

<<软件设计师考试考前串讲>>

图书基本信息

书名：<<软件设计师考试考前串讲>>

13位ISBN编号：9787121068690

10位ISBN编号：7121068699

出版时间：2008-9

出版时间：电子工业

作者：王勇

页数：384

字数：496000

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>

前言

全国计算机技术与软件专业技术资格（水平）考试是由国家人力资源和社会保障部、工业和信息化部组织和领导的国家级考试，考试具有很高的权威性，同时也决定了其考试范围的广度和深度都比较大，使许多考生在复习和准备上遇到了很多的难题。

虽然国家软考办、希赛IT教育研发中心陆续出版了一系列的有针对性的考试辅导教程，为考生复习和备考提供了基础性的帮助。

但是，由于考试范围十分广泛，内容量相当大，仍然无法完全满足考生的需求。

内容超值，针对性强 由于考试大纲规定的考试知识点体系庞大，对考生而言，要学习的内容很多，很难把考试大纲规定的知识点全部进行梳理和系统地学习。

为此，希赛IT教育研发中心组织有关专家对考试大纲和历年考试试题进行了深入的分析，在此基础上编写了本书，作为计算机技术与软件专业技术资格（水平）考试中的软件设计师级别的考试辅导指定教材。

就考试中经常出现的一些问题进行归纳和总结，其目的是希望能够压缩所有考试重点和难点知识，而不是囊括所有考试知识点。

其结果是让读者顺利通过考试，而不是获得满分。

本书根据软件设计师考试大纲，对历年考试中所有知识点进行了归类分析和总结，挖掘出了其中的考试重点和难点，指出了考试的命题方向以及每个知识点在实际考试中所占的分数比例。

根据希赛IT教育研发中心进行考试辅导和阅卷的经验，对其中的难点问题和考生容易出错的问题进行了详细的讨论。

本书在组织和写作上，倾注了作者们许多的精力和心血，将自己所有的心得和体会融入其中，相信能够对提高考生通过率，有效地完成“考试过关”提供帮助。

考生可通过阅读本书，迅速掌握考试重点和难点，解答问题的方法和技巧。

从希赛IT教育研发中心的研究结果来看，本书所总结的考试重点知识包括了实际考试时95%以上的考点，而这些知识点几乎在每次考试中都会出现。

因此，考生阅读本书，可得到事半功倍的效果，提高考试通过率。

作者权威，阵容强大 希赛是中国领先的互联网技术和IT教育公司，在互联网服务、图书出版、人才培养方面，希赛始终保持IT业界的领先地位。

希赛对国家信息化建设和软件产业化发展具有强烈的使命感，利用希赛网强大的平台优势，加强与促进IT人士之间的信息交流和共享，实现IT价值。

“希赛，影响IT”是全体希赛人不懈努力和追求的目标！

希赛IT教育研发中心是希赛公司属下的一个专门从事IT教育、教育产品开发、教育书籍编写的部门，在IT教育方面具有极高的权威性。

希赛IT教育研发中心的远程教育模式得到了教育部门的认可和推广，特别是在系统分析、需求分析、架构设计和项目管理的人才培养方面，有口皆碑。

同时，在高端课程方面，希赛IT教育研发中心开设的课程有软件工程系列、项目管理系列、企业管理系列，以及网络安全系列课程。

希赛IT教育研发中心负责中国IT教育门户学赛网的运维工作。

本书由希赛IT教育研发中心组编，由王勇主编，张友生审核了所有稿件。

全书共分13章，第1、8章由张友生编写，第2、3章由唐强编写，第4章由桂阳编写，第5章由邓子云编写，第6章由米安然编写，第7章由唐平编写，第9章由何玉云编写，第10章由谢顺编写，第11章由陈志风编写，第12、13章由王勇编写。

互动讨论，专家答疑 有关本书的意见反馈和咨询，读者可在希赛网社区“书评在线”版块中的“希赛IT教育研发中心”栏目上与作者进行交流。

对于读者的提问，作者们将会在线进行解答。

如果您在学习的过程中，碰到任何问题，或者您要了解最新的考试资讯和动态，进入“软考|资格水平考试”栏目，与广大考友进行交流和讨论。

<<软件设计师考试考前串讲>>

在这里，您可以和数十万名考生进行在线交流，讨论有关学习和考试的问题，讨论人生和职业规划的话题。

希赛IT教育研发中心拥有强大的师资队伍，为您提供全程的答疑服务，在线回答您的问题。

在线测试，心中有数 学赛网在线测试系统为考生准备了在线测试，其中有数十套全真模拟试题和考前密卷，考生可选择任何一套进行测试。

测试完毕，系统自动判卷，立即给出分数。

对于您做错的地方，系统会自动记忆，待您第二次参加测试时，可选择“试题复习”，这样，系统就会自动地把您原来做错的试题显示出来，供您重新测试，以加强记忆。

如此，您可利用学赛网在线测试系统检查自己的实际水平，加强考前训练，做到心中有数，考试不慌。

致谢 本书在编写的过程中参考了许多相关的资料和书籍，在此恕不一一列举，编者在此对这些参考文献的作者表示真诚的感谢。

同时，也感谢希赛教育的软件设计师学员，他们的想法和意见使本书更加贴近读者，是他们的支持，才使希赛教育得以发展壮大，成为中国著名的IT教育品牌。

由于编者水平有限，且本书涉及的知识点多，书中难免有不妥和错误之处，编者诚恳地期望各位考生和读者不吝指教和帮助，对此，我们将深为感激。

<<软件设计师考试考前串讲>>

内容概要

本书由希赛IT教育研发中心组织编写，作为计算机技术与软件专业技术资格（水平）考试中的软件设计师级别考试辅导培训教材。

根据最新的软件设计师考试大纲，对历年考试中所有知识点进行了归类分析和总结，挖掘出了其中的考试重点和难点，指出了考试的命题方向，以及每个知识点在实际考试中所占的分数比例。

根据作者进行考试辅导和阅卷的经验，对其中的难点问题进行了详细的讨论。

考生可通过阅读本书，迅速掌握考试重点和难点，解答问题的方法和技巧，得到事半功倍的效果，提高考试通过率。

<<软件设计师考试考前串讲>>

书籍目录

请在阅读完整本书后，利用此目录进行整体回顾，可随意将您自己的重点和难点写在“Handy Thinking”上。

第1章 软件设计师考试分析	1.1 软件设计师考试简介	1.1.1 考试简介	1.1.2 考试内容和形式
1.1.3 适合人群	1.2 历次考试知识点分布	1.2.1 上午考试知识点分布	1.2.2 下午考试知识点分布
1.3 考试大纲比较分析	1.4 考试复习要点	第2章 计算机硬件基础	2.1 考点分析
2.2.1 计算机的基本组成	2.2.2 Flynn的分类	2.2.3 并行处理	2.2.4 精简指令系统计算机
2.2.5 总线和接口	2.3 数据运算	2.3.1 各种码制	2.3.2 定点数和浮点数
2.4 寻址方式	2.3.1 各种码制	2.3.2 定点数和浮点数	2.3.3 逻辑运算
2.6.4 RAID	2.5 中断	2.6.1 主存储器	2.6.2 高速缓冲存储器
2.7.1 参数计算	2.6 存储体系	2.6.1 主存储器	2.6.2 高速缓冲存储器
2.7.2 影响流水性的主要因素	2.7.1 参数计算	2.7.2 影响流水性的主要因素	2.7.3 非线性流水
2.8.1 可靠性相关概念	2.8 性能评估	2.8.1 可靠性相关概念	2.8.2 可靠性计算
2.8.2 可靠性计算	2.8.1 可靠性相关概念	2.8.2 可靠性计算	2.8.3 容错
2.8.3 容错	2.8.1 可靠性相关概念	2.8.2 可靠性计算	2.8.3 容错
2.8.4 指令周期	2.8.1 可靠性相关概念	2.8.2 可靠性计算	2.8.3 容错
3.1 考点分析	3.2 线性表	3.2.1 栈	3.2.2 队列
3.2.1 栈	3.2 线性表	3.2.1 栈	3.2.2 队列
3.2.2 队列	3.2 线性表	3.2.1 栈	3.2.2 队列
3.2.3 链表	3.2.1 栈	3.2.2 队列	3.2.3 链表
3.2.5 广义表	3.3 二叉树	3.3.1 二叉树的性质	3.3.2 二叉树的遍历
3.3.1 二叉树的性质	3.3 二叉树	3.3.1 二叉树的性质	3.3.2 二叉树的遍历
3.3.2 二叉树的遍历	3.3 二叉树	3.3.1 二叉树的性质	3.3.2 二叉树的遍历
3.3.3 二叉树的遍历	3.3 二叉树	3.3.1 二叉树的性质	3.3.2 二叉树的遍历
3.4 排序	3.3.4 最优二叉树	3.3.5 平衡二叉树	3.4 排序
3.4.1 插入排序	3.4.3 交换排序	3.4.4 归并排序	3.4.5 基数排序
3.4.2 选择排序	3.4.3 交换排序	3.4.4 归并排序	3.4.5 基数排序
3.4.3 交换排序	3.4.3 交换排序	3.4.4 归并排序	3.4.5 基数排序
3.4.4 归并排序	3.4.3 交换排序	3.4.4 归并排序	3.4.5 基数排序
3.4.5 基数排序	3.4.3 交换排序	3.4.4 归并排序	3.4.5 基数排序
3.4.6 排序算法的比较	3.5 查找	3.5.1 顺序查找	3.5.2 散列表
3.5 查找	3.5 查找	3.5.1 顺序查找	3.5.2 散列表
3.5.1 顺序查找	3.5 查找	3.5.1 顺序查找	3.5.2 散列表
3.5.2 散列表	3.5 查找	3.5.1 顺序查找	3.5.2 散列表
3.6 图	3.6.1 图的基本概念	3.6.2 图的存储结构	3.6.3 图的遍历
3.6.1 图的基本概念	3.6 图	3.6.1 图的基本概念	3.6.2 图的存储结构
3.6.2 图的存储结构	3.6 图	3.6.1 图的基本概念	3.6.2 图的存储结构
3.6.3 图的遍历	3.6 图	3.6.1 图的基本概念	3.6.2 图的存储结构
3.6.4 拓扑排序	3.6.5 最短路径	3.6.6 关键路径	3.7 常用算法设计
3.6.5 最短路径	3.6.5 最短路径	3.6.6 关键路径	3.7 常用算法设计
3.6.6 关键路径	3.6.5 最短路径	3.6.6 关键路径	3.7 常用算法设计
3.7 常用算法设计	3.7.1 算法设计	3.7.2 迭代法	3.7.3 穷举搜索法
3.7.1 算法设计	3.7 常用算法设计	3.7.2 迭代法	3.7.3 穷举搜索法
3.7.2 迭代法	3.7 常用算法设计	3.7.2 迭代法	3.7.3 穷举搜索法
3.7.3 穷举搜索法	3.7 常用算法设计	3.7.2 迭代法	3.7.3 穷举搜索法
3.7.4 递归法	3.7.4 递归法	3.7.5 分治法	3.7.6 动态规划法
3.7.5 分治法	3.7.4 递归法	3.7.5 分治法	3.7.6 动态规划法
3.7.6 动态规划法	3.7.4 递归法	3.7.5 分治法	3.7.6 动态规划法
3.7.7 回溯法	3.7.8 贪婪法	3.7.9 分支限界法	3.7.10 概率算法
3.7.8 贪婪法	3.7.8 贪婪法	3.7.9 分支限界法	3.7.10 概率算法
3.7.9 分支限界法	3.7.8 贪婪法	3.7.9 分支限界法	3.7.10 概率算法
3.7.10 概率算法	3.7.8 贪婪法	3.7.9 分支限界法	3.7.10 概率算法
第4章 操作系统	4.1 考点分析	4.2 存储管理	4.2.1 虚拟存储器的分类
4.1 考点分析	4.1 考点分析	4.2 存储管理	4.2.1 虚拟存储器的分类
4.2 存储管理	4.1 考点分析	4.2 存储管理	4.2.1 虚拟存储器的分类
4.2.1 虚拟存储器的分类	4.1 考点分析	4.2 存储管理	4.2.1 虚拟存储器的分类
4.2.2 局部性原理	4.2.1 虚拟存储器的分类	4.2.2 局部性原理	4.2.3 虚存管理
4.2.3 虚存管理	4.2.1 虚拟存储器的分类	4.2.2 局部性原理	4.2.3 虚存管理
4.3 进程管理	4.2.2 局部性原理	4.2.3 虚存管理	4.3 进程管理
4.3.1 进程的状态	4.3.1 进程的状态	4.3.2 信号量与PV操作	4.3.3 死锁问题
4.3.2 信号量与PV操作	4.3.1 进程的状态	4.3.2 信号量与PV操作	4.3.3 死锁问题
4.3.3 死锁问题	4.3.1 进程的状态	4.3.2 信号量与PV操作	4.3.3 死锁问题
4.4 文件管理	4.4.1 文件的概念	4.4.2 树形目录结构	4.4.3 存储空间管理
4.4.1 文件的概念	4.4 文件管理	4.4.1 文件的概念	4.4.2 树形目录结构
4.4.2 树形目录结构	4.4 文件管理	4.4.1 文件的概念	4.4.2 树形目录结构
4.4.3 存储空间管理	4.4 文件管理	4.4.1 文件的概念	4.4.2 树形目录结构
4.4.4 管道	4.4.3 存储空间管理	4.4.4 管道	4.5 作业管理
4.5 作业管理	4.4.3 存储空间管理	4.4.4 管道	4.5 作业管理
4.5.1 作业的概念	4.5.1 作业的概念	4.5.2 处理机调度	4.5.3 作业调度算法
4.5.2 处理机调度	4.5.1 作业的概念	4.5.2 处理机调度	4.5.3 作业调度算法
4.5.3 作业调度算法	4.5.1 作业的概念	4.5.2 处理机调度	4.5.3 作业调度算法
4.6 设备管理	4.6.1 设备管理的功能	4.6.2 设备驱动程序	4.6.3 虚拟设备
4.6.1 设备管理的功能	4.6 设备管理	4.6.1 设备管理的功能	4.6.2 设备驱动程序
4.6.2 设备驱动程序	4.6 设备管理	4.6.1 设备管理的功能	4.6.2 设备驱动程序
4.6.3 虚拟设备	4.6 设备管理	4.6.1 设备管理的功能	4.6.2 设备驱动程序
4.7 Shell程序	4.7 Shell程序	4.7 Shell程序	4.7 Shell程序
第5章 程序设计语言	5.1 考点分析	5.2 基本概念	5.3 语言的分类
5.1 考点分析	5.1 考点分析	5.2 基本概念	5.3 语言的分类
5.2 基本概念	5.1 考点分析	5.2 基本概念	5.3 语言的分类
5.3 语言的分类	5.1 考点分析	5.2 基本概念	5.3 语言的分类
5.4 控制结构	5.4.1 常量和变量	5.4.2 三种控制结构	5.4.3 函数调用
5.4.1 常量和变量	5.4 控制结构	5.4.1 常量和变量	5.4.2 三种控制结构
5.4.2 三种控制结构	5.4 控制结构	5.4.1 常量和变量	5.4.2 三种控制结构
5.4.3 函数调用	5.4 控制结构	5.4.1 常量和变量	5.4.2 三种控制结构
5.5 文法	5.5.1 文法的定义	5.5.2 文法的基本概念	5.6 自动机与正规式
5.5.1 文法的定义	5.5 文法	5.5.1 文法的定义	5.5.2 文法的基本概念
5.5.2 文法的基本概念	5.5 文法	5.5.1 文法的定义	5.5.2 文法的基本概念
5.6 自动机与正规式	5.5.2 文法的基本概念	5.6.1 有限自动机	5.6.2 正规表达式
5.6.1 有限自动机	5.6.1 有限自动机	5.6.2 正规表达式	5.6.3 正规文法
5.6.2 正规表达式	5.6.1 有限自动机	5.6.2 正规表达式	5.6.3 正规文法
5.6.3 正规文法	5.6.1 有限自动机	5.6.2 正规表达式	5.6.3 正规文法
第6章 计算机网络	6.1 考点分析	6.2 网络体系结构	6.2.1 OSI参考模型
6.1 考点分析	6.1 考点分析	6.2 网络体系结构	6.2.1 OSI参考模型
6.2 网络体系结构	6.1 考点分析	6.2 网络体系结构	6.2.1 OSI参考模型
6.2.1 OSI参考模型	6.1 考点分析	6.2 网络体系结构	6.2.1 OSI参考模型
6.2.2 TCP/IP协议族	6.2.1 OSI参考模型	6.2.2 TCP/IP协议族	6.2.3 IP地址和子网掩码
6.2.3 IP地址和子网掩码	6.2.1 OSI参考模型	6.2.2 TCP/IP协议族	6.2.3 IP地址和子网掩码
6.3 网络系统建设	6.2.2 TCP/IP协议族	6.2.3 IP地址和子网掩码	6.3 网络系统建设
6.3.1 传输介质	6.3.1 传输介质	6.3.2 网络设备	6.4 组网技术
6.3.2 网络设备	6.3.1 传输介质	6.3.2 网络设备	6.4 组网技术
6.3.3 网络系统建设	6.3.1 传输介质	6.3.2 网络设备	6.4 组网技术
6.4 组网技术	6.3.1 传输介质	6.3.2 网络设备	6.4 组网技术
6.4.1 网络系统建设	6.4.1 网络系统建设	6.4.2 网络管理工具	6.5 网络管理
6.4.2 网络管理工具	6.4.1 网络系统建设	6.4.2 网络管理工具	6.5 网络管理
6.4.3 接入网技术	6.4.2 网络管理工具	6.4.3 接入网技术	6.5 网络管理
6.5 网络管理	6.4.3 接入网技术	6.4.3 接入网技术	6.5 网络管理
6.5.1 代理服务器	6.5.1 代理服务器	6.5.2 网络管理工具	6.6 网络应用
6.5.2 网络管理工具	6.5.1 代理服务器	6.5.2 网络管理工具	6.6 网络应用
6.6 网络应用	6.5.2 网络管理工具	6.5.3 电子政务	6.6.1 可扩展置标语言
6.6.1 可扩展置标语言	6.6.1 可扩展置标语言	6.6.2 开放式网络服务体系	6.6.3 电子政务
6.6.2 开放式网络服务体系	6.6.1 可扩展置标语言	6.6.2 开放式网络服务体系	6.6.3 电子政务
6.6.3 电子政务	6.6.1 可扩展置标语言	6.6.2 开放式网络服务体系	6.6.3 电子政务
6.6.4 电子商务	6.6.3 电子政务	6.6.4 电子商务	6.6.5 万维网
6.6.5 万维网	6.6.3 电子政务	6.6.4 电子商务	6.6.5 万维网
6.6.6 电子邮件	6.6.5 万维网	6.6.6 电子邮件	6.6.7 DNS
6.6.7 DNS	6.6.5 万维网	6.6.6 电子邮件	6.6.7 DNS
6.6.8 IIS	6.6.7 DNS	6.6.8 IIS	6.6.9 VOD
6.6.9 VOD	6.6.7 DNS	6.6.8 IIS	6.6.9 VOD
第7章 数据库系统	7.1 考点分析	7.2 数据库模型	7.2.1 数据库系统的三级模式
7.1 考点分析	7.1 考点分析	7.2 数据库模型	7.2.1 数据库系统的三级模式
7.2 数据库模型	7.1 考点分析	7.2 数据库模型	7.2.1 数据库系统的三级模式
7.2.1 数据库系统的三级模式	7.2 数据库模型	7.2.1 数据库系统的三级模式	7.2.2 数据模型的分类
7.2.2 数据模型的分类	7.2.1 数据库系统的三级模式	7.2.2 数据模型的分类	7.2.3 关系模型
7.2.3 关系模型	7.2.1 数据库系统的三级模式	7.2.2 数据模型的分类	7.2.3 关系模型
7.3 关系代数	7.3.1 集合运算	7.3.2 关系运算	7.3.3 元组演算
7.3.1 集合运算	7.3 关系代数	7.3.1 集合运算	7.3.2 关系运算
7.3.2 关系运算	7.3 关系代数	7.3.1 集合运算	7.3.2 关系运算
7.3.3 元组演算	7.3.1 集合运算	7.3.2 关系运算	7.3.3 元组演算
7.4 SQL语言	7.4.1 数据表操作	7.4.2 视图操作	7.4.3 完整性约束
7.4.1 数据表操作	7.4.1 数据表操作	7.4.2 视图操作	7.4.3 完整性约束
7.4.2 视图操作	7.4.1 数据表操作	7.4.2 视图操作	7.4.3 完整性约束
7.4.3 完整性约束	7.4.1 数据表操作	7.4.2 视图操作	7.4.3 完整性约束
7.5 函数依赖与范式	7.5.1 函数依赖	7.5.2 模式分解	7.5.3 多值依赖与4NF
7.5.1 函数依赖	7.5 函数依赖与范式	7.5.1 函数依赖	7.5.2 模式分解
7.5.2 模式分解	7.5 函数依赖与范式	7.5.1 函数依赖	7.5.2 模式分解
7.5.3 多值依赖与4NF	7.5 函数依赖与范式	7.5.1 函数依赖	7.5.2 模式分解
7.6 E-R模型设计	7.6.1 E-R图的画法	7.6.2 E-R图的集成	7.6.3 E-R图向关系模式的转换
7.6.1 E-R图的画法	7.6 函数依赖与范式	7.6.1 E-R图的画法	7.6.2 E-R图的集成
7.6.2 E-R图的集成	7.6.1 E-R图的画法	7.6.2 E-R图的集成	7.6.3 E-R图向关系模式的转换
7.6.3 E-R图向关系模式的转换	7.6.1 E-R图的画法	7.6.2 E-R图的集成	7.6.3 E-R图向关系模式的转换
7.7 事务处理	7.7.1 事务的概念	7.7.2 事务的隔离性	7.7.3 事务的并发控制
7.7.1 事务的概念	7.7 事务处理	7.7.1 事务的概念	7.7.2 事务的隔离性
7.7.2 事务的隔离性	7.7 事务处理	7.7.1 事务的概念	7.7.2 事务的隔离性
7.7.3 事务的并发控制	7.7.1 事务的概念	7.7.2 事务的隔离性	7.7.3 事务的并发控制
第8章 系统开发和运行维护	8.1 考点分析	8.2 系统开发模型	8.2.1 开发生命周期模型
8.1 考点分析	8.1 考点分析	8.2 系统开发模型	8.2.1 开发生命周期模型
8.2 系统开发模型	8.1 考点分析	8.2 系统开发模型	8.2.1 开发生命周期模型
8.2.1 开发生命周期模型	8.2 系统开发模型	8.2.1 开发生命周期模型	8.2.2 系统开发方法论
8.2.2 系统开发方法论	8.2.1 开发生命周期模型	8.2.2 系统开发方法论	8.2.3 数据字典
8.2.3 数据字典	8.2.1 开发生命周期模型	8.2.2 系统开发方法论	8.2.3 数据字典
8.3 需求分析	8.3.1 需求分析概述	8.3.2 数据流图	8.3.3 数据字典
8.3.1 需求分析概述	8.3 需求分析	8.3.1 需求分析概述	8.3.2 数据流图
8.3.2 数据流图	8.3.1 需求分析概述	8.3.2 数据流图	8.3.3 数据字典
8.3.3 数据字典	8.3.1 需求分析概述	8.3.2 数据流图	8.3.3 数据字典
8.4 软件设计	8.4.1 软件设计阶段	8.4.2 软件设计活动	8.4.3 结构化设计
8.4.1 软件设计阶段	8.4 软件设计	8.4.1 软件设计阶段	8.4.2 软件设计活动
8.4.2 软件设计活动	8.4.1 软件设计阶段	8.4.2 软件设计活动	8.4.3 结构化设计
8.4.3 结构化设计	8.4.1 软件设计阶段	8.4.2 软件设计活动	8.4.3 结构化设计
8.5 程序编写	8.5.1 程序设计风格	8.5.2 程序效率	8.6 软件测试
8.5.1 程序设计风格	8.5 程序编写	8.5.1 程序设计风格	8.5.2 程序效率
8.5.2 程序效率	8.5.1 程序设计风格	8.5.2 程序效率	8.6 软件测试
8.6 软件测试	8.5.2 程序效率	8.6.1 测试的类型	8.6.2 测试的阶段
8.6.1 测试的类型	8.6.1 测试的类型	8.6.2 测试的阶段	8.6.3 性能测试
8.6.2 测试的阶段	8.6.1 测试的类型	8.6.2 测试的阶段	8.6.3 性能测试
8.6.3 性能测试	8.6.1 测试的类型	8.6.2 测试的阶段	8.6.3 性能测试
8.6.4 第三方测试	8.6.3 性能测试	8.6.4 第三方测试	8.6.5 面向对象的测试
8.6.5 面向对象的测试	8.6.3 性能测试	8.6.4 第三方测试	8.6.5 面向对象的测试
8.6.6 面向对象的测试	8.6.4 第三方测试	8.6.5 面向对象的测试	8.6.6 面向对象的测试
第9章 面向对象方法	9.1 考点分析	9.2 面向对象分析	9.2.1 Coad/Yourdon方法
9.1 考点分析	9.1 考点分析	9.2 面向对象分析	9.2.1 Coad/Yourdon方法
9.2 面向对象分析	9.1 考点分析	9.2 面向对象分析	9.2.1 Coad/Yourdon方法
9.2.1 Coad/Yourdon方法	9.2 面向对象分析	9.2.1 Coad/Yourdon方法	9.2.2 Booch方法
9.2.2 Booch方法	9.2.1 Coad/Yourdon方法	9.2.2 Booch方法	9.2.3 UML方法
9.2.3 UML方法	9.2.1 Coad/Yourdon方法	9.2.2 Booch方法	9.2.3 UML方法
9.3 面向对象语言	9.3.1 类和对象	9.3.2 封装和消息	9.3.3 多态性
9.3.1 类和对象	9.3 面向对象语言	9.3.1 类和对象	9.3.2 封装和消息
9.3.2 封装和消息	9.3.1 类和对象	9.3.2 封装和消息	9.3.3 多态性
9.3.3 多态性	9.3.1 类和对象	9.3.2 封装和消息	9.3.3 多态性

<<软件设计师考试考前串讲>>

性	9.4 统一建模语言	9.4.1 UML的结构	9.4.2 UML的图形	9.4.3 用例之间的关系	9.5 设计模式	9.5.1 设计模式的概念	9.5.2 设计模式的组成	9.5.3 设计模式
类之间的关系	9.5.4 设计模式目录的内容	第10章 多媒体基础		10.1 考点分析	10.2 多媒体基础		10.3 设计模式	
方法分类	10.2.1 多媒体计算机	10.2.2 媒体的分类	10.2.3 存储媒体	10.2.4 同步化多媒体集成语言		10.3 压缩编码技术		
准	10.3.1 数据压缩的基础	10.3.2 数据压缩技术的分类	10.3.3 数据压缩标准		10.4 音频数据		10.4.1 音频技术概述	
10.5 颜色空间	10.4.2 音频数据存储和传输	10.4.3 音频数据格式		10.5.1 亮度、色调和饱和度		10.5.2 颜色空间		10.5.3 图形与图像
数据	10.6.1 视频文件格式	10.6.2 流媒体		第11章 安全性知识		11.1 考点分析	11.2 密钥技术	
11.2.1 数据加密技术	11.2.2 认证技术	11.2.3 数字证书	11.2.4 Kerberos	11.3 安全体系		11.3.1 防火墙		
11.3.2 入侵检测	11.3.3 病毒和木马扫描		11.3.4 安全扫描		11.3.5 日志审计		11.3.6 公共密钥基础设施	
计系统	11.4 网络攻击		11.4.1 网络攻击的类型		11.4.2 常见的网络攻击		11.5 虚拟专用网	
攻击	11.5.1 VPN的基本要求		11.5.2 隧道技术		第12章 标准化知识		12.1 标准化法	
分析	12.2.1 标准的制定		12.2.2 标准的表示		12.3 软件标准		12.3.1 我国软件开发标准	
知识	12.3.2 ISO 9000标准族		12.4 构件标准		12.4.1 CORBA基础知识		12.4.2 EJB基础知识	
的保护对象	12.4.3 DCOM基础知识		第13章 知识产权知识		13.1 考点分析	13.2 专利法		13.2.1 专利法
的确定	13.2.2 确定专利权人		13.2.3 专利权		13.3 著作权法		13.3.1 著作权法客体	
竞争	13.3.2 著作权法主体		13.3.3 著作权		13.4 计算机软件保护条例		13.4.1 保护对象	
	13.4.2 软件著作权人确定		13.4.3 软件著作权		13.4.4 法律责任		13.5 反不正当竞争法	
	13.5.1 什么是反不正当竞争		13.5.2 法律责任		13.5.3 商业秘密		2008年5月软件设计师考试试题分析与解答	
							主要参考文献	

<<软件设计师考试考前串讲>>

章节摘录

第1章 软件设计师考试分析 全国计算机技术与软件专业技术资格(水平)考试(本书将简称为“软考”)已经迎来了20周年的纪念,该考试为中国的软件产业发展培养了数以万计的专业人才。

由于软考证书的含金量很高,所以得到了用人单位的广泛认可。

但是,正是因为含金量高,软考的难度比较大,所以需要考生具备扎实的理论基础和一定的实践经验。

软件设计师就是软考中的一个级别,属于中级(对应于工程师职称资格)。

1.1 软件设计师考试简介 本节就软考的历史、软件设计师考试的内容和形式,以及考试适合的人群做简单的介绍,使读者对软件设计师考试有大致地了解。

1.1.1 考试简介 软考是由国家人力资源和社会保障部、工业和信息化部组织的国家级考试,是我国IT行业的考试第一品牌,有“IT国考”之称。

其目的是科学、公正地对全国计算机技术与软件专业技术人员进行职业资格、专业技术资格认定和专业技术水平测试。

软考在全国范围内已经实施了20年,到2008年为止,累计参加考试的人数已超过200万人次。该考试由于其权威性,所以得到了社会及用人单位的广泛认同,并为推动我国信息产业特别是软件产业的发展和提高各类IT人才的素质做出了积极的贡献。

根据国家人力资源和社会保障部、工业和信息化部文件(国人部发[2003]39号),软考纳入全国专业技术人员职业资格证书制度的统一规划。

通过考试获得证书的人员,表明其已具备从事相应专业岗位工作的水平和能力,用人单位可根据工作需要从获得证书的人员中择优聘任相应专业技术职务(技术员、助理工程师、工程师、高级工程师)。

计算机技术与软件专业实施全国统一考试后,不再进行相应专业技术职务任职资格的评审工作。

因此,这种考试既是职业资格考试,又是专业技术资格考试。

报考任何级别不限学历、不限资历条件,考生可根据自己熟悉的专业情况和水平选择适当的级别报考。

程序员、软件设计师、系统分析师、网络工程师、数据库系统工程师级别的考试已与日本相应级别的考试实现互认,程序员和软件设计师级别考试还与韩国相应级别的考试实现了互认,以后还将扩大考试互认的级别以及互认的国家。

软考分5个专业类别,分别是计算机软件、计算机网络、计算机应用技术、信息系统和信息服务。

每个专业又分3个层次,分别是高级资格(高级工程师)、中级资格(工程师)和初级资格(助理工程师、技术员)。

对每个专业、每个层次,设置了若干种资格考试。

软考每年组织2次,每年上半年(一般在5月份)和下半年(一般在11月份),考试的资格层次不尽相同。

大部分资格每年举行1次考试。

初级与中级资格考试一般分上、下午两场(分别考基础知识和应用技术两个科目),高级资格考试分上午一场(综合知识科目)和下午两场(案例分析科目和论文科目)。

大部分科目考试采用笔试形式,信息处理技术员应用技术科目需要上机考试。

同一级别的各个科目全部及格才能算合格。

软件设计师属于计算机软件方向的中级资格,对应的职称资格为工程师。

1.1.2 考试内容和形式 软件设计师考试分为两个科目,分别是计算机与软件工程知识和软件设计技术。

计算机与软件工程知识一般在上午考试,考试时间为150分钟。

考试内容涉及计算机硬件、数据结构与算法、操作系统、程序设计语言、计算机网络、数据库技术、

<<软件设计师考试考前串讲>>

多媒体知识、系统开发和运行维护、安全性知识、标准化知识、知识产权、计算机专业英语等课程。试题形式为单项选择题，一共75道选择题，每道试题1分，满分为75分。

软件设计技术一般在下午考试，考试时间为150分钟，考试内容涉及数据库设计、数据流图设计、算法设计、程序设计。

试题形式为问答题和填空题，一般有7道试题，其中第1~4题为必答题，第5~7题为选答题（三道试题分别为C语言、C++语言和Java语言程序设计题，由考生选做一题），每道试题15分，满分为75分。

软件设计师考试的合格标准一般为45分，即两个科目均需达到45分，任何一个科目低于45分的，都算不合格。

1.1.3 适合人群 从2004年起，软考已经正式成为了“以考代评”的职称资格考试，也就是通过了本场考试，就可以获得国家人事部门颁发的职称资格证书，其含金量不言而喻。

从1.1.2节的考试内容与范围的分析中，读者也会发现整个知识结构完整、系统化，这对于有志于成为数据库专业人才的读者来说，其实也是一个系统化学习、提高的机会，通过考试来提升自己，实现“以考促学”，也是一件十分有意义的事。

一般来说，软件设计师考试适合于以下人群：（1）专业的计算机软件从业人员，特别是软件设计专业人员。

首先可以获得职称资格，其次又可以完善自己的知识结构，以便更好地提高水平。

（2）有志成为计算机软件专业人员的学生、爱好者。

由于软件设计师考试比较系统化、能够贴近应用，并且比较通用化，因此是这类人员的首选。

不过，要注意的是，对于这样的考生而言，实践经验缺乏，对于下午考试的通过来说是一个不小的压力。

（3）正在从事程序设计的人员。

对于程序员来说，能够逐步积累经验，往软件设计师方向发展，是一个必经的步骤。

程序员通过参加软件设计师的考试，可以一举两得，一方面学习软件设计技术，另一方面拿一个企业和社会认可的软件设计师证书。

1.2 历次考试知识点分布 从新的软考开始，到2008年5月止，软件设计师一共进行了8次考试。本节把历次考试试题进行一次系统的梳理，使读者对考试知识范围和重点有一个更加明确的理解，以便更有针对性的进行复习。

1.2.1 上午考试知识点分布 各次上午考试的知识点分布情况分别如表1-1至表1-8所示，其中表中各知识点后面括号中的数字表示该知识点的分数。

（1）数据流图设计几乎是每次考试必考的内容，主要考查平衡原则，缺少多余的数据流、数据字典条目等。

（2）数据库设计是每次考试必考的内容，主要考查E-R图、关系模式、主键、范式，个别情况考SQL语言等。

（3）面向对象设计几乎是每次考试必考的内容，主要考查用例图、顺序图、类图、多重度等。

（4）算法设计主要考查程序流程图，或者简单的算法，通常使用C语言描述。

（5）从2007年以后的试题来看，程序设计试题通常是三者选一（C语言的试题、C++语言的试题和Java语言的试题），侧重于基本语法的考查。

也就是说，在整个软件设计师考试中，C语言是必须要掌握的，而C++语言和Java语言则是可选的。有关这些程序设计语言的知识，本书不做介绍，请考生学习希赛教育的《C语言程序设计辅导》视频教程，以及《Java程序设计培训》视频教程或《面向对象程序设计（C++版）辅导》视频教程。

（6）有关新技术的应用，主要考查范围限定在《系统分析师技术指南》（张友生主编，清华大学出版社）书籍中。

1.3 考试大纲比较分析 希赛教育组织辅导老师把2004年发布的考试大纲和2008年修订的考试大纲进行了详细的比较分析，限于篇幅，具体比较过程就不再重复。

通过分析，新修订的考试大纲并没有增加和删除实质性的知识点，只是把2004版考试大纲中的知识点提炼得更加明确、更加适当而已，删除了一些冗余的叙述，修改了一些含糊的叙述，而基本知识点仍然保持不变。

<<软件设计师考试考前串讲>>

因此, 1.2节中关于考试知识点分布的分析, 完全适合2008版考试大纲的考试。

1.4 考试复习要点 根据1.2节的分析, 我们可以得出软件设计师考试中的重要考点。每次考试的试题基本上都围绕在这些考点来命题。因此, 命题中知识重复的比例和概率都很大。

根据希赛IT教育研发中心辅导和培训的经验, 在复习的时候, 考生应该从宏观层面上掌握考试的脉络。

本书的宗旨就是要对经常考查的一些知识点进行精讲, 考生结合本书进行复习将能够确保针对性。

(1) 辅导教程是基础。

考生应该紧抓试题分布要点, 针对自己的弱点, 有重点地进行复习。

并且应该通过阅读《软件设计师考试考点分析与真题详解》(希赛IT教育研发中心组编, 电子工业出版社出版)来加强自己的弱项。

该辅导教程完全按照考试大纲的要求, 把考试大纲规定的知识点逐一落实到相关章节中。

(2) 视频教程是核心。

希赛教育《软件设计师视频教程》和《软件设计师考试串讲视频教程》由希赛顾问团具有丰富的软件开发和信息系统建设经验的专家编写和录制, 该教程对考试所涉及的重要考点进行了详细的讲解, 确保考生对各章知识脉络、内容了然于胸。

在视频教程的讲解过程中, 采用了“理论+实践经验+例题解答”的方法, 整个课程生动、风趣, 可学习性极强。

(3) 历年试题是重点。

通过练习往年的试题, 对自己容易出错的地方进行往返练习, 借助《软件设计师考试试题分类精解》(希赛IT教育研发中心组编, 电子工业出版社出版)、《软件设计师考试真题解析视频》(希赛教育视频教程)来帮助自己进行消化。

在学赛网(<http://www.educity.cn>)的“在线测试”频道, 考生可以对历年试题进行在线测试, 以帮助建立考试的感觉。

(4) 希赛辅导是保证。

希赛IT教育研发中心有6年的软考辅导经验, 厚积薄发, 助学员通过考试。

希赛教育拥有权威的师资, 实行个性化辅导, 家教式服务, 老师为每一位学员制订个性化学习计划和批改作业。

希赛教育有自成体系的辅导教材和资料, 使学习更具系统性, 辅导更具针对性。

学员在学习和备考的过程中, 碰到任何问题, 都可以在线向老师提问, 老师会在线解答。

同时, 学员可以通过练习高质量的模拟试题, 查漏补缺, 发现自己的弱点, 然后有针对性地进行复习, 确保通过考试。

.....

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>