第 7 章：软件工程基础知识　　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

考点脉络

软件工程是软件开发的指导思想，方法体系，所以这属于软件设计师需要掌握的核心内容。

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

（1）软件工程知识：软件生存周期与软件生存周期模型、软件开发方法、软件开发项目管理、软件开发工具与软件开发环境。

（2）系统分析基础知识：系统分析的主要步骤、机构化分析方法。

（3）系统设计基础知识：概要设计与详细设计的基本任务、系统设计的基本原理、系统模块结构设计、结构化设计方法、面向数据结构的设计方法、系统详细设计。

（4）系统实施基础知识：系统实施的基本内容、程序设计方法、程序设计的基本模块、系统测试、系统转换。

（5）系统运行和维护基础知识：系统可维护性的概念、系统维护的类型、系统评价的概念和类型

（6）软件质量管理基础知识：软件质量特性（ISO/IEC 9126软件质量模型）、软件质量保证、软件复杂性的概念及度量方法（McCabe度量法）、软件评审（设计质量评审、程序质量评审）、软件容错技术。

（7）软件过程改进基础知识：软件能力成熟度模型CMM、统一过程（UP）与极限编程（XP）的基本概念。

从历年的考试情况来看，本章的考点主要集中以下方面。

在软件生命周期与开发模型中，主要考查UP（统一过程）、XP（敏捷方法）以及传统软件开发模型的特点。

在系统开发方法论中，主要考查数据流图绘制原则、内聚与耦合。

在软件测试中，主要考查测试的阶段、白盒测试、黑盒测试、McCabe环路复杂度。

在软件质量保证中，主要考查软件质量特性。

在软件过程改进中，主要考查CMM每个阶段的特点及关键过程域

在项目管理中，主要考查项目管理相关基本概念、Pert图、风险管理。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

第 7 章：软件工程基础知识　　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

考点精讲

1．开发生命周期模型

系统开发的生命周期是指一个系统历经计划、分析、设计、编程、测试、维护直至淘汰的整个

过程。生命周期阶段的划分通常可以采用以下三种方法：

Boehm划分法：计划（问题定义、可行性研究）、开发（需求分析、总体设计、详细设计、编码、测试）、运行（维护）三大阶段。

国标（GB8566-1988）划分法：可行性研究与计划、需求分析、概念设计、详细设计、实现、组装测试、确认测试、使用和维护。，并在《GB/T8566-1995信息技术——软件生存期过程》中定义了获取过程、供应过程、开发过程、运行过程、维护过程、管理过程、支持过程七个部分。

RUP划分法：分为初始、细化、构造、移交四个主要阶段。

为了指导软件的开发，可以用不同的方式将软件生命周期中的所有开发活动组织起来，从而形成不同的软件开发模型。

2．开发模型

软件开发模型是几乎每次考试都会考查的重要知识点，在此，我们主要掌握各种开发模型的特点与其应用场合。

瀑布模型：严格遵循软件生命周期各阶段的固定顺序，一个阶段完成再进入另一个阶段。其优点是：可以使过程比较规范化，有利于评审；缺点在于：过于理想，缺乏灵活性，容易产生需求偏差。所以瀑布模型的应用场合为：需求明确的项目、二次开发项目以及与原型法配合使用。

快速原型模型：采用了一种动态定义需求的方法，通过快速地建立一个能够反映用户主要需求的软件原型，让用户在计算机上使用它，了解其概要，再根据反馈的结果进行修改，因此能够充分体现用户的参与和决策。原型化人员对原型的实施很重要，衡量他们的重要标准是能否从用户的模糊描述中快速地获取实际的需求。所以快速原型模型很好的弥补了瀑布模型的缺陷，它适合于需求不够明确的项目。

演化模型：也是一种原型化开发，但与快速原型不同的是，快速原型模型在获得真实需求时，就将抛弃原型。而演化模型则不然，它将从初始的模型中逐渐演化为最终软件产品，是一种“渐进式”原型法。其应用场合也是需求不明确的项目。

增量模型：它采用的是一种“递增式”模型，它将软件产品划分成为一系列的增量构件，分别进行设计、编码、集成和测试。相对于原型法而言，这种模型其实是从系统开发的另一个方面看待问题，原型法关注点是“制作一个原型”，而增量模型的关注点是“系统的功能模块不是一次完成的，而是一块一块开发，以增加的方式进行的”。在现实开发中，我们会发现，一个项目开发过程既用了原型模型也用了增量模型。所以增量模型仍有利于进行需求不明确的项目开发。

螺旋模型：结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。

喷泉模型：主要用于描述面向对象的开发过程，最核心的特点是迭代。所有的开发活动没有明显的边界，允许各种开发活动交叉进行。

UP：既是一个统一的软件开发过程，是一个通用过程框架，可以应付种类广泛的软件系统、不同的应用领域、不同的组织类型、不同的性能水平和不同的项目规模。UP是基于构件的，这意味着利用它开发的软件系统是由构件构成的，构件之间通过定义良好的接口相互联系。在准备软件系统所有蓝图的时候，UP使用的是统一建模语言UML。与其他软件过程相比，UP具有三个显著的特点：用例驱动、以基本架构为中心、迭代和增量。

UP中的软件过程在时间上被分解为四个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和交付阶段。每个阶段结束时都要安排一次技术评审，以确定这个阶段的目标是否已经满足。如果评

审结果令人满意，就可以允许项目进入下一个阶段。

UP一般用于大型软件的开发。

敏捷开发：从敏捷开发一词的敏捷可以看出，该方法是一种轻量级的开发方法。这种开发方法的主要思想是：传统的软件工程方法文档量太“重”了，现在需要进行减负，所以将不必要的文档都去掉，这就形成了敏捷开发。具体一点讲，敏捷方法包括：XP（极限编程）、自适应开发、水晶方法、特性驱动开发等。这些方法中，最著名的是XP方法（当然，XP方法最著名，并非因为方法本身很好，而是提出该方法的人，是个牛人）。

在XP方法中，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。十二个最佳实践：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、编码标准。

版权方授权希赛网发布，侵权必究

第 7 章：软件工程基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

一点一练

试题1

软件开发的增量模型__(1)__。

（1）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2

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__(2)__。

（2）A．瀑布模型　B．原型模型　C．V模型　D．螺旋模型

试题3

以下关于喷泉模型的叙述中，不正确的是__(3)__。

（3）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4

为了有效地捕获系统需求，应采用__(4)__。

（4）A．瀑布模型　B．V模型　C．原型模型　D．螺旋模型

试题5

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__(5)__最佳实践。

（5）A．隐喻　　B．重构　　C．小型发布　　D．持续集成

版权方授权希赛网发布，侵权必究

　　作者：希赛教育软考学院　来源：希赛网　　2014年05月05日

解析与答案

试题1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1答案

（1）B

试题2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2答案

（2）A

试题3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3答案

（3）D

试题4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4答案

（4）C

试题5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5答案

（5）C

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

系统开发方法论

系统开发方法包括：结构化方法、原型法、面向对象方法。其中原型法仅用于需求分析阶段，且在上一节已有描述，在此再赘述。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

考点精讲

1. 结构化方法

结构化方法是一种传统的软件开发方法，它是由结构化分析、结构化设计和结构化程序设计三部分有机组合而成的。它的基本思想：把一个复杂问题的求解过程分阶段进行，而且这种分解是自顶向下，逐层分解，使得每个阶段处理的问题都控制在人们容易理解和处理的范围内。

（1）结构化分析基础

结构化分析即我们平时所说的需求分析阶段，该阶段的主要任务是：解决“做什么”的问题。

结构化分析方法是以自顶向下，逐步求精为基点，以一系列经过实践的考验被认为是正确的原理和技术为支撑，以数据流图（“第11章数据流图技术”将详细论述相关内容），数据字典，结构化语言，判定表，判定树等图形表达为主要手段，强调开发方法的结构合理性和系统的结构合理性的软件分析方法。

（2）结构化设计基础

结构化设计是以结构化分析阶段所产生的成果为基础，进一步自顶而下、逐步求精和模块化的过程。其主要工作内容是进行概要设计与详细设计（注：在结构化方法诞生时，架构设计还未提出，所以在此没有架构设计相关内容）。

概要设计的主要任务是把需求分析得到的数据流图转换为软件结构和数据结构。设计软件结构的具体任务是：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等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。显然，概要设计建立的是目标系统的逻辑模型，与计算机无关。在概要设计过程中常常会使用结构图（包括模块、调用、数据）、层次图、HIPO（层次图加输入/处理/输出图）来描述程序的结构。

详细设计是对概要设计的一个细化，就是详细设计每个模块实现算法，所需的局部结构。经常使用的工具包括程序流程图、盒图、PAD图（问题分析图）、PDL（伪码）。

（3）模块设计原则

使用结构化设计方法进行设计时，需要遵循一定的原则，总结起来主要是两个方面：信息隐蔽与模块独立。

信息隐蔽指在设计和确定模块时，使得一个模块内包含信息（过程或数据），对于不需要这些信息的其他模块来说，是不能访问的。这样做的好处是：让所有的操作都通过标准的接口进行，这样避免了随意调用模块内部变量产生的混乱与错误。通过信息隐蔽可以提高软件的可修改性、可测试性和可移植性。所以不仅在结构化方法中有信息隐蔽原则，在面向对象方法中，也有相应机制进行信息隐蔽。

模块独立是指每个模块完成一个相对独立的特定子功能，并且与其他模块之间的联系最简单。保持模块的高度独立性，也是在设计时的一个很重要的原则。通常我们用耦合（模块之间联系的紧密程度）和内聚（模块内部各元素之间联系的紧密程度）两个标准来衡量，我们的目标是高内聚、低耦合。

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如表7-1所示。

表7-1模块的内聚类型

内聚类型	描述
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

而此相对应，模块的耦合类型通常也分为7种，根据耦合度从低到高排序如表7-2所示。

表7-2模块的耦合类型

耦合类型	描述
非直接耦合	没有直接联系，互相不依赖对方
数据耦合	借助参数表传递简单数据
标记耦合	一个数据结构的一部分借助于模块接口被传递
控制耦合	模块间传递的信息中包含用于控制模块内部逻辑的信息
外部耦合	与软件以外的环境有关
公共耦合	多个模块引用同一个全局数据区
内容耦合	一个模块访问另一个模块的内部数据 一个模块不通过正常入口转到另一模块的内部 两个模块有一部分程序代码重叠 一个模块有多个入口

除了满足以上两大基本原则之外，通常在模块分解时还需要注意：保持模块的大小适中；尽可能减少调用的深度；直接调用该模块的个数应该尽量大，但调用其他模块的个数则不宜过大；保证模块是单入口、单出口的；模块的作用域应该在之内；功能应该是可预测的。

2. 面向对象基本概念

面向对象开发方法与结构化开发有着本质的差异，这种方法引入了“对象”的概念，将数据和方法封装在一起，提高了模块的聚合度，降低了耦合度，更大程度上支持软件复用。是现在最流行和最具有发展前景的软件开发方法。在了解该方法的具体内容之前，需要对面向对象的一系列概念有所了解。

下面介绍考试中，经常会考到一系列面向对象的基本概念。

（1）对象

在计算机系统中，对象是指一组属性及这组属性上的专用操作的封装体，它由对象标识（名称）、属性（状态、数据、成员变量，也可以是另一个对象）和服务（操作、行为、方法）三个要素组成。对象是系统中用来描述客观事物的一个实体，，它们被封装为一个整体，以接口的形式对外提供服务。

（2）类

类是一组具有相同属性和方法的对象的集合。一个类中的每个对象都是这个类的一个实例（instance）。在系统分析和设计时，通常要把注意力集中在类上，而不是具体的对象上。每个类一般都有实例，没有实例的类是抽象类。抽象类不能被实例化（不能用new关键字去产生对象），抽象方法只需声明，而不需实现。是否建立了丰富的类库是衡量一个OO程序设计语言成熟与否的重要标志之一。

（3）继承与泛化

继承是面向对象方法中重要的概念，用来说明特殊类（子类）与一般类（父类）的关系，通常

使用泛化来说明一般类与特殊类之间的关系，它们之间是一对多关系。

（4）封装。面向对象系统中的封装单位是对象，对象之间只能通过接口进行信息交流，对象外部不能对对象中的数据随意地进行访问。封装的目的是使对象的定义和实现分离，这样，就能减少耦合，类内部的实现可以自由改变而不会影响其他的类或对象。同时，类具有严密的接口保护，使对象的属性或服务不会随意地被使用，对象的状态易于控制，可靠性随之增强。

（5）多态性

多态（多种形式）性则是指一般类中定义的属性或服务被特殊类继承后，可以具有不同的数据类型或表现出不同的行为，通常是使用重载和改写两项技术来实现的。

（6）模板类

模板类也称为类属类，它用来实现参数多态机制。一个类属类是关于一组类的一个特性抽象，它强调的是这些类的成员特征中与具体类型无关的那些部分，而用变元来表示与具体类型有关的那些部分。

（7）消息和消息通信

消息就是向对象发出的服务请求，它通常包括提供服务的对象标识、消息名、输入信息和回答信息。消息通信则是面向对象方法学中的一个重要原则，它与对象的封装原则密不可分，为对象间提供了唯一合法的动态联系的途径。

3. 面向对象开发方法

面向对象开发方法更接近于人类的自然思维。人类在认识和理解现实世界中普遍运用的三个构造法则是区分对象及其属性、区分整体对象及其组成部分、区分及形成不同对象类。而面向对象正是基于对象及属性、类属及成员、整体及其部分这些概念基础之上的。因而它必然更容易被理解和运用。面向对象的方法将对象的属性及服务视为一个整体。　　这更符合客观世界的规律，从而使其理解与实现起来更加容易，进一步减少维护的费用。

面向对象开发方法的发展经历了一个漫长的周期，先后提出了多个不同的方法。这些方法中，最有影响力的是：OMT方法、Coad/Yourdon方法、OOSE方法，而这些方法又进一步组合，形成了大家所熟知的UML（关于UML在后面的章节将详细论述）。

正是由于有多种不同的方法，所以它们强调的重点与方法的特色也各有不同，因此这里我们只对方法中具有代表性的或是都需要完成的一些内容进行论述。

（1）OOA

OOA即面向对象的分析，它的任务是了解问题域所涉及的对象、对象间的关系和操作，然后构造问题的对象模型。问题域是指一个包含现实世界事物与概念的领域，这些事物和概念与所设计的系统要解决的问题有关。在这个过程中，抽象是最本质和最重要的方法。针对不同的问题，可以选择不同的抽象层次，过简或过繁都会影响到对问题的本质属性的了解和解决。

（2）OOD

OOD即面向对象的设计，它在分析对象模型的基础上，设计各个对象、对象之间的关系（例如，层次关系、继承关系等）和通信方式（例如，消息模式）等，其主要作用是对OOA的结果作进一步的规范化整理，以便能够被OOP直接接受。

面向对象的设计需要遵循一系列的设计原则：

单一职责原则：设计目的单一的类；

开放-封闭原则：对扩展开放，对修改封闭；

李氏(Liskov)替换原则：子类可以替换父类；

依赖倒置原则：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程；

接口隔离原则：使用多个专门的接口比使用单一的总接口要好；

组合重用原则：要尽量使用组合，而不是继承关系达到重用目的；

迪米特(Demeter)原则(最少知识法则)：一个对象应当对其他对象有尽可能少的了解。

（3）OOP

OOP指系统功能的编码，实现在OOD阶段所规定的各个对象所应完成的任务。它包括每个对象的内部功能的实现，确立对象哪一些处理能力应在哪些类中进行描述，确定并实现系统的界面、输出的形式等。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

一点一练

试题1

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__(1)__内聚。

（1）A．顺序　B．过程　C．逻辑　D．功能

试题2

确定软件的模块划分及模块之间的调用关系是__(2)__阶段的任务。

（2）A．需求分析　B．概要设计　C．详细设计　D．编码

试题3

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__(3)__。

（3）A．数据耦合　B．标记耦合　C．公共耦合　D．内容耦合

试题4

采用面向对象开发方法时，对象是系统运行时基本实体。以下关于对象的叙述中，正确的是__(4)__。

（4）A．对象只能包括数据（属性）

B．对象只能包括操作（行为）

C．对象一定有相同的属性和行为

D．对象通常由对象名、属性和操作三个部分组成

试题5

面向对象分析的第一步是__(5)__。

（5）A．定义服务 B．确定附加的系统约束

C．确定问题域 D．定义类和对象

试题6

开-闭原则（Open-Closed Principle，OCP）是面向对象的可复用设计的基石。开-闭原则是指

一个软件实体应当对__(6)__开放，对__(7)__关闭；里氏替换原则（Liskov Substitution Principle，LSP ）是指任何__(8)__可以出现的地方，__(9)__一定可以出现。依赖倒转原则（Dependence Inversion Principle，DIP）就是要依赖于__(10)__而不依赖于__(11)__，或者说要针对接口编程，不要针对实现编程。

（6）A．修改　　B．扩展　　C．分析　　D．设计

（7）A．修改　　B．扩展　　C．分析　　D．设计

（8）A．变量　　B．常量　　C．基类对象　D．子类对象

（9）A．变量　　B．常量　　C．基类对象　D．子类对象

（10）A．程序设计语言　B．建模语言　C．实现　　D．抽象

（11）A．程序设计语言　B．建模语言　C．实现　　D．抽象

试题7

以下关于封装在软件复用中所充当的角色的叙述，正确的是__(12)__。

（12）A．封装使得其他开发人员不需要知道一个软件组件内部如何工作

B．封装使得软件组件更有效地工作

C．封装使得软件开发人员不简要编制开发文档

D．封装使得软件组件开发更加容易

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

第 7 章：软件工程基础知识　　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

解析与答案

试题1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1答案

（1）C

试题2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2答案

（2）B

试题3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3答案

（3）D

试题4分析

对象通常由对象名、属性和操作三个部分组成，对象不一定都有相同的属性和行为。

试题4答案

（4）D

试题5分析

面向对象分析的任务是了解问题域所涉及的对象、对象间的关系和操作，然后构造问题的对象模型。问题域是指一个包含现实世界事物与概念的领域，这些事物和概念与所设计的系统要解决的问题有关。因此面向对象分析的第一步是要确定问题域。

试题5答案

（5）C

试题6分析

开-闭原则要求一个软件实体应当对扩展开放，对修改关闭。也就是说，我们在设计一个模块的时候，应当使这个模块可以在不被修改的前提下被扩展，换句话说就是，应当可以在不必修改源代码的情况下改变这个模块的行为。

里氏代换原则要求子类型必须能够替换它们的基类型，所以在里氏代换原则中，任何可基类对象可以出现的地方，子类对象也一定可以出现。

依赖倒转原则是：要依赖于抽象，不要依赖于具体。也就是常说的要针对接口编程，不要针对实现编程。

试题6答案

（6）B（7）A（8）C（9）D（10）D（11）C

试题7分析

封装是面向对象技术的三大特点之一，封装的目的是使对象的定义和实现分离，这样，就能减少耦合。封装可以使得其他开发人员不需要知道一个软件组件内部是如何工作的，只需要使用该组件提供的接口来完成交互即可，如果在另外一个地方需要完成同样的功能，我们就可以将该组件使

用在另外一个地方，这样提供了软件的复用性。

试题7答案

（12）A

版权方授权希赛网发布，侵权必究

第 7 章：软件工程基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

软件测试与维护

软件测试与维护在每次考试中会考2-4分，考试试题以基本概念为主，本节将介绍相关内容。

版权方授权希赛网发布，侵权必究

第 7 章：软件工程基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

考点精讲

1．软件测试基础

软件测试是在规定的条件下对程序进行操作，以发现程序错误，衡量软件品质，并对其是否能满足设计要求进行评估的过程。测试的目的是发现尽可能多的缺陷。

为了让软件测试更有效率与效果，测试应遵循以下原则：

（1）尽早、不断的进行测试；

（2）程序员避免测试自己设计的程序；

（3）既要选择有效、合理的数据，也要选择无效、不合理的数据；

（4）修改后应进行回归测试；

（5）尚未发现的错误数量与该程序已发现错误数成正比。

2．测试阶段

软件测试按阶段划分，可分为：单元测试，集成测试，确认测试，系统测试。

（1）单元测试

单元测试，也称模块测试，通常可放在编程阶段，由程序员对自己编写的模块自行测试，检查模块是否实现了详细设计说明书中规定的功能和算法。单元测试主要发现编程和详细设计中产生的错误，单元测试计划应该在详细设计阶段制定。

单元测试期间着重从以下几个方面对模块进行测试：模块接口、局部数据结构、重要的执行通路、出错处理通路、边界条件等。

（2）集成测试

集成测试，也称组装测试，它是对由各模块组装而成的程序进行测试，主要目标是发现模块间的接口和通信问题。例如，数据穿过接口可能丢失；一个模块对另一个模块可能由于疏忽而造成有

害影响；把子功能组合起来可能不产生预期的主功能；个别看来是可以接受的误差可能积累到不能接受的程度；全程数据结构可能有问题等。集成测试主要发现设计阶段产生的错误，集成测试计划应该在概要设计阶段制定。

集成的方式可分为非渐增式和渐增式。

非渐增式集成是先测试所有的模块，然后一下子把所有这些模块集成到一起，并把庞大的程序作为一个整体来测试。这种测试方法的出发点是可以“一步到位”，但测试者面对众多的错误现象，往往难以分清哪些是“真正的”错误，哪些是由其他错误引起的“假性错误”，诊断定位和改正错误也十分困难。非渐增式集成只适合一些非常小的软件。

渐增式集成是将单元测试和集成测试合并到一起，它根据模块结构图，按某种次序选一个尚未测试的模块，把它同已经测试好的模块组合在一起进行测试，每次增加一个模块，直到所有模块被集成在程序中。这种测试方法比较容易定位和改正错误，目前在进行集成测试时已普遍采用渐增式集成。

（3）确认测试

确认测试（Validation Testing）主要依据软件需求说明书检查软件的功能、性能及其他特征是否与用户的需求一致。确认测试计划应该在需求分析阶段制定。

软件配置复查是确认测试的另一项重要内容。复查的目的是保证软件配置的所有成分都已齐全，质量符合要求，文档与程序完全一致，具有完成软件维护所必须的细节。

如果一个软件是为某个客户定制的，最后还要由该客户来实施验收测试（acceptance testing），以便确认其所有需求是否都已得到满足。由于软件系统的复杂性，在实际工作中，验收测试可能会持续到用户实际使用该软件之后的相当长的一段时间。

如果一个软件是作为产品被许多客户使用的，不可能也没必要由每个客户进行验收测试。绝大多数软件开发商都使用被称为α（Alpha）测试和β（Beta）测试的过程，来发现那些看起来只有最终用户才能发现的错误。

α测试由用户在开发者的场所进行，并且在开发者的指导下进行测试。开发者负责记录发现的错误和使用中遇到的问题。也就是说，α测试是在“受控的”环境中进行的。

β测试是在一个或多个用户的现场由该软件的最终用户实施的，开发者通常不在现场，用户负责记录发现的错误和使用中遇到的问题并把这些问题报告给开发者。也就是说，β测试是在“非受控的”环境中进行的。

经过确认测试之后的软件通常就可以交付使用了。

（4）系统测试

系统测试的对象是完整的、集成的计算机系统，系统测试的目的是在真实系统工作环境下，验证完整的软件配置项能否和系统正确连接，并满足系统/子系统设计文档和软件开发合同规定的要求。系统测试的技术依据是用户需求或开发合同，除应满足一般测试的准入条件外，在进行系统测试前，还应确认被测系统的所有配置项已通过测试，对需要固化运行的软件还应提供固件。

一般来说，系统测试的主要内容包括功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、安装与反安装测试等，其中，最重要的工作是进行功能测试与性能测试。功能测试主要采用黑盒测试方法，性能测试主要验证软件系统在承担一定负载的情况下所表现出来的特性是否符合客户的需要，主要指标有响应时间、吞吐量、并发用户数和资源利用率等。

3．白盒测试

白盒测试，又称透明盒测试、结构测试等，软件测试的主要方法之一。测试应用程序的内部结构或运作，而不是测试应用程序的功能（即黑盒测试）。在白盒测试时，以编程语言的角度来设计测试案例。测试者了解待测试程序的内部结构、算法等信息，这是从程序设计者的角度对程序进行的测试。

白盒测试可以应用于单元测试、集成测试和系统的软件测试流程，可测试在集成过程中每一单元之间的路径，或者主系统跟子系统中的测试。

白盒测试主要是从覆盖源程序语句的详尽程度来分析。逻辑覆盖标准包括以下不同的覆盖标准：语句覆盖、判定覆盖、条件覆盖、条件判定组合覆盖、多条件覆盖、路径覆盖。

语句覆盖：为了暴露程序中的错误，程序中的每条语句至少应该执行一次。该方法会选择足够多的测试数据，使被测程序中每条语句至少执行一次。语句覆盖是很弱的逻辑覆盖。

判定覆盖：比语句覆盖稍强的覆盖标准是判定覆盖。判定覆盖的含义是：设计足够的测试用例，使得程序中的每个判定至少都获得一次“真值”或“假值”，或者说使得程序中的每一个取“真”分支和取“假”分支至少经历一次，因此判定覆盖又称为分支覆盖。

条件覆盖：在设计程序中，一个判定语句是由多个条件组合而成的复合判定。为了更彻底地实现逻辑覆盖，可以采用条件覆盖的标准。条件覆盖的含义是：构造一组测试用例，使得每一判定语句中每个逻辑条件的可能值至少满足一次。

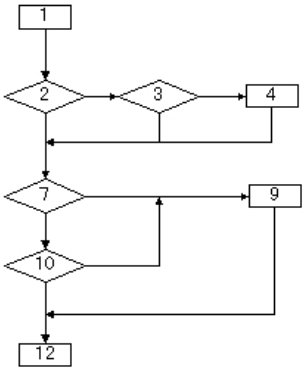
多条件覆盖：多条件覆盖也称条件组合覆盖，它的含义是：设计足够的测试用例，使得每个判定中条件的各种可能组合都至少出现一次。显然满足多条件覆盖的测试用例是一定满足判定覆盖、条件覆盖和条件判定组合覆盖的。

路径覆盖：最全面的覆盖。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）。

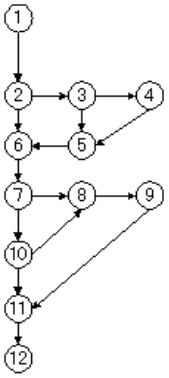
4．McCabe复杂度

McCabe复杂度属于白盒测试技术。McCabe复杂度包括环路复杂度（Cyclomatic complexity）、基本复杂度、模块设计复杂度、设计复杂度和集成复杂度等。在软件设计师的考试中，要求掌握环路复杂度。

McCabe度量标准是将程序的流程图转化为有向图，也就是控制流图，然后以图论的知识和计算方法来衡量软件的质量。如图7-1（a）对于应控制流程图是7-1（b）。



（a）程序流程图



（b）控制流程图

图7-1程序流程图和对应的控制流图

将程序流程图转换成控制流程图以后，就可以计算其环路复杂度了。

计算有向图G的环路复杂度公式为：

V(G)=m-n+2

说明：其中V(G)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。

在图7-1（b）中，有向弧数为15，结点数为12，所以环路复杂度为：15-12+2=5。

5．黑盒测试

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

常用的黑盒测试法包括：等价类划分、边值分析、错误推测和因果图。

（1）等价类划分

在设计测试用例时，等价类划分是用得最多的一种黑盒测试方法。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；反之，如果等价类中的一个输入数据不能检测出某个错误，那么等价类中的其他输入数据也不能检测出这一错误（除非这个等价类的某个子集还属于另一等价类）。

如果一个等价类内的数据是符合（软件需求说明书）要求的、合理的数据，则称这个等价类为有效等价类。有效等价类主要用来检验软件是否实现了软件需求说明书中规定的功能。

如果一个等价类内的数据是不符合（软件需求说明书）要求的、不合理或非法的数据，则称这个等价类为无效等价类。无效等价类主要用来检验软件的容错性。

黑盒测试中，利用等价类划分方法设计测试用例的步骤如下。

①根据软件的功能说明，对每一个输入条件确定若干个有效等价类和若干个无效等价类，并为每个有效等价类和无效等价类编号。

②设计一个测试用例，使其覆盖尽可能多的尚未被覆盖的有效等价类。重复这一步，直至所有的有效等价类均被覆盖。

③设计一个测试用例，使其覆盖一个尚未被覆盖的无效等价类。重复这一步，直至所有的无效等价类均被覆盖。

应当特别注意，无效等价类用来测试非正常的输入数据，因此每个无效等价类都有可能查出软

件中的错误，所以要为每个无效等价类设计一个测试用例。

（2）边值分析

经验表明，软件在处理边界情况时最容易出错。设计一些测试用例，使软件恰好运行在边界附近，暴露出软件错误的可能性会更大一些。

通常，每一个等价类的边界，都应该着重测试，选取的测试数据应该恰好等于、稍小于或稍大于边界值。

将等价类划分法和边值分析法结合使用，更有可能发现软件中的错误。

（3）错误推测

使用等价类划分和边值分析技术，有助于设计出具有代表性的、容易暴露软件错误的测试方案。但是，不同类型不同特征的软件通常又有一些特殊的容易出错的地方。错误推测法主要依靠测试人员的经验和直觉，从各种可能的测试方案中选出一些最可能引起程序出错的方案。

（4）因果图

因果图法是根据输入条件与输出结果之间的因果关系来设计测试用例的，它首先检查输入条件的各种组合情况，并找出输出结果对输入条件的依赖关系，然后为每种输出条件的组合设计测试用例。

6．软件维护

软件维护主要是指根据需求变化或硬件环境的变化对应用程序进行部分或全部的修改。在修改时应充分利用源程序，修改后应填写程序改登记表，并在程序变更通知书上写明新旧程序的不同之处。

软件维护是整个软件生命周期中最长的一个阶段，同时他也花费了整个软件生命周期中最多的成本，20世纪80年代末用于软件维护的花费约为整个软件生命周期总花费的75%，而且还在逐年上升。

软件维护通常可分为以下四种类型：

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尚未发现的错误。这方面的维护工作量要占整个维护工作量的17%～21%。所发现的错误有的不太重要，不影响系统的正常运行，其维护工作可随时进行；而有的错误非常重要，甚至影响整个系统的正常运行，其维护工作必须制定计划，进行修改，并且要进行复查和控制。

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。这方面的维护工作量占整个维护工作量的18%～25%。由于目前计算机硬件价格的不断下降，各类系统软件层出不穷，人们常常为改善系统硬件环境和运行环境而产生系统更新换代的需求；企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。这些因素都将导致适应性维护工作的产生。进行这方面的维护工作也要像系统开发一样，有计划、有步骤地进行。

完善性维护：这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。这些功能对完善系统功能是非常必要的。另外，还包括对处理效率和编写程序的改进，这方面的维护占整个维护工作的50%～60%，比重最大。

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。例如：目前的网络带宽够用，但新系统一旦上线，将面临带宽不够的风险，此时进行带宽的扩容就是一种预防性维护。这方

面的维护工作量占整个维护工作量的4%左右。

在软件维护这个主题中，另一个值得注意的方面是：软件的可维护性问题。软件的可维护性是指理解、改正、改动、改进软件的难易程度。根据Boehm质量模型，通常影响软件可维护性的因素有可理解性、可测试性和可修改性。

（1）可理解性

可理解性是指维护人员理解软件的结构、接口、功能和内部过程的难易程度。

（2）可测试性

可测试性是指测试和诊断软件错误的难易程度。

（3）可修改性

可修改性是指修改软件的难易程度。

为了提高软件的可维护性，在软件生命周期的各个阶段都必须充分考虑维护问题。先进的软件工程方法是软件可维护的基础保证。

面向对象方法学的对象封闭机制、消息通信机制、继承机制和多态机制从根本上提高了软件的可理解性、可测试性和可修改性。

结构化设计的几条主要原则，如模块化、信息隐蔽、高内聚、低耦合等，对于提高软件的可理解性、可测试性和可修改性也都有重要的作用。

另外，书写详细正确的文档、书写源文件的内部注解、使用良好的编程语言、具有良好的程序设计风格，也有助于提高软件的可理解性。使用先进的测试工具、保存以前的测试过程和测试用例，则有助于提高软件的可测试性。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

一点一练

试题1

在白盒测试法中，__(1)__是最弱的覆盖准则。图7-2至少需要__(2)__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。

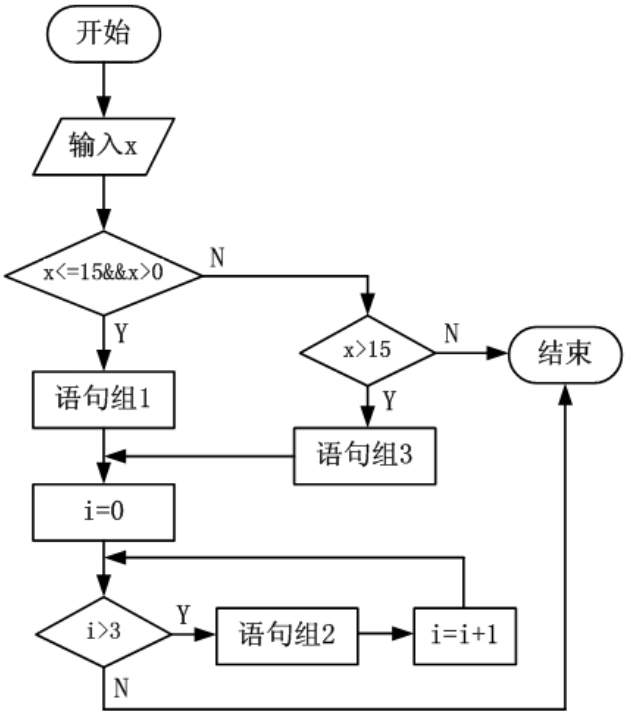


图7-2程序流程图

（1）A．语句　B．条件　C．判定　D．路径

（2）A．1　B．2　C．3　D．4

试题2

不属于黑盒测试技术的是__(3)__。

（3）A．错误猜测　B．逻辑覆盖　C．边界值分析　D．等价类划分

试题3

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__(4)__不是好的测试用例。

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

试题4

如图7-3所示的逻辑流，最少需要__(5)__个测试用例可实现语句覆盖。

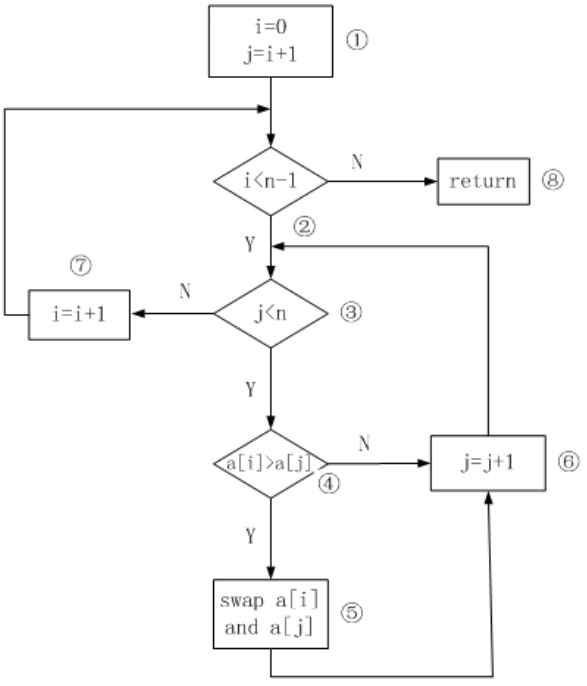


图7-3程序流程图

（5）A．1　　B．2　　C．3　　D．5

试题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__(6)__。

（6）A．功能测试　　B．性能测试　　C．回归测试　　D．验收测试

试题6

采用McCabe度量法计算图7-4的环路复杂性为__(7)__。

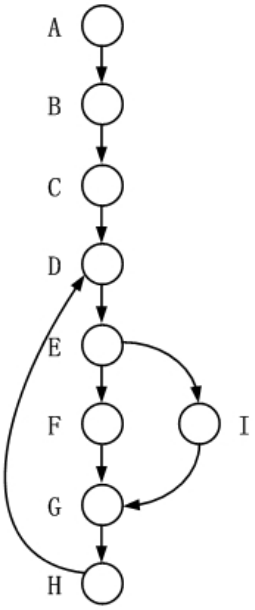


图7-4环路复杂性图

（7）A．2　B．3　C．4　D．5

试题7

以下关于软件测试的叙述中，正确的是__(8)__。

（8）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8

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__(9)__维护。

（9）A．正确性　B．完善性　C．适应性　D．预防性

试题9

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__(10)__维护。

（10）A．正确性　B．适应性　C．完善性　D．预防性

试题10

软件系统的可维护性评价指标不包括__(11)__。

（11）A．可理解性　　B．可测试性　　C．扩展性　　D．可修改性

版权方授权希赛网发布，侵权必究

 作者：希赛教育软考学院 来源：希赛网 2014年05月05日

解析与答案

试题1分析

关于逻辑覆盖相关知识请参看本节考点精讲部分。

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1答案

（1）A（2）C

试题2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2答案

（3）B

试题3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3答案

（4）D

试题4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4答案

（5）A

试题5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5答案

（6）C

试题6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6答案

（7）B

试题7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7答案

（8）C

试题8分析

本题考查软件维护的类型，关于软件维护类型详细概念请参看本节考点精讲部分。

试题8答案

（9）C

试题9分析

本题考查软件维护的类型，关于软件维护类型详细概念请参看本节考点精讲部分。

通过对概念的学习，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9答案

（10）C

试题10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10答案

（11）C

版权方授权希赛网发布，侵权必究

第 7 章：软件工程基础知识 作者：希赛教育软考学院 来源：希赛网 2014年05月05日

软件质量保证与软件过程改进

软件质量就是软件与显性与隐性需求相一致的程度。

软件质量保证，就是保证软件产品充分满足消费者要求的质量而进行的有计划、有组织的活动。它主要包括质量方针的制定和展开、质量保证方针和质量保证标准的制定、质量保证体系的建立和管理、明确各阶段的质量保证工作、各阶段的质量评审、确保设计质量、重要质量问题的提出与分析、总结实现阶段的质量保证活动、整理面向用户的文档和说明书、产品质量和质量保证系统的鉴定、质量信息的收集分析及使用。而软件过程改进同样要靠标准作为指导方针，所以在本节将介绍到很多与质量相关的标准。

版权方授权希赛网发布，侵权必究

第 7 章：软件工程基础知识 作者：希赛教育软考学院 来源：希赛网 2014年05月05日

考点精讲

1．软件质量特性标准

为了能够统一地描述软件质量特性，形成了许多质量特性标准，其中最常用的有国际通用的ISO/IEC 9126软件质量模型和Mc Call软件质量模型。

（1）IEO/IEC 9126模型

IEO/IEC 9126模型是考试当中经常考查的一个知识点，考查方式主要是在选择题中，选出一个质量属性类中不属于（或属于）这一类的子特性，所以要求考生掌握此标准中的质量属性分类与其子特性。值得注意的是IEO/IEC 9126模型等价于我国的国家标准《GB/T 16120—1996 软件产品评价、质量特性及其使用指南》。

表7-3《IEO/IEC 9126模型》

类	子特性	说明
功能性	与功能及其指定的性质有关的一组软件质量	
	适合性	规定任务提供一组功能的能力，以及这组功能的适合程度
功能性	准确性	系统满足需求规格说明和用户目标的程度
	互操作性	同其他指定系统的协同工作能力
	依从性	软件服从有关标准、约定、法规及类似规定的程度
	安全性	避免对程序及数据的非授权故意或意外访问的能力
可靠性	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	
	成熟性	由软件故障引起失效的频度
	容错性	在错误或违反指定接口情况下维护指定性能水平的能力
	易恢复性	在故障发生后重新建立性能水平，恢复数据的能力和时间
易使用性	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	
	易理解性	用户理解该软件系统的难易程度
	易学习性	用户学习使用该软件系统的难易程度
	易操作性	用户操作该软件系统的难易程度
效率	衡量在规定条件下软件的性能水平和所用资源量之间的关系的属性	
	时间特性	响应和处理时间及软件执行其功能时的吞吐量
	资源特性	软件执行其功能时，所使用的资源量及持续使用时间
可维护性	与软件维护的难易程度相关的一组软件属性	
	易分析性	诊断缺陷或失效原因、判定待修改程度的难易程度
	易更改性	修改、排错或适应环境变化的难易程度
	稳定性	修改造成难以预料的后果的风险程度
	易测试性	测试已修改软件的难易程度
可移植性	与从某一环境转移到另一环境的能力有关的属性	
	适应性	软件无须采用特殊处理就能够适应不同规定环境程度
	易安装性	在指定环境下安装软件的难易程度
	一致性	软件服从与可移植性有关的标准或约束的程度
	易替换性	软件在特定软件环境中用来替代指定的其他软件的可能性和难易程度

（2）McCall质量模型

McCall质量模型从软件运行、软件修改和软件转移三个方面来考查软件的质量，其体系如图7-5所示。

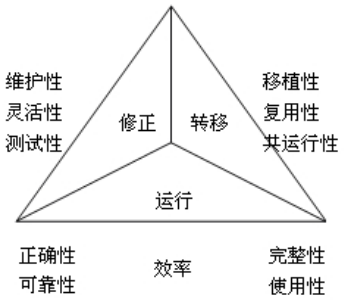


图7-5 McCall质量模型体系

表7-4 Mc Call质量模型体系

类别	质量特性	含义	直观描述
运行	正确性	程序能够满足规格说明和完成用户业务目标的程度	它做了该做的事吗
	可靠性	程序能够按要求的精确度实现其预期功能的程度	它总能准确工作吗
	效率	程序实现其功能所需要的计算资源量	需要资源多吗
	完整性	软件或数据不受未授权人控制的程度	它是安全的吗
	使用性	学习、操作程序、为其准备输入数据、解释其输出的工作量	它可用吗
修正	维护性	对运行的程序找到错误并排错的工作量	它可调整吗
	测试性	为保证程序执行规定功能所需的测试工作量	它可测试吗
	灵活性	修改运行的程序所需的工作量	它可修改吗
转移	移植性	将程序从一种硬件硬置和/或环境转移到另一硬件配置和/或环境所需工作量	可以在另一台机器上使用它吗
	复用性	程序可被用于与其实现功能相关的其他应用问题的程度	可以重复使用它的某些部分吗
	共运行性	让系统与另一系统协同运行所需的工作量	是否能与外系统连接

2．软件技术评审

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。它通常会采用系统化、严密的过程，包括制订计划、总体会议、做准备、开会、返工、追踪和因果分析。应培训审查者，依赖于缺陷检查表和其他错误分析技术；不是由作者来主持主审，而是由主持人陈述。

制订计划：确定谁参加，一般需要将人数控制在7人之内；确定准备哪些内容。

总体会议：确定评审的背景、假设及目标。

做准备：评审员预先阅读评审的材料。

评审会议：主持人引导，根据预先准备的检查表进行评审。

返工：评审会议是为了发现问题，问题应在会后进行返工。

跟踪：确定错误已经修改，并通知所有的评审人。

3．能力成熟度模型集成（CMMI）

能力成熟度模型集成融合了多种模型，形成了组织范围内过程改进的单一集成模型，其主要目的是消除不同模型之间的不一致和重复，降低基于模型进行改进的成本。CMMI有两种表示法：阶段表示法和连续式表示法。这两种表示方法各有优缺点，均采用统一的24个过程域，它们在逻辑上是等价的，对同一个组织采用两种模型分别进行CMMI评估，得到的结论应该是相同的。但在考试时，往往涉及的是阶段表示法，所以在此仅对阶段表示法进行说明。

阶段式模型基本沿袭CMM模型框架，仍保持五个成熟等级，但关键过程域做了一些调整和扩充，如表7-5所示。

表7-5过程域的阶段式分组

成熟度等级	过程域
可重复级（二级）	需求管理、项目计划、配置管理、项目监督与控制、供应商合同管理、度量和分析、过程和产品质量保证
已定义级（三级）	需求开发、技术解决方案、产品集成、验证、确认、组织级过程焦点、组织级过程定义、组织级培训、集成项目管理、风险管理、集成化的团队、决策分析和解决方案、组织级集成环境
已管理级（四级）	组织级过程性能、定量项目管理
优化级（五级）	组织级改革与实施、因果分析和解决方案

当组织通过了某一等级过程域中的全部过程，即意味着该组织的成熟度达到了这一等级。利用阶段式模型对组织进行成熟度度量，概念清晰、易于理解、便于操作。

版权方授权希赛网发布，侵权必究

　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

一点一练

试题1

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__(1)__不属于产品运行方面的质量特性。

（1）A．正确性　B．可靠性　C．效率　D．灵活性

试题2

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__(2)__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。

（2）A．易测试性 B．易分析性

C．稳定性 D．易改变性

试题3

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__(3)__质量特性。

（3）A．功能性　B．可靠性　C．可维护性　D．易使用性

试题4

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__(4)__。

（4）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5

关于过程改进，以下叙述中不正确的是__(5)__。

（5）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6

软件产品的可靠性并不取决于__(6)__。

（6）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

试题7

软件__(7)__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。

（7）A．可靠性　B．可用性　C．可维护性　D．可伸缩性

试题8

高质量的文档所应具有的特性中，不包括__(8)__。

（8）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

版权方授权希赛网发布，侵权必究

解析与答案

试题1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1答案

（1）D

试题2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2答案

（2）A

试题3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3答案

（3）A

试题4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4答案

（4）D

试题5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5答案

（5）D

试题6分析

软件产品的可靠性可描述为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，它主要体现在软件的成熟性、容错性和易恢复性方面。其中成熟性描述的是由软件故障引起软件失效的频度；容错性描述的是在错误或违反指定接口情况下维护指定性能水平的能力；易恢复性描述的是在故障发生后重新建立性能水平，恢复数据的能力和时间。软件产品的可靠性不取决于软件产品的使用方式。

试题6答案

（6）C

试题7分析

软件的可靠性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。

软件的可用性是指软件在特定使用环境下为特定用户用于特定用途时所具有的有效性。

软件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。

软件的可伸缩性是指是否可以通过运行更多的实例或者采用分布式处理来支持更多的用户。

试题7答案

（7）A

试题8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8答案

（8）D

版权方授权希赛网发布，侵权必究

软件项目管理

软件项目本身是复杂的，如果没有仔细地计划，复杂的项目是不可能成功的。一个计划良好的项目将受到有效的控制，进展明显，而参加该项目的人员都会得到支持以进行其工作。软件项目本身也具有风险性，如果没有有效的风险管理也是不能成功的。

通常软件工程项目的管理比其他工程项目的管理更困难，这是因为：

软件产品不可见。开发的进度及产品的质量是否符合要求不明显，比较难进行把握。

没有标准的软件过程。尽管近几年来“软件过程改进”领域有许多进步，但由于团队、人员的个性化因素，还不存在一个放之四海皆真理的标准化软件过程。

大型软件项目常常是一次性项目。由于这些项目都是“前无古人”的，因此缺乏可以借鉴的历史经验。

在软件设计师考试中，项目管理知识考查的分值较少，同时知识点比较集中，所以本节不会面面俱到的介绍项目管理知识，而是取了几个知识点来进行说明。主要包括网络计划技术中的一些计算问题以及项目管理相关的一些基本概念。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

第 7 章：软件工程基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

考点精讲

1. 项目管理九大知识领域

在PMBOK中提出了项目管理九大知识领域，此后这九大知识领域为业界普遍认同。九大知识领域中有四大核心领域：范围管理、时间管理、成本管理、质量管理，四大辅助领域：人力资源、沟通管理、风险管理、采购管理以及负责整体协调的整体管理。

整体管理：整体管理负责项目的全生命周期管理、全局性管理和综合性管理。全生命周期管理意味着项目整体管理过程负责管理项目的启动阶段直到项目收尾阶段的整个项目生命周期。全局性管理意味着项目整体管理过程负责管理项目的整体包括项目管理工作、技术工作和商务工作等。综合性管理意味着项目整体管理过程负责管理项目的需求、范围、进度、成本质量、人力资源、沟通、风险和采购。

范围管理：范围管理确定在项目内包括什么工作和不包括什么工作，由此界定的项目范围在项目的全生命周期内可能因种种原因而变化，项目范围管理也要管理项目范围的这种变化。

时间管理：时间管理又称进度管理，时间管理的职责是保证项目的所有工作都在一个指定的时间内完成。

成本管理：成本管理就是要确保在批准的预算内完成项目。

质量管理：质量管理是指确定质量方针、目标和职责，并通过质量体系中的质量计划、质量控制、质量保证和质量改进来使其实现的所有管理职能的全部活动。

人力资源管理：人力资源管理是为项目正常开展，而进行的招聘（选择）项目组人员并进行有效组织、考核绩效支付报酬、进行有效激励、结合组织与个人需要进行有效开发以便实现最优组织

绩效的全过程。

沟通管理：沟通管理是指项目组为项目干系人之间进行良好沟通而进行的管理活动。其内容包括沟通计划编制、信息分发、绩效报告、项目干系人管理。

风险管理：所谓风险管理，就是要在风险成为影响项目成功的威胁之前，识别、着手处理并消除风险的源头。

采购管理：采购管理是为完成项目工作，从项目团队外部购买或获取所需的产品、服务或成果的过程。

2. 网络计划技术

用网络分析的方法编制的计划称为网络计划，它是一种编制大型工程项目进度计划的有效方法。计划借助于网络表示各项工作与所需要的时间，以及各项工作的相互关系。通过网络分析研究工程费用与工期的关系，并找出在编制计划及计划执行过程中的关键路径，这种方法称为关键路径法（Critical Path Method，CPM）。还有一种方法，也是应用网络分析方法与网络计划，但它注重于对各项工作安排的评价和审查，这种方法称为计划评审技术（Program Evaluation and Review Technique，PERT）。

（1）关键路径

在现代管理中，人们常用有向图来描述和分析一项工程的计划和实施过程，一项工程常被分为多个小的子工程，这些子工程被称为活动。在有向图中，若以顶点表示活动，弧表示活动之间的先后关系，这样的图简称为AOV（Activity On Vertex）网；若以顶点表示事件，弧表示活动，权表示完成该活动所需的时间（称为活动历时或持续时间），这样的图称为AOE（Activity On Edge）网。例如，图7-6表示一个具有10个活动的某个工程的AOE网。图中有7个顶点，分别表示事件1～7，其中1表示工程开始状态，7表示工程结束状态。

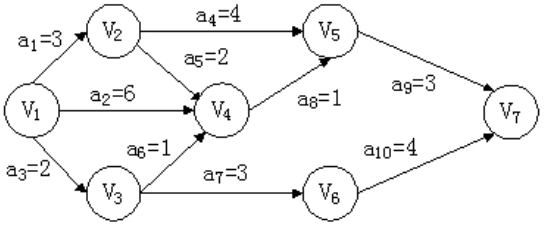


图7-6 AOE网络的例子

因AOE网中的某些活动可以并行地进行，所以完成工程的最少时间是从开始顶点到结束顶点的最长路径长度，称从开始顶点到结束顶点的最长路径为关健路径（临界路径），关键路径上的活动为关键活动。

为了找出给定的AOE网络的关键活动，从而找出关键路径，先定义几个重要的量：

（1）$V_e(j)$、$V_l(j)$：顶点j事件最早、最迟发生时间。

（2）e(i)、l(i)：活动i最早、最迟开始时间。

从源点V_1到某顶点V_j的最长路径长度，称为事件V_j的最早发生时间，记作$V_e(j)$。$V_e(j)$也是以V_j为起点的出边<V_j,V_k>所表示的活动a_i的最早开始时间e(i)。

在不推迟整个工程完成的前提下，一个事件V_j允许的最迟发生时间，记作$V_l(j)$。显然，$l(i)=V_l(j)-$(a_i所需时间)，其中j为a_i活动的终点。满足条件l(i)=e(i)的活动为关键活动，关键活动所组成的路径称为关键路径。

求顶点V_j的$V_e(j)$和$V_l(j)$可按以下两步来做：

（1）由源点开始向汇点递推

$$\begin{cases} V_e(1)=0 \\ V_e(j)=Max\{V_e(i)+d(i,j)\}, <V_i,\ V_j>\in E_1, 2\le j\le n \end{cases}$$

其中，E_1是网络中以V_j为终点的入边集合。

（2）由终点（汇点）开始向源点递推

$$\begin{cases} V_l(n)=V_e(n) \\ V_l(j)=Min\{V_l(k)-d(j,k)\}, <V_j,\ V_k>\in E_2, 2\le j\le n-1 \end{cases}$$

其中，E_2是网络中以V_j为起点的出边集合。

要求一个AOE网的关键路径，一般需要根据以上变量列出一张表格，逐个检查。例如，求图7-6所示的AOE网的关键路径的表格如表7-6所示。

表7-6 求关键路径的过程

V_j	$V_e(j)$	$V_l(j)$	a_i	e(i)	l(i)	l(i)-e(i)
V_1	0	0	$a_1(3)$	0	0	0
V_2	3	3	$a_2(6)$	0	0	0
V_3	2	3	$a_3(2)$	0	1	1
V_4	6	6	$a_4(4)$	3	3	0
V_5	7	7	$a_5(2)$	3	4	1
V_6	5	6	$a_6(1)$	2	5	3
V_7	10	10	$a_7(3)$	2	3	1
			$a_8(1)$	6	6	0
			$a_9(3)$	7	7	0
			$a_{10}(4)$	5	6	1

因此，图7-6的关键活动为a_1，a_2，a_4，a_8和a_9，其对应的关键路径有两条，分别为$V_1\rightarrow V_2\rightarrow V_5\rightarrow V_7$和$V_1\rightarrow V_4\rightarrow V_5\rightarrow V_7$，长度都是10。

当然，上面所描述的方法用来求关键路径有一定的复杂度，需要进行大量的计算才能得到结果，这是一种严谨的方法，可以应对复杂的AOE网络图。，然而考试中涉及的图往往很简单，此时再用这种方法来解决问题，会浪费不必要的时间，所以可以直接用“目测法”快速解决问题。所谓“目测法”，就是直接观察AOE网络中，从起点到终点，哪条路径最长，最长的那条路径，便是关键路径，而最长路径的长度，就是关键路径长度。

3. 甘特图

甘特图（Gantt图）用水平线段表示任务的工作阶段；线段的起点和终点分别对应着任务的开工时间和完成时间；线段的长度表示完成任务所需的时间。图7-7给出了一个具有5 个任务的甘特图。如果这5条线段分别代表完成任务的计划时间，则在横坐标方向附加一条可向右移动的纵线。它可随着项目的进展，指明已完成的任务(纵线扫过的)和有待完成的任务(纵线尚未扫过的)。我们从甘特图上可以很清楚地看出各子任务在时间上的对比关系。

在甘特图中，每一任务完成的标准，不是以能否继续下一阶段任务为标准，而是必须交付应交付的文档与通过评审为标准。因此在甘特图中，文档编制与评审是软件开发进度的里程碑。甘特图的优点是标明了各任务的计划进度和当前进度，能动态地反映软件开发进展情况。缺点是难以反映多个任务之间存在的复杂的逻辑关系。

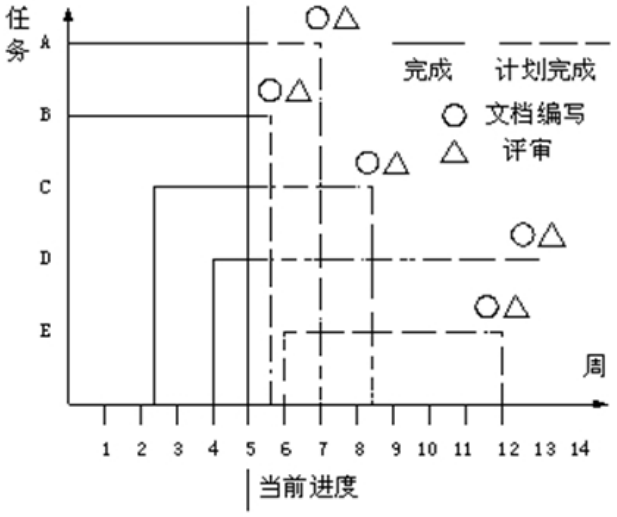


图7-7甘特图

4. 风险管理

风险是指“损失或伤害的可能性”。所以风险包含这些特征：不确定性、针对未来、会带来损失。

项目风险管理通常包括风险识别、风险估计（风险评估）和风险驾驭（风险控制）三个主要活动。风险识别的主要工作是找到潜在风险并将其文档化，它包括项目风险、技术风险和商业风险三种。风险估计则是通过对各种风险发生的可能性和破坏性这两个方面进行评估，并将它们按优先级进行排列。风险驾驭则是指利用某种技术，如原型化、软件自动化、软件心理学、可靠性工程学等方法设法避开风险。

当在软件工程环境中考虑风险时，主要是基于关心未来、关心变化、关心选择这三个概念提出的。项目风险关系着项目计划的成败，而商业风险关系着软件的生存能力。在进行软件工程分析时，项目管理人员要进行四种风险评估活动，包括建立表示风险概率的尺度，描述风险引起的后果，估计风险影响的大小，确定风险估计的正确性。

在风险管理时，经常会应用风险曝光度（Risk Exposure）这个概念，它是计算方法是风险出现的概率乘以风险可能造成的损失。假设正在开发的软件项目可能存在一个未被发现的错误，而这个错误出现的概率是0.8%，给公司造成的损失将是2000000元，那么这个错误的风险曝光度就应为2000000×0.8%=16000元。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

第 7 章：软件工程基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月05日

一点一练

试题1

图7-8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__(1)__在关键路径上。

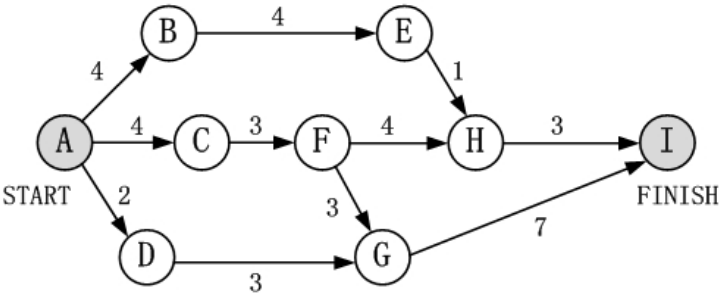


图7-8活动网络图

（1）A．B B．C C．D D．H

试题2

__(2)__最不适于采用无主程序员组的开发人员组织形式。

（2）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3

若软件项目组对风险采用主动的控制方法，则__(3)__是最好的风险控制策略。

（3）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

试题4

甘特图（Gantt图）不能__(4)__。

（4）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

试题5

包含8个成员的开发小组的沟通路径最多有__(5)__条。

（5）A．28　　B．32　　C．56　　D．64

试题6

下列关于风险的叙述不正确的是：风险是指__(6)__。

（6）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

试题7

图7-9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__(7)__在关键路径上。

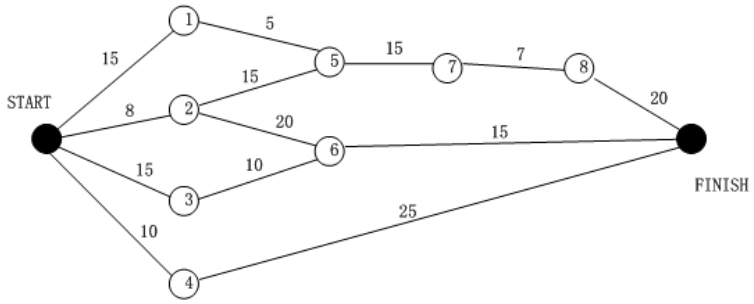


图7-9活动图

（7）A．1　　B．2　　C．3　　D．4

版权方授权希赛网发布，侵权必究

上一节　　本书简介　　下一节

解析与答案

试题1分析

本题主要考查关键路径求解的问题。

从开始顶点到结束顶点的最长路径为关健路径，关键路径上的活动（事件）为关键活动（事件）。

在本题中找出的最长路径是A->C->F->G-->I，其长度为4+3+3+7=17，而其它任何路径的长度都比这条路径小，因此我们可以知道事件C在关键路径上。

试题1答案

（1）B

试题2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。

试题2答案

（2）C

试题3分析

风险避免即放弃或不进行可能带来损失的活动或工作。例如，为了避免洪水风险，可以把工厂建在地势较高、排水方便的地方，这是一种主动的风险控制方法。

风险监控是指在决策主体的运行过程中，对风险的发展与变化情况进行全程监督，并根据需要进行应对策略的调整。

风险管理是指在一个肯定有风险的环境里把风险减至最低的管理过程。对于风险我们可以转移，可以规避，但不能消除。

试题3答案

（3）A

试题4分析

甘特图是一种能清晰描述每个任务的开始和截止时间，能有效获得任务并行进行的信息的项目管理工具。

试题4答案

（4）D

试题5分析

在知道小组成员后，求沟通路径可按公式n*（n-1）/2求解，那么题目告诉我们开发小组有8个成员，即存在的沟通路径为8*（8-1）/2=28条。

试题5答案

（5）A

试题6分析

本题主要我们对风险概念的理解。

目前，对风险大致有两种定义：一种定义强调了风险表现为不确定性；而另一种定义则强调风险表现为损失的不确定性。其中广义的定义是：风险表现为损失的不确定性，说明风险产生的结果可能带来损失、获利或是无损失也无获利。

从风险的定义我们不难看出，风险是可能发生的事件，而且是会带来损失的事件，人为对其干预，可能会减少损失。

试题6答案

（6）B

试题7分析

本题主要考查关键路径求解的问题。

从开始顶点到结束顶点的最长路径为关健路径（临界路径），关键路径上的活动为关键活动。

在本题中找出的最长路径是Start->2->5->7->8->Finish，其长度为8+15+15+7+20=65，而其它任何路径的长度都比这条路径小，因此我们可以知道里程碑2在关键路径上。

试题7答案

（7）B

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

考前冲刺

试题1

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__(1)__进行开发。

（1）A．瀑布模型　B．演化模型　C．螺旋模型　D．原型模型

试题2

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，__(2)__最适合开发此项目。

（2）A．原型模型　B．瀑布模型　C．V模型　D．螺旋模型

试题3

图7-10所示的逻辑流程实现折半查找功能，最少需要__(3)__个测试用例可以覆盖所有的可能路径。

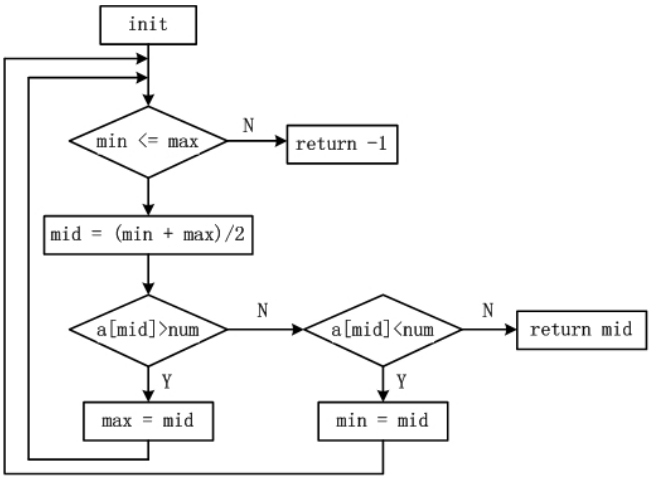


图7-10 程序流程图

（3）A．1　　B．2　　C．3　　D．4

试题4

根据McCabe度量法，图7-11的复杂性度量值为__(4)__。

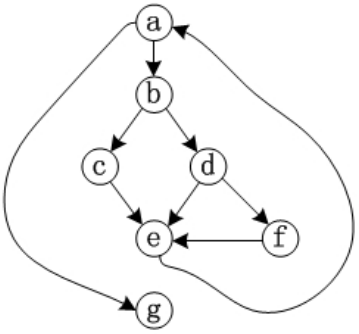


图7-11程序复杂性图

（4）A．4　　B．5　　C．6　　D．7

试题5

进行防错性程序设计，可以有效地控制__(5)__维护成本。

（5）A．正确性　B．适应性　C．完善性　D．预防性

试题6

在软件维护阶段，为软件的运行增加监控设施属于__(6)__维护。

（6）A．改正性　　B．适应性　　C．完善性　　D．预防性

试题7

以下关于过程改进的叙述中，错误的是__(7)__。

（7）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8

软件复杂性度量的参数不包括__(8)__。

（8）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9

以下关于软件系统文档的叙述中，错误的是__(9)__。

（9）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10

图7-12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__(10)__。

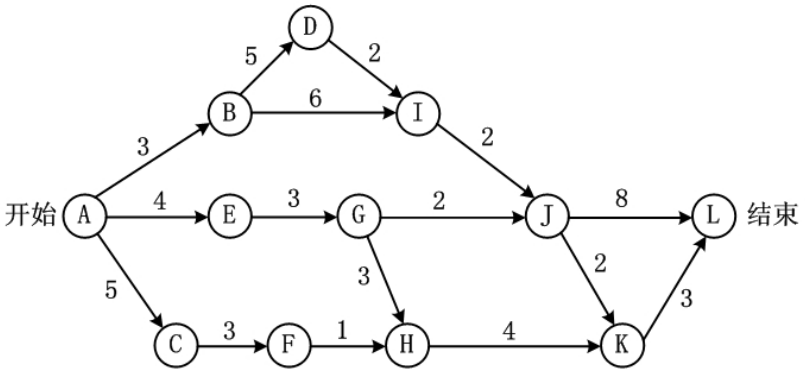


图7-12活动网络图

（10）A． 20　　B． 19　　C． 17　　D． 16

试题11

在软件开发过程中进行风险分析时，__(11)__活动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。

（11）A．风险识别　　B．风险预测　　C．风险评估　　D．风险控制

试题12

以下关于风险管理的叙述中，不正确的是__(12)__。

（12）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

试题13

下列关于项目估算方法的叙述不正确的是__(13)__。

（13）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

试题14

在有些程序设计语言中，过程调用和响应调用需执行的代码的绑定直到运行时才进行，这种绑定称为__(14)__。

（14）A．静态绑定　　B．动态绑定　　C．过载绑定　D．强制绑定

试题15

一个类是__(15)__。在定义类时，将属性声明为private的目的是__(16)__。

（15）A．一组对象的封装B．表示一组对象的层次关系

C．一组对象的实例D．一组对象的抽象定义

（16）A．实现数据隐藏，以免意外更改

B．操作符重载

C．实现属性值不可更改

D．实现属性值对类的所有对象共享

试题16

在面向对象软件开发中，封装是一种__(17)__技术，其目的是使对象的使用者和生产者分离。

（17）A．接口管理　　B．信息隐藏　　C．多态　　D．聚合

试题17

下列关于一个类的静态成员的描述中，不正确的是__(18)__。

（18）A．类的静态方法只能访问该类的静态数据成员

B．静态数据成员可被该类的所有方法访问

C．该类的对象共享其静态数据成员的值

D．该类的静态数据成员的值不可修改

版权方授权希赛网发布，侵权必究

上一节　　本书简介　　下一节

习题解析

试题1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1答案

（1）A

试题2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2答案

（2）B

试题3分析

测试用例的个数为判断数+1，所以本题中需要的测试用例个数为4。

试题3答案

（3）B

试题4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4答案

（4）A

试题5分析

本章7.4节有关于维护类型的详细描述，解答本题时，请参看相关内容。本题中正确性维护其实就是改正性维护。通过概念可以得知，预防性维护是在错误出现之前进行维护修改，这样可以避免一些错误的出现，所以能有效控制正确性维护成本（此外，我们需要了解一点：一个错误，在他爆发前发现，并修改，成本比较低。而一旦等到错误出现，再找到错误进行修改，成本会成倍上涨）。

试题5答案

（5）A

试题6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5%。

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50%。

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护。

试题6答案

（6）C

习题解析

软件过程的改变进行计划、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7答案

（7）B

试题8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[illegible]。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[illegible]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8答案

（8）B

试题9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9答案

（9）C

试题10分析

本题主要考查求关键路径。从开始顶点到结束顶点的最长路径为关健路径（临界路径），关键路径上的活动为关键活动。

在本题中找出的最长路径是A->B>D->I->J->L，其长度为3+5+2+2+8=20，而其它任何路径的长度都比这条路径小，因此我们可以知道关键路径的长度为20。

试题10答案

（10）A

试题11分析

在进行风险分析时，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。

试题11答案

（11）D

试题12分析

对风险排优先级是根据风险的曝光度来进行的，曝光度等于风险的产生后果乘以风险发生的概率。

试题12答案

（12）A

试题13分析

项目估算的常用方法主要有专家判断法、启发式法和机器学习法等。

专家判断法是指向学有专长、见识广博并有相关经验的专家进行咨询、根据他们多年来的实践经验和判断能力对计划项目作出预测的方法。很显然，采用这种方法容易受到专家经验和主观性的影响。

启发式方法使用一套相对简单、通用、有启发性的规则进行估算的方法，它具有参数难以确定、精确度不高等特点。

机器学习方法是一种基于人工智能与神经网络技术的估算方法，它难以描述训练数据的特征和确定其相似性。

而无论采用哪种估算方法，估算得到的结果都是大概的，而不是精确的。

试题13答案

（13）D

试题14分析

动态绑定是指在执行期间（非编译期）判断所引用对象的实际类型，根据其实际的类型调用其相应的方法。而静态绑定是指在程序编译期就完成的绑定。

试题14答案

（14）B

试题15分析

类与对象的关系是抽象与具体的关系，类是一组对象的抽象，而对象是类的实例。在定义类时，将属性声明为private，即只允许自身对其进行访问、修改等操作，而外界不可见，从而达到实现数据隐藏，以免意外更改的母的。

试题15答案

（15）D（16）A

试题16分析

面向对象系统中的封装单位是对象，对象之间只能通过接口进行信息交流，对象外部不能对对象中的数据随意地进行访问。封装的目的是使对象的定义和实现分离，这样，就能减少耦合，类内部的实现可以自由改变而不会影响其他的类或对象。

试题16答案

（17）B

试题17分析

类的静态成员与一般的类成员不同，静态成员与对象的实例无关，只与类本身有关。它们一般用来实现类要封装的功能和数据，但不包括特定对象的功能和数据。静态成员和普通数据成员区别较大，体现在下面几点：

（1）普通数据成员属于类的一个具体的对象，只有对象被创建了，普通数据成员才会被分配内

存。而静态数据成员属于整个类，即使没有任何对象创建，类的静态数据成员变量也存在。

（2）外部访问类的静态成员只能通过类名来访问。

（3）类的静态成员函数无法直接访问普通数据成员（可以通过类的指针等作为参数间接访问），而类的任何成员函数都可以访问类的静态数据成员。

（4）类的静态方法只能访问该类的静态数据成员。

另外，静态成员和类的普通成员一样，也具有public、protected、private3种访问级别，也可以具有返回值及被修改等性质。

试题17答案

（18）D

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

考点脉络

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

（1）信息系统安全基础知识

（2）信息系统安全管理

（3）保障完整性与可用性的措施

（4）加密与解密机制基础知识

（5）风险管理（风险分析、风险类型、抗风险措施和内部控制）

（6）计算机安全相关的法律、法规基础知识

从历年的考试情况来看，本章的考点主要集中于加密解密技术、网络安全、计算机病毒等方面。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节

安全基础技术

在软件设计师的考试中，经常会考到与安全相关的一些基础概念及技术原理，如：区分对称与非对称算法，什么情况下用哪种密钥加密解密、信息摘要的用途等。下面将详细介绍这些相关技术。

版权方授权希赛网发布，侵权必究

上一节　本书简介　下一节