

<<全国计算机等级考试四级教程>>

图书基本信息

书名：<<全国计算机等级考试四级教程>>

13位ISBN编号：9787040222968

10位ISBN编号：7040222965

出版时间：2007-9

出版范围：高等教育

作者：教育部考试中心 编

页数：380

字数：580000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<全国计算机等级考试四级教程>>

前言

当今，人类正在步入一个以智力资源的占有和配置，知识生产、分配和使用为最重要因素的知识经济时代，也就是小平同志提出的“科学技术是第一生产力”的时代。

科教是经济发展的基础，知识是人类创新的源泉。

基础研究的科学发现、应用研究的原理探索 and 开发研究的技术发明，三者之间的联系愈来愈紧密，转换周期日趋缩短。

世界各国的竞争已成为以经济为基础、以科技（特别是高科技）为先导的综合国力的竞争。

在高科技中，信息科学技术是知识高度密集、学科高度综合，具有科学与技术融合特征的学科。

它直接渗透到经济、文化和社会的各个领域，迅速改变着人们的观念、生活和社会的结构，是当代发展知识经济的支柱之一。

在信息科学技术中，微电子是基础，计算机硬件及通信设施是载体，计算机软件是核心。

软件是人类知识的固化，是知识经济的基本表征，软件已成为信息时代的新型“物理设施”。

人类抽象的经验、知识正逐步由软件予以精确地体现。

在信息时代，软件是信息化的核心，国民经济和国防建设、社会发展、人民生活都离不开软件，软件无处不在。

软件产业是增长最快的朝阳产业，是具有高额附加值、高投入 / 高产出、无污染、低能耗的绿色产业。

软件产业的发展将推动知识经济的进程，促进从注重量的增长向注重质的提高的方向发展，是典型的知识型产业。

软件产业是关系到国家经济安全和文化安全，体现国家综合实力，决定21世纪国际竞争地位的战略产业。

为了适应知识经济发展的需要，大力推动信息产业的发展，需要在全民中普及计算机的基本知识，广开渠道，培养和造就一批又一批能熟练运用计算机和软件技术的各行各业的专门人才。

1994年，原国家教委（现教育部）推出了全国计算机等级考试，它是一种重视应试人员对计算机和软件的实际掌握能力的考试。

它不限制报考人员的学历背景，任何年龄段的人员都可以报考。

这就为培养各行各业计算机的应用人才开辟了一条广阔的道路。

1994年是推出计算机等级考试的第一年，当年参加考试的有1万余人；而2008年，年报考人数已近418万人。

截至2009年上半年，全国计算机等级考试共开考29次，考生人数累计达3122万，其中有1170万人获得了不同级别的计算机等级证书。

<<全国计算机等级考试四级教程>>

内容概要

本书是按照教育部考试中心颁布的《全国计算机等级考试四级软件测试工程师考试大纲（2007年版）》的有关要求，由教育部考试中心组织，在全国计算机等级考试委员会指导下，邀请有关专家编写的。

内容包括软件测试的基本概念、软件测试在软件开发过程中的地位、代码走查与评审、结构覆盖（白盒）测试、功能（黑盒）测试、单元测试和集成测试、系统测试、软件性能测试和可靠性测试、面向对象软件测试、Web应用软件测试以及兼容性测试、构件测试、极限测试和文档测试等。

本书还对软件测试过程和管理进行了详细阐述，讨论了了软件自动化测试技术和有关工具，介绍了我国软件测试的现行标准和测试文档规范，最后结合软件测试过程管理平台QEsuite和软件分析与测试工具QESAT/C介绍了软件测试工程的实践经验。

本书内容丰富翔实，理论和实践并重，实用性强，不仅可作为四级软件测试工程师考试的标准教材，而且可用作高校计算机与软件工程专业的教材，也可作为软件测试人员实用的培训教材和技术参考书。

书籍目录

第1章 软件测试的基本概念 1.1 软件质量的概念 1.1.1 软件质量的定义 1.2.2 软件质量的属性
1.2.3 软件质量的模型 1.2.4 软件质量的度量 1.2 软件测试的概念 1.2.1 软件测试的定义与目的
1.2.2 软件测试的原则 1.3 软件的缺陷与错误 1.3.1 软件缺陷的定义和类型 1.3.2 软件缺陷的级别
1.3.3 软件缺陷产生的原因 1.3.4 软件缺陷的构成 1.3.5 修复软件缺陷的代价 1.4 软件测试
的经济学与心理学 1.4.1 软件测试的心理学 1.4.2 软件测试的经济学 1.5 软件质量保证 1.5.1
软件质量保证概要 1.5.2 软件质量保证活动的实施 1.5.3 SQA与软件测试的关系 本章小结第2章
软件测试类型及其在软件开发过程中的地位 2.1 软件开发阶段 2.1.1 软件生存周期 2.1.2 软件测
试的生存周期模型 2.1.3 测试信息流 2.2 规划阶段的测试 2.2.1 目标阐述 2.2.2 需求分析 2.2.3
功能定义 2.2.4 规划阶段进行的测试 2.3 设计阶段的测试 2.3.1 外部设计 2.3.2 内部设计
2.3.3 设计阶段的测试 2.3.4 伪代码分析 2.4 编程阶段的测试 2.4.1 白盒测试与黑盒测试 2.4.2
结构测试与功能测试 2.4.3 路径测试:覆盖准则 2.4.4 增量测试与大突击测试 2.4.5 自顶向下测
试与自底向上测试 2.4.6 静态测试与动态测试 2.4.7 性能测试 2.5 回归测试 2.6 运行和维护阶
段的测试 本章小结第3章 代码检查、走查与评审 3.1 桌上检查 3.1.1 桌上检查的检查项目 3.1.2
对程序代码做静态错误分析 3.2 代码检查 3.2.1 特定的角色和职责 3.2.2 代码检查过程 3.2.3
用于代码检查的错误列表 3.3 走查 3.3.1 特定的角色和职责 3.3.2 走查的过程 3.3.3 走查中
的静态分析技术 3.4 同行评审 3.4.1 为什么需要评审 3.4.2 同行评审的角色和职能 3.4.3 同行
评审的内容 3.4.4 评审的方法和技术 本章小结第4章 覆盖率(白盒)测试 4.1 覆盖率概念 4.2
逻辑覆盖 4.2.1 语句覆盖 4.2.2 判定覆盖(分支覆盖) 4.2.3 条件覆盖 4.2.4 条件/判定覆盖
4.2.5 条件组合覆盖 4.2.6 路径覆盖 4.2.7 ESTCA覆盖 4.2.8 LCSAJ覆盖 4.3 路径测试 4.3.1
分支结构的路径测试 4.3.2 循环结构的路径测试 4.3.3 Z路径覆盖与基本路径测试 4.4 数据流测
试 4.4.1 定义/使用测试的几个定义 4.4.2 定义/使用测试举例 4.4.3 定义/使用路径测试覆盖
指标 4.5 基于覆盖的测试用例选择 4.5.1 如何使用覆盖率 4.5.2 使用最少测试用例来达到覆盖 本
章小结第5章 功能(黑盒)测试 5.1 等价类测试 5.1.1 等价类的概念 5.1.2 等价类测试的类型
5.1.3 等价类测试的原则 5.1.4 等价类方法测试用例设计举例 5.2 边界值分析 5.2.1 边界值分析
的概念 5.2.2 选择测试用例的原则 5.2.3 边界值方法测试用例设计举例 5.3 基于判定表的测试
5.3.1 判定表的概念 5.3.2 基于判定表的测试用例设计举例 5.4 基于因果图的测试 5.4.1 因
果图的适用范围 5.4.2 用因果图生成测试用例 5.4.3 因果图法测试用例设计举例 5.5 基于状态
图的测试 5.5.1 功能图及其符号 5.5.2 功能图法设计测试用例举例 5.6 基于场景的测试
5.6.1 基本流和备选流 5.6.2 场景法设计测试用例举例 5.7 其他黑盒测试用例设计技术 5.7.1
规范(规格)导出法 5.7.2 内部边界值测试法 5.7.3 错误猜测法 5.7.4 基于接口的测试 5.7.5
基于故障的测试 5.7.6 基于风险的测试 5.7.7 比较测试 本章小结第6章 单元测试和集成测试 6.1
单元测试的基本概念 6.1.1 单元测试的定义和目标 6.1.2 单元测试与集成测试、系统测试的区别
6.1.3 单元测试环境 6.2 单元测试策略 6.2.1 自顶向下的单元测试策略 6.2.2 自底向上的单元测
试策略 6.2.3 孤立测试 6.2.4 综合测试 6.3 单元测试分析 6.3.1 模块接口 6.3.2 局部数据结构
6.3.3 独立路径 6.3.4 出错处理 6.3.5 边界条件 6.3.6 其他测试分析的指导原则 6.4 单元测
试的测试用例设计原则 6.4.1 单元测试的测试用例设计步骤 6.4.2 单元测试中的白盒测试与黑盒测
试 6.5 集成测试的基本概念 6.5.1 集成测试的定义 6.5.2 集成测试与系统测试的区别 6.5.3
集成测试与开发的关系 6.5.4 集成测试重点 6.5.5 集成测试层次 6.5.6 集成测试环境 6.6
集成测试的策略 6.6.1 基于分解的集成策略 6.6.2 基于功能的集成 6.6.3 基于调用图的集成
6.6.4 基于路径的集成 6.6.5 基于进度的集成 6.6.6 基于风险的集成 6.7 集成测试分析
6.7.1 体系结构分析 6.7.2 模块分析 6.7.3 接口分析 6.7.4 可测试性分析 6.7.5 集成测
试策略的分析 6.7.6 常见的集成测试故障 6.8 集成测试的测试用例设计 本章小结第7章 系统
测试 7.1 系统测试概念 7.1.1 什么是系统测试 7.1.2 系统测试与单元测试、集成测试的区别
7.1.3 系统测试的组织和分工 7.1.4 系统测试分析 7.1.5 系统测试环境 7.2 系统测试的方
法 7.2.1 功能测试 7.2.2 协议一致性测试 7.2.3 性能测试 7.2.4 压力测试 7.2.5 容量测

试 7.2.6 安全性测试 7.2.7 失效恢复测试 7.2.8 备份测试 7.2.9 GUI测试 7.2.10 健壮性测试 7.2.11 兼容性测试 7.2.12 易用性测试 7.2.13 安装测试 7.2.14 文档测试 7.2.15 在线帮助测试 7.2.16 数据转换测试 7.3 系统测试的实施 7.3.1 确认测试 7.3.2 测试和测试 7.3.3 验收测试 7.3.4 回归测试 7.3.5 系统测试问题总结、分析 7.4 如何做好系统测试 本章小结第8章 软件性能测试和可靠性测试 8.1 软件性能测试的基本概念 8.1.1 什么是软件性能 8.1.2 软件性能的测试 8.2 软件性能测试的执行 8.2.1 性能测试的过程与组织 8.2.2 性能分析 8.2.3 性能测试的自动化 8.3 软件可靠性的概念 8.3.1 错误、缺陷、故障与失效 8.3.2 软件可靠性定义 8.3.3 软件可靠性指标 8.3.4 软件可靠性与硬件可靠性的区别 8.4 软件可靠性测试的执行 8.4.1 软件可靠性测试的过程 8.4.2 软件可靠性预测 8.5 软件可靠性分析方法 本章小结第9章 面向对象软件的测试 9.1 面向对象软件测试的问题 9.1.1 面向对象的基本特点引起的测试问题 9.1.2 面向对象程序的测试组织问题 9.2 面向对象软件的测试模型及策略 9.2.1 面向对象软件的测试模型 9.2.2 面向对象分析的测试 9.2.3 面向对象设计的测试 9.2.4 面向对象编程的测试 9.2.5 面向对象程序的单元测试 9.2.6 面向对象程序的集成测试 9.2.7 面向对象软件的系统测试 9.3 面向对象程序的单元测试 9.3.1 方法层次的测试 9.3.2 类层次的测试 9.3.3 类树层次的测试 9.4 面向对象程序的集成测试 9.4.1 面向对象程序的集成测试策略 9.4.2 针对类间连接的测试 9.5 面向对象软件的系统测试 9.5.1 功能测试 9.5.2 其他系统测试 本章小结第10章 Web应用软件测试 10.1 Web应用软件 10.1.1 Web应用软件的概念 10.1.2 Web应用软件的特点 10.1.3 Web应用软件的基本结构 10.1.4 Web应用软件的常用开发技术 10.2 应用服务器的分类和特征 10.2.1 三层和多层体系结构 10.2.2 应用服务器的分类 10.2.3 应用服务器对Web应用软件测试的影响 10.3 Web应用软件的测试策略 10.3.1 表示层的测试 10.3.2 业务层的测试 10.3.3 数据层的测试 10.3.4 层间的集成测试 10.4 Web应用软件的系统测试技术 10.4.1 功能测试 10.4.2 性能测试 10.4.3 易用性测试 10.4.4 内容测试 10.4.5 安全性测试 10.4.6 接口测试 10.5 Web应用软件的系统安全检测与防护 10.5.1 入侵检测 10.5.2 漏洞扫描 10.5.3 安全策略 本章小结第11章 其他测试 11.1 兼容性测试 11.1.1 硬件兼容性测试 11.1.2 软件兼容性测试 11.1.3 数据兼容性测试 11.2 易用性测试 11.2.1 易安装性测试 11.2.2 功能易用性测试 11.2.3 用户界面测试 11.3 构件测试 11.3.1 与测试相关的构件特性 11.3.2 构件测试的主要困难 11.3.3 构件测试涉及的人员 11.3.4 构件测试的常用方法 11.4 极限测试 11.4.1 极限编程基础 11.4.2 极限测试 11.5 文档测试 11.5.1 文档测试的范围 11.5.2 用户文档的内容 11.5.3 用户文档的测试 本章小结第12章 软件测试过程和管理 12.1 软件测试过程 12.1.1 测试过程的概念 12.1.2 测试过程的抽象模型 12.1.3 测试阶段中的测试活动 12.2 测试过程组织与管理 12.2.1 软件测试过程管理的特点 12.2.2 软件测试过程管理的原则 12.2.3 软件测试过程的人员组织 12.3 测试计划的制定 12.3.1 测试计划的整体目标 12.3.2 定义软件测试项目的标准 12.3.3 测试实施策略的制定 12.3.4 测试计划活动的细分 12.3.5 测试计划的要点 12.3.6 编制测试计划的技巧 12.4 测试方案的确定 12.4.1 软件测试设计和开发主要内容 12.4.2 软件测试设计要点 12.4.3 测试用例的设计方法 12.4.4 测试用例的管理 12.4.5 测试开发 12.5 测试环境管理 12.5.1 测试环境的定义 12.5.2 测试环境是测试的基础 12.5.3 测试环境的各要素 12.6 测试执行的管理 12.6.1 测试用例执行的跟踪 12.6.2 缺陷的跟踪和管理 12.6.3 与项目组外部人员的沟通 12.6.4 测试执行结束 12.7 测试质量分析 12.7.1 评估系统测试的覆盖程度 12.7.2 软件缺陷分析方法 12.8 测试文档管理 12.8.1 测试文档的分类管理 12.8.2 测试文档的存储和共享 12.8.3 文档模板 12.9 测试的复用与维护 12.9.1 测试过程的复用与维护 12.9.2 测试用例和软件缺陷报告的复用 本章小结第13章 软件自动化测试 13.1 自动化测试的原理与方法 13.2 自动化测试的限制 13.3 自动化测试用例的生成 13.3.1 脚本的作用、质量和编写原则 13.3.2 脚本的基本结构 13.4 测试执行自动化 13.4.1 前处理 13.4.2 后处理 13.4.3 自动化测试过程 13.5 测试结果比较自动化 13.5.1 自动比较的基本概念 13.5.2 动态比较 13.5.3 执行后比较 13.6 测试工具的分类与选择 13.6.1 测试工具的分类 13.6.2 测试工具的选择 13.7 测试工具的主流产品介绍 13.7.1 面向开发的单元测试工具 13.7.2 负载和性能测试工具 13.7.3 GUI功能测试工具 13.7.4 基于Web应用的测试工具 13.7.5 软件测试管理和其他工具 13.7.6 其他测试工具 本章小结第14章 软件测试的标准和文

<<全国计算机等级考试四级教程>>

档 14.1 软件测试的标准 14.1.1 软件测试规范 14.1.2 软件测试文档编制规范 14.2 软件测试文档示例 14.2.1 软件测试文档示例 14.2.2 软件测试常用模板 本章小结第15章 软件测试实践 15.1 软件测试过程管理实践 15.1.1 测试的策划 15.1.2 测试的设计 15.1.3 测试的执行 15.1.4 软件缺陷(问题)报告和软件缺陷(问题)生命周期 15.1.5 测试的总结 15.1.6 测试过程的质量控制 15.1.7 QESuite软件测试过程管理平台 15.2 白盒测试实践 15.2.1 QESAT/C简介 15.2.2 被测程序link.c说明 15.2.3 测试准备 15.2.4 静态分析 15.2.5 动态测试 本章小结附录参考文献

章节摘录

插图：11.5.3用户文档的测试1.测试要点在用户文档测试中需要注意以下几个方面的问题：首先，由于用户文档测试中的技术含量相对较低，往往得不到开发人员的重视。

但实际上用户文档测试的回报率其实并不低，许多用户文档的缺陷都可能给开发组织带来巨大损失。因此，好的项目需要对用户文档测试给予关注，并为其留出充足的时间和资金预算。

其次，用户文档通常要求书写人员既要了解程序的特性又要了解用户的特性，但在实际中往往难以做到。

在用户文档测试时，需要针对文档书写人员的特点确定测试人员，尽可能做到两者互补，即如果书写人员不了解用户，测试人员应该了解用户；如果书写人员不了解程序，测试人员应该了解程序。

另外，由于用户文档没有相应的规约，测试人员必须与书写人员密切合作才能完成对文档的修改和更新。

再次，对于需要印刷的文档（如宣传材料、包装材料以及各种手册等），测试时需要特别注意文档与程序的一致性。

由于印刷需要较长的周期和较高的费用，如果文档中存在较多的错误或不一致，要么让用户因文档问题而抱怨，要么不得不把已印刷的文档销毁，重新印刷修改过的文档（这既增加了成本又推迟了软件的发布时间）。

最后，用户文档测试的重点不应放在文字校对上，而应该放在查找错误和问题上，例如功能错误、易用性问题、程序与手册的描述不一致等。

编辑推荐

《全国计算机等级考试四级教程:软件测试工程师(2010年版)》是由高等教育出版社出版的。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>