案时，由于”较少的驱动模块和桩模块的编写工作量“条款需要运用一定技巧，并非一定会减少，所以选其相对更合适。

**试题答案**

（34）C（35）D

# **试题29(2016年上半年试题36)**

采用McCabe度量法计算下图所示程序的环路复杂性为（  ）。



（36）A．1   
B.2   
C.3   
D.4

**试题分析**

环形复杂度V(G)=E-N+2，其中，E是流图中边的条数，N是结点数。  
V(G)=E-N+2=11-10+2=3

**试题答案**

（36）C

# **试题30(2015年下半年试题15-16)**

若用户需求不清晰且经常发生变化，但系统规模不太大且不太复杂，则最适宜采用（  ）开发方法，对于数据处理领域的问题，若系统规模不太大且不本复杂，需求变化也不大，则最适宜采用（  ）开发方法。

（15）A．结构化   
B.Jackson   
C.原型化   
D.面向对象   
  
（16）A．结构化   
B.Jackson   
C.原型化   
D.面向对象

**试题分析**

在本题的两个空中，第1个空比较容易，由于题目明确说明“用户需求不清晰且经常发生变化”所以只有原型化方法适用。而第2空需求变化不大的情况下，其实多种模型都可用，所以另一条线索成为解题关键，即“数据处理领域问题”。结构化方法的基本特征是：自顶向下，逐层分解，也适合于大型的数据处理系统，所以用他最合适。

**试题答案**

（15）C（16）A

# **试题31(2015年下半年试题29)**

某开发小组欲开发一个超大规模软件：使用通信卫星，在订阅者中提供、监视和控制移动电话通信，则最不适宜采用（  ）过程模型。

（29）A．瀑布   
B.原型   
C.螺旋   
D.喷泉

**试题分析**

需要开发的是大型软件系统，大型软件系统的需求往往难以在前期确定，所以瀑布模型最不适合。

**试题答案**

（29）A

# **试题32(2015年下半年试题30)**

（  ）开发过程模型以用户需求为动力，以对象为驱动，适合于面向对象的开发方法。

（30）A．瀑布   
B.原型   
C.螺旋   
D.喷泉

**试题分析**

备选答案中的几种开发模型中，只有喷泉模型属于面向对象开发模型。

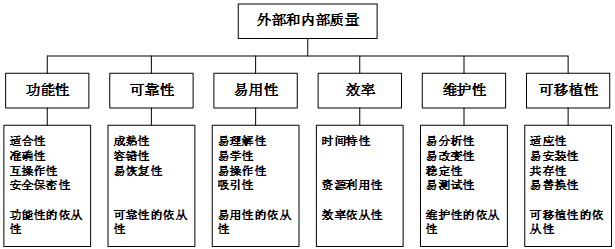
**试题答案**

（30）D

# **试题33(2015年下半年试题31)**

在ISO/IEC软件质量模型中，易使用性的子特性不包括（  ）。  
（31）A．易理解性   
B.易学性   
C.易操作性   
D.易分析性

**试题分析**



**试题答案**

（31）D

# **试题34(2015年下半年试题32)**

在进行子系统结构设计时，需要确定划分后的子系统模块结构，并画出模块结构图。该过程不需要考虑（  ）。

（32）A．每个子系统如何划分成多个模块   
B.每个子系统采用何种数据结构和核心算法   
C.如何确定子系统之间、模块之间传送的数据及其调用关系   
D.如何评价并改进模块结构的质量

**试题分析**

系统模块结构设计的任务是确定划分后的子系统的模块结构，并画出模块结构图，这个过程中必须考虑这样几个问题：每个子系统如何划分成若干个模块；如何确定子系统之间、模块之间传送的数据及其调用关系；如何评价并改进模块结构的质量；如何从数据流图导出模块结构图。

**试题答案**

（32）B

# **试题35(2015年下半年试题33)**

数据流图中某个加工的一组动作依赖于多个逻辑条件的取值，则用（  ）能够清楚地表示复杂的条件组合与应做的动作之间的对应关系。

（33）A．流程图   
B.NS盒图   
C.形式语言   
D.决策树

**试题分析**

NS盒图是用于取代传统流程图的一种描述方式。NS盒图形象直观，具有良好的可见度。例如循环的范围、条件语句的范围都是一目了然的，所以容易理解设计意图，为编程、复查、选择测试用例、维护都带来了方便。它能清楚的表达复杂条件组合与应做的动作之间的对应关系。

**试题答案**

（33）B

# **试题36(2015年下半年试题34)**

根据软件过程活动对软件工具进行分类，则逆向工程工具属于（  ）工具。  
（34）A．软件开发   
B.软件维护   
C.软件管理   
D.软件支持

**试题分析**

逆向工程是在软件维护时，由于缺少文档资料，而对软件的一种分析。

**试题答案**

（34）B

# **试题37(2015年下半年试题35-36)**

若用白盒测试方法测试以下代码，并满足条件覆盖，则至少需要（  ）个测试用例。采用McCabe度量法算出该程序的环路复杂性为（  ）。

Int find \_max（int i,int j,int k）{

    int max;

    if(i>j)then

                 if（i>k）then max =i；

                 else max=k；

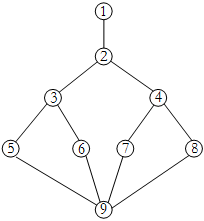
         else if（j>k）then max =j；

                  else max=k；

    }

（35）A．3   
B.4   
C.5   
D.6   
  
（36）A．1   
B.2   
C.3   
D.4

**试题分析**

要计算McCabe复杂度需要先绘制出图：  


**试题答案**

（35）B（36）D

# **试题38(2015年上半年试题15)**

软件工程的基本要素包括方法、工具和（  ）。  
（15）A．软件系统   
B.硬件系统   
C.过程   
D.人员

**试题分析**

本题考查软件工程的基本概念。

软件工程是一种层次化的技术，从底向上分别为质量、过程、方法和工具。任何工程方法必须以有组织的质量承诺为基础。软件工程的基础是过程，过程是将技术结合在一起的凝聚力，使得计算机软件能够被合理地和及时地开发，过程定义了一组关键过程区域，构成了软件项目管理控制的基础；方法提供了建造软件在技术上需要“如何做”， 它覆盖了一系列的任务。方法也依赖于一些基本原则，这些原则控制了每一个技术区域 而且包含建模活动和其他描述技术；工具对过程和方法提供了自动或半自动的支持，如：计算机辅助软件工程（CASE）。软件工程的基本要素包括方法、工具和过程。

**试题答案**

（15）C

# **试题39(2015年上半年试题16)**

在（  ）设计阶段选择适当的解决方案，将系统分解为若干个子系统，建立整个系统的体系结构。  
（16）A．概要  
B.详细  
C.结构化  
D.面向对象

**试题分析**

本题考查软件工程的基本概念。  
软件设计的任务是基于需求分析的结果建立各种设计模型，给出问题的解决方案。从工程管理的角度，可以将软件设计分为两个阶段：概要设计阶段和详细设计阶段。结构化设计方法中，概要设计阶段进行软件体系结构的设计、数据设计和接口设计；详细设计阶段进行数据结构和算法的设计。面对对象设计方法中，概要设计阶段进行体系结构设计、初步的类设计/数据设计、结构设计；详细设计阶段进行构件设计。  
结构化设计和面向对象设计是两种不通过的设计方法，结构化设计根据系统的数据流图进行设计，模块体现为函数、过程及子程序；面向对象设计基于面向对象的基本概念进行，模块体现为类、对象和构件等。

**试题答案**

（16）A

# **试题40(2015年上半年试题29)**

某公司计划开发一种产品，技术含量很高，与客户相关的风险也很多，则最适于采用（  ）开发过程模型。  
（29）A．瀑布   
B.原型   
C.增量   
D.螺旋

**试题分析**

这些模型中仅有螺旋模型考虑风险因素。

**试题答案**

（29）D

# **试题41(2015年上半年试题30)**

在敏捷过程的方法中（  ）认为每一个不同的项目都需要一套不同的策略、约定和方法论。  
（30）A．极限编程（XP）   
B.水晶法(Crystal)   
C.并列争球法(Scrum)   
D.自适应软件开发(ASD)

**试题分析**

水晶方法，Crystal ，是由 Alistair Cockburn 和 Jim Highsmith 建立的敏捷方法系列，其目的是发展一种提倡“机动性的”[1] 方法，包含具有共性的核心元素，每个都含有独特的角色、过程模式、工作产品和实践。Crystal 家族实际上是一组经过证明、对不同类型项目非常有效的敏捷过程，它的发明使得敏捷团队可以根据其项目和环境选择最合适的 Crystal 家族成员。

极限编程（Extreme Programming，XP）是一门针对业务和软件开发的规则，它的作用在于将两者的力量集中在共同的、可以达到的目标上。它是以符合客户需要的软件为目标而产生的一种方法论，XP使开发者能够更有效的响应客户的需求变化，哪怕是在软件生命周期的后期。它强调，软件开发是人与人合作进行的过程，因此成功的软件开发过程应该充分利用人的优势，而弱化人的缺点，突出了人在软件开发过程中的作用。极端编程属于轻量级的方法，认为文档、架构不如直接编程来的直接。

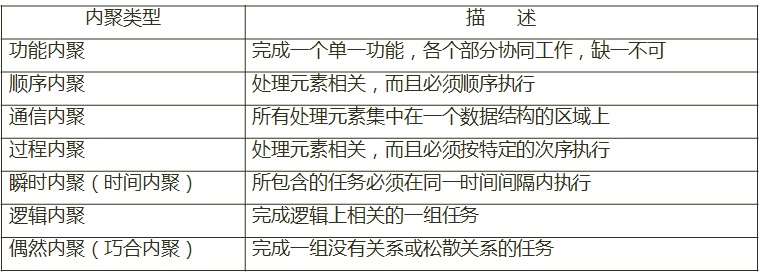
**试题答案**

（30）B

# **试题42(2015年上半年试题32)**

某模块实现两个功能：向某个数据结构区域写数据和从该区域读数据。该模块的内聚类型为（  ）内聚。  
（32）A．过程   
B.时间   
C.逻辑   
D.通信

**试题分析**



**试题答案**

（32）D

# **试题43(2015年上半年试题34)**

自底向上的集成测试策略的优点包括（  ）。  
（34）A．主要的设计问题可以在测试早期处理   
B.不需要写驱动程序   
C.不需要写桩程序   
D.不需要进行回归测试

**试题分析**

1、自顶向下集成

优点：较早地验证了主要控制和判断点；按深度优先可以首先实现和验证一个完整的软件功能；功能较早证实，带来信心；只需一个驱动，减少驱动器开发的费用；支持故障隔离。

缺点：柱的开发量大；底层验证被推迟；底层组件测试不充分。

适应于产品控制结构比较清晰和稳定；高层接口变化较小；底层接口未定义或经常可能被修改；产口控制组件具有较大的技术风险，需要尽早被验证；希望尽早能看到产品的系统功能行为。

2、自底向上集成

优点：对底层组件行为较早验证；工作最初可以并行集成，比自顶向下效率高；减少了桩的工作量；支持故障隔离。

缺点：驱动的开发工作量大；对高层的验证被推迟，设计上的错误不能被及时发现。

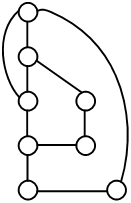
适应于底层接口比较稳定；高层接口变化比较频繁；底层组件较早被完成。

**试题答案**

（34）C

# **试题44(2015年上半年试题35)**

采用McCabe度量法计算下列程序图的环路复杂性为（  ）。



（35）A．2   
B.3   
C.4   
D.5

**试题分析**

点数：8，边数：10。

10-8+2=4。

**试题答案**

（35）C

# **试题45(2015年上半年试题36)**

以下关于软件可维护性的叙述中，不正确的是“可维护性（  ）”。  
（36）A．是衡量软件质量的一个重要特性  
B.不受软件开发文档的影响  
C.是软件开发阶段各个时期的关键目标  
D.可以从可理解性、可靠性、可测试性、可行性、可移植性等方面进行度量

**试题分析**

本题考查维护方面的基础知识。  
软件交付给用户使用后到软件报废之前都属于软件维护阶段。软件系统的可维护性可以定义为：维护人员理解、改正、改动和改进该软件的难易程度。提供软件可维护性是开发软件系统所有步骤的关键目的，是衡量软件质量的一种重要特性，可以从可理解性、可靠性、可测试性、可行性、可移植性等方面进行度量。良好的软件开发文档可以有效地提高软件的可维护性。

**试题答案**

（36）B

# **试题46(2014年下半年试题15)**

以下关于结构化开发方法的叙述中，不正确的是（  ）。  
（15）A．总的指导思想是自顶向下，逐层分解  
B.基本原则是功能的分解与抽象  
C.与面向对象开发方法相比，更适合于大规模、特别复杂的项目  
D.特别适合于数据处理领域的项目

**试题分析**

结构化开发方法是一种面向数据流的开发方法，其基本思想是软件功能的分解和抽象。结构化开发方法又称生命周期法，是迄今为止最传统、应用最广泛的一种信息系统开发方法。结构化开发方法采用系统工程的思想和工程化的方法，按用户至上的原则，结构化、模块化、自顶向下地对信息系统进行分析与设计。该方法严格按照信息系统开发的阶段性开展设计工作，每个阶段都产生一定的设计成果，通过评估后再进入下一阶段开发工作。因此，结构化开发方法具有以下优点：

• 开发工作的顺序性、阶段性适合初学者参与软件的开发

• 开发工作的阶段性评估可以减少开发工作重复性和提高开发的成功率

• 该方法有利于提高系统开发的正确性、可靠性和可维护性

• 具有完整的开发质量保证措施

结构化开发方法存在的不足主要是开发周期太长，个性化开发阶段的文档编写工作量过大或过于烦琐,无法发挥开发人员的个性化开发 能力。一般来说，结构化开发方法主要适用于组织规模较大、组织结构相对稳定的企业，这些大型企业往往业务处理过程规范、信息系统数据需求非常明确，在一定时期内需求变化不大。

**试题答案**

（15）C

# **试题47(2014年下半年试题16)**

模块A、B和C都包含相同的5个语句，这些语句之间没有联系。为了避免重复把这5个语句抽取出来组成一个模块D，则模块D的内聚类型为（  ）内聚。  
（16）A．功能  
B.通信  
C.逻辑  
D.巧合

**试题分析**

功能内聚：完成一个单一功能，各个部分协同工作，缺一不可。

顺序内聚：处理元素相关，而且必须顺序执行。

通信内聚：所有处理元素集中在一个数据结构的区域上。

过程内聚：处理元素相关，而且必须按特定的次序执行。

瞬时内聚：所包含的任务必须在同一时间间隔内执行（如初始化模块）。

逻辑内聚：完成逻辑上相关的一组任务。

偶然内聚：完成一组没有关系或松散关系的任务。

**试题答案**

（16）D

# **试题48(2014年下半年试题29)**

以下关于增量模型的叙述中，正确的是（  ）。  
（29）A．需求被清晰定义  
B.可以快速构造核心产品  
C.每个增量必须要进行风险评估  
D.不适宜商业产品的开发

**试题分析**

增量模型融合了瀑布模型的基本成分（重复应用）和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时，第1个增量往往是核心的产品，即第1个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。由于能够在较短的时间内向用户提交一些有用的工作产品，因此能够解决用户的一些急用功能。由于每次只提交用户部分功能，用户有较充分的时间学习和适应新的产品。

对系统的可维护性是一个极大的提高，因为整个系统是由一个个构件集成在一起的，当需求变更时只变更部分部件，而不必影响整个系统。

增量模型存在以下缺陷：

1） 由于各个构件是逐渐并入已有的软件体系结构中的，所以加入构件必须不破坏已构造好的系统部分，这需要软件具备开放式的体系结构。

2） 在开发过程中，需求的变化是不可避免的。增量模型的灵活性可以使其适应这种变化的能力大大优于瀑布模型和快速原型模型，但也很容易退化为边做边改模型，从而使软件过程的控制失去整体性。

3）如果增量包之间存在相交的情况且未很好处理，则必须做全盘系统分析，这种模型将功能细化后分别开发的方法较适应于需求经常改变的软件开发过程。

**试题答案**

（29）B

# **试题49(2014年下半年试题30)**

以下关于CMM的叙述中，不正确的是（  ）。  
（30）A．CMM是指软件过程能力成熟度模型  
B.CMM棍据软件过程的不同成熟度划分了5个等级，其中，1级被认为成熟度最高，5级被认为成熟度最低  
C.CMMI的任务是将已有的几个CMM模型结合在一起，使之构造成为“集成模型”  
D.采用更成熟的CMM模型，一般来说可以提高最终产品的质量

**试题分析**

CMM（Capability Maturity Model）是能力成熟度模型的缩写，CMM是国际公认的对软件公司进行成熟度等级认证的重要标准。CMM共分五级。在每一级中，定义了达到该级过程管理水平所应解决的关键问题和关键过程。每一较低级别是达到较高级别的基础。其中五级是最高级，即优化级，达到该级的软件公司过程可自发地不断改进，防止同类问题二次出现；四级称为已管理级，达到该级的软件公司已实现过程的定量化；三级为已定义级，即过程实现标准化；二级为可重复级，达到该级的软件公司过程已制度化，有纪律，可重复；一级为初始级，过程无序，进度、预算、功能和质量等方面不可预测。

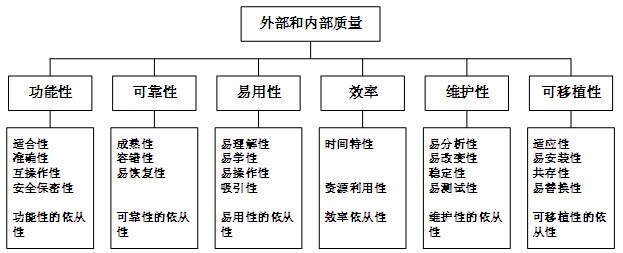
**试题答案**

（30）B

# **试题50(2014年下半年试题31)**

在ISO/IEC软件质量模型中，可靠性是指在规定的二段时间内和规定的条件下，软件维持在其性能水平的能力；其子特性不包括（  ）。  
（31）A．成熟性  
B.容错性  
C.易恢复  
D.可移植性

**试题分析**



**试题答案**

（31）D

# **试题51(2014年下半年试题32)**

在软件开发过程中，系统测试阶段的测试目标来自于（  ）阶段。  
（32）A．需求分析  
B.概要设计  
C.详细设计  
D.软件实现

**试题分析**

测试的目标是需求分析开始。从需求开始时就已经确定了产品的功能，在那些的用户场景、用户使用的功能。

**试题答案**

（32）A

# **试题52(2014年下半年试题33)**

以下关于文档的叙述中，不正确的是（  ）。  
（33）A．项目相关人员可以通过文档进行沟通  
B.编写文档会降低软件开发的效率  
C.编写高质量文档可以提高软件开发的质量  
D.文档是软件的不可或缺的部分

**试题分析**

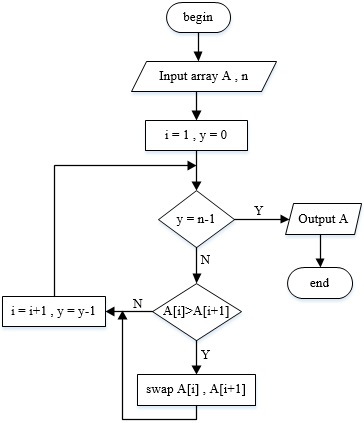
编写文档有助于提高软件开发的效率而不是降低。

**试题答案**

（33）B

# **试题53(2014年下半年试题34-35)**

下图所示的程序流程图中有（  ）条不同的简单路径。采用McCabe度量法计算该程序图的环路复杂性为（  ）。



（34）A．3  
B.4  
C.5  
D.6  
  
（35）A．3  
B.4  
C.5  
D.6

**试题分析**

环形复杂度定量度量程序的逻辑复杂度。描绘程序控制流的流图之后，可以用下述3种方法中的任何一种来计算环形复杂度。

（1）流图中的区域数等于环形复杂度。

（2）流图G的环形复杂度V(G)=E-N+2，其中，E是流图中边的条数，N是结点数。

（3）流图G的环形复杂度V(G)=P+1，其中，P是流图中判定结点的数目。

这种环路度量法，计算的思路是这样的：它是考虑控制的复杂程度，即条件选择的分支繁杂程度。

图中有3次简单的判断。故3条简单路径，形成3块环形区域，区域复杂度为3。

**试题答案**

（34）A（35）A

# **试题54(2014年下半年试题36)**

以下关于软件维护和可维护性的叙述中，不正确的是（  ）。  
（36）A．软件维护要解决软件产品交付用户之后运行中发生的各种问题  
B.软件的维护期通常比开发期长得多，其投入也大得多  
C.进行质量保证审查可以提高软件产品的可维护性  
D.提高可维护性是在软件维护阶段考虑的问题

**试题分析**

在软件开发的各个阶段都需考虑软件的可维护性。

**试题答案**

（36）D

# **试题55(2014年上半年试题15)**

以下关于结构化开发方法的叙述中，不正确的是（  ）。

（15）A．将数据流映射为软件系统的模块结构  
B.一般情况下，数据流类型包括变换流型和事务流型  
C.不同类型的数据流有不同的映射方法  
D.一个软件系统只有一种数据流类型

**试题分析**

面向数据流的设计是以需求分析阶段产生的数据流图为基础，按一定的步骤映射成软件结构，因此又称结构化设计。该方法由美国IBM公司L.Constantine和E.Yourdon等人于1974年提出，与结构化分析(SA)衔接，构成了完整的结构化分析与设计技术，是目前使用最广泛的软件设计方法之一。

各种软件系统，不论DFD如何庞大和复杂，一般可分为变换型和事务型，一个软件系统既可以只有一种数据流类型，也可以是两种数据流烈性。在结构化设计中，可以将数据流映射为软件系统的模块结构，不同类型的数据流有不同的映射方法。

**试题答案**

（15）D

# **试题56(2014年上半年试题16)**

模块A提供某个班级某门课程的成绩给模块B，模块B计算平均成绩、最高分和最低分，将计算结果返回给模块A，则模块B在软件结构图中属于（  ）模块。

（16）A．传入  
B.传出  
C.变换  
D.协调

**试题分析**

传入模块：从下属模块取得数据，经处理再将其传送给上级模块。      
传出模块：从上级模块取得数据，经处理再将其传送给下属模块。

变换模块：从上级模块取得数据，进行特定的处理，转换成其他形式，再传送给上级模块。

**试题答案**

（16）C

# **试题57(2014年上半年试题29)**

以下关于统一过程UP的叙述中，不正确的是（  ）。

（29）A．UP是以用例和风险为驱动，以架构为中心，迭代并且增量的开发过程   
B.UP定义了四个阶段，即起始、精化、构建和确认阶段   
C.每次迭代都包含计划、分析、设计、构造、集成、测试以及内部和外部发布   
D.每个迭代有五个核心工作流

**试题分析**

什么是 Rational 统一过程（ Rational Unified Process）?  
Rational Unified Process 是软件工程的过程。它提供了在开发组织中分派任务和责任的纪律化方法。它的目标是在可预见的日程和预算前提下，确保满足最终用户需求的高质量产品。  
统一过程模型是一种“用例驱动，以体系结构为核心，迭代及增量”的软件过程框架，由UML方法和工具支持。

RUP把一个项目分为四个不同的阶段：  
构思阶段 ：包括用户沟通和计划活动两个方面，强调定义和细化用例，并将其作为主要模型。  
细化阶段 ：包括用户沟通和建模活动，重点是创建分析和设计模型，强调类的定义和体系结构的表示。  
构建阶段 ：将设计转化为实现，并进行集成和测试。  
移交阶段 ：将产品发布给用户进行测试评价，并收集用户的意见，之后再次进行迭代修改产品使之完善。

**试题答案**

（29）B

# **试题58(2014年上半年试题30)**

某公司要开发一个软件产品，产品的某些需求是明确的，而某些需求则需要进一步细化。由于市场竞争的压力，产品需要尽快上市，则开发该软件产品最不适合采用（  ）模型。

（30）A．瀑布   
B.原型   
C.增量   
D.螺旋

**试题分析**

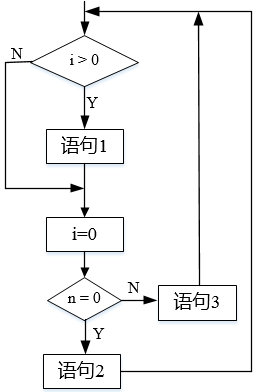
瀑布模型（Waterfall Model） 是一个项目开发架构，开发过程是通过设计一系列阶段顺序展开的，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈，因此，如果有信息未被覆盖或者发现了问题，那么最好 “返回”上一个阶段并进行适当的修改，项目开发进程从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。包括软件工程开发、企业项目开发、产品生产以及市场销售等构造瀑布模型。

**试题答案**

（30）A

# **试题59(2014年上半年试题32)**

采用McCabe度量法计筧下列程序图的环路复杂性为（  ）。



（32）A．2   
B.3   
C.4   
D.5

**试题分析**

McCabe度量法是通过定义环路复杂度，建立程序复杂性的度量，它基于一个程序模块的程序图中环路的个数。计算有向图 G 的环路复杂性的公式为：VG=m-n+2，其中 VG是有向图 G 中的环路个数，m是 G 中的有向弧数，n 是 G 中的节点数。  
图中m为8，n为6，则m-n+2=4。

**试题答案**

（32）C

# **试题60(2014年上半年试题33)**

以下关于文档的叙述中，不正确的是（  ）。

（33）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.文档也是软件产品的一部分，没有文档的软件就不能称之为软件   
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

**试题分析**

软件文档的作用   
在软件生产过程中，总是产生和使用大量的信息。软件文档在产品的开发过程中起着重要的作用。 提高软件开发过程的能见度。把开发过程中发生的事件以某种可阅读的形式记录在文档中。   
管理人员可把这些记载下来的材料作为检查软件开发进度和开发质量的依据，实现对软件开发的工程管理。 ? 提高开发效率。软件文档的编制，使得开发人员对各个阶段的工作都进行周密思考、全盘权衡、减少返工。并且可在开发早期发现错误和不一致性，便于及时加以纠正。 ? 作为开发人员在一定阶段的工作成果和结束标志。 ? 记录开发过程中有关信息，便于协调以后的软件开发、使用和维护。   
提供对软件的运行、维护和培训的有关信息，便于管理人员、开发人员、操作人员、用户之间的协作、交流和了解。使软件开发活动更科学、更有成效。   
便于潜在用户了解软件的功能、性能等各项指标，为他们选购符合自己需要的软件提供依据。   
从某种意义上来说，文档是软件开发规范的体现和指南。 ? 按规范要求生成一整套文档的过程，就是按照软件开发规范完成一个软件开发的过程。   
 所以，在使用工程化的原理和方法来指导软件的开发和维护时，应当充分注意软件文档的编制和管理。

**试题答案**

（33）A

# **试题61(2014年上半年试题34)**

某搜索引擎在使用过程中，若要增加接受语音输入的功能，使得用户可以通过 语音输入来进行搜索，此时应对系统进行（  ）维护。

（34）A．正确性   
B.适应性   
C.完善性   
D.预防性

**试题分析**

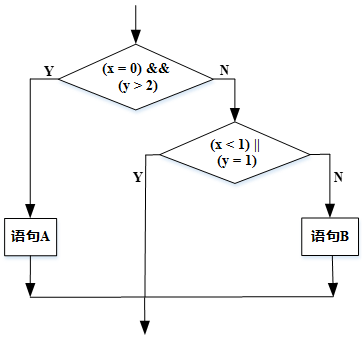
在系统运行过程中，软件需要维护的原因是多样的，根据维护的原因不同，可以将软件维护分为以下四种：  
（1）改正性维护。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程就称为改正性维护。  
（2）适应性维护。在使用过程中，外部环境（新的硬、软件配置）、数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化。为使软件适应这种变化，而去修改软件的过程就称为适应性维护。  
（3）完善性维护。在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动称为完善性维护。  
（4）预防性维护。这是指预先提高软件的可维护性、可靠性等，为以后进一步改进软件打下良好基础。通常，预防性维护可定义为“把今天的方法学用于昨天的系统以满足明天的需要”。也就是说，采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编码和测试。

**试题答案**

（34）C

# **试题62(2014年上半年试题35-36)**

采用白盒测试方法对下图进行测试，设计了4个测试用例：①（x=0,y=3）， ②(x=1,y=2)， ③(x=-1,y=2)，④(x=3,y=1)。至少需要测试用例①②才能完成（  ）覆盖，至少需要测试用例①②③或①②④才能完成（  ）覆盖。



（35）A．语句   
B.条件   
C.判定\条件   
D.路径   
  
（36）A．语句   
B.条件   
C.判定\条件   
D.路径

**试题分析**

语句覆盖（StatementCoverage） 又称行覆盖（LineCoverage），段覆盖（SegmentCoverage），基本块覆盖（BasicBlockCoverage），这是最常用也是最常见的一种覆盖方式，就是度量被测代码中每个可执行语句是否被执行到了，只管覆盖代码中的执行语句，却不考虑各种分支的组合等等。  
路径覆盖的含义是，选取足够多的测试数据，使程序的每条可能路径都至少执行一次（如果程序图中有环，则要求每个环至少经过一次）。

**试题答案**

（35）A（36）D

# **试题63(2013年下半年试题15)**

以下关于数据流图中基本加工的叙述，不正确的是（  ）。

（15）A．对每一个基本加工，必须有一个加工规格说明   
B.加工规格说明必须描述把输入数据流变换为输出数据流的加工规则   
C.加工规格说明必须描述实现加工的具体流程   
D.决策表可以用来表示加工规格说明

**试题分析**

对基本加工的说明有三种描述方式：结构化语言、判断表（决策表）、判断树（决策树）。基本加工逻辑描述的基本原则为：

1、对数据流图的每一个基本加工，必须有一个基本加工逻辑说明。

2、基本加工逻辑说明必须描述基本加工如何把输入数据流变换为输出数据流的加工规则。

3、加工逻辑说明必须描述实现加工的策略而不是实现加工的细节。

4、加工逻辑说明中包含的信息应是充足的，完备的，有用的，无冗余的。

**试题答案**

（15）C

# **试题64(2013年下半年试题16)**

在划分模块时，一个模块的作用范围应该在其控制范围之内。若发现其作用范围不在其控制范围内，则（  ）不是适当的处理方法。

（16）A．将判定所在模块合并到父模块中，使判定处于较高层次   
B.将受判定影响的模块下移到控制范围内   
C.将判定上移到层次较高的位置   
D.将父模块下移，使判定处于较高层次

**试题分析**

一个模块的作用范围（或称影响范围）指受该模块内一个判定影响的所有模块的集合。

一个模块的控制范围指模块本身以及其所有下属模块（直接或间接从属于它的模块）的集合。

一个模块的作用范围应在其控制范围之内，且判定所在的模块应在其影响的模块在层次上尽量靠近。如果再设计过程中，发现模块作用范围不在其控制范围之内，可以用“上移判点”或“下移受判断影响的模块，将它下移到判断所在模块的控制范围内”的方法加以改进。

**试题答案**

（16）D

# **试题65(2013年下半年试题29)**

（  ）开发过程模型最不适用开发初期对软件需求缺乏准确全面认识的情况。  
（29）A．瀑布  
B.演化  
C.螺旋  
D.增量

**试题分析**

瀑布模型是一种经典的开发模型，开发过程是通过设计一系列阶段顺序展开的，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈，因此，如果有信息未被覆盖或者发现了问题，那么最好 “返回”上一个阶段并进行适当的修改，项目开发进程从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。

瀑布模型的突出缺点是不适应用户需求的变化。

**试题答案**

（29）A

# **试题66(2013年下半年试题30)**

（  ）不是增量式开发的优势。  
（30）A．软件可以快速地交付  
B.早期的增量作为原型，从而可以加强对系统后续开发需求的理解  
C.具有最高优先级的功能首先交付，随着后续的增量不断加入，这就使得更重要的功能得到更多的测试  
D.很容易将客户需求划分为多个增量

**试题分析**

增量式开发的主要优点包括：

1、由于能够在较短的时间内向用户提交一些有用的工作产品，因此能够解决用户的一些急用功能。

2、由于每次只提交用户部分功能，用户有较充分的时间学习和适应新的产品。

3、对系统的可维护性是一个极大的提高，因为整个系统是由一个个构件集成在一起的，当需求变更时只变更部分部件，而不必影响整个系统。

主要缺点包括：

1、由于各个构件是逐渐并入已有的软件体系结构中的，所以加入构件必须不破坏已构造好的系统部分，这需要软件具备开放式的体系结构。

2、在开发过程中，需求的变化是不可避免的。增量模型的灵活性可以使其适应这种变化的能力大大优于瀑布模型和快速原型模型，但也很容易退化为边做边改模型，从而使软件过程的控制失去整体性。

3、如果增量包之间存在相交的情况且未很好处理，则必须做全盘系统分析，这种模型将功能细化后分别开发的方法较适应于需求经常改变的软件开发过程。

**试题答案**

（30）D

# **试题67(2013年下半年试题31)**

在对程序质量进行评审时，模块结构是一个重要的评审项，评审内容中不包括（  ）。  
（31）A．数据结构  
B.数据流结构  
C.控制流结构  
D.模块结构与功能结构之间的对应关系

**试题分析**

模块结构评审时，主要包括以下方面的评审：

1、控制流结构：规定了处理模块与处理模块之间的流程关系。检查处理模块之间的控制转移关系与控制转移形式（调用方式）。

2、数据流结构：规定了数据模块是如何被处理模块进行加工的流程关系。检查处理模块与数据模块之间的对应关系；处理模块与数据模块之间的存取关系，如建立、删除、查询、修改等。

3、模块结构与功能结构之间的对应关系：包括功能结构与控制流结构的对应关系；功能结构与数据流结构的对应关系；每个模块的定义（包括功能、输入与输出数据）。

**试题答案**

（31）A

# **试题68(2013年下半年试题32)**

SEI能力成熟度模型（SEICMM）把软件开发企业分为5个成熟度级别，其中（  ）重点关注产品和过程质量。  
（32）A．级别2：重复级  
B.级别3：确定级  
C.级别4：管理级  
D.级别5：优化级

**试题分析**

CMM的分级结构及其过程描述：

（1）初始级：软件过程的特点是无秩序或说无定规的，有时甚至是混乱的。软件过程定义几乎处于无章法、无步骤可循的状态，软件产品所取得的成功往往依赖于极个别人的努力和机遇。

（2）可重复级：已建立了基本的项目管理过程，可用于对成本、进度和功能特性进行跟踪。对类似的应用项目，有章可循并能重复以往所取得的成功。

（3）已定义级：用于管理的和工程的软件过程均已文档化、标准化，并形成了整个软件组织的标准软件过程。全部项目均采用与实际情况相吻合的、适当修改后的标准软件过程来进行操作。

（4）已管理级：软件过程和产品质量有详细的度量标准。软件过程和产品质量得到了定量的认识和控制。

（5）优化级：通过对来自过程、新概念和新技术等方面的各种有用信息的定量分析，能够不断地、持续地对促进过程进行改进。

除第一级外，每一级都设定了一组目标，如果达到了这组目标，则表明达到了这个成熟级别，自然可以向上一更为成熟的高一级别迈进。CMM体系不主张跨级别的进化，因为从第二级开始，每一个低级别的实现均是更高级别实现的基础。

**试题答案**

（32）C

# **试题69(2013年下半年试题33)**

系统可维护性的评价指标不包括（  ）。  
（33）A．可理解性  
B.可测试性  
C.可移植性  
D.可修改性

**试题分析**

系统的可维护性的评价指标包括：可理解性、可测试性、可修改性。

可理解性：指别人能理解系统的结构、界面、功能和内部过程的难易程度。模块化、详细设计文档、结构化设计和良好的高级程序设计语言等，都有助于提高可理解性。

可测试性：诊断和测试的容易程度取决于易理解的程度。

可修改性：诊断和测试的容易程度与系统设计所制定的设计原则有直接关系。模块的耦合、内聚、作用范围与控制范围的关系等，都对可修改性有影响。

**试题答案**

（33）C

# **试题70(2013年下半年试题34)**

 逆向工程从源代码或目标代码中提取设计信息，通常在原软件生命周期的（  ）阶段进行。

（34）A．需求分析   
B.软件设计   
C.软件测试   
D.软件维护

**试题分析**

逆向工程产品设计可以认为是一个从产品到设计的过程。简单地说，逆向工程产品设计就是根据已经存在的产品，反向推出产品设计数据（包括各类设计图或数据模型）的过程。这个过程一般是在软件交付使用之后进行，所以是在原软件生命周期的软件维护阶段进行。

**试题答案**

（34）D

# **试题71(2013年下半年试题35)**

一个程序根据输入的年份和月份计算该年中该月的天数，输入参数包括年份（正整数）、月份（用1~12表示）。若用等价类划分测试方法进行测试，则（  ）不是一个合适的测试用例（分号后表示测试的输出）。

（35）A．（2013，1，31）   
B.（0，1，‘错误’）   
C.（0，13，‘错误’）   
D.（2001，-1，‘错误’）

**试题分析**

测试用例编写一般原则：

1、设计一个新的测试用例，使其尽可能多地覆盖尚未被覆盖的有效等价类，重复这一步，直到所有的有效等价类都被覆盖为止；

2、设计一个新的测试用例，使其仅覆盖一个尚未被覆盖的无效等价类，重复这一步，直到所有的无效等价类都被覆盖为止。

在本题中，C选项同时覆盖了两个无效等价类，所以不符合测试用例编写的一般原则。

**试题答案**

（35）C

# **试题72(2013年下半年试题36)**

（  ）不是单元测试主要检查的内容。

（36）A．模块接口   
B.局部数据结构   
C.全局数据结构   
D.重要的执行路径

**试题分析**

在单元测试开始时，应该对通过所有被测模块的数据流进行测试。

①输入参数是否与形式参数是否一致。

②调用模块的实参与被调用模块形参是否一致。

③全局变量在各模块中的定义和用法是否一致

④外部输入、输出等等。

**试题答案**

（36）C

# **试题73(2013年上半年试题15)**

统一过程模型是一种“用例和风险驱动以架构为中心迭代并且增量”的开发过程，定义了不同阶段及其制品其中精化阶段关注（  ）。

（15）A．项目的初始活动   
B.需求分析和架构演进   
C.系统的构建，产生实现模型   
D.软件提交方面的工作，产生软件增量

**试题分析**

   精化阶段的目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。

**试题答案**

（15）B

# **试题74(2013年上半年试题18)**

“软件产品必须能够在3秒内对用户请求作出响应”属于软件需求中的（  ）。

（18）A．功能需求   
B.非功能需求   
C.设计约束   
D.逻辑需求

**试题分析**

软件需求中对软件产品的相应时间，吞吐量，价格等属性的要求都属于非功能性需求。

**试题答案**

（18）B

# **试题75(2013年上半年试题19)**

某项目为了修正一个错误而进行了修改。错误修正后，还需要进行（  ）以发现这一修正是否引起原本正确运行的代码出错。

（19）A．单元测试   
B.接受测试   
C.安装测试   
D.回归测试

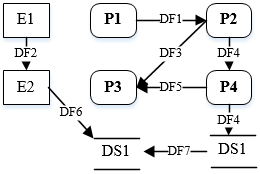
**试题分析**

 回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

**试题答案**

（19）D

# **试题76(2013年上半年试题29)**

l  在如下所示的数据流图中，共存在（  ）个错误  


（29）A．4   
B.6   
C.8   
D.9

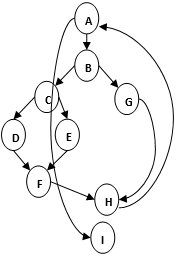
**试题分析**

数据流图的设计原则：  
①  数据守恒原则  
②  守恒加工原则  
③  对于每个加工，必须既有输入数据流，又有输出数据流  
④  外部实体与外部实体之间不存在数据流  
⑤  外部实体与外部存储之间不存在数据流  
⑥  数据存储与数据存储之间不存在数据流  
⑦  父图与子图的平衡原则  
⑧  数据流与加工有关，且必须经过加工  
DF2违背④，DF6违背⑤，DF7违背⑥，P1和P3缺少数据流违背③，P4违背②，因此共有六个错误。

**试题答案**

（29）B

# **试题77(2013年上半年试题30-31)**

软件的复杂性主要体现在程序的复杂性。（  ）是度量软件复杂性的一个主要参数。若采用McCabe度量法计算环路复杂性，则对于下图所示的程序图，其环路复杂度为（  ）。  


（30）A．代码行数   
B.常量的数量   
C.变量的数量   
D.调用的库函数的数量   
  
（31）A．2   
B.3   
C.4   
D.5

**试题分析**

代码行数度量法以程序的总代码行数作为程序复杂性的度量值。  
McCabe度量法先画出程序图，然后采用公式V(G)=m-n+2计算环路复杂度。所以环路复杂度为11-9+2=4。

**试题答案**

（30）A（31）C

# **试题78(2013年上半年试题32)**

（  ）不属于软件设计质量评审。

（32）A．功能与模块之间的对应关系   
B.软件规格说明是否符合用户的要求   
C.软件是否具有可测试性   
D.软件是否具有良好的可靠性

**试题分析**

 (1)评价软件的规格说明是否合乎用户的要求，即[总体设计](http://baike.baidu.com/view/228947.htm" \t "http://www.educity.cn/jiaocheng/_blank)思想和设计方针是否正确，需求规格说明是否得到了用户或单位上级机关的批准；需求规格说明与软件的[概要设计](http://baike.baidu.com/view/551728.htm" \t "http://www.educity.cn/jiaocheng/_blank)规格说明是否一致等。  
(2)评审可靠性，即是否能避免输入异常(错误或超载等)、硬件失效及[软件失效](http://baike.baidu.com/view/3740235.htm" \t "http://www.educity.cn/jiaocheng/_blank)所产生的失效，一旦发生应能及时采取代替手段或恢复手段。  
(3)评审保密措施实现情况，即是否提供对使用系统资格、对特定数据的使用资格及特殊功能的使用资格进行检查，在查出有违反使用资格情况后，能否向系统管理人员报告有关信息，是否提供对系统内重要数据加密的功能。  
(4)评审操作特性实施情况，即操作命令和操作信息的恰当性，输入数据与输入[控制语句](http://baike.baidu.com/view/1359886.htm" \t "http://www.educity.cn/jiaocheng/_blank)的恰当性，输出数据的恰当性，应答时间的恰当性等。  
(5)评审性能实现情况。  
(6)评审软件是否具有可修改性、可扩充性、可互换性和可移植性。  
(7)评审软件是否具有[可测试性](http://baike.baidu.com/view/2194552.htm" \t "http://www.educity.cn/jiaocheng/_blank)。  
(8)评审软件是否具有复用性。  
因此A选项不属于软件质量评审。

**试题答案**

（32）A

# **试题79(2013年上半年试题33)**

在软件维护中，由于企业的外部市场环境和管理需求的变化而导致的维护工作，属于（  ）维护

（33）A．正确性   
B.适应性   
C.完善性   
D.预防性

**试题分析**

①改正性维护改正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。  
②适应性维护适应性维护是指使用软件适应信息技术变化和管理需求变化而进行的修改。  
③完善性维护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。

**试题答案**

（33）B

# **试题80(2013年上半年试题34)**

在对软件系统进行评价时，需要从信息系统的组成部分、评价对象和经济学角度出发进行综合考虑以建立起一套指标体系理论架构。从信息系统评价对象出发，对于用户方来说，他们所关心的是（  ）。

（34）A．用户需求和运行质量   
B.系统外部环境   
C.系统内部结构   
D.系统质量和技术水平

**试题分析**

 在对软件系统进行评价时，从系统评价对象出发，用户方所关心的是用户需求和运行质量平。

**试题答案**

（34）A

# **试题81(2013年上半年试题35)**

在设计测试用例时，应遵循（  ）原则。

（35）A．仅确定测试用例的输入数据，无需考虑输出结果   
B.只需检验程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

**试题分析**

测试用例要包括欲测试的功能、应输入的数据和预期的输出结果。测试用例应覆盖有效合理输入，也要包含不合理、失效的输入。

**试题答案**

（35）C

# **试题82(2013年上半年试题36)**

单元测试中，检查模块接口时，不需要考虑（  ）。

（36）A．测试模块的输入参数和形式参数的个数、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

**试题分析**

 在单元测试开始时，应该对通过所有被测模块的数据流进行测试。  
①输入参数是否与形式参数是否一致。  
②调用模块的实参与被调用模块形参是否一致。  
③全局变量在各模块中的定义和用法是否一致  
④外部输入、输出等等。

**试题答案**

（36）D

# **试题83(2012年下半年试题15)**

以下关于数据流图的叙述中，不正确的是（  ）。

（15）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

**试题分析**

本题主要考查数据流图的平衡原则。  
对于数据流图，必须要保持父图与子图平衡，且每个加工必须要有输入数据流和输出数据流，而每条数据流都至少有一端是加工，而且还要能保持数据守恒（即相关数据流经过加工后得到相应的输出数据流）。

**试题答案**

（15）C

# **试题84(2012年下半年试题18)**

在软件设计阶段，划分模块的原则是：一个模块的（  ）。

（18）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

**试题分析**

模块的作用范围是指受该模块内部一个判定影响的所以模块集合，只有某一模块内含有依赖于该判定的操作，那么该模块就在该判定的作用范围内。  
模块的控制范围包括该模块本身以及该模块的所有下属模块。控制范围完全取决于系统的结构，与模板本身的功能并没有多大关系。  
在系统设计中，对于模块的作用范围和控制范围有两条规则：  
（1）对于任何一个判断，其作用范围应该是这个判断所在模块的控制范围的一个子集。换言之，所有受判断影响的模块应该从属于做出判断的那个模块。  
（2）受模块M判定影响的模块，最好局限于模块M本身或其直接下属模块。  
因此，一个模块的作用范围应该在其控制范围之内。

**试题答案**

（18）A

# **试题85(2012年下半年试题29)**

某开发小组欲开发一个规模较大、需求较明确的项目。开发小组对项目领域熟悉且该项目与小组开发过的某一项目相似，则适宜采用（  ）开发过程模型。

（29）A．瀑布   
B.演化   
C.螺旋   
D.喷泉

**试题分析**

根据题目的描述，我们不难理解，对于开发小组而言，该项目是需求比较明确的一个项目，在这种情况下，最适合采用瀑布模型来开发。而演化模型适合初始时需求不明确的项目；螺旋模型特别加入了风险分析，适合风险高的大项目；而喷泉模型的特点是迭代和无间隙性，是一种面向对象开发的模型。因此本题答案选A。

**试题答案**

（29）A

# **试题86(2012年下半年试题30)**

敏捷开发方法中，（  ）认为每一种不同的项目都需要一套不同的策略、约定和方法论。

（30）A．极限编程（XP）   
B.水晶法（Crystal）   
C.并列争球法（ Scrum）   
D.自适应软件开发（ASD）

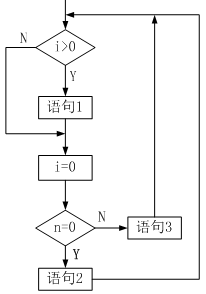
**试题分析**

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法，常见的敏捷开发方法有极限编程法、水晶法、并列争球法和自适应软件开发方法。  
极限编程是一种轻量级的开发方法，它提出了四大价值观：沟通、简单、反馈、勇气。五大原则：快速反馈、简单性假设、逐步修改、提倡更改、优质工作。  
水晶法强调经常交付，认为每一种不同的项目都需要一套不同的策略、约定和方法论。  
并列争球法的核心是迭代、增量交付，按照30天进行迭代开发交付可实际运行的软件。  
自适应软件开发的核心是三个非线性的，重迭的开发阶段：猜测、合作、学习。

**试题答案**

（30）B

# **试题87(2012年下半年试题31)**

采用McCabe度量法计算下图的环路复杂性为（  ）。  


（31）A．2   
B.3   
C.4   
D.5

**试题分析**

本题考查环路复杂性的计算，是考试中常考的一个知识点。对于这类试题，我们可以采用结点数-边的条数+2这个公式来计算。在本题给出的图中，结点个数为6，而边的条数为8条（注意最上面的表和语句2过去的边是重合的，因此可视为同一条边），因此环路复杂性为4。

**试题答案**

（31）C

# **试题88(2012年下半年试题33)**

以下关于文档的叙述中，不正确的是（  ）。  
（33）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  
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

**试题分析**

软件文档是软件产品的一部分，可以说，没有文档的软件不能称之为软件。因为软件开发文档是软件开发使用和维护中的必备资料。它能提高软件开发的效率，保证软件的质量，而且在软件的使用过程中有指导、帮助、解惑的作用，尤其在维护工作中，文档是不可或缺的资料。  
但是，正是因为软件文档占有突出的地位，所以在开发过程中，需要花大量的时间来编制和整理文档。而文档描述的内容不仅包括软件的使用范围及相关的操作命令，还包括很多其他的内容。

**试题答案**

（33）A

# **试题89(2012年下半年试题34)**

由于信用卡公司升级了其信用卡支付系统，导致超市的原有信息系统也需要做相应的修改工作，该类维护属于（  ）。  
（34）A．正确性维护  
B.适应性维护  
C.完善性维护  
D.预防性维护

**试题分析**

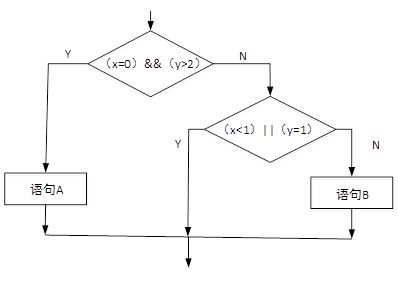
根据引起软件维护的原因，软件维护通常可分为以下四种类型：  
（1）改正性维护。是指在使用过程中发现了隐蔽的错误后，为了诊断和改正这些隐蔽错误而修改软件的活动。该类维护一般占总维护工作量的25%。  
（2）适应性维护。是指为了适应变化了的环境而修改软件的活动。该类维护一般占总维护工作量的20%。  
（3）完善性维护。是指为了扩充或完善原有软件的功能或性能而修改软件的活动。该类维护一般占总维护工作量的50%。  
（4）预防性维护。是指为了提高软件的可维护性和可靠性、为未来的进一步改进打下基础而修改软件的活动。该类维护一般占总维护工作量的5%。  
在本题中，很显然，改变系统为了适应环境的改变，因此属于适应性维护。

**试题答案**

（34）B

# **试题90(2012年下半年试题35-36)**

用白盒测试方法对下图所示的程序进行测试，设计了4个测试用例：①（x=0，y=3）、②（x=1，y=2）、③（x=-1，y=2）和④（x=3，y=1）。测试用例①②实现了（  ）覆盖；若要完成路径覆盖，则可用测试用例（  ）。



（35）A．语句  
B.条件  
C.判定  
D.路径  
  
（36）A．①②  
B.②③  
C.①②③  
D.①③④

**试题分析**

本题主要考查白盒测试。  
当x=0，y=3时，程序流程图中的第一个判定取值为真，且其中的两个条件也都取值为真，然后程序执行语句A。  
当x=1，y=2时，程序流程图中的第一个判定取值为假，且其中的两个条件也都取值为假。然后程序执行第二个判定，结果取假，且第二个判定中的条件也都取值为假。  
当x=-1，y=2时，程序流程图中的第一个判定取值为假，且其中的两个条件也都取值为假。然后程序执行第二个判定，结果取真，且第二个判定中的条件x<1取真，y=1取假。  
当x=3，y=1时，程序流程图中的第一个判定取值为假，且其中的两个条件也都取值为假。然后程序执行第二个判定，结果取真，且第二个判定中的条件x<1取假，y=1取真。  
综上所述，可以知道测试用例①②实现了语句覆盖，即图中的每条语句都至少执行了一次。如果要实现路径覆盖，即每条路径至少执行一次，根据分析可以知道，应该是①②③或者①②④组合。

**试题答案**

（35）A（36）C

# **试题91(2012年上半年试题15)**

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（  ）。

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

**试题分析**

数据流图主要由实体、数据存储、处理过程和数据流四部分组成。在顶层数据流图中，将系统描述成一个处理过程，而其它的是与该处理过程相关的输入输出流，因此顶层数据流图描述了系统的输入与输出。

**试题答案**

（15）B

# **试题92(2012年上半年试题16)**

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（  ）内聚

（16）A．顺序   
B.过程   
C.逻辑   
D.功能

**试题分析**

内聚是指模块内部各元素之间联系的紧密程度，模块的内聚类型通常可以分为7种，根据内聚度从高到低排序分别如下：  
功能内聚：完成一个单一功能，各个部分协同工作，缺一不可。  
顺序内聚：处理元素相关，而且必须顺序执行。  
通信内聚：所有处理元素集中在一个数据结构的区域上。  
过程内聚：处理元素相关，而且必须按特定的次序执行。  
瞬时内聚：所包含的任务必须在同一时间间隔内执行（如初始化模块）。  
逻辑内聚：完成逻辑上相关的一组任务。  
偶然内聚：完成一组没有关系或松散关系的任务。

**试题答案**

（16）C

# **试题93(2012年上半年试题18)**

（  ）最不适于采用无主程序员组的开发人员组织形式。

（18）A．项目开发人数少（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

**试题分析**

大规模项目最不适于采用无主程序员组的开发人员组织形式，因此大项目需要主程序员来整合各模块程序。

**试题答案**

（18）C

# **试题94(2012年上半年试题29)**

软件开发的增量模型（  ）。

（29）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

**试题分析**

增量模型是一种非整体开发的模型，该模型具有较大的灵活性，适合于软件需求不明确的一种模型。使用该模型开发产品，一般是尽快构造出可运行的产品，然后在该产品的基础上再增加需要的新的构建，使产品更趋于完善。

**试题答案**

（29）B

# **试题95(2012年上半年试题30)**

假设某软件公司与客户签订合同开发一个软件系统，系统的功能有较清晰的定义，且客户对交付时间有严格要求，则该系统的开发最适宜采用（  ）。

（30）A．瀑布模型   
B.原型模型   
C.V模型   
D.螺旋模型

**试题分析**

瀑布模型严格遵循软件生命周期各阶段的固定顺序：计划、分析、设计、编程、测试和维护，上一阶段完成后才能进入到下一阶段。瀑布模型的优点是：可强迫开发人员采用规范的方法；严格规定了各阶段必须提交的文档；要求每个阶段结束后，都要进行严格的评审。但瀑布模型过于理想化，而且缺乏灵活性，无法在开发过程中逐渐明确用户难以确切表达或一时难以想到的需求。该模型比较适合于需求明确，对交付时间有严格要求的开发。  
原型模型基于这样一种客观事实：并非所有的需求在系统开发之前都能准确地说明和定义。因此，它不追求也不可能要求对需求的严格定义，而是采用了动态定义需求的方法。它适用于需求不明确的开发环境。  
螺旋模型综合了瀑布模型和演化模型的优点，还增加了风险分析。螺旋模型包含了四个方面的活动：制订计划、风险分析、实施工程、客户评估。采用螺旋模型时，软件开发沿着螺旋线自内向外旋转，每转一圈都要对风险进行识别和分析，并采取相应的对策。螺旋模型比较适合大规模的开发，它对风险控制有很高的要求。  
综上所述，要满足题目描述的要求，应该采用瀑布模型开发最适宜。

**试题答案**

（30）A

# **试题96(2012年上半年试题31)**

某企业由于外部市场环境和管理需求的变化对现有软件系统提出新的需求，则对该软件系统进行的维护属于（  ）维护。

（31）A．正确性   
B.完善性   
C.适应性   
D.预防性

**试题分析**

改正性维护：在软件交付使用后，必然会有一部分隐藏的错误被带到运行阶段来。这些隐藏下来的错误在某些特定的使用环境下就会暴露出来。为了纠正这些错误而对软件进行的维护工作就是改正性维护。

适应性维护：随着计算机的飞速发展，外部环境(新的硬、软件配置)或数据环境(数据库、数据格式、数据输入∕输出方式、数据存储介质)或应用环境可能发生变化，为了使软件适应这种变化，而去修改软件的过程就叫做适应性维护。

完善性维护：在软件的使用过程中，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。这种情况下进行的维护活动叫做完善性维护。

预防性维护：为了提高软件的可维护性、可靠性等而提出的一种维护类型，它为以后进一步改进软件打下良好基础。通常，预防性维护定义为：“把今天的方法学用于昨天的系统以满足明天的需要”。也就是说，采用先进的软件工程方法对需要维护的软件或软件中的某一部分(重新)进行设计、编制和测试。

综上所述，本题的正确答案应选C。

**试题答案**

（31）C

# **试题97(2012年上半年试题32)**

McCall软件质量模型从软件产品的运行、修正和转移三个方面确定了11个质量特性，其中（  ）不属于产品运行方面的质量特性。

（32）A．正确性   
B.可靠性   
C.效率   
D.灵活性

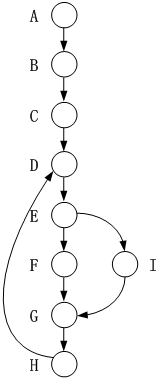
**试题分析**

McCall软件质量模型从软件产品的运行、修正和转移三个方面确定了11个质量特性，其中运行方面包含了正确性、可靠性、效率、完整性、使用性这些质量特性。修正方面包含了维护性、测试性、灵活性这3个质量特性。转移方面包含了维护性移植性、复用性、共运行性这3个质量特性。

**试题答案**

（32）D

# **试题98(2012年上半年试题33)**

采用McCabe度量法计算下列程序图的环路复杂性为（  ）。  


（33）A．2   
B.3   
C.4   
D.5

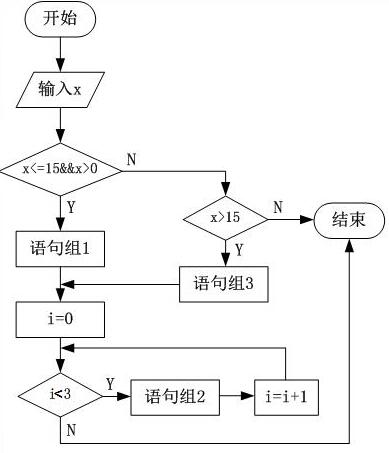
**试题分析**

McCabe度量法是一种基于程序控制流的复杂性度量方法。采用这种方法要先画出程序图，然后采用公式V（G）=m-n+2计算环路复杂度。其中，m是图G中弧的个数，n是图G中的结点数。本题图中结点数为9，边数为10，所以环路复杂度为10-9+2=3。

**试题答案**

（33）B

# **试题99(2012年上半年试题34-35)**

在白盒测试法中，（  ）是最弱的覆盖准则。下图至少需要（  ）个测试用例才可以完成路径覆盖，语句组2不对变量i进行操作。  
  


（34）A．语句   
B.条件   
C.判定   
D.路径   
  
（35）A．1   
B.2   
C.3   
D.4

**试题分析**

白盒测试常用的技术是逻辑覆盖，即考查用测试数据运行被测程序时对程序逻辑的覆盖程度。主要的覆盖标准有六种：语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、组合条件覆盖和路径覆盖。  
（1）语句覆盖  
语句覆盖是指选择足够多的测试用例，使得运行这些测试用例时，被测程序的每个语句至少执行一次。很显然，语句覆盖是一种很弱的覆盖标准。  
（2）判定覆盖  
判定覆盖又称分支覆盖，它的含义是，不仅每个语句至少执行一次，而且每个判定的每种可能的结果（分支）都至少执行一次。判定覆盖比语句覆盖强，但对程序逻辑的覆盖程度仍然不高。  
（3）条件覆盖  
条件覆盖的含义是，不仅每个语句至少执行一次，而且使判定表达式中的每个条件都取到各种可能的结果。条件覆盖不一定包含判定覆盖，判定覆盖也不一定包含条件覆盖。  
（4）判定/条件覆盖  
同时满足判定覆盖和条件覆盖的逻辑覆盖称为判定/条件覆盖。它的含义是，选取足够的测试用例，使得判定表达式中每个条件的所有可能结果至少出现一次，而且每个判定本身的所有可能结果也至少出现一次。  
（5）条件组合覆盖  
条件组合覆盖的含义是，选取足够的测试用例，使得每个判定表达式中条件结果的所有可能组合至少出现一次。显然，满足条件组合覆盖的测试用例，也一定满足判定/条件覆盖。因此，条件组合覆盖是上述五种覆盖标准中最强的一种。然而，条件组合覆盖还不能保证程序中所有可能的路径都至少经过一次。  
（6）路径覆盖  
路径覆盖的含义是，选取足够的测试用例，使得程序的每条可能执行到的路径都至少经过一次（如果程序中有环路，则要求每条环路径至少经过一次）。路径覆盖实际上考虑了程序中各种判定结果的所有可能组合，因此是一种较强的覆盖标准。  
本题的第二问比较复杂，关键是我们要清楚程序流程图中有几条路径，然后为每条路径设置一个测试用例。  
在本题给出的程序流程图中，如果我们将判定“x<=15&&x>0”规定为第1个判定，“x>15”规定为第2个判定，“i<3”规定为第3个判定的话，那么本题中的路径有：第1个判定为N，第2个判定为N；第1个判定为N，第2个判定为Y，第3个判定一直为Y，后再取N（这里要注意，由于在第3个判定前，给i赋初值为0，因此这个判定每次都是先为Y，循环后再为N）；第1个判定为Y，第3个判定先为Y，然后再为N。总共只有这3条路径，因此至少需要设置3个测试用例来满足路径覆盖。

**试题答案**

（34）A（35）C

# **试题100(2012年上半年试题36)**

根据ISO/IEC 9126软件质最模型中对软件质量特性的定义，可维护性质量特性的（  ）子特性是指与为确认经修改软件所需努力有关的软件属性。

（36）A．易测试性   
B.易分析性   
C.稳定性   
D.易改变性

**试题分析**

可维护性质量特性是指与软件维护的难易程度相关的一组软件属性，它包含了易分析性、稳定性、易测试性和易改变性4个子特性。其中：  
易分析性是描述诊断缺陷或失效原因、判定待修改程度的难易程度的特性。  
稳定性是描述修改造成难以预料的后果的风险程度，风险程度越低，稳定性越好。  
易测试性是描述测试已修改软件的难易程度的特性。  
易改变性是描述修改、排错或适应环境变化的难易程度。  
本题中，是说与为确认经修改软件所需努力有关的软件属性，也就是说要确认修改后的软件是否正确所要付出的努力，这应该是易测试性所描述的内容，因此本题答案选A。

**试题答案**

（36）A

# **试题101(2011年下半年试题15)**

确定软件的模块划分及模块之间的调用关系是（  ）阶段的任务。

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

**试题分析**

需求分析阶段的任务主要是要解决系统做什么的问题，即弄清楚问题的要求，包括需要输入什么数据，要得到什么结果，最后应输出什么。  
概要设计的主要任务是把需求分析得到的结果转换为软件结构和数据结构，即将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面、确定数据的结构特性、以及数据库的设计等。  
详细设计是在概要设计的基础上更细致的设计，它包括具体的业务对象设计、功能逻辑设计、界面设计等工作。详细设计是系统实现的依据，需要更多地考虑设计细节。  
编码即编写程序代码，具体实现系统。

**试题答案**

（15）B

# **试题102(2011年下半年试题16)**

利用结构化分析模型进行接口设计时，应以（  ）为依据。  
（16）A．数据流图   
B.实体--关系图   
C.数据字典   
D.状态--迁移图

**试题分析**

数据流图是结构化分析模型需求分析阶段得到的结果，描述了系统的功能，在进行接口设计时，应以它为依据。

**试题答案**

（16）A

# **试题103(2011年下半年试题29)**

以下关于喷泉模型的叙述中，不正确的是（  ）。

（29）A．喷泉模型是以对象作为驱动的模型，适合于面向对象的开发方法   
B.喷泉模型克服了瀑布模型不支持软件重用和多项开发活动集成的局限性   
C.模型中的开发活动常常需要重复多次，在迭代过程中不断地完善软件系统   
D.各开发活动（如分析、设计和编码）之间存在明显的边界

**试题分析**

喷泉模型主要用于描述面向对象的开发过程。喷泉一词体现了面向对象开发过程的迭代和无间隙特征。迭代意味着模型中的开发活动常常需要多次重复，每次重复都会增加或明确一些目标系统的性质，但却不是对先前工作结果的本质性改动。无间隙是指在开发活动（如分析、设计、编程）之间不存在明显的边界，而是允许各开发活动交叉、迭代地进行。

**试题答案**

（29）D

# **试题104(2011年下半年试题30)**

 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（  ）进行开发。

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

**试题分析**

本题主要考查各开发模型的基本特征，适用场合。  
瀑布模型严格遵循软件生命周期各阶段的顺序进行软件开发，适用于需求非常明确的的场合，而在本题中，题目告诉我们是用新技术开发一个新系统来替代老系统，属于二次开发，需求应该非常明确，因此适合采用瀑布模型。

**试题答案**

（30）A

# **试题105(2011年下半年试题31)**

 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（  ）质量特性。

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

**试题分析**

功能性是指与功能及其指定的性质有关的一组软件质量；可靠性是指衡量在规定的一段时间内和规定条件下维护性能水平的一组软件质量；可维护性是指与软件维护的难易程度相关的一组软件属性；易使用性是指与使用难易程度及规定或隐含用户对使用方式所做的评价相关的属性。  
综上所述，可以知道将每个用户的数据和其他用户的数据隔离开，是考虑了软件的功能性质量特性。

**试题答案**

（31）A

# **试题106(2011年下半年试题32)**

 在软件评审中，设计质量是指设计的规格说明书符合用户的要求。设计质量的评审内容不包括（  ）。

（32）A．软件可靠性   
B.软件的可测试性   
C.软件性能实现情况   
D.模块层次

**试题分析**

设计质量评审的内容主要有：软件需求规格说明、软件可靠性、软件是否具有可修改性、可扩充性、可互换性、可移植性、可测试行和可重用性及软性性能的实现情况等。

**试题答案**

（32）D

# **试题107(2011年下半年试题33)**

针对应用在运行期的数据特点，修改其排序算法使其更高效，属于（  ）维护。

（33）A．正确性   
B.适应性   
C.完善性   
D.预防性

**试题分析**

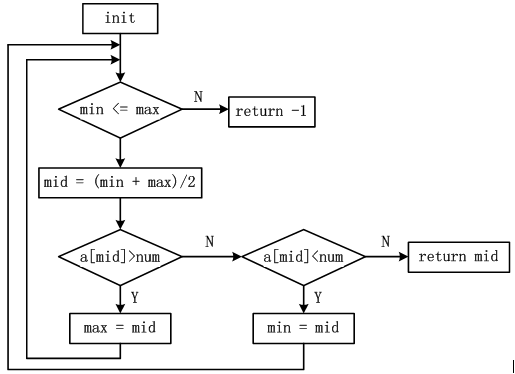
改正性维护也叫正确性维护，是指在使用过程中发现了隐蔽的错误后，为了诊断和改正这些隐蔽错误而修改软件的活动。  
适应性维护是指为了适应变化了的环境而修改软件的活动。  
完善性维护是指为了扩充或完善原有软件的功能或性能而修改软件的活动。  
预防性维护是指为了提高软件的可维护性和可靠性、为未来的进一步改进打下基础而修改软件的活动。  
综上所述，可知针对应用在运行期的数据特点，修改其排序算法使其更高效属于完善性维护。

**试题答案**

（33）C

# **试题108(2011年下半年试题34)**

 下图所示的逻辑流实现折半查找功能，最少需要（  ）个测试用例可以覆盖所有的可能路径。



（34）A．1   
B.2   
C.3   
D.4

**试题分析**

折半查找是在一组有序的数（假设为递增顺序）中查找一个数的算法，其思想是：将待查找的数与数组中间位置mid的数进行比较，若相等，则查找成功；若大于中间位置的数，则在后半部分进行查找；若小于中间位置的数，则在前半部分进行查找。直到查找成功，返回所查找的数的位置，或者失败，返回-1。设计一个查找成功的测试用例，可以覆盖除了return-1之外的所有语句和路径；设计一个查找失败的测试用例，可以覆盖除了return mid之外的所有语句和路径。因此，最少需要2个测试用例才可以覆盖所有路径。

**试题答案**

（34）B

# **试题109(2011年下半年试题35)**

在某班级管理系统中，班级的班委有班长、副班长、学习委员和生活委员，且学生年龄在15～25岁。若用等价类划分来进行相关测试，则（  ）不是好的测试用例。

（35）A．（队长，15）  
B.（班长，20）  
C.（班长，15）  
D.（队长，12）

**试题分析**

所谓等价类就是某个输入域的集合，对于一个等价类中的输入值来说，它们揭示程序中错误的作用是等效的。也就是说，如果等价类中的一个输入数据能检测出一个错误，那么等价类中的其他输入数据也能检测出同一个错误。  
在本题中一个设计用例包含两个输入条件，一个是班委，另一个是年龄，从四个选项来看，D选项中的两个输入都不是有效数据，如果用这个用例检测出了一个错误，那么也不能确定是由哪个输入条件引起的，因此其不是一个好的测试用例。

**试题答案**

（35）D

# **试题110(2011年下半年试题36)**

进行防错性程序设计，可以有效地控制（  ）维护成本。

（36）A．正确性  
B.适应性  
C.完善性  
D.预防性

**试题分析**

改正性维护也叫正确性维护，是指在使用过程中发现了隐蔽的错误后，为了诊断和改正这些隐蔽错误而修改软件的活动。  
适应性维护是指为了适应变化了的环境而修改软件的活动。  
完善性维护是指为了扩充或完善原有软件的功能或性能而修改软件的活动。  
预防性维护是指为了提高软件的可维护性和可靠性、为未来的进一步改进打下基础而修改软件的活动。  
进行防错性程序设计，可以有效地控制正确性维护的成本。

**试题答案**

（36）A

# **试题111(2011年上半年试题16)**

模块A直接访问模块B的内部数据，则模块A和模块B的耦合类型为（  ）。

（16）A．数据耦合   
B.标记耦合   
C.公共耦合   
D.内容耦合

**试题分析**

    本题主要考查耦合的基本内容。  
    耦合是指两个或两个以上的模块相互依赖于对方的一个量度。它可以分为非直接耦合、数据耦合、标记耦合、控制耦合、外部耦合、公共耦合及内容耦合等。  
    当一个模块直接修改或操作另一个模块的数据或者直接转入另一个模块时，就发生了内容耦合。所以本题的答案选D。

**试题答案**

（16）D

# **试题112(2011年上半年试题29)**

为了有效地捕获系统需求，应采用（  ）。

（29）A．瀑布模型   
B.V模型   
C.原型模型   
D.螺旋模型

**试题分析**

瀑布模型严格遵循软件生命周期各阶段的固定顺序进行软件开发，其优点是可强迫开发人员采用规范的方法；严格规定了各阶段必须提交的文档；要求每个阶段结束后，都要进行严格的评审；而其缺点是过于理想化，缺乏灵活性，无法在开发过程中逐渐明确用户难以确切表达或一时难以想到的需求。  
V模型是一种典型的测试模型，该模型强调开发过程中测试贯穿始终。  
原型模型基于这样一种客观事实：并非所有的需求在系统开发之前都能准确地说明和定义。因此，它不追求也不可能要求对需求的严格定义，而是采用了动态定义需求的方法。原型模型适用于需求不够明确的项目，它能有效地捕获系统需求。  
螺旋模型综合了瀑布模型和演化模型的优点，还增加了风险分析。采用螺旋模型时，软件开发沿着螺旋线自内向外旋转，每转一圈都要对风险进行识别和分析，并采取相应的对策。

**试题答案**

（29）C

# **试题113(2011年上半年试题30)**

关于过程改进，以下叙述中不正确的是（  ）。

（30）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

**试题分析**

    软件过程改进的实施对象是软件企业的软件过程，也就是软件产品的生产过程，其中还也包括软件维护之类的维护过程。  
    在本题各选项的描述中，A、B、C都是正确的，D不正确。  
    CMMI是Capability Maturity Model Integration的简称，即能力成熟度模型集成，它是在CMM的基础上发展起来的。CMMI是一种过程改进模型，它不仅支持阶段性过程改进，而且还支持连续性过程改进。

**试题答案**

（30）D

# **试题114(2011年上半年试题33)**

高质量的文档所应具有的特性中，不包括（  ）。

（33）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\*\*文档\*\*节”的方式引用

**试题分析**

    本题主要考查文档管理的相关内容。高质量的文档应具有针对性、精确性和完整性等特性。即文档编制应考虑读者对象群；文档的行文应该十分确切，不能出现多义性的描述；任何文档都应当是完整的、独立的，应该自成体系。  
选项D描述的显然不符合高质量文档的要求。

**试题答案**

（33）D

# **试题115(2011年上半年试题34)**

在软件维护阶段，为软件的运行增加监控设施属于（  ）维护。

（34）A．改正性   
B.适应性   
C.完善性   
D.预防性

**试题分析**

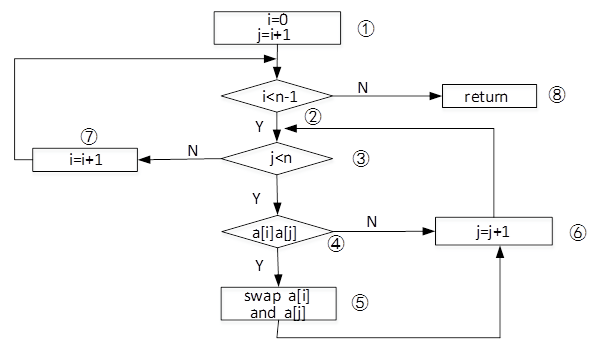
根据引起软件维护的原因不同，软件维护通常可分为以下四种类型：  
改正性维护。是指在使用过程中发现了隐蔽的错误后，为了诊断和改正这些隐蔽错误而修改软件的活动。该类维护一般占总维护工作量的25%。  
适应性维护。是指为了适应变化了的环境而修改软件的活动。该类维护一般占总维护工作量的20%。  
完善性维护。是指为了扩充或完善原有软件的功能或性能而修改软件的活动。该类维护一般占总维护工作量的50%。  
预防性维护。是指为了提高软件的可维护性和可靠性、为未来的进一步改进打下基础而修改软件的活动。该类维护一般占总维护工作量的5%。  
而本题所描述的为软件的运行增加监控设施属于完善性维。

**试题答案**

（34）C

# **试题116(2011年上半年试题35)**

下图所示的逻辑流，最少需要（  ）个测试用例可实现语句覆盖。



（35）A．1   
B.2   
C.3   
D.5

**试题分析**

    语句覆盖是一种白盒测试，它是指选择足够多的测试用例，使得运行这些测试用例时，被测程序的每个语句至少执行一次。显然，语句覆盖是一种很弱的覆盖标准。  
    根据题目给出的逻辑图，程序的出口只有⑧，路径主要有①②⑧、①②③⑦②⑧、①②③④⑥③⑦②⑧及①②③④⑤⑥③⑦②⑧四条。那么很显然，路径①②③④⑤⑥③⑦②⑧覆盖了所有的语句，因此本题答案选A。

**试题答案**

（35）A

# **试题117(2011年上半年试题36)**

在改正当前故障的同时可能会引入新的故障，这时需要进行（  ）。

（36）A．功能测试   
B.性能测试   
C.回归测试   
D.验收测试

**试题分析**

回归测试是指修改了当前故障后，重新进行测试以确认修改没有引入新的错误或导致其他的错误。因此本题答案选C。

**试题答案**

（36）C

# **试题118(2010年下半年试题15)**

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，（  ）最适合开发此项目。

（15）A．原型模型   
B.瀑布模型   
C.V模型   
D.螺旋模型

**试题分析**

本题主要考查我们对各开发模型的理解。  
瀑布模型具有可强迫开发人员采用规范的方法；严格规定了各阶段必须提交的文档；要求每个阶段结束后，都要进行严格的评审等优点。但瀑布模型过于理想化，而且缺乏灵活性，无法在开发过程中逐渐明确用户难以确切表达或一时难以想到的需求。  
原型模型一般是基于需求不容易明确这一事实：并非所有的需求在系统开发之前都能准确地说明和定义。因此，它不追求也不可能要求对需求的严格定义，而是采用了动态定义需求的方法。快速原型法适用于需求不够明确的项目。  
V模型一种典型的测试模型，该模型强调开发过程中测试贯穿始终。  
螺旋模型综合了瀑布模型和演化模型的优点，并在此基础上还增加了风险分析。采用螺旋模型时，软件开发沿着螺旋线自内向外旋转，每转一圈都要对风险进行识别和分析，并采取相应的对策。  
本题描述的是一个大型项目，那么对文档的要求应该相应较高，而且具备了相关领域及类似规模系统的开发经验，那么系统的需求应该较明确，综合考虑，应该采用瀑布模型。

**试题答案**

（15）B

# **试题119(2010年下半年试题18)**

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（  ）最佳实践。

（18）A．隐喻   
B.重构   
C.小型发布   
D.持续集成

**试题分析**

12个最佳实践分别是：计划游戏，小型发布，隐喻，简单设计，测试先行，重构，结对编程，集体代码所有制，持续集成，每周工作40小时，现场客户及编码标准。其中系统的设计要能够尽可能早交付属于小型发布。  
小型发布可以使得集成更频繁，客户获得的中间结果越频繁，反馈也就越频繁，客户就能够实时地了解项目的进展情况，从而提出更多的意见，以便在下一次迭代中计划进去，以实现更高的客户满意度。

**试题答案**

（18）C

# **试题120(2010年下半年试题30)**

以下关于过程改进的叙述中，错误的是（  ）。

（30）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、改进和监控四个部分   
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

**试题分析**

软件过程改进/过程改进（SPI）帮助软件企业对其软件(制作)过程的改变(进)进行计划、(措施)制定以及实施。它的实施对象就是软件企业的软件过程，也就是软件产品的生产过程，当然也包括软件维护之类的维护过程，而对于其他的过程并不关注。软件过程改进不是一次性就能达到最终目标，而是需要反复进行的。在软件过程改进时，如果发现问题，需转化为软件过程改进计划。

**试题答案**

（30）B

# **试题121(2010年下半年试题31)**

软件复杂性度量的参数不包括（  ）。

（31）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

**试题分析**

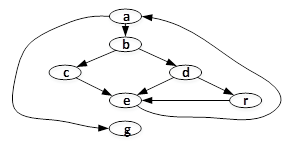
软件复杂性主要表现在程序的复杂性。程序的复杂性主要指模块内程序的复杂性。它直接关联到软件开发费用的多少、开发周期长短和软件内部潜伏错误的多少。同时它也是软件可理解性的另一种度量。  
软件复杂性度量的参数很多，主要有：  
    (1) 规模，即总共的指令数，或源程序行数。  
    (2) 难度，通常由程序中出现的操作数的数目所决定的量来表示。  
    (3) 结构，通常用于程序结构有关的度量来表示。  
    (4) 智能度，即算法的难易程度。

**试题答案**

（31）B

# **试题122(2010年下半年试题32)**

根据McCabe度量法，以下程序图的复杂性度量值为（  ）。



（32）A．4   
B.5   
C.6   
D.7

**试题分析**

   常见的程序复杂性度量法主要有McCabe度量法和代码行度量法。其中McCabe度量法是一种基于程序控制流的复杂性度量方法。McCabe定义的程序复杂性度量值又称环路复杂度，它基于一个程序模块的程序图中环路的个数。计算有向图G的环路复杂性的公式：  
  V(G)＝m－n＋2  
    其中，V（G）是有向图G中的环路个数 ，m是图G中有向弧个数，n是图G中结点个数。在本题中m的值应该是9，而n的值是7.所以根据公式最后计算的结果为4。

**试题答案**

（32）A

# **试题123(2010年下半年试题33)**

软件系统的可维护性评价指标不包括（  ）。

（33）A．可理解性   
B.可测试性   
C.扩展性   
D.可修改性

**试题分析**

软件系统的可维护性是指与软件维护的难易程度相关的一组软件属性。它的评价指标有可理解性、可修改性、可测试性及稳定性等。

**试题答案**

（33）C

# **试题124(2010年下半年试题34)**

以下关于软件系统文档的叙述中，错误的是（  ）。

（34）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

**试题分析**

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，通过它可以提高软件开发的可见度，提高软件开发的效率以及便于用户理解软件的功能、性能等各项指标。

**试题答案**

（34）C

# **试题125(2010年下半年试题35)**

以下关于软件测试的叙述中，正确的是（  ）。

（35）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

**试题分析**

    软件测试的目的就是在软件投入生产性运行之前，尽可能多地发现软件产品（主要是指程序）中的错误和缺陷。软件测试是软件质量保证的主要手段之一。  
    一个好的测试用例是极有可能发现至今为止尚未发现的错误的测试用例。一次成功的测试是发现了至今为止尚未发现的错误的测试。一个高效的测试是指用少量的测试用例，发现被测软件尽可能多的错误。软件测试不能说明软件中不存在错误。

**试题答案**

（35）C

# **试题126(2010年下半年试题36)**

不属于黑盒测试技术的是（  ）。

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

**试题分析**

黑盒测试又称功能测试。它把软件看做一个不透明的黑盒子，完全不考虑（或不了解）软件的内部结构和处理算法，它只检查软件功能是否能按照软件需求说明书的要求正常使用，软件是否能适当地接收输入数据并产生正确的输出信息，软件运行过程中能否保持外部信息（例如文件和数据库）的完整性等。  
常用的黑盒测试技术包括等价类划分、边界值分析、错误推测和因果图等。

**试题答案**

（36）B

# 十九.历年真题解析

## 2005年上半年

●下列叙述中，与提高软件可移植性相关的是\_\_\_(18)\_\_\_\_

（18）A．选择时间效率高的算法 B．尽可能减少注释

C．选择空间效率高的算法 D．尽量用高级语言编写系统中对效率要求不高的部分

●在系统转换的过程中，旧系统和新系统并行工作一段时间，再由新系统代替旧系统的策略称为\_\_(19)\_\_；在新系统全部正式运行前，一部分一部分地代替旧系统的策略称为\_(20)\_\_。

（19）A．直接转换 　B．位置转换　 C．分段转换 　D．并行转换

（20）A．直接转换 　 B．位置转换 　C．分段转换 　 D．并行转换

●下列要素中，不属于DFD的是\_\_(21)\_。当使用DFD对一个工资系统进行建模时，\_(22)\_\_可以被认定为外部实体。

（21）A.加工 B.数据流 C.数据存储 D.联系

（22）A.接收工资单的银行 B.工资系统源代码程序 C.工资单 D.工资数据库的维护

●在系统验收测试中，\_(23)\_是在一个模拟的环境下使用模拟数据运行系统；\_\_(24)\_是在一个实际环境中使用真实数据运行系统。（23）A．验证测试 B．审计测试 C．确认测试 D．模块测试

（24）A．验证测试 B．审计测试 C．确认测试 D．模块测试  
●采用瀑布模型进行系统开发的过程中，每个阶段都会产生不同的文档。以下关于产生这些文档的描述中，正确的是\_\_\_(25)\_\_\_。

（25）A．外部设计评审报告在概要设计阶段产生 B．集成测试计划在程序设计阶段产生  
C．系统计划和需求说明在详细设计阶段产生D．在进行编码的同时，独立地设计单元测试计划

## 2005年下半年

●在开发一个系统时，如果用户对系统的目标是不很清楚，难以定义需求，这时最好使用　（6）。

（6）A. 原型法 B. 瀑布模型 C. V-模型 D. 螺旋模型

●应该在（7）阶段制定系统测试计划。

（7）A. 需求分析 B. 概要设计 C. 详细设计 D. 系统测试

●代码走查(code walkthrough)和代码审查(code inspection)是两种不同的代码评审方法，这两种方法的主要区别是\_(14)\_\_。

(14)A.在代码审查中由编写代码的程序员来组织讨论，而在代码走查中由高级管理人员来领导评审小组的活动  
　　B.在代码审查中只检查代码中是否有错误，而在代码走查中还要检查程序与设计文档的一致性  
　　C.在代码走查中只检查程序的正确性，而在代码审查中还要评审程序员的编程能力和工作业绩  
　　D.代码审查是一种正式的评审活动，而代码走查的讨论过程是非正式的

●在软件项目管理中可以使用各种图形工具来辅助决策，下面对Gannt图的描述不正确的是\_\_(15)\_\_\_。

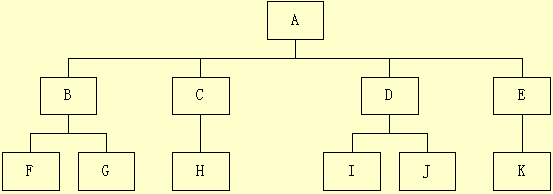
(15)A.Gannt图表现各个活动的顺序和它们之间的因果关系 B.Gannt图表现哪些活动可以并行进行

C.Gannt图表现了各个活动的起始时间 D.Gannt图表现了各个活动完成的进度

●一个故障已经被发现，而且也被排除了，为了检查修改是否引起了其它故障，这时应该进行\_(19)\_\_。

(19)A.程序走查 B.退化测试 C.软件评审 D.接收测试

●一个系统的模块结构图如下所示，用{×,×,×}表示这个系统的测试模块组合。下面的选项中\_\_\_(20)\_\_\_表示自顶向下的测试，\_\_\_\_(21)\_\_\_表示三明治式测试。



供选择的答案：

(20)A.{A}{A,B,C,D,E}{A,B,C,D,E,F,G,H,I,J,K}

B.{F}{G}{H}{I}{J}{K}{B,F,G}{C,H}{D,I,J}{E,K}{A,B,C,D,E,F,G,H,I,J,K}

C.{K}{J}{I}{H}{G}{F}{B}{C}{D}{E}{A,B,C,D,E,F,G,H,I,J,K}

D.{A}{F}{G}{H}{I}{J}{K}{B,F,G}{C,H}{D,I,J}{E,K}{A,B,C,D,E,F,G,H,I,J,K}

(21)A.{A}{A,B,C,D,E}{A,B,C,D,E,F,G,H,I,J,K}

B.{F}{G}{H}{I}{J}{K}{B,F,G}{C,H}{D,I,J}{E,K}{A,B,C,D,E,F,G,H,I,J,K}

C.{K}{J}{I}{H}{G}{F}{B}{C}{D}{E}{A,B,C,D,E,F,G,H,I,J,K }

D.{A}{F}{G}{H}{I}{J}{K}{B,F,G}{C,H}{D,I,J}{E,K}{ A,B,C,D,E,F,G,H,I,J,K }

●以下内容中，\_\_\_(42)\_\_\_应写入操作手册。

(42)A.描述系统对各种输入数据的处理方法　　B.说明系统升级时厂商提供的服务  
　　C.描述系统处理过程的各个界面　　 D.说明系统各部分之间的接口关系

●某软件工程项目各开发阶段工作量的比例如下表所示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求分析 | 概要设计 | 详细设计 | 编码 | 测试 |
| 0.29 | 0.13 | 0.17 | 0.10 | 0.31 |

假设当前已处于编码阶段，3000行程序已完成了1200行，则该工程项目开发进度已完成的比例是\_\_\_\_(43)\_\_\_。

(43)A.29% B.45% C.59% D.63%

●关于维护软件所需的成本，以下叙述正确的是\_\_\_(45)\_\_\_。

(45)A．纠正外部和内部设计错误比纠正源代码错误需要更大的成本

B.与需求定义相比，源代码的文字量大得多，所以源代码的维护成本更高

C.用户文档需要经常更新，其维护成本超过了纠正设计错误的成本

D.需求定义的错误会在设计时被发现并纠正，因此需求定义纠错的成本小于源代码纠错的成本

## 2006年上半年

●使用LOC (lines of code)度量软件规模的优点是\_\_\_(9)\_\_\_。

(9)A. 容易计算　B. 与使用的编程语言相关　 C. 与使用的开发模型有关　　 D. 在设计之前就可以计算出LOC

●渐增式开发方法有利于\_\_\_(4)\_\_\_。

(4)A. 获取软件需求　B. 快速开发软件　C. 大型团队开发　D.商业软件开发

● 在软件项目管理中可以使用各种图形工具来辅助决策，下面对Gantt图的描述中，不正确的是\_\_\_(15)\_\_\_。

(15)A．Gantt图表现了各个活动的持续时间　　 B．Gantt图表现了各个活动的起始时间

C．Gantt图反映了各个活动之间伪依赖关系 D．Gantt图表现了完成各个活动的进度

● 模块的耦合度描述了\_\_\_(16)\_\_\_。

(16)A．模块内各种元素结合的程度　B．模块内多个功能之间的接口

　　C．模块之间公共数据的数量　　D．模块之间相互关联的程度

● 内聚是一种指标，表示一个模块\_\_\_(17)\_\_\_。

(17)A．代码优化的程度　　　　B．代码功能的集中程度

　 C．完成任务时及时程度 　 D．为了与其他模块连接所要完成的工作量

● 在软件项目开发过程中，评估软件项目风险时，\_\_\_(18)\_\_\_与风险无关。  
(18)A．高级管理人员是否正式承诺支持该项目 B．开发人员和用户是否充分理解系练的需求  
 C．最终用户是否同意部署已开发的系统 D．开发需要的资金是否能按时到位

●最主要的软件质量度量指标有正确性、可维护性、完整性和可用性，下面关于这些质量指标的描述中正确的是\_\_\_(26)\_\_\_。  
(26)A．软件维护的工作量比开发阶段的工作量小 B．正确性用每千行代码的故障(fault)数来度量  
 C．软件完整性是指软件功能与需求符合的程度 D．可用性与用户的操作效率和用户对软件的主观评价有关

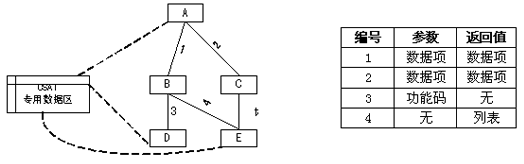
● 正式的技术评审FTR(Formal Technical Review)是软件工程师组织的软件质量保证活动，下面关于FTR指导原则中不正确的是\_\_(28)\_\_。  
(28)A．评审产品，而不是评审生产者的能力　　B．要有严格的评审计划，并遵守日程安排  
 C．对评审中出现的问题要充分讨论，以求彻底解决　　D．限制参与者人数，并要求评审会之前做好准备

● 两个小组独立地测试同一个程序，第一组发现25个错误，第二组发现30个错误，在两个小组发现的错误中有15个是共同的，那么可以估计程序中的错误总数是 (29) 个。  
(29)A．25　　B．30　　　C．50　　D．60

● 下面有关测试的说法正确的是\_\_\_(30)\_\_。  
(30)A．测试人员应该在软件开发结束后开始介入 B．测试主要是软件开发人员的工作  
 C．要根据软件详细设计中设计的各种合理数据设计测试用例D．严格按照测试计划进行，避免测试的随意性

● 在绘制数据流图时，要遵循的一个原则是父图与子图的平衡，所谓平衡是指\_\_\_(31)\_\_\_。  
(31)A．父图和子图都不得改变数据流的性质　　B．子图不改变父图数据流的—致性  
　 C．父图的输入/输出数据流与子图的输入/输出数据流一致　　D．子图的输出数据流完全由父图的输入数据流确定

● 下图中的程序由A、B、C、D、E 5个模块组成，下表中描述了这些模块之间的接口，每一个接口有一个编号。此外，模块A、D和E都要引用一个专用数据区。那么A和E之间耦合关系是\_\_\_(32)\_\_\_。



(32)A．公共耦合　 B．数据耦合　　　　C．内容耦合　 D．无耦合

## 2006年下半年

●常见的软件开发模型有瀑布模型、演化模型、螺旋模型、喷泉模型等。其中（5）模型适用于需求明确或很少变更的项目，（6）主要用来描述面向对象的软件开发过程。

（5） A瀑布模型 B.演化模型 C.螺旋模型 D.喷泉模型 （6）A．瀑布模型 B.演化模型 C.螺旋模型 D.喷泉模型

●软件能力成熟度模型（CMM）是目前国际上最流行、最实用的软件生产过程标准和软件企业成熟度的等级认证标准。该模型将软件能力成熟度自低到高依次划分为初始级、可重复级、已定义级、已管理级、优化级。从（17）开始，要求企业建立基本的项目管理过程的政策和管理规程，使项目管理过程有章可循。

（17）A．初始级 B. 可重复级 C. 已定义级 D. 已管理级

●软件项目开发成本的估算依据，通常是开发成本估算模型，常用的模型有：IBM模型②Putnam模型③基本COCOMO模型④中级COCOMO模型⑤高级OCOMO模型其中（18）都是静态单变量模型。

（18）A．①② B. ②④⑤ C. ①③ D. ③④⑤

●过程（UP）的基本特征是“用例驱动，以架构为中心的和受控的迭代式增量开发”。UP将一个周期的开发过程化分为4个阶段，其中 (26) 提交结果包含了系统架构。

（26）A.先启阶段 B.精化阶段 C.构建阶段 D.提交阶段

●某软件在运行初期运行在Windows NT环境中，现因某种原因，该软件需要在UNIX环境中运行，而且必须完成同样的功能。为适应这个要求，软件本身需要进行修改，而所需的工作量取决于该软件的（27）

（27）A.可扩充性 B.可靠性 C.复用性 D.可移植性

●按照ISO/IEC 9126软件质量度量模型定义，一个软件可靠性的子特性包括（28）：

（28）A.容错性和安全性 B.容错性和适应性 C.容错性和易恢复性 D.易恢复性和安全性

●（29）详细描述软件的功能、性能和用户界面，以使用户了解如何使用软件。

（29）A.概要设计说明书 B.详细设计说明书 C.用户手册 D.用户需求说明书

●各类软件维护活动中，（30）维护占整个维护工作的比重最大。

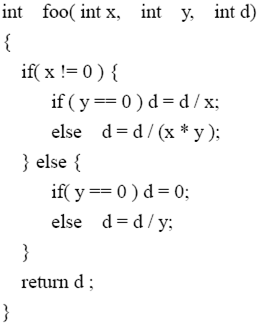
（30） A.完善性 B.改正性 C.适应性 D.预防性

●软件的测试通常分为单元测试、组装测试、确认测试、系统测试四个阶段进行。（32）属于确认测试阶段的活动。

（32）A.设计评审 B.代码审查 C.结构测试 D.可靠性测试

●给出C语言程序, 当用路径覆盖法测试时，至少需要设计（31）个测试用例。

(31) A.3 B. 4 C. 5 D. 8



## 2007年上半年

●结构化开发方法中，数据流图是 (15) 阶段产生的结果（2）。

(15) A．需求分析 B．总体设计 C．详细设计 D．程序编码

●CVS是一种 (16) 工具.

(16) A.需求分析 B.编译 C.程序编码 D.版本控制

●以下关于原型化开发方法的叙述中,不正确的是 (17) .

(17) A.原型化方法适应于需求不明确的软件开发 B.在开发过程中,可以废弃不用早期构造的软件原型

C.原型化方法可以直接开发出最终产品 D.原型化方法利用于确认各项系统服务的可用性

●通常在软件的（18）活动中无需用户参与．

(18)A.需求分析 Ｂ.维护 Ｃ.编码　Ｄ．测试

●进行软件项目的风险分析时,风险避免、风险监控和风险管理及意外事件计划是 (19) 活动中需要考虑的问题.

(19) A.风险识别　 Ｂ.风险预测　Ｃ.风险评估 Ｄ.风险控制

●统一过程（UP）是一种用例驱动的迭代式增量开发过程，每次迭代中主要的工程流包括捕获需求，分析，设计，实现和测试等。这种软件过程的用例图（Use Case Diagram）是通过（29）得到的。

(29)A捕获需求 B分析 C设计 D实现

●序言性注释是指在每个程序或模块开头的一段说明，起辅助理解程序的作用，一般包括：程序的表示，名称和版本号，程序功能描述，接口于界面描述，输入，输出数据说明，开发历史，与运算环境有关的信息等。下列叙述中不属于序言性注释的是 30 。

(30)A.程序对硬件，软件资源的要求 B.重要变量和参数设计

C.嵌入在程序中的SQL语句 D.程序开发的原作者，审查者，修改者，编程日期等。

●软件（32）的提高，有利于软件可靠性的提高。

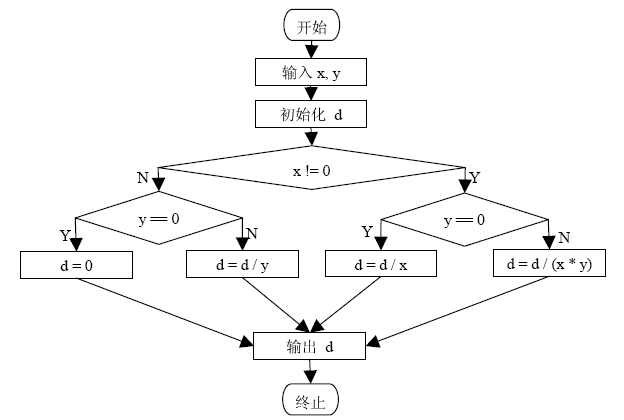
（32）A．存储效率 B．执行效率 C．容错性 D．可移植性

●LOC是软件规模的一种量度，它表示 34

(34)A.软件功能数 B.源代码行数 C.每单位成本数 D.工作量

## 2007年下半年

●阅读下列流程图



●当用判定覆盖法进行测试时，至少需要设计 35 个测试用例

（35）A．2 B.4 C.6 D.8

●在某大学学生学籍管理信息系统中，假设学生年龄的输入范围为16～40,则根据黑盒测试中的等价类划分技术，下面划分正确的是 37

(37)A.可划分为2个有效等价类，2个无效等价类 B.可划分为1个有效等价类，2个无效等价类

C.可划分为2个有效等价类，1个无效等价类 D.可划分为1个有效等价类，1个无效等价类

● 在软件开发中， （29） 不能用来描述项目开发的进度安排。在其他三种图中，可用 （30） 动态地反映项目开发进展情况。

（29）A. 甘特图 B. PERT图 C. PERT/CPM图 D. 鱼骨图（30）A. 甘特图 B. PERT图 C. PERT/CPM图 D. 鱼骨图

● 选择软件开发工具时，应考虑功能、 （31） 、稳健性、硬件要求和性能、服务和支持。

（31）A. 易用性 B. 易维护性 C. 可移植性 D. 可扩充性

● 内聚性和耦合性是度量软件模块独立性的重要准则，软件设计时应力求 （32） 。

（32）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

● 对于如下的程序流程，当采用语句覆盖法设计测试案例时，至少需要设计（34）个测试案例。



（34）A. 1 B. 2 C. 3 D. 4

● 为验证程序模块A是否正确实现了规定的功能，需要进行 （35） ；为验证模块A能否与其他模块按照规定方式正确工作，需要进行 （36） 。

（35）A. 单元测试 B. 集成测试 C. 确认测试 D. 系统测试（36）A. 单元测试 B. 集成测试 C. 确认测试 D. 系统测试

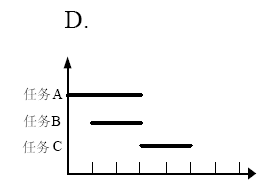
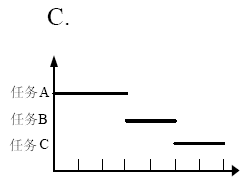
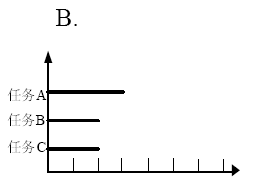
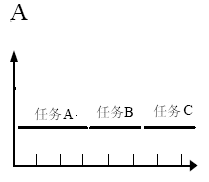
## 2008年上半年

●（15） 是一种面向数据流的开发方法，其基本思想是软件功能的分解和抽象。

（4）A . 结构化开发方法 B.Jackson 系统开发方法 C. Booch方法 D.UML（统一建模语言）

●某项目制定的开发计划中定义了3个任务，其中任务A首先开始，且需要3周完成，任务B必须在任务A启动1周后开始，且需要2周完成，任务C必须在任务A完成后开始，且需要2周完成。该项目的进度安排可用下面的甘特图（17）来描述。

（17）



●风险分析在软件项目开发中具有重要作用，包括风险识别、风险预测、风险评估和风险控制等。“建立风险条目检查表”是\_\_(18)\_\_\_\_时的活动，“描述风险的结果”是（19）时的活动。

（18）A.风险识别 B.风险预测 C.风险评估 D.风险控制（19）A.风险识别 B.风险预测 C.风险评估 D.风险控制

● 下列叙述中错误的是（22）.

（22）A. 面向对象程序设计语言可支持过程化的程序设计

B.给定算法的时间复杂性与实现该算法所采用的程序设计语言无关

C.与汇编语言相比，采用脚本语言编程可获得更高的运行效率

D.面向对象程序设计语言不支持对一个对象的成员变量进行直接访问

●某公司采用的软件开发过程通过了CMM2认证，表明该公司（30）。

（30）A.开发项目成效不稳定，管理混乱 B.对软件过程和产品质量建立了定量的质量目标

C.建立了基本的项目级管理制度和规程，可对项目的成本、进度进行跟踪和控制

D.可集中精力采用新技术新方法，优化软件过程

● 在软件设计和编码过程中，采取“\_（32）\_”的做法将使软件更容易理解和维护。

（32）A.良好的程序结构，有无文档均可 B.使用标准或规定之外的语句

C.编写详细正确的文档，采用良好的编程结构 D.尽量减少程序中的注释

●软件维护成本在软件成本中占较大比重。为降低维护的难度，可采取的措施有（33）。

（33）A.设计并实现没有错误的软件 B.限制可修改的范围 C.增加维护人员数量

D.在开发过程中就采取有利于维护的措施，并加强维护管理

●软件文档按照其产生和使用的范围可分为开发文档、管理文档和用户文档。其中开发文档不包括（34）。

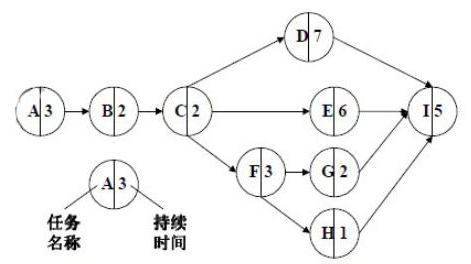
（34）A.软件需求说明 B.可行性研究报告 C.维护修改建议 D.项目开发计划

●软件测试是软件开发中不可缺少的活动，通常（35）在代码编写阶段进行。检查软件的功能是否与用户要求一致是（36）的任务。

（35）A.验收检测 B.系统检测 C.单元测试 D.集成测试（36）A.验收测试 B.系统测试 C.单元测试 D.集成测试

## 2008年下半年

●若一个项目由9个任务构成，其计划图（如下图所示）展示了任务之间的前后关系以及每个任务所需天数，该项目的关键路径是 （16） ，完成项目所需的最短时间是 （17） 天。



1. A. A→B→C→D→I B. A→B→C→E→I C. A→B→C→F→G→I D. A→B→C→F→H→I

（17） A. 16 B. 17 C. 18 D. 19

●RUP（Rational Unified Process）分为4个阶段，每个阶段结束时都有重要的里程碑，其中生命周期架构是在 （18） 结束时的里程碑。

（18）A.初启阶段 B.精化阶段 C.构建阶段 D.移交阶段

●在软件工程环境中进行风险识别时，常见的、已知的及可预测的风险类包括产品规模、商业影响等，与开发工具的可用性及质量相关的风险是 （19） 。

（19）A.客户特性 B.过程定义 C.开发环境 D.构建技术

●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初始级、可重复级、定义级、管理级和优化级。其中 （29） 对软件过程和产品都有定量的理解与控制。

（29）A.可重复级和定义级 B.定义级和管理级 C.管理级和优化级 D.定义级、管理级和优化级

●极限编程（XP）包含了策划、设计、编程和测试四个活动，其十二个最佳实践中的“持续集成”实践在 （30） 活动中进行。

(30）A.策划和设计 B.设计和编程 C.设计和测试 D. 编程和测试

● ISO/IEC 9126软件质量模型中第一层定义了六个质量特性，并为各质量特性定义了相应的质量子特性。子特性 （31） 属于可靠性质量特性。

（31）A.准确性 B.易理解性 C.成熟性 D. 易学性

● UP（统一过程）是用例驱动的、以架构为核心、迭代和增量的软件过程框架，它提供了一种 （32） 的特性。

（32）A.演进 B.敏捷 C.测试驱动 D.持续集成

●系统测试人员与系统开发人员需要通过文档进行沟通，系统测试人员应根据一系列文档对系统进行测试，然后将工作结果撰写成 （33） ，交给系统开发人员。

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

●系统的可维护性可以用系统的可维护性评系价统指测标试来衡量。系统的可维护性评价指标不包括 （34） 。

（34）A.可理解性 B.可修改性 C.准确性 D.可测试性

●某程序根据输入的三条线段长度，判断这三条线段能否构成三角形。以下6个测试中，（35） 两个用例属于同一个等价类。

①6、7、13； ②4、7、10； ③9、20、35； ④9、11、21； ⑤5、5、4； ⑥4、4、4。

（35）A.①② B.③④ C.⑤⑥ D. ①④

●在模拟环境下，常采用黑盒测试检验所开发的软件是否与需求规格说明书一致。其中有效性测试属于 （36） 中的一个步骤。

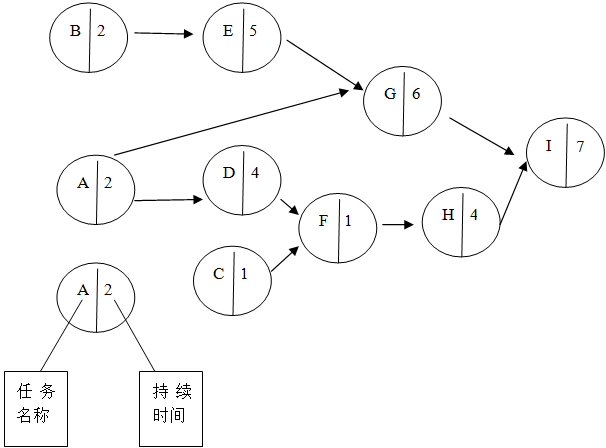
（36）A.单元测试 B.集成测试 C.确认测试 D. 系统测试

## 2009年上半年

●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（16）来表述“谁做”。

（16）A.角色 B.活动 C.制品 D. 工作流

●某项目主要由A～I任务构成，其计划图（如下图所示）展示了各任务之间的前后关系以及每个任务的工期（单位天），该项目的关键路径是（17），在不延误项目总工期的情况下，任务A最多可以推迟开始的时间是（18）天。



（17）A.A-->G-->I B.A-->D-->F-->H-->I C.B-->E-->G-->I D.C-->F-->H-->I

（18）A.0 B.2 C.5 D.7

● 软件风险一般包含（19）两个特性。

（19）A.救火和危机管理 B.已知风险和未知风险 C.不确定性和损失 D. 员工和预算

●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（29）。

(29)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

●RUP在每个阶段都有主要目标，并在结束时产生一些制品。在（30）结束时,产生 “在适当的平台上集成的软件产品”。

（30）A.初期阶段 B.精化阶段 C.构建阶段 D.移交阶段

●根据ISO/IEC 9126 软件质量度量模型定义，一个软件的时间和资源质量子特性属于（31）质量特性。

(31)A.功能 B.效率 C.可靠性 D.易使用性

●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（33）。

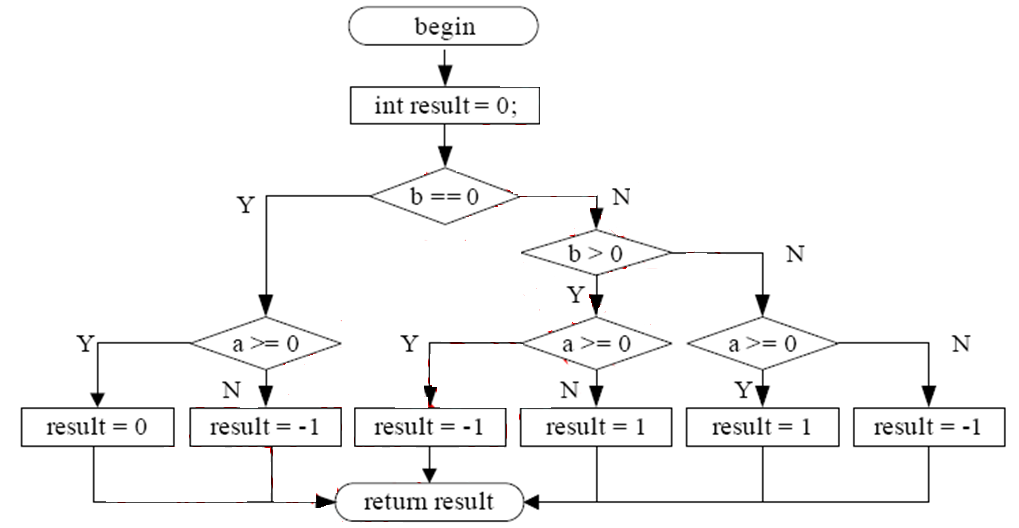
(33)A.系统开发合同 B.系统设计说明书 C.系统开发计划 D.系统测试报告

●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（34）出发，评价软件的结构和过程。

(34)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

● 当用分支覆盖法对以下流程图进行测试时，至少需要设计（35）个测试用例。



（35）A. 4 B. 5 C. 6 D. 8

●某银行为了使其网上银行系统能够支持信用卡多币种付款功能而进行扩充升级，这需要对数据类型稍微进行一些改变，这一状况需要对网上银行系统行进行（36）维护。

（36）A.正确性 B.适应性 C.完善性 D.预防性

## 2009年下半年

●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 （15） 来建立系统的逻辑模型，从而完成分析工作。

（15）A. ER 图 B. 数据流图 C. 程序流程图 D. 软件体系结构

●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， （16） 方法不属于面向对象方法。

(16)A.Booch B.Coad C.OMT D.Jackson

●确定构建软件系统所需要的人数时，无需考虑 （17）。

（17）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

●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。（18）最可能发现这一问题。

（18）A.单元测试 B.接受测试 C.回归测试 D.安装测试

●风险预测从两个方面评估风险，即风险发生的可能性以及 （19） 。

（19）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

●软件能力成熟度模型（CMM）的第4级（已管理级）的核心是（29）。

(29)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

●软件系统设计的主要目的是为系统制定蓝图，(30)并不是软件设计模型所关注的。

(30)A.系统总体结构 B.数据结构 C.界面模型 D.项目范围

●ISO/IEC 9126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（31）子特性。

（31）A.容错性 B.成熟性 C.易恢复性 D.易操作性

●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（33）和预算分配表等。

（33）A. PERT图 B.总体规划 C.测试计划 D.开发合同

●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（34）维护。

（34）A.正确性 B.适应性 C.完善性 D.预防性

●某系统重用了第三方组件（但无法获得其源代码），则应采用（35）对组件进行测试。

(35)A.基本路径覆盖 B.分支覆盖 C.环路覆盖 D.黑盒测试

●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、（36）。

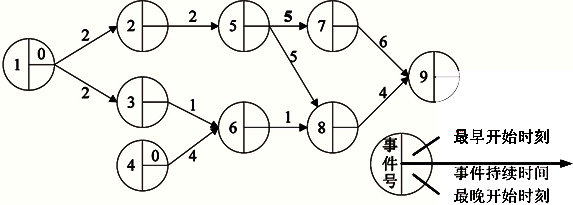
(36)A.好的计划 B.不断的发布 C.反馈和勇气 D.持续集成

## 2010年上半年

●使用白盒测试方法时，应根据（17）和指定的覆盖标准确定测试数据。

（17）A、程序内部逻辑 B、程序结构的复杂性 C、使用说明书 D、程序的功能

●进度安排的常用图形描述方法有Gantt图和PERT图。Gantt图不能清晰地描述（18）；PERT图可以给出哪些任务完成后才能开始另一些任务。下图所示的PERT图中，事件6的最晚开始时刻是（19）。



（18）A、每个任务从何时开始 B、每个任务到何时结束 C、每个任务的进展情况 D、各任务之间的依赖关系

（19）A、0 B、3 C、10 D、11

●对于一个大型软件来说，不加控制的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。 （29） 不属于配置数据库。

（29）A、开发库 B、受控库 C、信息库 D、产品库

●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 （30） 。

（30）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

●统一过程（UP）定义了初启阶段、精华阶段、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 （32） 的里程碑是生命周期架构。

（32）A、初启阶段 B、精华阶段 C、构建阶段 Ｄ、移交阶段

●程序的三种基本控制结构是 （33） 。

（33）A、过程、子程序和分程序 B、顺序、选择和重复 C、递归、堆栈和队列 D、调用、返回和跳转

● （34） 不属于软件配置管理的活动。

（34）A、变更标识 B、变更控制 C、质量控制 D、版本控制

●一个功能模块M1中的函数F1有一个参数需要接收指向整型的指针，但是在功能模块M2中调用F1时传递了一个整型值，在软件测试中， （35） 最可能测出这一问题。

（35）A、M1的单元测试 B、M2的单元测试 C、M1和M2的集成测试 D、确认测试

●确定系统边界和关系规范化分别在数据库设计的（51）阶段进行。

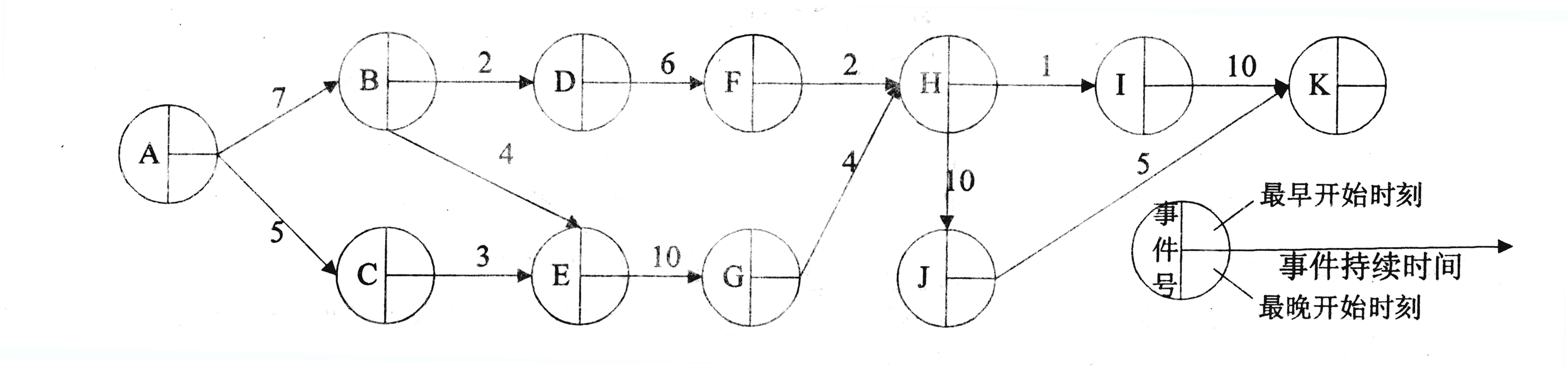
（51）A、需求分析和逻辑设计 B、需求分析和概念设计 C、需求分析和物理设计 D、逻辑设计和概念设计

## 2010年下半年

●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， （15） 最适合开发此项目。

（15）A.原型模型 B.瀑布模型 C.V模型 D.螺旋模型

●使用PERT图进行进度安排，不能清晰地描述 （16） ，但可以给出哪些任务完成后才能开始另一些任务。下面PERT图所示工程从A到中的关键路径是（17），（图中省略了任务的开始和结束时刻）。



（16）A.每个任务从何时开始 B.每个任务到何时结束 C.各任务之间的并行情况 D.各任务之间的依赖关系

（17）A.ABEGHIK B.ABEGHJK C.ACEGHIK D.ACEGHJK

●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（18）最佳实践。

（18）A.隐喻 B.重构 C.小型发布 D.持续集成

●在软件开发过程中进行风险分析时， （19）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。

（19）A .风险识别 B.风险预测 C.风险评估 D.风险控制

●以下关于过程改进的途述中，错误的是（30）。

（30）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、改进和监控四个部分

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

●软件复杂性度量的参数不包括（31）。

（31）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

●软件系统的可维护性评价指标不包括 （33） 。

（33）A.可理解性 B.可测试性 C.可扩展性 D.可修改性

●以下关于软件系统文档的途述中，错误的是 （34）。

（34）A.软件系统文档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

●以下关于软件测试的叙述中，正确的是 （35）。

（35）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

●不属于黑盒试技术的是（36）。

（36）A.错误猜测 B.逻辑覆盖 C.边界值分析 D.等价类划分

## 2011年上半年

● 模块A直接访问模块B的内部数据，则模块A和模块B的耦合类型为（16）。

（16）A. 数据耦合 B. 标记耦合 C. 公共耦合 D. 内容耦合

● 下列关于风险的叙述不正确的是：风险是指 （17） 。

（17）A. 可能发生的事件 B. 一定会发生的事件

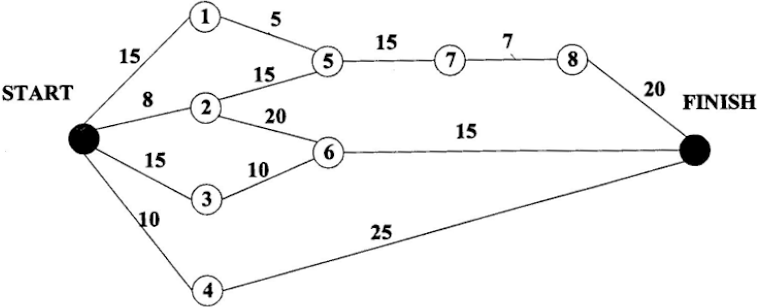
C. 会带来损失的事件 D. 可能对其进行干预，以减少损失的事件

● 下列关于项目估算方法的叙述不正确的是 （18）。

（18）A. 专家判断方法受到专家经验和主观性影响 B. 启发式方法（如COCOMO模型）的参数难以确定

C. 机器学习方法难以描述训练数据的特征和确定其相似性 D. 结合上述三种方法可以得到精确的估算结果

● 下图是一个软件项目的活动图，其中顶点表示项目里程碑，边表示包含的活动，边上的权重表示活动的持续时间，则里程碑（19）在关键路径上。



（19）A. 1 B. 2 C. 3 D. 4

● 为了有效地捕获系统需求，应采用 （29） 。

（29）A. 瀑布模型 B. V模型 C. 原型模型 D. 螺旋模型

● 关于过程改进，以下叙述中不正确的是 （30） 。

（30）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

● 软件产品的可靠性并不取决于（31） 。

（31）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

● 软件 （32）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。

（32）A. 可靠性 B. 可用性 C. 可维护性 D. 可伸缩性

● 高质量的文档所应具有特性中，不包括 （33） 。

（33）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

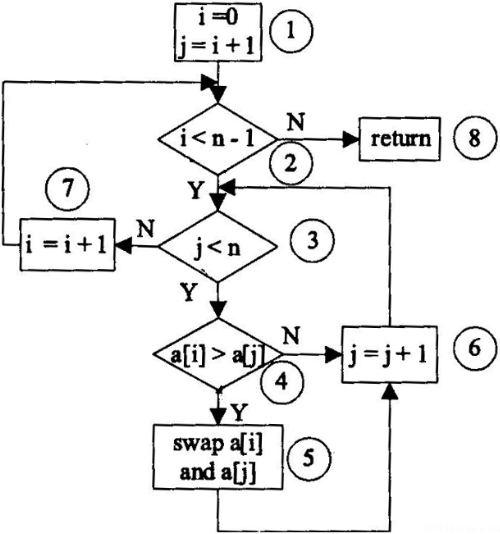
C. 完整性，任何文档都应当是完整的、独立的，应该自成体系

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\*\*文档\*\*节”的方式引用

● 在软件维护阶段，为软件的运行增加监控设施属于 （34） 维护。

（34）A. 改正性 B. 适应性 C. 完善性 D. 预防性

● 下图所示的逻辑流，最少需要 （35） 个测试用例可实现语句覆盖。



（35）A. 1 B. 2 C. 3 D. 5

● 在改正当前故障的同时可能会引入新的故障，这时需要进行 （36） 。

（36）A. 功能测试 B. 性能测试 C. 回归测试 D. 验收测试

## 2011年下半年

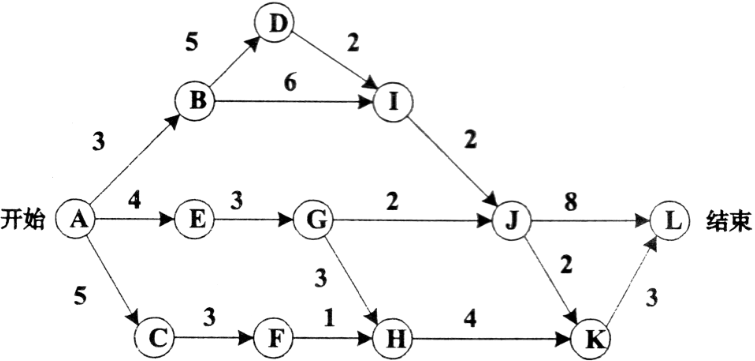
● 确定软件的模块划分及模块之间的调用关系是 （15） 阶段的任务。

（15）A. 需求分析 B. 概要设计 C. 详细设计 D. 编码

● 利用结构化分析模型进行接口设计时，应以 （16） 为依据。

（16）A. 数据流图 B. 实体—关系图 C. 数据字典 D. 状态—迁移图

● 下图是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的值表示完成活动所需要的时间，则关键路径长度为 （17） 。



（17）A. 20 B. 19 C. 17 D. 16

● 甘特图（Gantt图）不能 （18） 。

（18）A. 作为项目进度管理的一个工具 B. 清晰地描述每个任务的开始和截止时间

C. 清晰地获得任务并行进行的信息 D. 清晰地获得各任务之间的依赖关系

● 以下关于风险管理的叙述中，不正确的是 （19） 。

（19）A. 仅根据风险产生的后果来对风险排优先级

B. 可以通过改变系统性能或功能需求来避免某些风险

C. 不可能去除所有风险，但可以通过采取行动来降低或者减轻风险

D. 在项目开发过程中，需要定期地评估和管理风险

● 以下关于喷泉模型的叙述中，不正确的是 （29） 。

（29）A. 喷泉模型是以对象作为驱动的模型，适合于面向对象的开发方法

B. 喷泉模型克服了瀑布模型不支持软件重用和多项开发活动集成的局限性

C. 模型中的开发活动常常需要重复多次，在迭代过程中不断地完善软件系统

D. 各开发活动（如分析、设计和编码）之间存在明显的边界

● 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 （30） 进行开发。

（30）A. 瀑布模型 B. 演化模型 C. 螺旋模型 D. 原型模型

● 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（31）质量特性。

（31）A. 功能性 B. 可靠性 C. 可维护性 D. 易使用性

● 在软件评审中，设计质量是指设计的规格说明书符合用户的要求。设计质量的评审内容不包括 （32） 。

（32）A. 软件可靠性 B. 软件的可测试性

C. 软件性能实现情况 D. 模块层次

● 针对应用在运行期的数据特点，修改其排序算法使其更高效，属于（33）维护。

（33）A. 正确性 B. 适应性 C. 完善性 D. 预防性

● 进行防错性程序设计，可以有效地控制 （36） 维护成本。

（36）A. 正确性 B. 适应性 C. 完善性 D. 预防性

## 2012年上半年

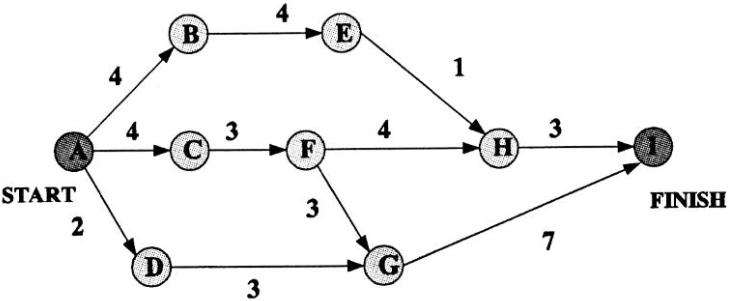
●数据流图(DFD)对系统的功能和功能之间的数据流进行建模，其中顶层数据流图描述了系统的 (15) 。

(15)A．处理过程 B．输入与输出 C．数据存储 D．数据实体

●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 (16) 内聚。

(16)A．顺序 B．过程 C．逻辑 D．功能

●下图是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的值表示完成活动所需要的时间，则 (17) 在关键路径上。



(17)A．B B．C C．D D．H

● (18) 最不适于采用无主程序员组的开发人员组织形式。

(18)A．项目开发人数少（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

●若软件项目组对风险采用主动的控制方法，则 (19) 是最好的风险控制策略。

(19)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

● 软件开发的增量模型 (29) 。

(29)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

● 假设某软件公司与客户签订合同开发一个软件系统，系统的功能有较清晰的定义，且客户对交付时间有严格要求，则该系统的开发最适宜采用 (30) 。

(30)A．瀑布模型 B．原型模型 C．V模型 D．螺旋模型

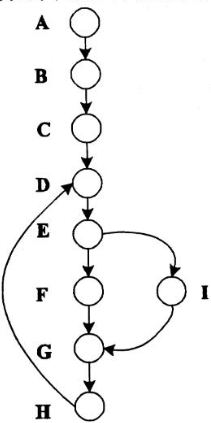
●某企业由于外部市场环境和管理需求的变化对现有软件系统提出新的需求，则对该软件系统进行的维护属于 (31) 维护。

(31)A．正确性 B．完善性 C．适应性 D．预防性

●McCall软件质量模型从软件产品的运行、修正和转移三个方面确定了11个质量特性，其中 (32) 不属于产品运行方面的质量特性。

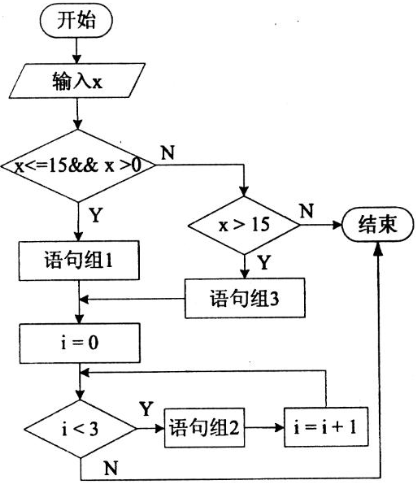
(32)A．正确性 B．可靠性 C．效率 D．灵活性

●采用McCabe度量法计算下列程序图的环路复杂性为 (33) 。



(33) A. 2 B. 3 C. 4 D. 5

●在白盒测试法中， (34) 是最弱的覆盖准则。下图至少需要 (35) 个测试用例才可以完成路径覆盖，语句组2不对变量i进行操作。



(34)A．语句 B．条件 C．判定 D．路径

(35)A．1 B．2 C．3 D．4

●根据ISO/IEC 9126软件质量模型中对软件质量特性的定义，可维护性质量特性的 (36) 子特性是指与为确认经修改软件所需努力有关的软件属性。

(36)A．易测试性 B．易分析性 C．稳定性 D．易改变性

## 2012年下半年

● 以下关于数据流图的叙述中，不正确的是 (15) 。

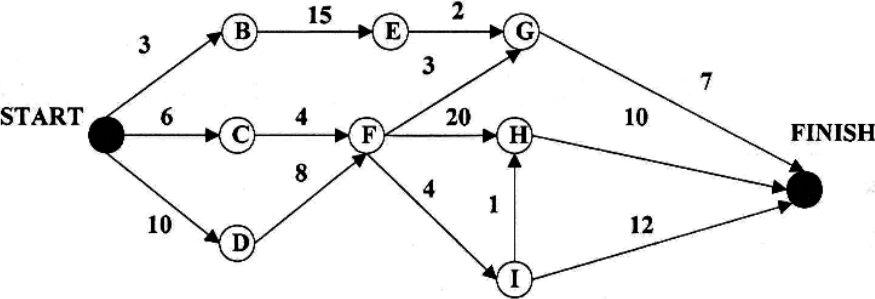
(15) A．每条数据流的起点或终点必须是加工 B．必须保持父图与子图平衡

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

● 某软件项目的活动图如下所示。图中顶点表示项目里程碑，连接顶点的边表示包含的活动，则里程碑(16)在关键路径上，活动FG的松弛时间为 (17) 。

(16) A．B B．C C．D D．I

(17) A．19 B．20 C．21 D．24



● 在软件设计阶段，划分模块的原则是：一个模块的(18)。

(18) 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

● 定义风险参照水准是 (19) 活动常用的技术。

(19) A．风险识别 B．风险预测 C．风险评估 D．风险控制

● 某开发小组欲开发一个规模较大、需求较明确的项目。开发小组对项目领域熟悉且该项目与小组开发过的某一项目相似，则适宜采用(29)开发过程模型。

(29) A．瀑布 B．演化 C．螺旋 D．喷泉

● 敏捷开发方法中， (30) 认为每一种不同的项目都需要一套不同的策略、约定和方法论。

(30) A．极限编程(XP) B．水晶法(Crystal) C．并列争球法( Scrum) D．自适应软件开发(ASD)

● 采用McCabe度量法计算下图的环路复杂性为 (31) 。



(31) A．2 B．3 C．4 D．5

● 在屏蔽软件错误的容错系统中，冗余附加技术的构成不包括 (32) 。

(32) A．关键程序和数据的冗余存储及调用 B．冗余备份程序的存储及调用

C．实现错误检测和错误恢复的程序 D．实现容错软件所需的固化程序

● 以下关于文档的叙述中，不正确的是 (33) 。

(33) 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

● 由于信用卡公司升级了其信用卡支付系统，导致超市的原有信息系统也需要做相应的修改工作，该类维护属于(34)。

(34) A．正确性维护 B．适应性维护 C．完善性维护 D．预防性维护

● 用白盒测试方法对下图所示的程序进行测试，设计了4个测试用例：①(x=0，y=3)、②(x=1，y=2)、③（x=-1，y=2）和④(x=3，y=1)。测试用例①②实现了 (35) 覆盖;若要完成路径覆盖，则可用测试用例 (36) 。



(35) A．语句 B．条件 C．判定 D．路径

(36) A．①② B．②③ C．①②③ D．①③④

## 2013年上半年

● 统一过程模型是一种“用例和风险驱动，以架构为中心，迭代并且增量”的开发过程，定义了不同阶段及其制品，其中精化阶段关注（15) 。

(15) A.项目的初创活动 B.需求分析和架构演进 C.系统的构建，产生实现模型 D.软件提交方面的工作，产生软件增量

● 在进行进度安排时，PERT图不能清晰地描述（16)，但可以给出哪些任务完成后才能开始另一些任务。某项目X包含任务A、B、……、J，其PERT图如下图所示（A=1表示任务A的持续时间是1天）,则项目X的关键路径是（17) 。



(16) A.每个任务从何时开始 B.每个任务到何时结束 C.各任务之间的并行情况 D.各任务之间的依赖关系

(17) A.A-D-H-J B.B-E-H-J C.B-F-J D.C-G-I-J

●“软件产品必须能够在3秒内对用户请求作出响应”属于软件需求中的（18)。

(18) A.功能需求 B.非功能需求 C.设计约束 D.逻辑需求

● 某项目为了修正一个错误而进行了修改。错误修正后，还需要进行（19)以发现这一修正是否引起原本正确运行的代码出错。

(19) A.单元测试 B.接受测试 C.安装测试 D.回归测试

● 在如下所示的数据流图中，共存在（29)个错误。



(29) A.4 B.6 C.8 D.9

● 软件的复杂性主要体现在程序的复杂性。（30)是度量软件复杂性的一个主要参数。若采用McCabe度量法计算环路复杂性，则对于下图所示的程序图，其环路复杂度为（31) 。



(30) A.代码行数 B.常量的数量 C.变量的数量 D.调用的库函数的数量

(31) A.2 B.3 C.4 D.5

● (32）不属于软件设计质量评审。

(32) A.功能与模块之间的对应关系 B.软件规格说明是否符合用户的要求

C.软件是否具有可测试性 D.软件是否具有良好的可靠性

● 在软件维护中，由于企业的外部市场环境和管理需求的变化而导致的维护工作，属于(33）维护。

(33) A.正确性 B.适应性 C.完善性 D.预防性

● 在对软件系统进行评价时，需要从信息系统的组成部分、评价对象和经济学角度出发进行综合考虑以建立起一套指标体系理论架构。从信息系统评价对象出发，对于用户方来说，他们所关心的是(34）。

(34) A.用户需求和运行质量 B.系统外部环境 C.系统内部结构 D.系统质量和技术水平

● 在设计测试用例时，应遵循（35）原则。

(35) A.仅确定测试用例的输入数据，无需考虑输出结果

B.只需检验程序是否执行了应有的功能，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

● 单元测试中，检查模块接口时，不需要考虑（36）。

(36) 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