

软件设计师考试 32 小时通关

主 编 薛大龙

副主编 邹月平 吴芳茜 严洪翔 孙烈阳



★ 软考题库 - 微信搜一搜

🔍 软考达人



中国水利水电出版社
www.waterpub.com.cn

• 北京 •

内 容 提 要

软件设计师考试是全国计算机技术与软件专业技术资格考试的一个中级考试,通过考试后方可拿到软件设计师证书,根据国人部发〔2003〕39号文件指示,该证书是以考代评的中级职称证书。

与普通的软考教材相比,本书在保证知识的系统性与完整性的基础上,在易学性、学习有效性等方面进行了大幅度的改进和提高。本书在全面分析知识点的基础上,对整个学习架构进行了科学重构,可以极大地提高学习的有效性;同时在每章的最后,配备了与本章内容相关的典型章节练习题;在此基础上,本书还配备了历年真题(包括典型单选题及分析、典型案例题及分析,分别应对软件设计师考试的两科考试),一站式解决考生们学习及练习的问题。考生可通过学习本书,抓住考试的重点,熟悉试题形式,掌握解答问题的方法和技巧等。

本书可作为考生备考软考中级资格“软件设计师”的学习教材,也可供各类培训班使用。

图书在版编目(CIP)数据

软件设计师考试32小时通关 / 薛大龙主编. -- 北京:
中国水利水电出版社, 2019.1
ISBN 978-7-5170-7012-2

I. ①软… II. ①薛… III. ①软件设计—资格考试—
自学参考资料 IV. ①TP311.5

中国版本图书馆CIP数据核字(2018)第234090号

策划编辑: 周春元 责任编辑: 周春元 加工编辑: 张天娇

书 名	软件设计师考试 32 小时通关 RUANJIAN SHEJISHI KAOSHI 32 XIAOSHI TONGGUAN
作 者	主 编 薛大龙 副主编 邹月平 吴芳茜 严洪翔 孙烈阳
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市鑫金马印装有限公司
规 格	184mm×240mm 16 开本 24.5 印张 582 千字
版 次	2019 年 1 月第 1 版 2019 年 1 月第 1 次印刷
印 数	0001—3000 册
定 价	68.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

序

大家常用“悲催”形容 IT 技术人，因为“一入 IT 深似海，从此学习无绝期”。但 IT 技术人员也是幸运的，因为在这个行业里，只要坚持学习，路永远就在脚下，梦想就在不远的地方——这是个最坏的年代，也是个最好的年代。

学习真的能改变命运吗？答案是肯定的。随着大数据、人工智能和深度学习的时代到来，中国出现了一个冲向全球 IT 产业巅峰的最佳时机，各条战线都在加大 IT 的投入，全力拥抱“互联网+”的浪潮。数据显示，按照目前的产业规划，中国目前 IT 高端技术人才的缺口高达近百万！可以说，经历了十余年辛勤的播种、耕耘，IT 技术人员已经迎来了郁郁葱葱的盛夏。

当然，盛夏的果实不是那么容易就结出来的。IT 发展到今天，技术的复杂度已经远非当年可比，从业者的学习压力更大，学习需求也更迫切，学习的容量和深度也和十年前不可同日而语。但同时，移动时代的来临使得随时随地学习成为可能，技术平台的发展使得“以人为本”的学习环境逐步深入人心。如何能够帮助 IT 技术人员随时随地、用最适合自己的方式学习，这是摆在所有 IT 技术人才培养从业者面前的一个必须解决的问题，也是 51CTO 学院一直孜孜探索的核心方向。

十二年来，51CTO 有幸和众多技术专家、讲师、机构一起用汗水铸成一个个视频课程、一篇篇经验文章、一本本技术书籍，为近 1500 万的注册用户提供了站式技术到家的学习服务，为中国 IT 技术人员的提升尽自己的一份力，也为帮助中国 IT 产业的快速推进贡献了自己的价值。

更为重要的是，在此过程中，51CTO 学院积累了许多关于 IT 技术人员提升和学习的新经验、新方法。以薛大龙老师的计算机技术与软件专业技术资格考试（以下简称软考）系列课程为例，目前已经陆续上线了视频课程（学员自行使用各种平台随时随地学习）、在线直播课程、微职位培训班（班级化体系学习）等多种形式，形成了一整套有效帮助学员提升技术水平、提高软考通过率的学习方法，三年多来，已经有效帮助了十几万 IT 技术人员。如今，我们在阅读 IT 技术书籍的时候依然是“宅”在家，但作者和老师也许就在身边，打开手机就可以面对面交流！

这样的探索，只是当前互联网学习模式发展的一个缩影。数千名和薛大龙老师同样默默奉献的老教师们，以及同 51CTO 学院一起并肩前行的学习平台，都在努力为中国 IT 技术人员的学习和提升贡献自己的力量，帮助他们用技术成就梦想。在我们看来，这已经不是一个个体、一个团队或一家公司的事情，而是事关国家发展、中华民族复兴的大事！

51CTO 副总裁、51CTO 学院联合创始人 邱文平

前言

为什么选择本书

软考历年的全国平均通过率一般不超过 20%。考试所涉及的知识范围较广，而考生一般又多忙于工作，仅靠官方教程，考生在有限的时间内很难领略及把握考试的重点、难点。

作为软考的命题人和阅卷人，多年来潜心研究软考的知识体系，对历年的软考试题进行了深入的分析、归纳与总结，并把这些规律性的东西融入到软考培训的教学当中，取得了非常显著的效果。但限于各方面条件，能够参加面授的考生还是相对少数，为了能让更多考生分享到我们的一些经验与成果，本人组织编写了本书。本书具有以下几个特点：

- **青出于蓝：**本书保留了普通教材知识的系统性和完整性的特点，但在易学性、学习有效性等方面进行了大幅度的改进和提高。
- **有的放矢：**通过对考试大纲的细致分析，本书让考试中的重点、难点及考生们在学习过程中容易忽略的知识点暴露无遗。
- **超高效率：**本书把我们团队中多名杰出讲师的软考教学经验、多年试题研究及命题规律经验融合在一起，形成了 32 小时超强学习架构。
- **一站式解决：**本书在以上基础上，还增加了典型单选题及解析、典型案例题及分析等众多内容，同时具备了教材与实战的功能。

本书作者不一般

本书由薛大龙担任主编，邹月平、吴芳茜、严洪翔、孙烈阳担任副主编。具体编写分工如下：孙烈阳负责第 1、10、13 小时，严洪翔负责第 2、7、8、9 小时，邹月平负责第 3~6、14、27~32 小时，吴芳茜负责第 11、12 小时，薛大龙负责第 15~26 小时。为了帮助考生们加强重要性高、代表性强、命题频率大、学一可得三的知识点，每章还配备了练习题。本书由薛大龙确定架构，由严洪翔统稿，由邹月平初审，由薛大龙终审。

薛大龙，北京理工大学博士研究生，多所大学客座教授，北京市评标专家，全国计算机技术与软件专业技术资格考试辅导教材编委会主任，多次参与全国软考的命题与阅卷，作为规则制定者非常熟悉命题要求、命题形式、命题难度、命题深度、命题重点及判卷标准等。

邹月平，面授名师、高级工程师、信息系统项目管理师、一级建造师、全国计算机技术与软件专业技术资格考试辅导教材编委会副主任。拥有多年培训经验，以其语言简练、逻辑清晰、善于在试题中把握要点、总结规律、提纲挈领，快速掌握知识要点，深得学员好评。

吴芳茜，高级工程师，北京市评标专家，曾担任中国软件评测中心事业部总经理，承担项

目管理与 IT 服务工作的电子政务、电子党务工程累计投资额超过 10 亿，多次参与全国软考试题支撑、教材编制及公开授课工作，参与国家标准《GB/T 19668.1-2014 信息技术服务 监理 第 1 部分：总则》的编写工作。

严洪翔，毕业于西安电子科技大学、浙江大学 MBA，23 年 IT “老兵”，高级工程师，信息系统项目管理师，就职于北明软件有限公司，组织、主持大型软件项目开发达数十项，具备丰富的研发管理经验。担任书籍《系统分析师考试 32 小时通关》副主编，参与书籍《系统架构设计师考试 32 小时通关》的编写。

孙烈阳，全国计算机技术与软件专业技术资格考试辅导教材编委会委员，就职于黄河水利委员会信息中心，曾作为项目总监、项目经理参与多个项目的技术与管理工作经验，具有丰富的项目技术和管理经验。担任书籍《系统分析师考试 32 小时通关》副主编，参与书籍《系统架构设计师考试 32 小时通关》的编写。

另外，参与本书编写和审核的还有刘阳、何鹏涛、兰帅辉、李海龙、李莉莉等专家，在此一并表示感谢。

给读者的学习提示

每天 1 小时，共 32 小时的时间，对我们每个人来说，都是挑战，尤其是对于离开学校多年的读者。但是，如果你坚持下来了，你会发现自己每天都有一个小的提升，通过量的积累，最终会有一个质的飞跃。尤其是拿到证书的喜悦心情，获得高级职称的自豪感，会让自己感觉到所有的努力都是值得的。

王国维在《人间词话》中说，古今之成大事业、大学问者，必经过三种境界：昨夜西风凋碧树，独上高楼，望尽天涯路，此第一境也；衣带渐宽终不悔，为伊消得人憔悴，此第二境也；众里寻他千百度，蓦然回首，那人却在，灯火阑珊处，此第三境也。

给我们的启示是：

（1）要确立一个目标，就是一次通过考试，千万不要想着这次不过还有下次，今天不学还有明天。然而，明日复明日，明日何其多。只有确立了这样的目标，你才会真正空出时间或挤出时间，来进行每天至少 1 小时的学习。

（2）为了通过考试，一定要努力、努力、再努力，考试不外乎是理解加记忆。想想当年的高考，那么多的数理化知识点，几千个英语单词，多少现代文和文言文，我们都记下来了，软考的知识点还不及高考的十分之一，经过考前 1~2 个月的突击是能够通过的，它比高考容易得多。

（3）当你学完本书，并把书中的知识点或理解，或记忆，或融会贯通后，你会发现，蓦然回首，你在获得证书的同时，还增加了这方面的知识和能力。

技术成就梦想，相信自己，只需努力！

致谢

感谢中国水利水电出版社万水分社周春元副总经理和张天娇编辑在本书的策划、选题的申报、写作大纲的确定以及编辑、出版等方面付出的辛勤劳动和智慧，给予了我们很多帮助。

本书适合谁

本书可以作为考生备考软考“软件设计师”的学习教材，也可以供各类培训班使用。考生可通过学习本书，掌握考试的重点，熟悉试题形式及解答问题的方法和技巧等。有关软件设计师考试的学习，读者可以加入读者群（QQ 群号：645979537）与作者面对面交流，如有疑问或建议，也可以发邮件至电子邮箱 PYXDL@163.com 与我们交流，我们会及时地回复。

编 者

2018 年 8 月

目 录

序
前言

第一篇 基础篇

第 1 小时 计算机系统知识	2	第 3 小时 数据结构与数据运算	33
1.0 章节考点分析	2	3.0 章节考点分析	33
1.1 计算机系统基础知识	3	3.1 线性结构	34
1.1.1 计算机系统硬件基本组成	3	3.1.1 线性表	34
1.1.2 CPU 的功能与组成	3	3.1.2 线性表的存储结构	34
1.1.3 数据表示	6	3.1.3 栈	35
1.1.4 校验码	6	3.1.4 队列	35
1.2 计算机体系结构	7	3.1.5 串	36
1.2.1 计算机体系结构的发展	7	3.2 非线性结构	37
1.2.2 存储系统	10	3.2.1 二维数组	37
1.2.3 输入/输出技术	14	3.2.2 三对角矩阵	37
1.2.4 总线结构	16	3.2.3 树	38
1.3 计算机安全	16	3.2.4 图	41
1.3.1 计算机安全概述	16	3.3 数据运算	42
1.3.2 加密技术和认证技术	18	3.3.1 查找	42
1.4 练习题	20	3.3.2 排序	43
第 2 小时 程序设计语言基础知识	24	3.4 练习题	46
2.0 章节考点分析	24	第 4 小时 操作系统知识	48
2.1 程序语言的基本概念	25	4.0 章节考点分析	48
2.2 程序设计的基本成分	25	4.1 OS 概述	50
2.3 汇编程序基本原理	26	4.2 进程管理	50
2.4 编译程序基本原理	27	4.3 存储管理	50
2.5 解释程序基本原理	30	4.4 设备管理	51
2.6 练习题	31	4.5 文件管理	51

4.6 作业管理	52	5.5 软件项目系统测试	66
4.7 练习题	52	5.5.1 系统测试与调试	67
第 5 小时 软件工程基础知识	56	5.5.2 传统软件的测试策略	67
5.0 章节考点分析	56	5.5.3 测试方法	68
5.1 软件工程概述	57	5.5.4 调试	68
5.1.1 计算机软件	57	5.6 软件项目管理	68
5.1.2 软件工程基本原理	57	5.6.1 项目管理涉及的范围	68
5.1.3 软件生存周期	57	5.6.2 项目估算	69
5.1.4 软件过程	58	5.6.3 进度管理	69
5.2 软件过程模型	60	5.6.4 软件项目组织	70
5.2.1 瀑布模型	60	5.6.5 软件质量管理	71
5.2.2 增量模型	61	5.6.6 软件配置管理	72
5.2.3 演化模型	62	5.6.7 软件风险管理	73
5.2.4 喷泉模型	62	5.7 软件度量	73
5.2.5 基于构件的开发模型	64	5.8 练习题	74
5.2.6 形式化方法模型	64	第 6 小时 结构化开发方法	78
5.2.7 统一过程模型	64	6.0 章节考点分析	78
5.3 软件项目需求分析	65	6.1 系统分析与设计概述	79
5.3.1 软件需求	65	6.2 结构化分析方法	82
5.3.2 需求分析原则	66	6.3 结构化设计方法	82
5.3.3 需求工程	66	6.4 WebApp 分析与设计	83
5.4 软件项目系统设计	66	6.5 用户界面设计	83
5.4.1 概要设计	66	6.6 练习题	83
5.4.2 详细设计	66		

第二篇 进阶篇

第 7 小时 面向对象技术	86	7.2 UML	91
7.0 章节考点分析	86	7.2.1 事物	91
7.1 面向对象基础	88	7.2.2 关系	91
7.1.1 面向对象基础的概念	88	7.2.3 UML 中的图	92
7.1.2 面向对象分析	89	7.3 设计模式	98
7.1.3 面向对象设计	89	7.3.1 设计模式的要素	98
7.1.4 面向对象程序设计	89	7.3.2 创建型设计模式	98
7.1.5 面向对象测试	90	7.3.3 结构型设计模式	98

7.3.4 行为设计模式	98	10.2.2 网络传输介质	125
7.4 练习题	98	10.2.3 TCP/IP 协议族	125
第 8 小时 算法设计与分析	102	10.2.4 Internet 及应用	127
8.0 章节考点分析	102	10.3 网络安全	128
8.1 算法的基本概念	104	10.3.1 网络安全概述	128
8.2 算法的分析基础	104	10.3.2 网络信息安全	128
8.3 分治法	105	10.3.3 防火墙技术	128
8.4 动态规划法	105	10.3.4 信息系统安全等级	131
8.5 贪心法	105	10.3.5 信息安全系统三维空间	132
8.6 回溯法	106	10.3.6 入侵检测系统	132
8.7 分支限界法	106	10.4 练习题	132
8.8 概率算法	106	第 11 小时 标准化和软件知识产权基础知识 ..	137
8.9 近似算法	106	11.0 章节考点分析	137
8.10 NP 完全性理论	106	11.1 标准化基础知识	138
8.11 练习题	107	11.2 知识产权基础知识	141
第 9 小时 数据库技术基础	109	11.3 练习题	145
9.0 章节考点分析	109	第 12 小时 软件系统分析与设计	146
9.1 数据库模型	110	12.0 章节考点分析	146
9.2 数据模型	111	12.1 结构化分析与设计	147
9.3 数据库操作	112	12.2 数据库分析与设计	147
9.4 数据库语言 SQL	114	12.3 面向对象分析与设计	150
9.5 数据库管理系统	115	12.4 算法分析与设计	151
9.6 关系数据库的规范化	116	12.5 面向过程的程序设计与实现	152
9.7 数据库的控制功能	116	12.6 面向对象的程序设计与实现	152
9.8 数据仓库和分布式数据库基础知识	118	12.7 练习题	152
9.9 练习题	119	第 13 小时 新技术	155
第 10 小时 网络与信息安全基础知识	122	13.0 章节考点分析	155
10.0 章节考点分析	122	13.1 大数据	156
10.1 网络概述	123	13.2 云计算	157
10.1.1 计算机网络的概念	123	13.3 物联网	157
10.1.2 计算机网络的分类	123	13.4 移动互联网	159
10.1.3 网络拓扑结构	123	13.5 工业互联网	160
10.1.4 ISO/OSI 网络体系结构	123	13.6 人工智能	160
10.2 网络设备及网络协议与标准	124	13.7 区块链	163
10.2.1 网络设备	125	13.8 练习题	163

第 14 小时 专业英语	167	14.3 开发方法	169
14.0 章节考点分析	167	14.4 新技术	170
14.1 软件文档	167	14.5 练习题	170
14.2 非功能需求	168		

第三篇 真题篇

第 15 小时 2016 年 5 月考试真题		试题六 (15 分)	224
(上午基础知识)	173	第 20 小时 2016 年 11 月真题解析	226
第 16 小时 2016 年 5 月考试真题		上午基础知识	226
(下午案例分析)	184	下午案例分析	236
试题一 (15 分)	184	试题一 (15 分)	236
试题二 (15 分)	186	试题二 (15 分)	236
试题三 (15 分)	187	试题三 (15 分)	237
试题四 (15 分)	188	试题四 (15 分)	237
试题五 (15 分)	190	试题五 (15 分)	237
试题六 (15 分)	192	试题六 (15 分)	238
第 17 小时 2016 年 5 月真题解析	194	第 21 小时 2017 年 5 月考试真题	
上午基础知识	194	(上午基础知识)	239
下午案例分析	202	第 22 小时 2017 年 5 月考试真题	
试题一 (15 分)	202	(下午案例分析)	251
试题二 (15 分)	202	试题一 (15 分)	251
试题三 (15 分)	203	试题二 (15 分)	253
试题四 (15 分)	204	试题三 (15 分)	254
试题五 (15 分)	204	试题四 (15 分)	256
试题六 (15 分)	205	试题五 (15 分)	258
第 18 小时 2016 年 11 月考试真题		试题六 (15 分)	260
(上午基础知识)	206	第 23 小时 2017 年 5 月真题解析	262
第 19 小时 2016 年 11 月考试真题		上午基础知识	262
(下午案例分析)	216	下午案例分析	271
试题一 (15 分)	216	试题一 (15 分)	271
试题二 (15 分)	217	试题二 (15 分)	271
试题三 (15 分)	219	试题三 (15 分)	272
试题四 (15 分)	220	试题四 (14 分)	273
试题五 (15 分)	222	试题五 (16 分)	273

试题六（16分）	273	试题六（15分）	330
第24小时 2017年11月考试真题		第32小时 2018年5月真题解析	333
（上午基础知识）	274	上午基础知识	333
第25小时 2017年11月考试真题		下午案例分析	341
（下午案例分析）	284	试题一（15分）	341
试题一（15分）	284	试题二（15分）	341
试题二（15分）	286	试题三（15分）	342
试题三（15分）	288	试题四（15分）	343
试题四（15分）	289	试题五（15分）	343
试题五（15分）	291	试题六（15分）	344
试题六（15分）	292	第30小时 2018年11月考试真题	
第26小时 2017年11月真题解析	294	（上午基础知识）	345
上午基础知识	294	第31小时 2018年11月考试真题	
下午案例分析	307	（下午案例分析）	356
试题一（共15分）	307	试题一（15分）	356
试题二（共15分）	308	试题二（15分）	358
试题三（15分）	309	试题三（15分）	359
试题四（15分）	309	试题四（15分）	360
试题五（共15分）	310	试题五（15分）	362
试题六（共15分）	310	试题六（15分）	364
第27小时 2018年5月考试真题		第32小时 2018年11月真题解析	367
（上午基础知识）	311	上午基础知识	367
第28小时 2018年5月考试真题		下午案例分析	377
（下午案例分析）	323	试题一（15分）	377
试题一（15分）	323	试题二（15分）	377
试题二（15分）	325	试题三（15分）	378
试题三（15分）	326	试题四（15分）	378
试题四（15分）	328	试题五（15分）	379
试题五（15分）	329	试题六（15分）	379

第一篇

基础篇

第1小时

计算机系统知识

1.0 章节考点分析

第1小时主要学习计算机系统基础知识、计算机体系结构、安全性、可靠性与系统性能评测基础知识等内容。

根据考试大纲，本小时的知识点会涉及单选题型和案例分析题型，约占2~8分。本小时的内容偏重于概念知识，根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，考查概念的知识点多数参照教材，扩展内容较少。本小时的架构如图1-1所示。

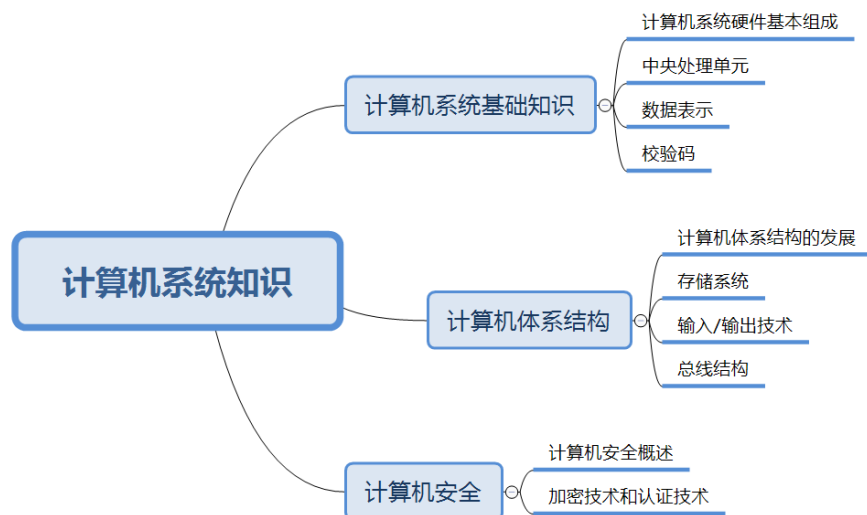


图 1-1 第1小时架构图



【导读小贴士】

计算机系统知识是最基础的知识点，在软件设计师考试中，一些选择题、案例分析题会来自本小时，除了理解的基础上掌握核心知识点外，还要学会灵活应用。

1.1 计算机系统基础知识

1.1.1 计算机系统硬件基本组成

计算机的基本硬件系统由运算器、控制器、存储器、输入设备和输出设备五大部件组成。运算器、控制器等部件被集成在一起统称为中央处理单元（Central Processing Unit, CPU）。存储器是计算机系统中的记忆设备，分为内部存储器和外部存储器。前者速度高、容量小，一般用于临时存放程序、数据及中间结果；而后者容量大、速度慢，可长期保存程序和数据。输入设备和输出设备合称为外部设备（简称外设），输入设备用于输入原始数据及各种命令，而输出设备则用于输出处理结果。

1.1.2 CPU 的功能与组成

1. CPU 的功能

CPU 的功能如图 1-2 所示。

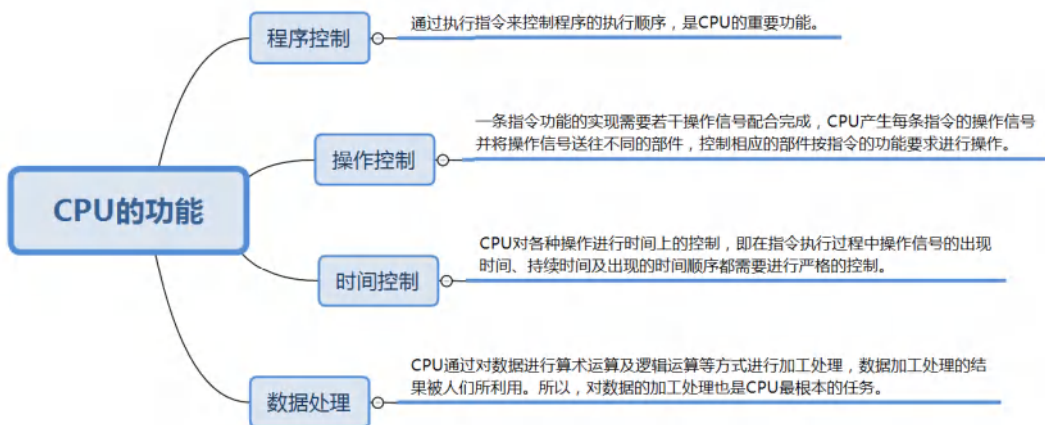


图 1-2 CPU 的功能

2. CPU 的组成

CPU 主要由运算器、控制器、寄存器组和内部总线等部件组成，如图 1-3 所示。

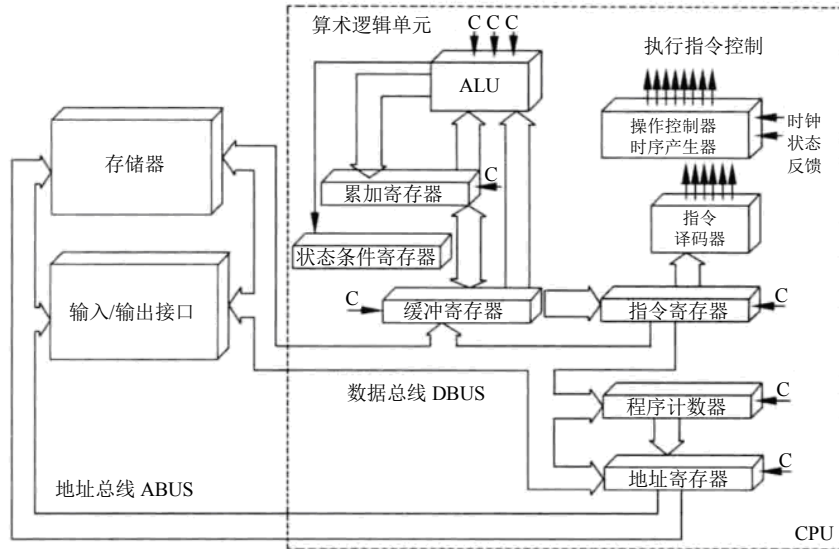


图 1-3 CPU 基本组成结构示意图

(1) 运算器。运算器的组成如图 1-4 所示。

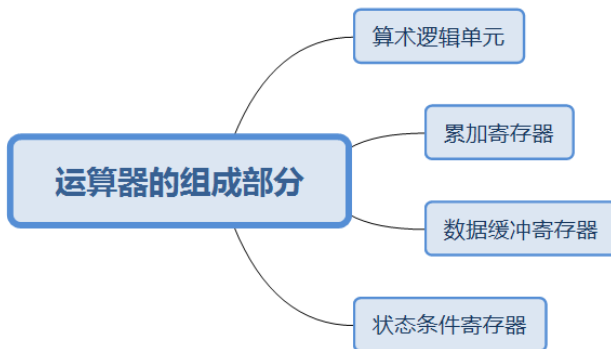


图 1-4 运算器的组成部分

运算器的主要功能如图 1-5 所示。

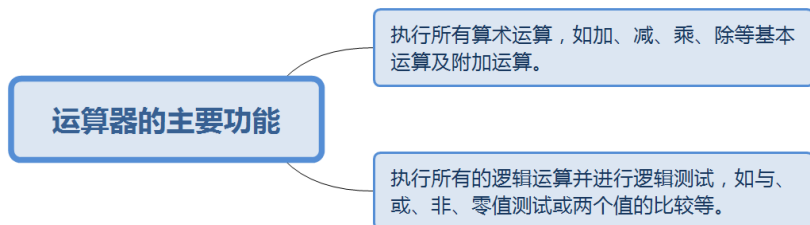


图 1-5 运算器的主要功能

运算器中各组成部件的功能如图 1-6 所示。

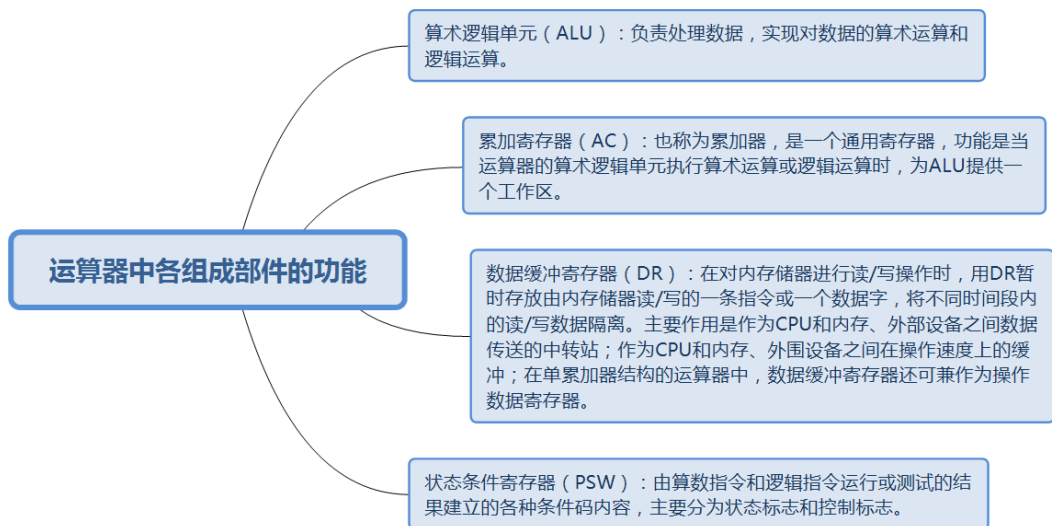


图 1-6 运算器中各组成部件的功能

(2) 控制器。运算器只能完成运算, 而控制器用于控制整个 CPU 的工作, 它决定了计算机运行过程的自动化。它不仅要保证程序的正确执行, 而且要能够处理异常事件。一般包括指令控制逻辑、时序控制逻辑、总线控制逻辑和中断控制逻辑等几个部分。

指令控制逻辑要完成取指令、分析指令和执行指令的操作, 过程分为取指令、指令译码、按指令操作码执行、形成下一条指令地址等步骤。

1) 指令寄存器 (IR)。当 CPU 执行一条指令时, 先把它从内存取到缓冲寄存器中, 再送入 IR 暂存, 指令译码器根据 IR 的内容产生各种微操作指令, 控制其他的组成部件工作, 完成所需的功能。

2) 程序计数器 (PC)。具有寄存信息和计数两种功能, 又称为指令计数器。程序的执行分为两种情况, 一是顺序执行, 二是转移执行。

3) 地址寄存器 (AR)。保存当前 CPU 所访问的内存单元的地址。

4) 指令译码器 (ID)。指令分为操作码和地址码两个部分, 为了执行任何给定的命令, 必须对操作码进行分析, 以便识别所有完成的操作。

时序控制逻辑要为每条指令按时间顺序提供应有的控制信号。总线逻辑是为多个功能部件服务的信息通路的控制电路。中断控制逻辑用于控制各种中断请求, 并根据优先级的高低对中断请求进行排队, 逐个交给 CPU 处理。

(3) 寄存器组。寄存器组分为专用寄存器和通用寄存器。运算器和控制器中的寄存器是专用寄存器, 其作用是固定的。通用寄存器的用途广泛, 并且由程序员规定其用途, 其数目因处理器的不同有所差异。

3. 多核 CPU

核心又称为内核，是 CPU 最主要的组成部分。CPU 所有的计算、接受/存储命令、处理数据都由核心执行。各种 CPU 核心都具有固定的逻辑结构，一级缓存、二级缓存、执行单元、指令级单元和总线接口等逻辑单元都会有合理的布局。

CPU 主要厂商 AMD 和 Intel 的双核技术在物理结构上有很大的不同。AMD 将两个内核做在一个 Die 上，通过直接架构连接起来，集成度更高；Intel 则是将放在不同核心上的两个内核封装在一起。因此，有人将 Intel 的方案称为“双芯”，将 AMD 的方案称为“双核”。

1.1.3 数据表示

各种数值在计算机中表示的形式称为机器数，特点是采用二进制计数制，数的符号用 0 和 1 表示，小数点则隐含，表示不占位置。机器数对应的实际数值称为数的真值。

设机器字长为 n ，各种码制下带符号数的范围如表 1-1 所示。

表 1-1 设机器字长为 n ，各种码制下带符号数的范围

码制	定点整数	定点小数
原码	$-(2^{n-1}-1) \sim +(2^{n-1}-1)$	$-(1-2^{-(n-1)}) \sim +(1-2^{-(n-1)})$
反码	$-(2^{n-1}-1) \sim +(2^{n-1}-1)$	$-(1-2^{-(n-1)}) \sim +(1-2^{-(n-1)})$
补码	$-2^{n-1} \sim +(2^{n-1}-1)$	$-1 \sim +(1-2^{-(n-1)})$
移码	$-2^{n-1} \sim +(2^{n-1}-1)$	$-1 \sim +(1-2^{-(n-1)})$

1.1.4 校验码

在计算机系统运行时，各部位之间要进行数据交换，为了确保数据在传送过程中正确无误，一是提高硬件电路的可靠性，二是提高代码的校验能力，包括查错和纠错。

常用的三种校验码如图 1-7 所示。

1. 奇偶校验码

奇偶校验码 (Parity Code) 是一种简单有效的校验方法。这种方法通过在编码中增加一位校验位，使编码中 1 的个数为奇数 (奇校验) 或偶数 (偶校验)，从而使码距变为 2。

2. 海明码

海明码 (Hamming Code) 是利用奇偶性来查错和纠错的校验方法。

3. 循环冗余校验码

循环冗余校验码 (Cyclic Redundancy Check, CRC) 由两部分组成，左边为信息码 (数据)，右边为校验码，如图 1-8 所示。

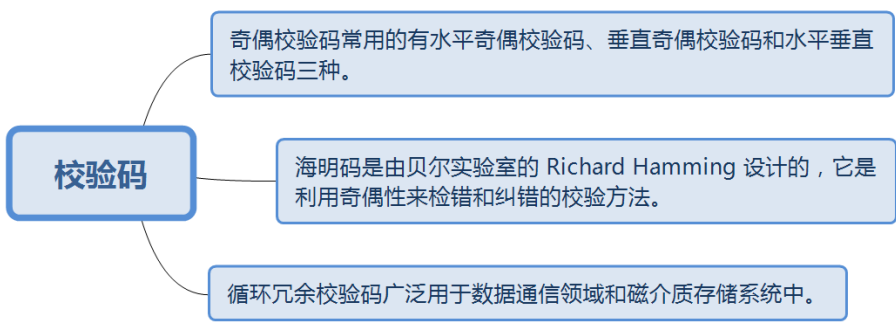


图 1-7 常用的三种校验码

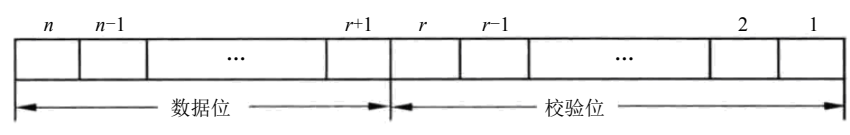


图 1-8 CRC 的代码格式

1.2 计算机体系结构

1.2.1 计算机体系结构的发展

1. 计算机系统结构概述

1964 年，阿姆达尔（G.M.Amdahl）在介绍 IBM360 系统时指出，计算机体系结构是站在程序员的角度所看到的计算机属性。

1982 年，梅尔斯（G.J.Myers）在其所著的《计算机体系结构的进展》一书中定义了组成计算机系统的若干层次。

1984 年，拜尔（J.L.Baer）在一篇题为《计算机体系结构》的文章中给出了一个含义更加广泛的定义：体系结构是由结构、组织、实现、性能四个基本方面组成的。

计算机体系结构、计算机组织和计算机实现三者的关系，如图 1-9 所示。

2. 计算机体系结构分类

- （1）从宏观上按处理机的数量分类，如图 1-10 所示。
- （2）从微观上按并行程度分类，如图 1-11 所示。

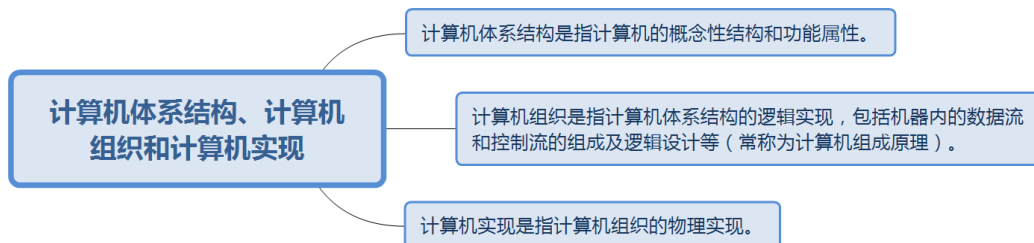


图 1-9 计算机体系结构、计算机组织和计算机实现

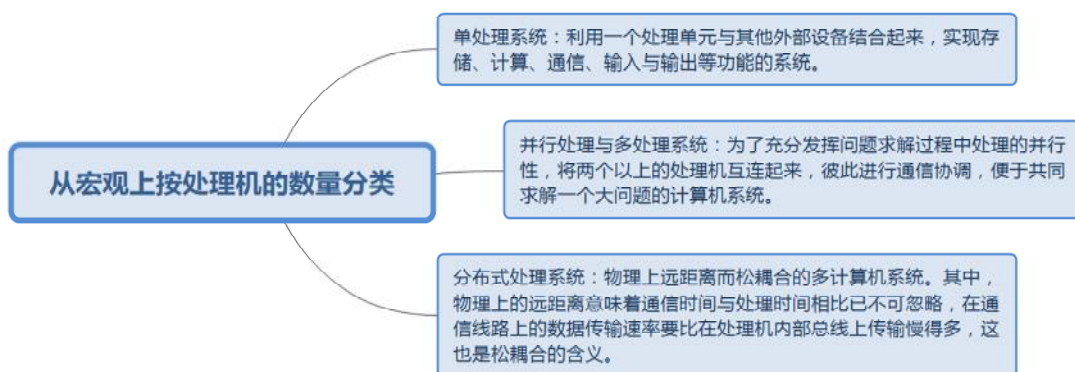


图 1-10 从宏观上按处理机的数量分类

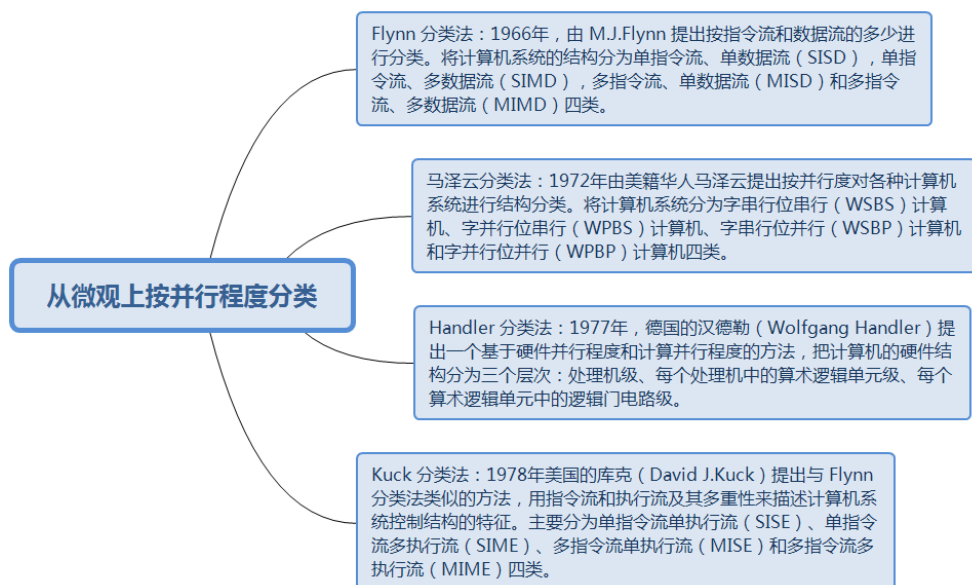


图 1-11 从微观上按并行程度分类

3. 指令系统

(1) 指令集体系结构的分类如图 1-12 所示。

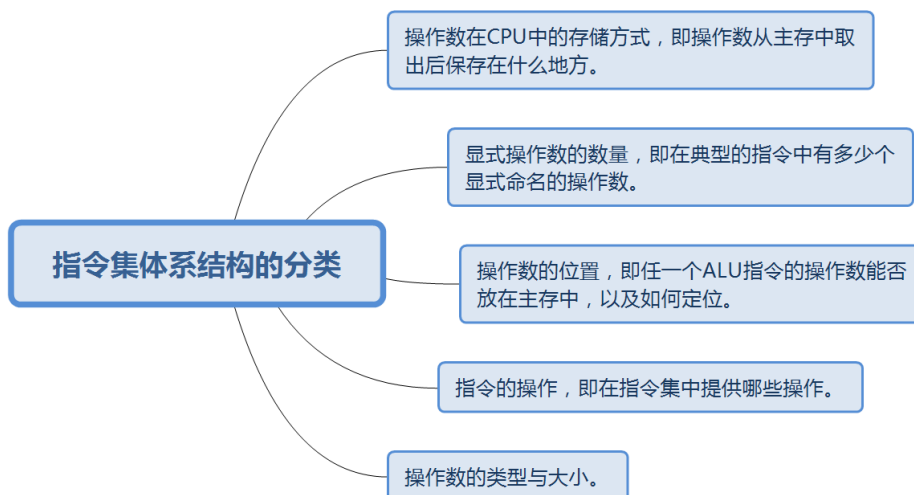


图 1-12 指令集体系结构的分类

(2) 复杂指令集计算机（CISC）的主要弊端如图 1-13 所示。

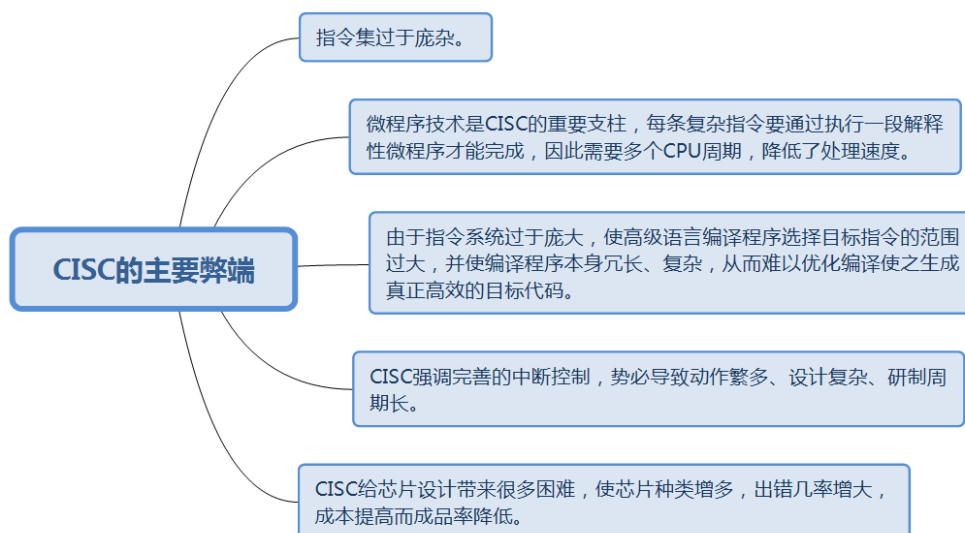


图 1-13 CISC 的主要弊端

(3) 指令系统的优化。面向高级语言的优化思路是尽可能缩小高级语言与机器语言之间的语义差异。面向操作系统的优化思路是进一步缩小操作系统与体系架构之间的语义差异。

精简指令集计算机（RISC）的关键技术如图 1-14 所示。

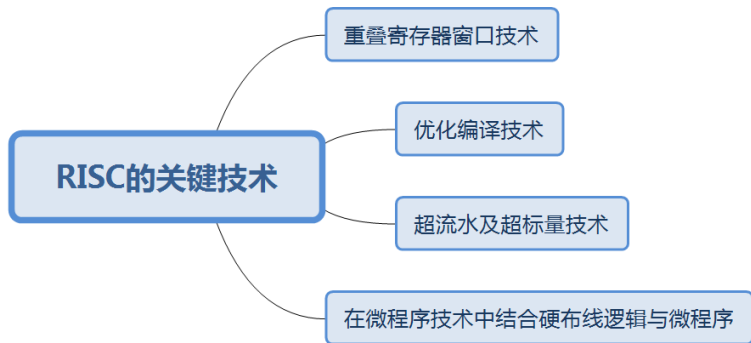


图 1-14 RISC 的关键技术

指令的流水处理如图 1-15 所示。

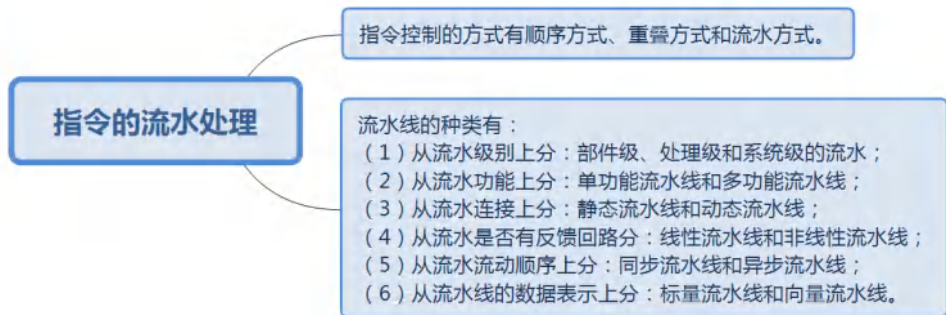


图 1-15 指令的流水处理

RISC 采用的流水技术有三种：超流水线、超标量和超长指令字。

（4）阵列处理机、并行处理机和多处理机的区别如下所述：

1）阵列处理机。将重复设置的多个处理单元（PU）按一定的方式连成阵列，在单个控制部件（CU）的控制下，对分配给自己的数据进行处理，并行地完成一条指令所规定的操作。

2）并行处理机。SIMD、MIMD 是典型的并行计算机，SIMD 有共享存储器和分布存储器两种形式，如图 1-16 所示。

3）多处理机。由多台处理机组成的系统，每台处理机有属于自己的控制部件，可执行独立的程序，共享一个主存储器和所有外部设备。

1.2.2 存储系统

1. 存储器的层次结构

存储器的层次结构如图 1-17 所示。

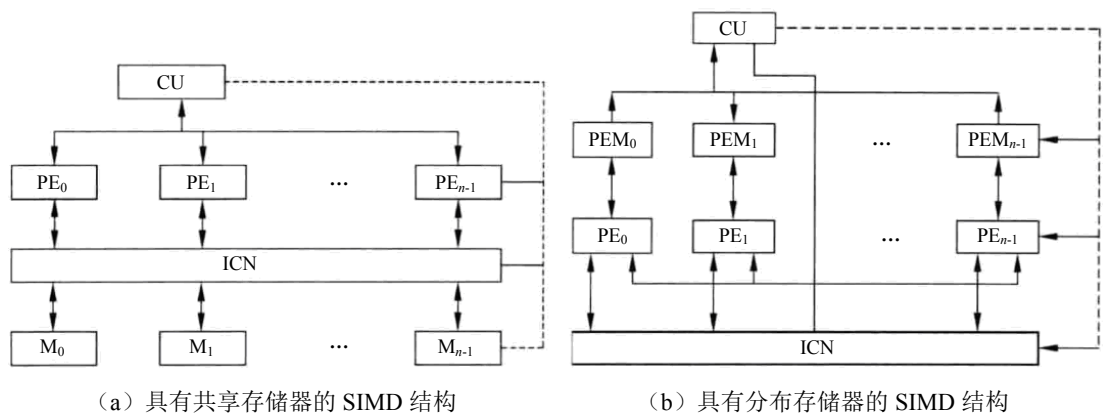


图 1-16 SIMD 结构

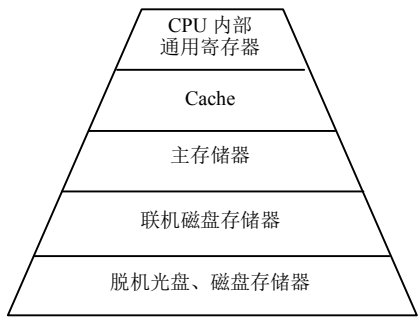


图 1-17 存储器的层次结构

2. 存储器的分类

存储器的分类如图 1-18 所示。

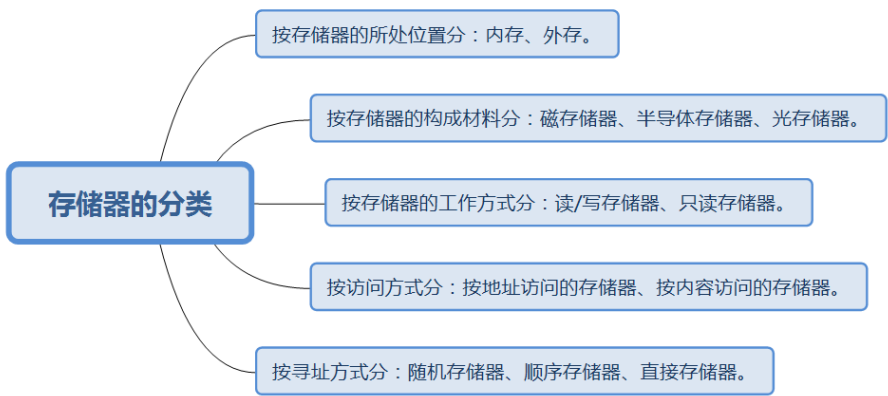


图 1-18 存储器的分类

3. 相联存储器

相联存储器是一种按内容访问的存储器，其结构如图 1-19 所示。

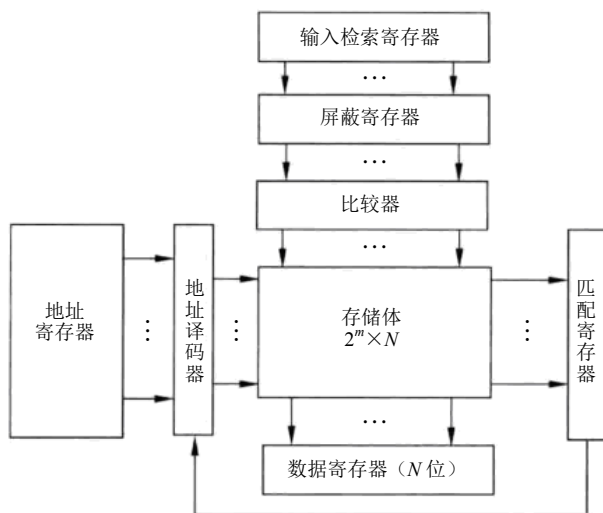


图 1-19 相联存储器结构示意图

4. 高速缓存

高速缓存（Cache）的组成部分、地址映像方法、替换算法、性能分析和多级 Cache 如图 1-20 所示。

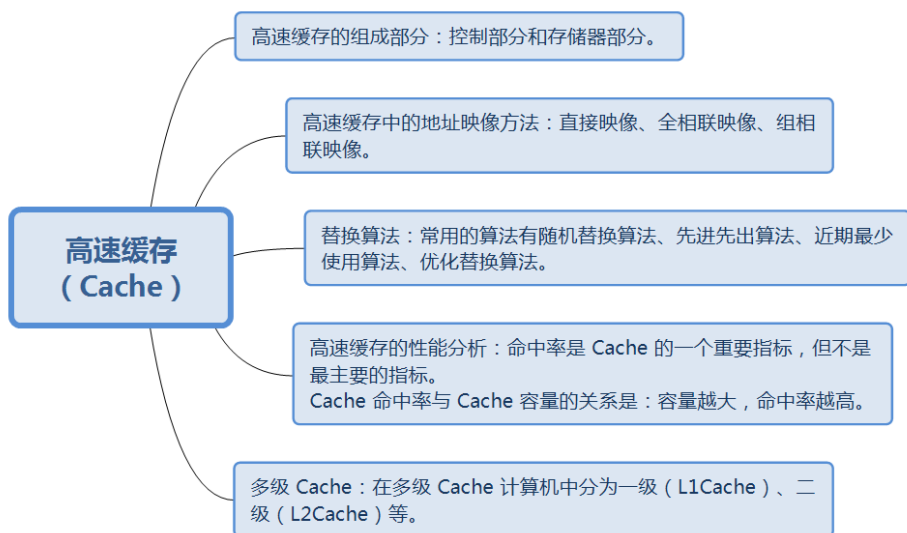


图 1-20 高速缓存

5. 虚拟存储器

虚拟存储（Virtual Memory）技术是把很大的程序（数据）分成许多较小的块，全部存储在辅存中。运行时把要用到的程序（数据）块先调入主存，并且把马上就要用到的程序块从主存调入高速缓存。这样就可以一边运行程序，一边进行所需程序（数据）块的调进或调出。虚拟存储器管理方式的分类如图 1-21 所示。

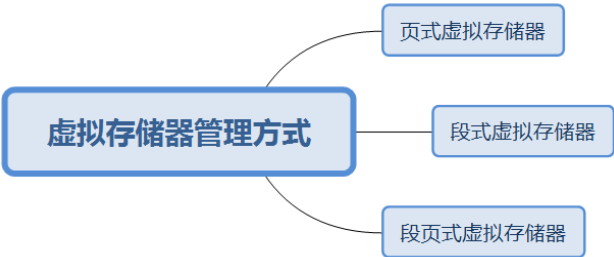


图 1-21 虚拟存储器管理方式的分类

6. 外存储器

常用的两种外存储器如图 1-22 所示。

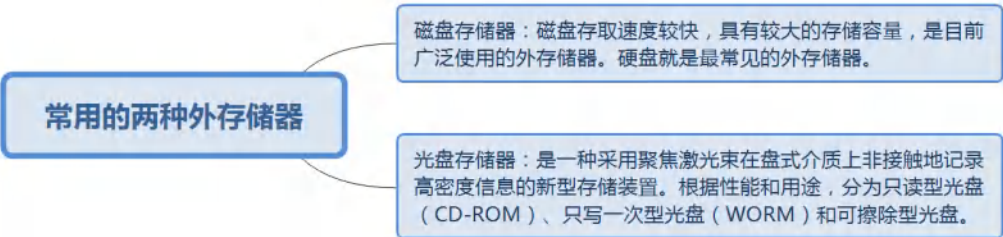


图 1-22 常用的两种外存储器

7. 磁盘阵列技术

磁盘阵列是由多台磁盘存储器组成的一个快速、大容量、高可靠的外存子系统，常见的磁盘阵列称为廉价冗余磁盘阵列（RAID）。常见的 RAID 如表 1-2 所示。

表 1-2 廉价冗余磁盘阵列

RAID 级别	说明
RAID-0	RAID-0 是一种不具备容错能力的磁盘阵列。由 n 个磁盘存储器组成的 0 级阵列，其平均故障间隙时间（MTBF）是单个磁盘存储器的 n 分之一，但数据传输率是单个磁盘存储器的 n 倍
RAID-1	RAID-1 是采用镜像容错改善可靠性的一种磁盘阵列
RAID-2	RAID-2 是采用海明码进行错误检测的一种磁盘阵列
RAID-3	RAID-3 减少了用于检验的磁盘存储器的个数，从而提高了磁盘阵列的有效容量（一般只有一个检验盘）

续表

RAID 级别	说明
RAID-4	RAID-4 是一种可独立地对组内各磁盘进行读/写的磁盘阵列, 该阵列也只用一个检验盘
RAID-5	RAID-5 是对 RAID-4 的一种改进, 它不设置专门的检验盘, 同一个磁盘既记录数据, 也记录检验信息, 这就解决了前面多个磁盘机争用一个检验盘的问题
RAID-6	RAID-6 磁盘阵列采用两级数据冗余和新的数据编码以解决数据恢复问题, 使其在两上磁盘出现故障时仍然能够正常工作。在进行写操作时, RAID-6 分别进行两个独立的检验运算, 形成两个独立的冗余数据, 写入两个不同的磁盘

1.2.3 输入/输出技术

1. 微型计算机中常用的内存与接口的编址方法

常用的内存与接口的编址方法如图 1-23 所示。

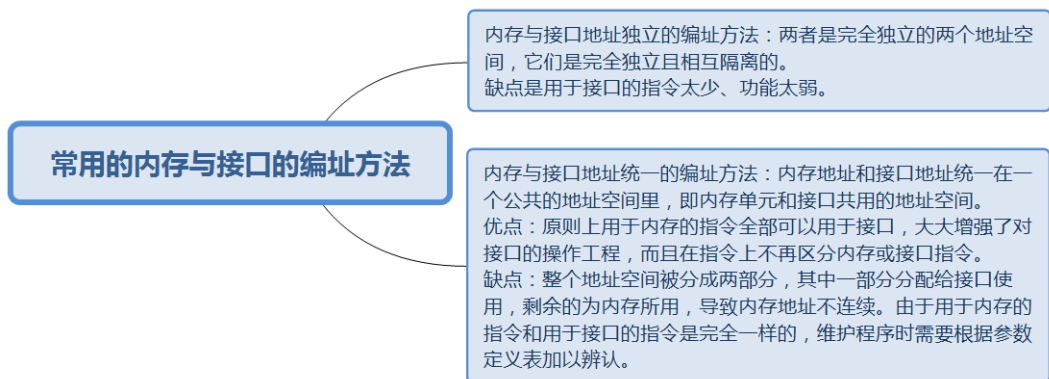


图 1-23 常用的内存与接口的编址方法

2. 直接程序控制

直接程序控制是指外设数据的输入/输出过程是在 CPU 执行程序的控制下完成的。直接程序控制分为两种情况，如图 1-24 所示。

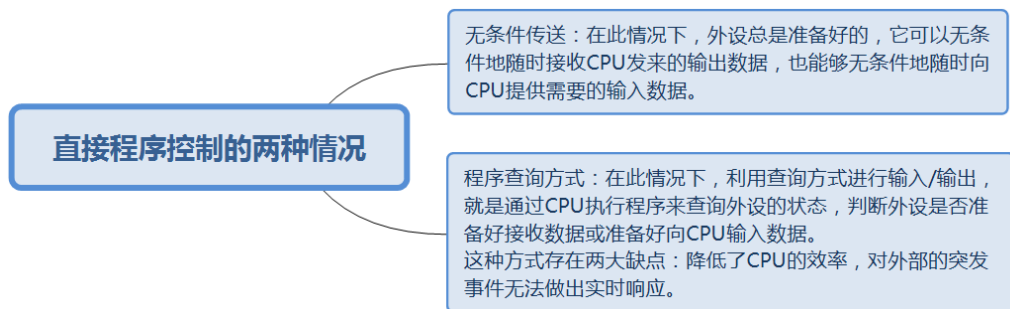


图 1-24 直接程序控制的两种情况

3. 中断方式

中断方式即由程序控制 I/O 的方法，缺点在于 CPU 必须等待 I/O 系统完成数据的传输任务，而且要定期查询 I/O 系统的状态，确认传输是否完成。因此，大大降低了整个系统的性能。

利用中断方式完成数据输入/输出的过程为：当 I/O 系统与外设系统交换数据时，CPU 无需等待，也不必去查询 I/O 的状态，从而可以去处理其他任务。当 I/O 系统准备好后，则发出中断请求信号通知 CPU，CPU 接到中断请求信号后，保存当前正在执行程序的状态，转入 I/O 中断服务程序的执行，完成与 I/O 系统的数据交换，然后再返回被打断的程序继续执行。

与程序控制方式相比，中断方式因为 CPU 无需等待而提高了效率。

(1) 中断处理方法如图 1-25 所示。

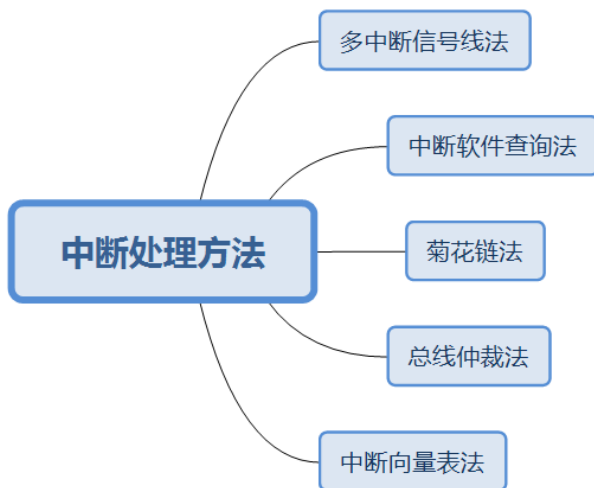


图 1-25 中断处理方法

(2) 在进行中断优先级控制时解决以下两种情况，如图 1-26 所示。

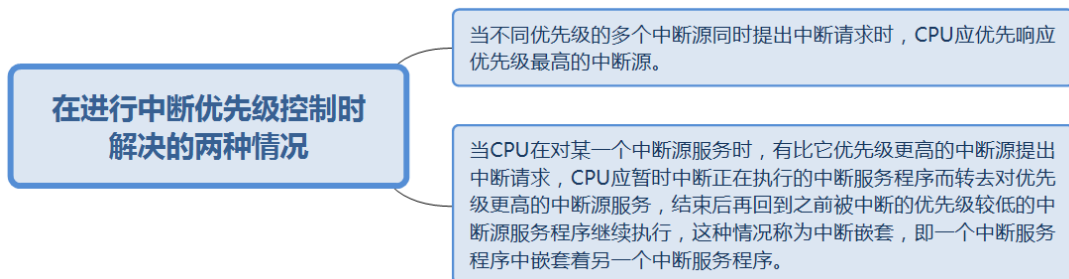


图 1-26 在进行中断优先级控制时解决的两种情况

4. 直接存储器存储方式

直接内存存取（DMA）传送的一般过程如图 1-27 所示。

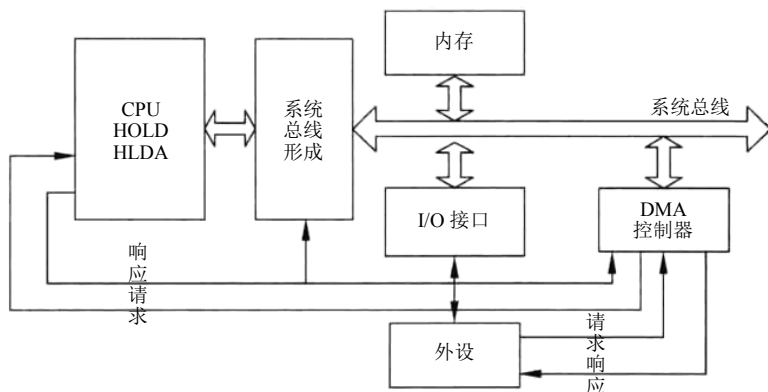


图 1-27 DMA 传送过程示意图

5. 输入/输出处理机（IOP）

IOP 数据传送的方式包括字节多路方式、选择传送方式和数组多路方式。

1.2.4 总线结构

1. 总线的定义与分类

一般来说，任何连接两个以上电子元器件的导线都可以称为总线。总线一般分为三类，分别是内部总线、系统总线、外部总线。

2. 系统总线

系统总线的性能直接影响计算机的性能。常见的标准包括 ISA 总线、EISA 总线、PCI 总线。

3. 外部总线

外部总线的标准较多，包括 RS-232C、SCSI 总线、USB、IEEE-1394、IEEE-488 等。

1.3 计算机安全

1.3.1 计算机安全概述

计算机安全指的是计算机资产安全，是要保证这些计算机资产不受自然和人为的有害因素的威胁和危害。计算机资产由系统资源和信息资源两大部分组成。系统资源主要包括软件、硬件、配套设施、文件资料，还包括相关的服务系统和业务工作人员。信息资源包括计算机系统中存储、处理和传输的各种信息。

1. 信息安全的基本要素

- (1) 机密性。确保信息不暴露给未授权的实体或进程。
- (2) 完整性。只有被允许的人才能修改数据，并能判断数据是否已被篡改。
- (3) 可用性。得到授权的实体在需要时可访问数据。
- (4) 可控性。可控制授权范围内的信息流向及行为方式。
- (5) 可审查性。对出现的安全问题提供调查的依据和手段。

2. 计算机的安全等级

计算机系统三类安全性是指技术安全性、管理安全性、政策法律安全性。重要的安全评估准则有：

- (1) 美国国防部和国家标准局推出的《可信计算机系统评估准则》(TCSEC)。
- (2) 加拿大的《可信计算机产品评估准则》(CTCPEC)。
- (3) 美国制定的《联邦(最低安全要求)评估准则》(FC)。
- (4) 欧洲英、法、德、荷四国国防部门信息安全机构联合制定的《信息技术安全评估准则》(ITSEC)。
- (5) 美国制定的《信息技术安全通用评估准则》(CC)。

3. 安全威胁

典型的安全威胁如表 1-3 所示。

表 1-3 典型的安全威胁

威胁	说明
授权侵犯	为某一特权使用一个系统，却将该系统用作其他未授权的目的
拒绝服务	对信息或其他资源的合法访问被无条件地拒绝，或者推迟与时间密切相关的操作
窃听	信息从被监视的通信过程中泄露出去
信息泄露	信息被泄露或暴露给某个未授权的实体
截获/修改	某一通信数据项在传输过程中被改变、删除或替代
假冒	一个实体（人或系统）假装成另一个实体
否认	参与某次通信交换的一方否认曾发生过此次交换
非法使用	资源被某个未授权的人或未授权的方式使用
人员疏忽	一个授权的人为了金钱或利益，或者由于粗心将信息泄露给未授权的人
完整性破坏	通过对数据进行未授权的创建、修改或破坏，使数据的一致性受到损坏
媒体清理	从废弃的或打印过的媒体中获得信息
物理入侵	一个入侵者通过物理控制而获得对系统的访问
资源耗尽	某一资源（如访问端口）被故意超负荷使用，导致其他用户的服务被中断

4. 影响数据安全的因素

主要分为内部因素和外部因素。

1.3.2 加密技术和认证技术

1. 加密技术

加密技术是最常用的安全保密手段,其中密钥加密技术的密码体制分为对称密钥体制和非对称密钥体制两种。相应地,对数据加密的技术分为对称加密(私人密钥加密)和非对称加密(公开密钥加密)。

(1) 对称加密算法。常用的对称加密算法如图 1-28 所示。

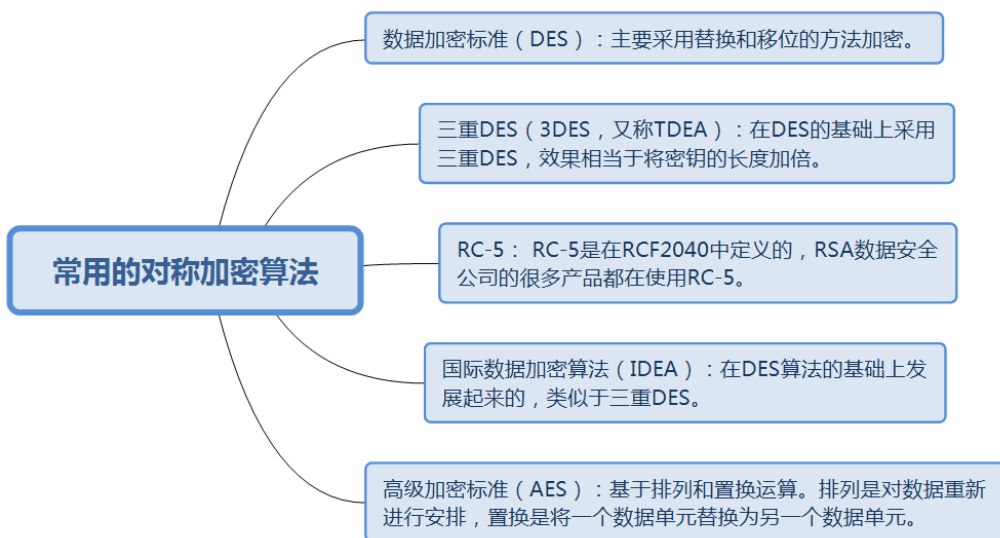


图 1-28 常用的对称加密算法

(2) 非对称加密算法。与对称加密算法不同,非对称加密算法需要两个密钥:公开密钥和私有密钥。两者是一对,如果用公开密钥对数据进行加密,只有用对应的私有密钥才能解密;如果用私有密钥对数据进行加密,只有用对应的公开密钥才能解密。因为加密和解密使用的是两个不同的密钥,所以这种算法称为非对称加密算法。非对称加密算法有两个不同的体制,如图 1-29 所示。

2. 认证技术

关键绩效指标 (PKI) 的基本构成部分如图 1-30 所示。数字签名如图 1-31 所示。

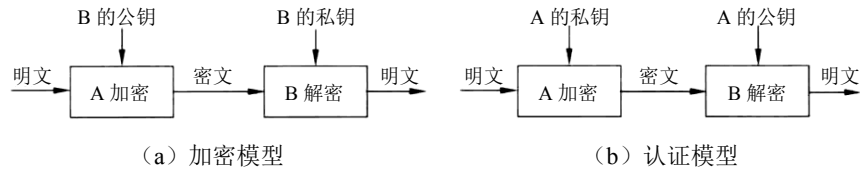


图 1-29 非对称加密算法的体制模型

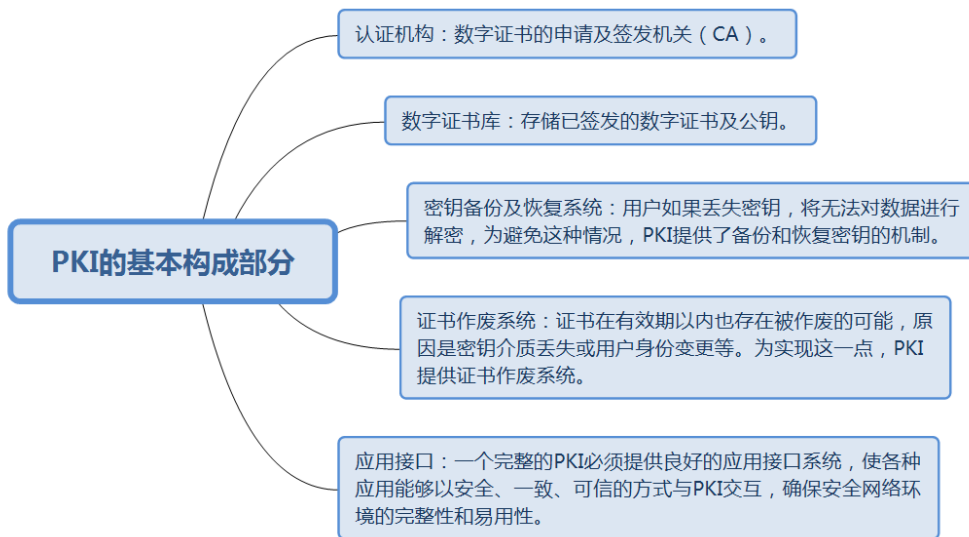


图 1-30 PKI 的基本构成部分

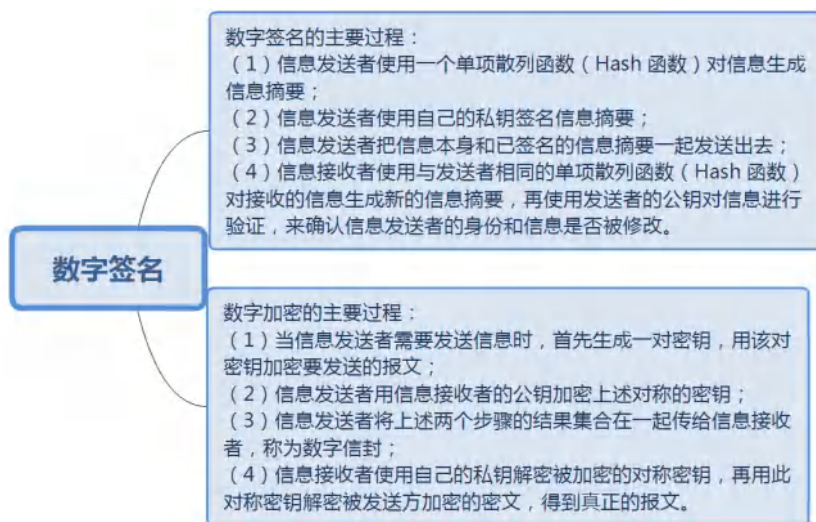


图 1-31 数字签名

1.4 练习题

1. 为实现程序指令的顺序执行, CPU _____ 中的值将自动加 1。
- A. 指令寄存器 (IR) B. 程序计数器 (PC)
C. 地址寄存器 (AR) D. 指令译码器 (ID)

解析:

本题考查 CPU 中相关寄存器的基础知识。

指令寄存器 (IR) 用来保存当前正在执行的指令。当执行一条指令时, 先把它从内存取到数据寄存器 (DR) 中, 然后再传送至 IR。为了执行任何给定的指令, 必须对操作码进行测试, 以便识别所要求的操作, 指令译码器 (ID) 就是做这项工作的。指令寄存器中操作码字段的输出就是指令译码器的输入。操作码一经译码后, 即可向操作控制器发出具体操作的特定信号。

地址寄存器 (AR) 用来保存当前 CPU 所访问的内存单元的地址。由于内存和 CPU 之间存在着操作速度上的差别, 所以必须使用地址寄存器来保持地址信息, 直到内存的读/写操作完成为止。

为了保证程序指令能够连续地执行下去, CPU 必须具有某些手段来确定下一条指令的地址, 程序计数器正起到这种作用, 所以通常又称为指令计数器。在程序开始执行前, 必须将它的起始地址, 即程序的一条指令所在的内存单元地址送入程序计数器 (PC), 因此 PC 的内容即是从内存提取的第一条指令的地址。当执行指令时, CPU 将自动修改 PC 的内容, 即每执行一条指令, PC 就增加一个量, 这个量等于指令所含的字节数, 以便使其总保持将要执行的下一条指令的地址。由于大多数指令都是按顺序来执行的, 所以修改的过程通常只是简单地对 PC 加 1。

答案: B

2. 以下关于计算机系统中中断概念的叙述中, 正确的是 _____。
- A. 由 I/O 设备提出的中断请求和电源掉电都是可屏蔽中断
B. 由 I/O 设备提出的中断请求和电源掉电都是不可屏蔽中断
C. 由 I/O 设备提出的中断请求是可屏蔽中断, 电源掉电是不可屏蔽中断
D. 由 I/O 设备提出的中断请求是不可屏蔽中断, 电源掉电是可屏蔽中断

解析:

本题考查中断的基础知识。

按照是否可以被屏蔽, 可以将中断分为两大类: 不可屏蔽中断 (又叫非屏蔽中断) 和可屏蔽中断。不可屏蔽中断源一旦提出请求, CPU 必须无条件响应, 而对可屏蔽中断源的请求, CPU 可以响应, 也可以不响应。典型的非屏蔽中断源的例子是电源掉电, 一旦出现, 必须立即无条件响应, 否则进行其他任何工作都是没有意义的。典型的可屏蔽中断源的例子是打印机中断, CPU 对打印机中断请求的响应可以快一些, 也可以慢一些, 因为让打印机等待是完全可以的。对于软中断, 它不受中断允许标志位 (IF 位) 的影响, 所以属于非屏蔽中断范畴。

答案: C

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

- A. 高内聚、高耦合
- C. 低内聚、高耦合

- B. 高内聚、低耦合
- D. 低内聚、低耦合

解析:

本题考查软件设计原则的基础知识。

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。耦合性和内聚性是模块独立性的两个定性标准, 在划分软件系统模块时, 尽量做到高内聚、低耦合, 提高模块的独立性。

答案: B

4. 关于 64 位和 32 位微处理器, 不能以二倍关系描述的是_____。

- A. 通用寄存器的位数
- C. 运算速度

- B. 数据总线的宽度
- D. 能同时运算的位数

解析:

本题考查计算机系统的基础知识。

计算机系统的运算速度受多种因素的影响, 64 位微处理器可以同时处理 64 位数据进行运算, 但不能说其速度是 32 位微处理器的两倍。

答案: C

5. 计算机指令一般包括操作码和地址码两个部分, 为分析执行一条指令, 其_____。

- A. 操作码应存入指令寄存器 (IR), 地址码应存入程序计数器 (PC)
- B. 操作码应存入程序计数器 (PC), 地址码应存入指令寄存器 (IR)
- C. 操作码和地址码都应存入指令寄存器 (IR)
- D. 操作码和地址码都应存入程序计数器 (PC)

解析:

本题考查指令系统的基础知识。

程序被加载到内存后开始运行, 当 CPU 执行一条指令时, 先把它从内存存储器取到缓冲寄存器 (DR) 中, 再送入 IR 暂存, 指令译码器根据 IR 的内容产生各种微操作指令, 控制其他的组成部件工作, 完成所需的功能。

程序计数器 (PC) 具有寄存信息和计数两种功能, 又称为指令计数器。程序的执行分为两种情况, 一是顺序执行, 二是转移执行。在程序开始执行前, 将程序的起始地址送入 PC, 该地址在程序加载到内存时确定, 因此 PC 的内容即是程序第一条指令的地址。执行指令时, CPU 将自动修改 PC 的内容, 以便使其总保持将要执行的下一条指令的地址。由于大多数指令都是按顺序来执行的, 所以修改的过程通常只是简单地对 PC 加 1。当遇到转移指令时, 后继指令的地址根据当前指令的地址加上一个向前或向后转移的位移量得到, 或者根据转移指令给出的直接转移地址得到。

答案: C

6. 中断向量可以提供_____。

- A. I/O 设备的端口地址 B. 所传送数据的起始地址
C. 中断服务程序的入口地址 D. 主程序的断电地址

解析:

计算机发生中断时,各个寄存器和程序计数器所指向的内容,或者其中所存的内容,可以暂时地存储在另一个地方。当执行完别的程序时,可以从暂存的地方将中断向量取出放入原来的位置,从而可以执行原来中断的程序,即中断向量可以描述中断服务程序的入口地址。

答案: C

7. 在 CPU 的寄存器中,_____对用户是完全透明的。

- A. 程序计数器 B. 指令寄存器 C. 状态寄存器 D. 通用寄存器

解析:

本题考查计算机系统的基础知识。

寄存器组是 CPU 中的一个重要组成部分,它是 CPU 内部的临时存储空间。寄存器既可以用来存放数据和地址,也可以存放控制信息或 CPU 工作时的状态。在 CPU 中增加寄存器的数量,可以使 CPU 把执行程序时所需的数据尽可能地放在寄存器中,从而减少访问内存的次数,提高其运行速度。但是,寄存器的数目也不能太多,除了增加成本外,寄存器地址编码的增加还会增加指令的长度。CPU 中的寄存器通常分为存放数据的寄存器、存放地址的寄存器、存放控制信息的寄存器、存放状态信息的寄存器和其他寄存器等类型。

程序计数器是存放指令地址的寄存器,其作用是:当程序按顺序执行时,每取出一条指令,程序计数器(PC)的内容自动增加一个值,指向下一条要取的指令。当程序出现转移时,则将转移地址送入 PC,然后由 PC 指向新的指令地址。

指令寄存器(IR)用于存放正在执行的指令,指令从内存取出后送入指令寄存器。其操作码部分经指令译码器送到微操作信号发生器,地址码部分指明参加运算的操作数的地址形成方式。在指令执行的过程中,指令寄存器中的内容保持不变。

状态字寄存器(PSW)用于保存指令执行完成后产生的条件码,如运算是否有溢出、结果为正还是为负、是否有进位等。此外,PSW 还保存中断和系统工作状态等信息。

通用寄存器组是 CPU 中的一组工作寄存器,运算时用于暂存操作数或地址。在程序中使用通用寄存器可以减少访问内存的次数,提高运算速度。

在汇编语言程序中,程序员可以直接访问通用寄存器以存取数据,可以访问状态字寄存器以获取有关数据处理结果的相关信息,可以通过相对程序计数器进行寻址,但是不能访问指令寄存器。

答案: B

8. CPU 中译码器的主要作用是进行_____。

- A. 地址译码 B. 指令译码 C. 数据译码 D. 选择多路数据至 ALU

解析:

本题考查计算机系统的基础知识。

CPU 中指令译码器的功能是对现行指令进行分析,确定指令类型和指令所要完成的操作及寻

址方式，并将相应的控制命令发往相关部件。

答案：B

9. 从认证中心 CA 获取用户 B 的数字证书，该证书用_____作数字签名，从用户 B 的数字证书中可以获得 B 的公钥。

- A. CA 的公钥 B. CA 的私钥 C. B 的公钥 D. B 的私钥

解析：

本题考查数字证书的基础知识。

用户的数字证书由某个可信的证书发放机构（Certification Authority，CA）建立，并由 CA 或用户将其放入公共目录中。在 X.509 标准中，一般格式的数字证书包含以下数据域：

- 版本号：用于区分 X.509 的不同版本。
- 序列号：由同一发行者（CA）发放的每个证书的序列号是唯一的。
- 签名算法：签署证书所用的算法及其参数。
- 发行者：建立和签署证书的 CA 的 X.509 名字。
- 有效期：包括证书有效期的起始时间和终止时间。
- 主体名：证书持有者的名称及其有关信息。
- 公钥：证书持有者的公钥及其使用方法。
- 发行者 ID：任选的名字唯一的标识证书的发行者。
- 主体 ID：任选的名字唯一的标识证书的持有者。
- 扩展域：添加的扩充信息。
- 认证机构的签名：用 CA 私钥对证书的签名。

从上述描述可知，数字证书用 CA 私钥作数字签名，从用户的数字证书中可以获得用户的公钥。

答案：B

第2小时

程序设计语言基础知识

2.0 章节考点分析

根据考试大纲，本小时知识点会涉及考试中的各种题型，上午的客观题占6分，下午的程序设计与设计模式的结合往往是最后一道大题。本小时的架构如图2-1所示。

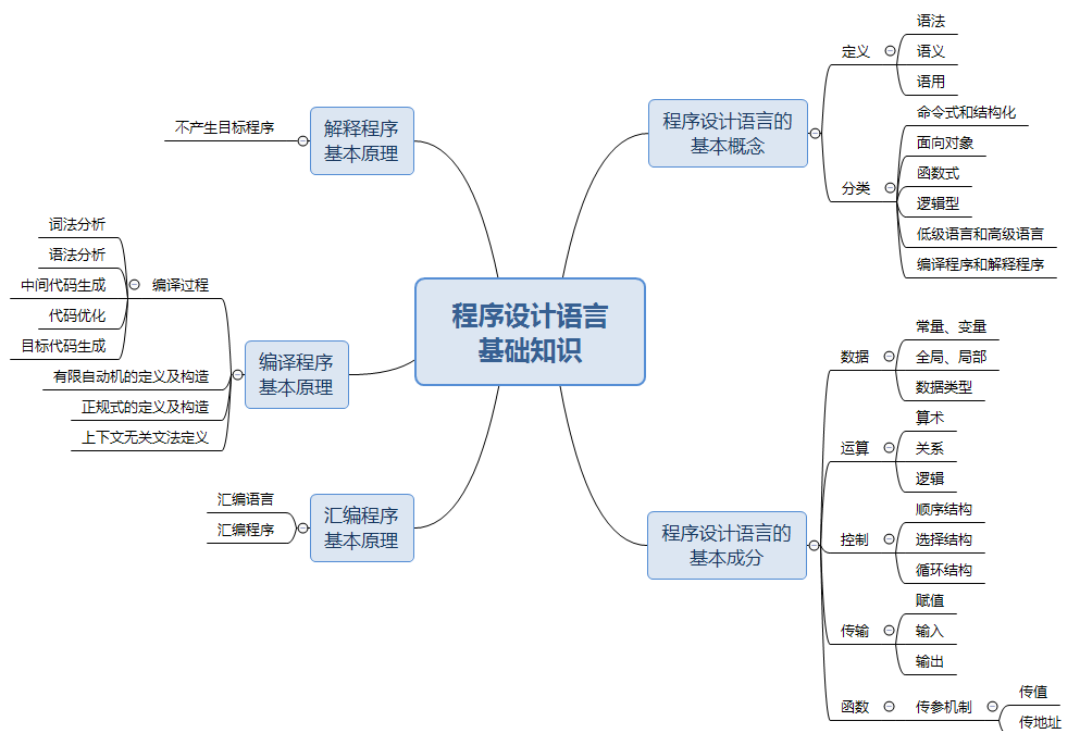


图 2-1 第 2 小时架构图



【导读小贴士】

程序语言之于程序员，就如同文字之于作家，画笔之于画家。正是因为有如此丰富多样的程序语言，才有今天的信息化时代。对于一个程序员而言，语言是工具，其编程思想才是灵魂，要深入灵魂、抓住本质。

2.1 程序语言的基本概念

(1) 低级语言与高级语言：

- 1) 低级语言：汇编。
 - 2) 高级语言：常见的有 Java、C、C++、PHP、Python、Delphi 等。
- (2) 翻译形式：汇编、解释、编译。
- (3) 程序设计语言的定义：语法、语义、语用。
- (4) 程序设计语言的分类：
- 1) 过程式（命令式和结构化）：FORTRAN、Pascal、C。
 - 2) 面向对象：Simula、Smalltalk、C++、Objective C、Java、Python。
 - 3) 函数式：lisp、python、scala。
 - 4) 逻辑型：Prolog。
 - 5) 脚本语言：shell、bat、js、python。

2.2 程序设计的基本成分

程序语言的基本成分包括数据、运算、控制和传输等。

(1) 数据成分：

- 1) 常量和变量。
 - 2) 全局量和局部量。
 - 3) 数据类型。
- (2) 运算成分：算式运算、关系运算、逻辑运算。
- (3) 控制成分：顺序结构、选择结构、循环结构。

分支结构示意图如图 2-2 所示，循环结构示意图如图 2-3 所示。

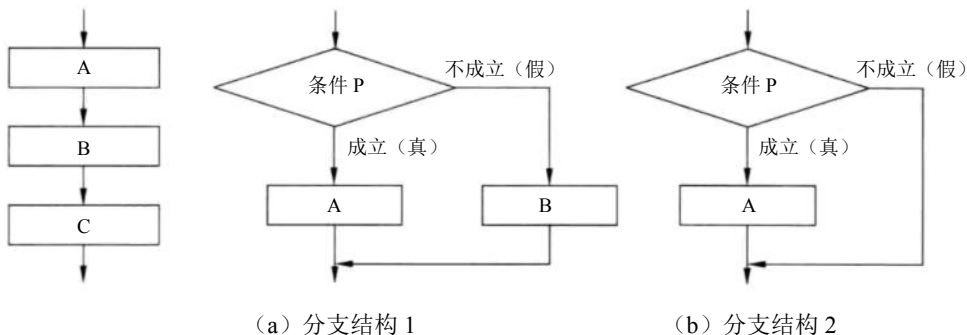


图 2-2 分支结构示意图

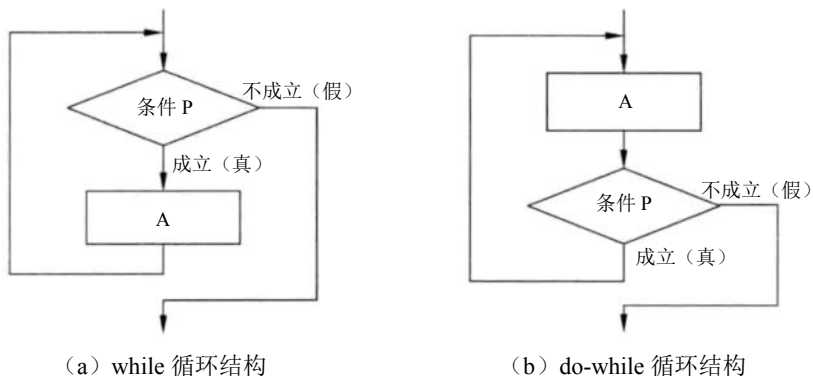


图 2-3 循环结构示意图

(4) 函数：定义、声明、调用（值调用、引用调用）。

2.3 汇编程序基本原理

1. 汇编语言

汇编语言是为特定的计算机或计算机系统设计的面向机器的符号化的程序设计语言。用汇编语言编写的程序称为汇编语言源程序。因为计算机不能直接识别和运行符号语言程序，所以要用专门的翻译程序——汇编程序进行翻译。用汇编语言编写程序要遵循所用语言的规范和约定。

汇编语言源程序由若干条语句组成，一个程序中可以有三类语句：指令语句、伪指令语句和宏指令语句。

2. 汇编程序

将汇编语言所编写的源程序翻译成机器指令程序。汇编程序一般需要两次扫描源程序才能完成翻译过程。

第一次扫描：检查语法错误，确定符号名字；建立使用的全部符号名字表；每一符号名字后跟

一对应值（地址或数）。

第二次扫描：在第一次扫描的基础上，将符号地址转换成真地址（代真）；利用操作码表将助记符转换成相应的目标码。

2.4 编译程序基本原理

1. 编译程序

编译程序的工作过程分为六个阶段，如图 2-4 所示。

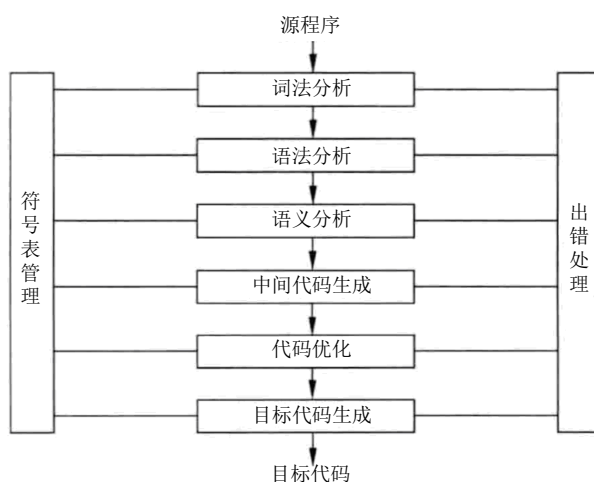


图 2-4 编译程序的工作过程

2. 文法

$$G = \{V_t, V_n, S, P\}$$

V_t 是一个非空有限集合的符号，它的每个元素称为终结符号； V_n 是一个非空有限集合的符号，它的每个元素称为非终结符号； S 称为文法 G 的开始符号； P 是一个非空有限集合，它的元素称为产生式。

- 1 型文法：又称为上下文有关文法。
- 2 型文法：又称为上下文无关文法。
- 3 型文法：又称为正规文法，使用最多。
- 0 型文法：短语文法。

3. 有限自动机

计算机控制系统的控制程序具有有限状态自动机（FA）的特征，可以用有限状态机理论来描述。

（1）确定有限自动机（DFA）。自动机的每个状态都有对字母表中所有符号的转移。一个确定的有限自动机是个五元组 (S, Σ, f, s_0, Z) ，其中：

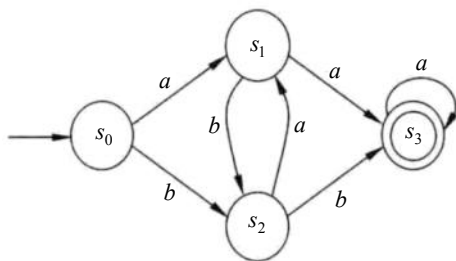
- S 是一个有限集，其每个元素称为一个状态。
- Σ 是一个有穷字母表，其每个元素称为一个输入字符。
- f 是 $S \times \Sigma \rightarrow S$ 上的单值部分映像。 $f(A, a) = Q$ 表示当前状态为 A 、输入为 a 时，将转换到下一状态 Q ，称 Q 为 A 的一个后继状态。
- $s_0 \in S$ ，是唯一的一个开始状态。
- Z 是非空的终止状态集合， $Z \subseteq S$ 。

一个 DFA 可以用两种直观的方式表示：状态转换图和状态转换矩阵。状态转换图简称为转换图，是一个有向图。DFA 中的每个状态对应转换图中的一个节点，DFA 中的每个转换函数对应图中的一条有向弧，若转换函数为 $f(A, a) = Q$ ，则该有向弧从节点 A 出发，进入节点 Q ，字符 a 是弧上的标记。

【例 2.1】已知有 DFA $M_1 = (\{s_0, s_1, s_2, s_3\}, \{a, b\}, f, S, \{s_3\})$ ，其中 f 为：

$$f(s_0, a) = s_1, f(s_0, b) = s_2, f(s_1, a) = s_3, f(s_1, b) = s_2, f(s_2, a) = s_1, f(s_2, b) = s_3, f(s_3, a) = s_3$$

与 DFA M_1 对应的状态图如图 2-5 (a) 所示，其中，双圈表示的节点是终态节点。状态转换矩阵可以用一个二维数组 M 表示，矩阵元素 $M[A, a]$ 的行下标表示状态，列下标表示输入字符， $M[A, a]$ 的值是当前状态为 A 、输入为 a 时，应转换到的下一状态。与 DFA M_1 对应的状态转换矩阵如图 2-5 (b) 所示。在转换矩阵中，一般以第一行的行下标所对应的状态作为初态，而终态则需要特别指明。



(a) 状态转换图

	a	b
s_0	s_1	s_2
s_1	s_3	s_2
s_2	s_1	s_3
s_3	s_3	—

(b) 状态转换矩阵

图 2-5 确定有限自动机 (DFA)

(2) 非确定有限自动机 (NFA)。自动机的状态对字母表中的每个符号可以有转移，也可以没有转移，对一个符号甚至可以有多个转移。一个不确定的有限自动机也是一个五元组，它与确定有限自动机的区别如下。

1) f 是 $S \times \Sigma \rightarrow 2S$ 上的映像。对于 S 中的一个给定状态及输入符号，返回一个状态的集合，即当前状态的后继状态不一定是唯一确定的。

2) 有向弧上的标记可以是 ε 。

【例 2.2】已知有 NFA $N = (\{s_0, s_1, s_2, s_3\}, \{a, b\}, f, S, \{s_3\})$ ，其中 f 为：

$f(s_0,a)=s_0, f(s_0,b)=s_1, f(s_1,a)=s_0, f(s_1,b)=s_2, f(s_2,a)=s_2, f(s_2,b)=s_3$
 与 NFA M_2 对应的状态转换图和状态转换矩阵如图 2-6 所示。

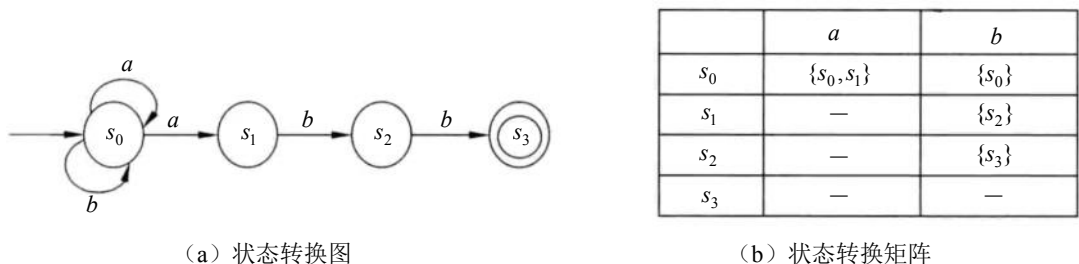


图 2-6 非确定有限自动机（NFA）

4. 正规表达式

对于字母表 Σ ，其上的正规式及其表示的正规集可以递归定义如下。

- (1) ε 是一个正规式，它表示集合 $L(\varepsilon)=\{\varepsilon\}$ 。
- (2) 若 a 是 Σ 上的字符，则 a 是一个正规式，它所表示的正规集为 $\{a\}$ 。
- (3) 若正规式 r 和 s 分别表示正规集 $L(r)$ 和 $L(s)$ ，则：
 - 1) $r \mid s$ 是正规式，表示集合 $L(r) \cup L(s)$ 。
 - 2) $r \cdot s$ 是正规式，表示集合 $L(r)L(s)$ 。
 - 3) r^* 是正规式，表示集合 $(L(r))^*$ 。
 - 4) (r) 是正规式，表示集合 $L(r)$ 。

仅由有限次地使用上述三个步骤定义的表达式才是 Σ 上的正规式，其中运算符 “ $\mid$ ” “ $\cdot$ ” “ * ” 分别称为 “或” “连接” 和 “闭包”。在正规式的书写中，连接运算符 “ $\cdot$ ” 可省略。运算符的优先级从高到低的顺序排列为 “ * ” “ $\cdot$ ” “ $\mid$ ”。

设 $\Sigma=\{a,b\}$ ，表 2-1 列出了 Σ 上的一些正规式和相应的正规集。

表 2-1 正规式和相应的正规集

正规式	正规集
ab	字符串 ab 构成的集合
$a b$	字符串 a 、 b 构成的集合
a^*	由 0 个或多个 a 构成的字符串集合
$(a b)^*$	所有字符 a 和 b 构成的串的集合
$a(a b)^*$	以 a 为首字符的 a 、 b 字符串的集合
$(a b)^*abb$	以 abb 结尾的 a 、 b 字符串的集合

若两个正规式表示的正规集相同，则认为二者等价，两个等价的正规式 U 和 V 记为 $U=V$ 。例

如, $b(ab)^*=(ba)^*b$, $(a|b)^*=(a^*b^*)^*$ 。设 U 、 V 和 W 均为正规式, 正规式的代数性质如表 2-2 所示。

表 2-2 正规式的代数性质

$U V = V U$	$(UV)W = U(VW)$
$(U V)W = U (VW)$	$\varepsilon U = U\varepsilon = U$
$U(V W) = UV UW$	$V^* = (V^+ \varepsilon)$
$(U V)W = UW VW$	$V^{**} = V^*$

5. 正规式与有限自动机之间的转换

(1) 有限自动机转为正规式的过程如图 2-7 所示。

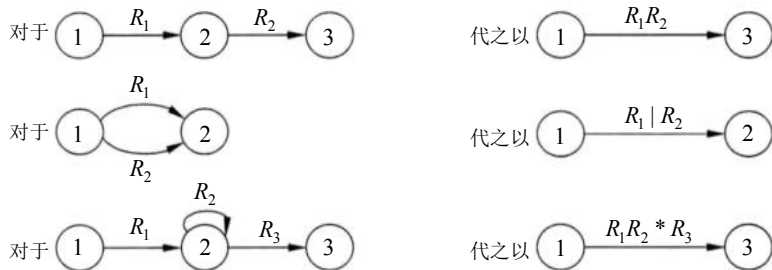


图 2-7 有限自动机转为正规式

(2) 正规式转为有限自动机的过程如图 2-8 所示。

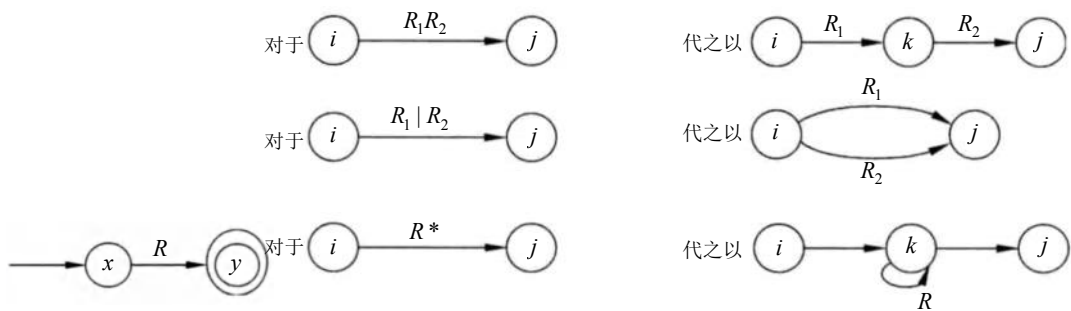


图 2-8 正规式转为有限自动机

2.5 解释程序基本原理

解释程序是另一种语言处理程序, 在词法、语法和语义分析方面与编译程序的工作原理基本相同, 但是在运行用户程序时, 它直接执行源程序或源程序的内部形式。因此, 解释程序不产生源程序的目标程序, 这是它和编译程序的主要区别。图 2-9 显示了解释程序实现高级语言的三种方式。

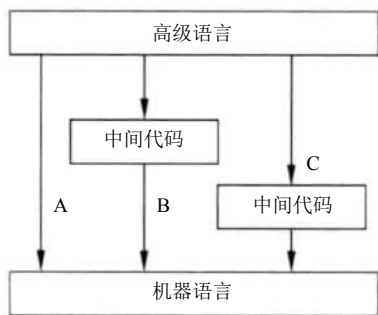


图 2-9 解释程序实现高级语言的三种方式

2.6 练习题

1. 程序运行过程中常使用参数在函数（过程）间传递信息，引用调用传递的是实参的_____。

- A. 地址 B. 类型 C. 名称 D. 值

解析：

程序运行时，对函数的调用一般有两种形式：传值调用和引用调用。

传值调用：形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变。

引用调用：形参取的是实参的地址，即相当于实参存储单元的地址引用，因此，其值的改变同时就改变了实参的值。

答案：A

2. 已知文法 $G: S \rightarrow A_0 | B_1, A \rightarrow S_1 | 1, B \rightarrow S_0 | 0$ ，其中 S 是开始符号，从 S 出发可以推导出_____。

- A. 所有由 0 构成的字符串 B. 所有由 1 构成的字符串
C. 某些 0 和 1 个数相等的字符串 D. 所有 0 和 1 个数不同的字符串

解析：

对于文法可以推导出的字符串分析，考试一般可以对文法举例，然后总结规律。以本题的文法为例，可以产生的字符串包括：

- (1) 10 推导过程： $S \rightarrow A_0, A \rightarrow 1$ 。
(2) 01 推导过程： $S \rightarrow B_1, B \rightarrow 0$ 。
(3) 1010 推导过程： $S \rightarrow A_0, A \rightarrow S_1, S \rightarrow A_0, A \rightarrow 1$ 。

至此，可以了解到，选项 A、B、D 的描述都是不正确的。

答案：C

3. 算数表达式 $a+(b-c)*d$ 的后缀式是_____（-、+、*表示算数的减、加、乘运算，运算符的优先级和结合性遵循惯例）。

- A. $bc-d*a+$ B. $abc-d*+$ C. $ab+c-d*$ D. $abcd-*+$

解析:

本题要求算术表达式的后缀式, 解决该类问题的方法是将算术表达式构造成一棵二叉树, 然后对二叉树进行后序遍历得到后缀式。题目中的算术表达式可以构造为如图 2-10 所示的语法树。

对二叉树进行后序遍历的结果为 $abc-d*+$ 。

答案: B

4. 将高级语言程序翻译为机器语言的过程中, 常引入中间代码, 其好处是_____。

- A. 有利于反编译处理
- B. 有利于进行与机器无关的优化
- C. 尽早发现语法错误
- D. 可以简化语法和语义分析

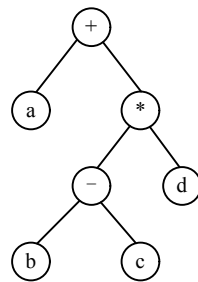


图 2-10 语法树

解析:

“中间代码”是一种简单且含义明确的记号系统, 可以有若干种形式, 它们的共同特征是与具体的机器无关, 此时所做的优化一般建立在对程序的控制流和数据流分析的基础之上, 与具体的机器无关。

答案: B

5. 对高级语言程序进行编译的过程中, 有限自动机 (NFA 或 DFA) 是进行_____的适当工具。

- A. 词法分析
- B. 语法分析
- C. 语义分析
- D. 出错处理

解析:

语言中具有独立含义的最小语法单位是符号 (单词), 如标识符、无符号常数和界限符等。词法分析的任务是把构成源程序的字符串转换成单词符号序列。有限自动机是一种识别装置的抽象概念, 它能准确地识别正规集。有限自动机分为两类: 确定的有限自动机 (DFA) 和不确定的有限自动机 (NFA)。

答案: A

6. 弱类型语言 (动态类型语言) 是指不需要进行变量/对象类型声明的语言。_____属于弱类型语言。

- A. Java
- B. C/C++
- C. Python
- D. C#

解析:

弱/强类型指的是语言类型系统的类型检查的严格程度, 动态类型和静态类型则指变量与类型的绑定方法。静态类型是指编译器在编译源程序期间执行的类型检查, 动态类型是指编译器 (虚拟机) 在程序运行时执行的类型检查。简单地说, 在声明了一个变量之后, 不能改变其类型的语言, 是静态语言; 能够随时改变其类型的语言, 是动态语言。弱类型相对于强类型来说, 类型检查更不严格, 比如说允许变量类型的隐式转换、允许强制类型转换等。

答案: C

第3小时

数据结构与数据运算

3.0 章节考点分析

第3小时主要学习数据的线性结构、数据的非线性结构、数据的运算等内容。

根据考试大纲，本小时知识点会涉及单选题型，约占5分。根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，除了要掌握有关概念的知识点，同时又要掌握适当的扩展内容。本小时的架构如图3-1所示。

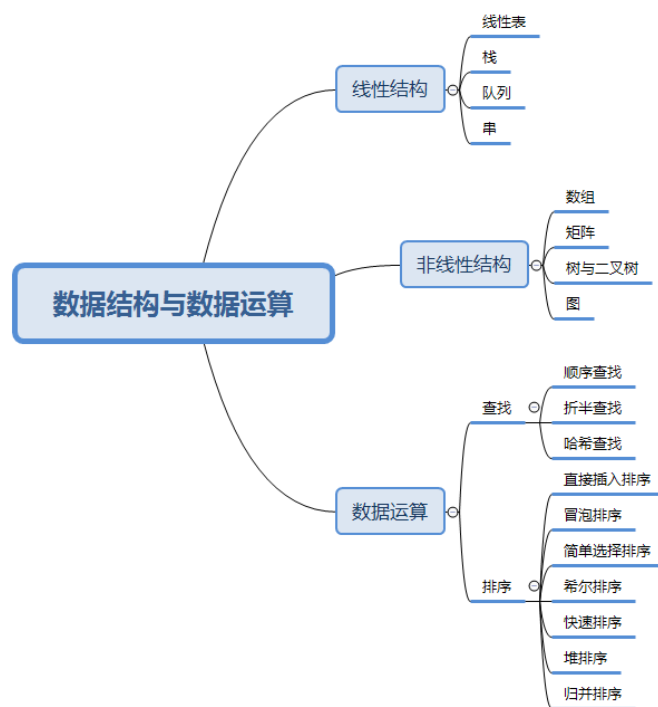


图 3-1 第3小时架构图



【导读小贴士】

数据的结构可以分为线性结构、非线性结构与数据的运算三个部分。本小时选择了一些软件设计师考试经常会涉及的数据结构及它们的存储方式，另外，数据运算也是高频考点。广大考生需要重点掌握的是这几种典型数据结构的特点、存储的方式及查找和排序算法的有关定义。数据结构这部分还有一些更加深入的知识点，出于篇幅限制没有一一列出，考生们可以参考其他资料作为补充知识点来学习。

3.1 线性结构

3.1.1 线性表

一个线性表是最简单、最基本的数据结构。线性表常采用顺序存储和链式存储，主要的基本操作是插入、删除和查找等。

一个线性表是 n ($n \geq 0$) 个元素的有限序列，通常表示为 $(a_1, a_2, \dots, a_n)$ 。非空线性表的特点为：①存在唯一的一个称作“第一个”的元素；②存在唯一的一个称作“最后一个”的元素；③除第一个元素外，序列中的每个元素均有一个直接前驱；④除最后一个元素外，序列中的每个元素均有一个直接后继。

3.1.2 线性表的存储结构

1. 顺序存储

线性表的顺序存储是指用一组地址连续的存储单元依次存储线性表中的数据元素，从而使得逻辑上相邻的两个元素在物理位置上也相邻，如图 3-2 所示。

优点：可以随机存取表中元素。

缺点：插入与删除需要移动元素。

a_1	a_2	a_3	...	a_{n-1}	a_n
-------	-------	-------	-----	-----------	-------

图 3-2 线性表的顺序存储

2. 链式存储

链式存储通过指针链接起来的结点来存储数据元素，结构如图 3-3 所示。

优点：插入与删除不需要移动元素。

缺点：不能对数据元素进行随机访问。

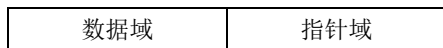


图 3-3 链式存储

链式存储结构有单链表、循环链表（循环单链表、循环双链表）等，其中单链表只能从头结点开始往后顺序遍历整个链表，而循环单链表可以从表中的任一结点开始遍历整个链表。

3.1.3 栈

栈又称为“后进先出”（LIFO）的线性表。在栈中进行插入和删除操作的一端称为栈顶，另一端称为栈底，如图 3-4 所示。



图 3-4 栈的示意图



图 3-5 队列的示意图

3.1.4 队列

队列是一种“先进先出”（FIFO）的线性表，即在表的一端插入元素，在另一端删除元素，如图 3-5 所示。在队列中允许插入元素的一端称为队尾（rear），允许删除元素的一端称为队头（front）。

1. 循环队列

由于顺序队列的存储空间是提前设定的，所以队尾指针会有一个上限值，当队尾指针达到上限时便无法通过修改队尾指针来添加元素。如若将顺序队列设置成环状结构，就可以维持入队、出队操作。设置循环队列 Q 的容量为 MAXSIZE，初始队列为空，而且 Q.rear 和 Q.front 都等于 0，如图 3-6（a）所示。

元素入队时，修改队尾指针 $Q.rear=(Q.rear+1)\%MAXSIZE$ ，如图 3-6（b）所示。

元素出队时，修改队头指针 $Q.front=(Q.front+1)\%MAXSIZE$ ，如图 3-6（c）所示。

循环队列约定以“队尾指针所指位置的下一个位置是队头指针”来表示“队列满”的情况，如图 3-6（d）所示。

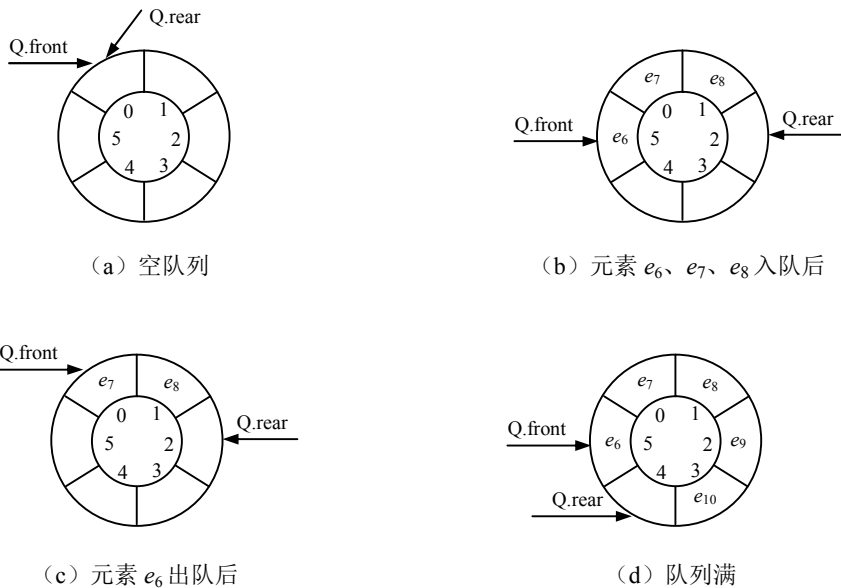


图 3-6 循环队列

2. 双端队列

某双端队列如图 3-7 所示，要求元素进出队列必须在同一端口，即从 A 端进入的元素必须从 A 端出、从 B 端进入的元素必须从 B 端出。

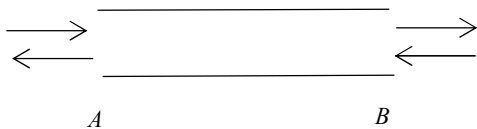


图 3-7 双端队列

3.1.5 串

串是仅由字符构成的有限序列，是一种线性表，一般记为 $s = "a_1a_2...a_n"$ ($n > 0$)，其中， s 是串的名称，用单引号括起来的字符序列是串值。串的基本概念包括：

(1) 空串：长度为 0 的串称为空串，空串不包含任何字符。

(2) 空格串：由一个或多个空格组成的串。

(3) 子串：由串中任意长度的连续字符构成的序列称为子串。含有子串的串称为主串。空串是任意串的子串。

3.2 非线性结构

3.2.1 二维数组

如图 3-8 所示，二维数组的特点如下：

- (1) 数据元素数目固定，一旦定义了一个数组结构，就不再有元素个数的增减变化。
- (2) 数据元素具有相同的类型。
- (3) 数据元素的下标关系具有上下界约束且下标有序。

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

图 3-8 二维数组

二维数组的存储结构可以分为以行为主序和以列为主序。

设每个元素占用 L 个单元， m 、 n 为数组的行数和列数， $Loc(a_{11})$ 表示元素 a_{11} 的地址。

以行为主序优先存储的地址计算公式为：

$$Loc(a_{ij}) = Loc(a_{11}) + ((i-1) \times n + (j-1)) \times L$$

以列为主序优先存储的地址计算公式为：

$$Loc(a_{ij}) = Loc(a_{11}) + ((j-1) \times m + (i-1)) \times L$$

3.2.2 三对角矩阵

若以行为主序将 n 阶三对角矩阵 $A_{n \times n}$ 的非 0 元素 a_{ij} 存储在一维数组 $B[k]$ ($1 \leq k \leq 3 \times n - 2$) 中，如图 3-9 所示，则元素位置之间的对应关系为：

$$k = 3 \times (i-1) - 1 + j - i + 1 + 1 = 2i + j - 2 \quad (1 \leq i, j \leq n)$$

$$A_{n \times n} = \begin{bmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & 0 \\ & a_{3,2} & a_{3,3} & a_{3,4} & & \\ & & \vdots & \vdots & \vdots & \\ & & & a_{i,i-1} & a_{i,i} & a_{i,i+1} \\ & & & & \vdots & \vdots & \vdots \\ & & & & & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

图 3-9 三对角矩阵示意图

3.2.3 树

树是 n ($n \geq 0$) 个结点的有限集合, 当 $n=0$ 时, 集合为空, 称为空树。在任意一非空树中 ($n > 0$), 有且仅有一个称为根的结点, 其余结点可以分成 m ($m \geq 0$) 个不相交的有限结点的集合 $T_1, T_2, \dots, T_m$, 其中每个 T_i 又都是一棵树, 并且称为根的子树, 如图 3-10 所示。

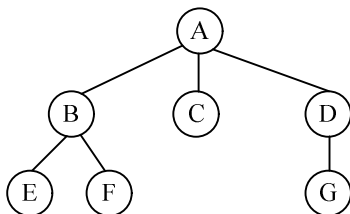


图 3-10 树结构示意图

1. 树的基本概念

(1) 双亲、孩子和兄弟。结点的子树的根称为该节点的孩子结点; 相应地, 该结点称为其子结点的双亲。具有相同双亲的结点互为兄弟。

(2) 结点的度。一个结点拥有子树的个数称为该结点的度。例如, 如图 3-10 中, A 的度为 3, B 的度为 2, C 的度为 0, D 的度为 1。

(3) 叶子结点。叶子结点是指度为 0 的结点。例如, 图 3-10 中的 E、F、C、G 都是叶子结点。

(4) 内部结点。除根结点外, 度不为 0 的结点称为内部结点。例如, 图 3-10 中的 B、D 都是内部结点。

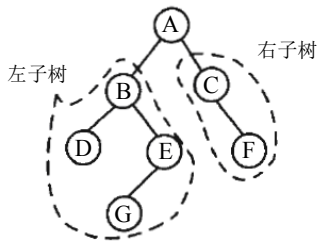
(5) 结点的层次。例如, 图 3-10 中的 A 在第一层, B、C、D 在第二层, E、F、G 在第三层。

(6) 树的深度。一棵树的最大层数为该树的深度 (或高度)。例如, 图 3-10 中的树的深度为 3。

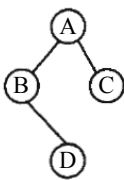
(7) 有序/无序树。如果树中各结点的各个子树是从左到右有序排列且不能交换时, 则称该树为有序树, 否则称为无序树。

2. 二叉树

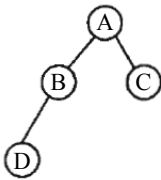
二叉树与普通树的区别在于, 二叉树中结点的子树分为左子树和右子树, 如图 3-11 所示。



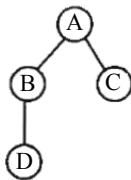
(a) 二叉树



(b) 二叉树中结点 B 的左子树为空



(c) 二叉树中结点 B 的右子树为空



(d) 普通树中结点 B 有一棵子树

图 3-11 二叉树与普通树

3. 满二叉树与完全二叉树

如果一个二叉树的层数为 K ，结点总数为 2^k-1 个，则它就是满二叉树，如图 3-12 (a) 所示。

在一个深度为 h 的完全二叉树中，除第 h 层外（最后一层），其他各层都是满的。第 h 层所有的结点都必须从左到右依次放置，不能留空，如图 3-12 (b) 所示。图 3-12 (c) 为非完全二叉树。

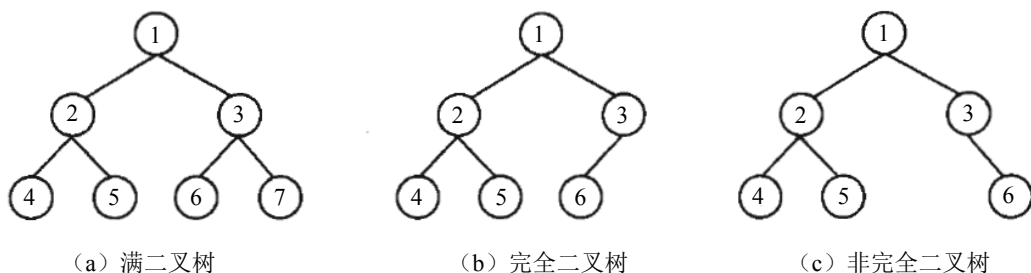


图 3-12 满二叉树、完全二叉树、非完全二叉树

4. 二叉树的性质

- (1) 在二叉树的第 i 层上最多有 2^{i-1} 个结点 ($i \geq 1$)。
- (2) 深度为 k 的二叉树最多有 2^k-1 个结点 ($k \geq 1$)。
- (3) 对于任何一棵二叉树，如果其叶子结点数为 n_0 ，度为 2 的结点数为 n_2 ，则 $n_0 = n_2 + 1$ 。
- (4) 具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。
- (5) 如果对一棵有 n 个结点的完全二叉树的结点按层序编号（从第 1 层到 $\lceil \log_2 n \rceil + 1$ 层，每层从左到右），则对任一结点 i ($1 \leq i \leq n$) 有：

- 1) 如果 $i=1$ ，则该结点是二叉树的根，无双亲；如果 $i > 1$ ，则该结点是 $i/2$ 。
- 2) 如果 $2i \leq n$ ，则该结点左子树的编号是 $2i$ ；否则，无左子树。
- 3) 如果 $2i+1 \leq n$ ，则该结点右子树的编号是 $2i+1$ ；否则，无右子树。

5. 二叉树的存储结构

(1) 二叉树的顺序存储。在采用顺序存储时，完全二叉树与一般二叉树相比节省了空间，这是因为一般二叉树需要添加一些“虚结点”而造成了空间的浪费，如图 3-13 所示。

(2) 二叉树的链式存储。二叉树的链式存储可以分为二叉链表和三叉链表的存储结构，如图 3-14 所示。

6. 二叉树的遍历

遍历是按某种策略访问树中的每个结点且仅访问一次的过程。二叉树的遍历可以分为前序、中序、后序和层次遍历四种形式。这四种遍历方式产生的结果如表 3-1 所示。

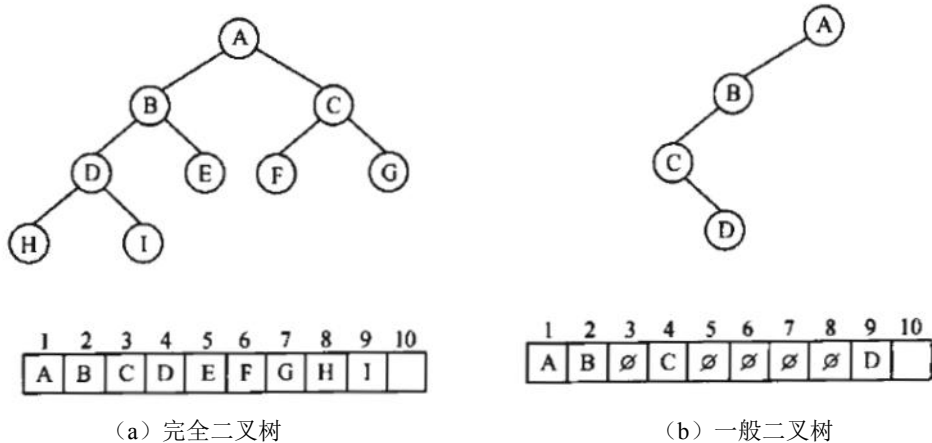


图 3-13 二叉树的顺序存储

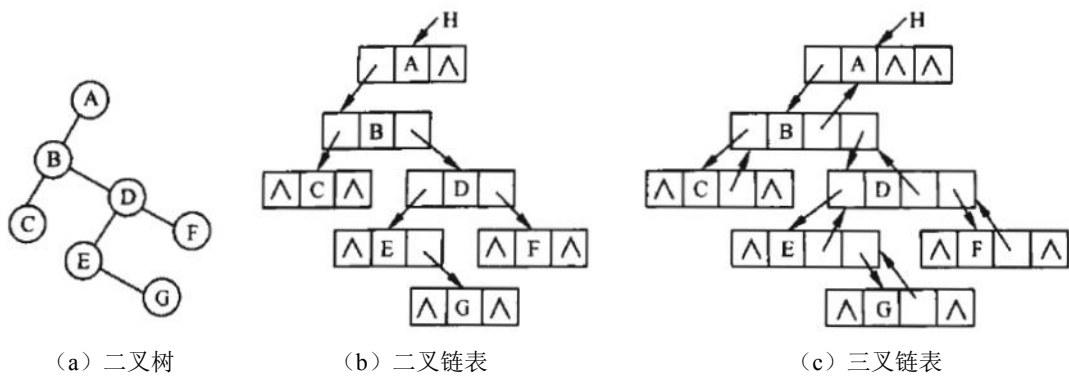


图 3-14 二叉树的链式存储

表 3-1 二叉树的遍历方式

前序遍历 (根、左、右)	1、2、4、5、7、8、3、6	
中序遍历 (左、根、右)	4、2、7、8、5、1、3、6	
后序遍历 (左、右、根)	4、8、7、5、2、6、3、1	
层次遍历	1、2、3、4、5、6、7、8	

7. 最优二叉树 (哈夫曼树)

哈夫曼树又称为最优二叉树, 是一种带权路径长度最短的二叉树。所谓树的带权路径长度, 就是树中所有的叶子结点的权值乘上其到根结点的路径长度 (若根结点为 0 层, 叶子结点到根结点的路径长度为叶子结点的层数)。树的带权路径长度记为 $WPL=(W_1 \times L_1+W_2 \times L_2+W_3 \times L_3+\cdots+W_n \times L_n)$, N 个权值 $W_i (i=1,2,\cdots,n)$ 构成一棵有 N 个叶结点的二叉树, 相应的叶结点的路径长度为 $L_i (i=1,2,\cdots,n)$ 。由此可以证明哈夫曼树的带权路径长度是最小的。

3.2.4 图

图是由集合 V 和 E 构成的二元组, 记作 $G=(V,E)$, 其中, V 是图中顶点的非空有限集合, E 是图中边的有限集合, 如图 3-15 所示。



图 3-15 图

1. 图的存储结构

(1) 邻接矩阵。邻接矩阵表示法是指用一个矩阵来表示图中顶点之间的关系。对于具有 n 个顶点的图 $G=(V,E)$, 其邻接矩阵是一个 n 阶方阵且满足:

$$A[i][j]=\begin{cases} 1, & \text{若}(v_i,v_j)\text{或}\langle v_i,v_j \rangle\text{是 } E \text{ 中的边} \\ 0, & \text{若}(v_i,v_j)\text{或}\langle v_i,v_j \rangle\text{不是 } E \text{ 中的边} \end{cases}$$

有向图与无向图及其邻接矩阵如图 3-16 所示。

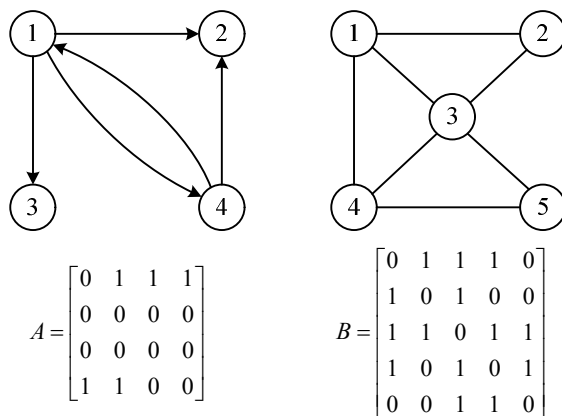


图 3-16 有向图与无向图及其邻接矩阵

(2) 邻接表。邻接表表示法是指为图中的每一个顶点建立一个单链表。有向图及其邻接表如图 3-17 所示。

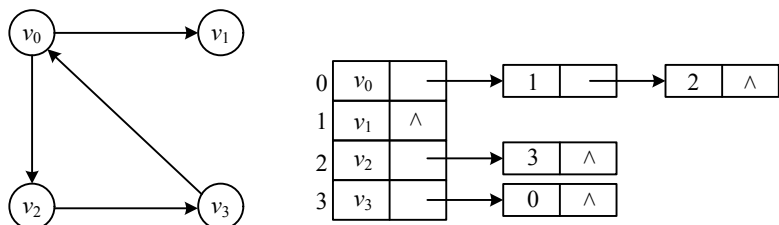


图 3-17 有向图及其邻接表

2. 图的遍历

图的遍历可以分为深度优先遍历、广度优先遍历两种方式,深度优先遍历类似于树的前序遍历,而广度优先遍历则相当于树的层次遍历。

3.3 数据运算

3.3.1 查找

1. 顺序查找

将待查的元素从头到尾与表中元素进行比较,如果存在,则返回成功;否则,查找失败。此方法效率不高。顺序查找的平均查找长度为 $(n+1)/2$ 。

2. 二分查找

二分查找的前提是元素有序(一般是升序),基本思路是拿中间元素 $A[m]$ 与要查找的元素 x 进行比较,如果相等,则表示找到;如果 $A[m]$ 比 x 大,那么要找的元素一定在 $A[m]$ 前边(左边);如果 $A[m]$ 比 x 小,那么要找的元素一定在 $A[m]$ 后边(右边)。每进行一次查找,数组规模减半。反复将子数组规模减半或使当前子数组为空,直到发现要查找的元素。

3. 哈希查找

设关键序列为 47、34、13、12、52、38、33、27、3,哈希表长为 11,哈希函数为 $\text{Hash}(key)=key \bmod 11$,则有:

$$\text{Hash}(47) = 47 \bmod 11 = 3, \text{Hash}(34) = 34 \bmod 11 = 1, \text{Hash}(13) = 13 \bmod 11 = 2,$$

$$\text{Hash}(12) = 12 \bmod 11 = 1, \text{Hash}(52) = 52 \bmod 11 = 8, \text{Hash}(38) = 38 \bmod 11 = 5,$$

$$\text{Hash}(33) = 33 \bmod 11 = 0, \text{Hash}(27) = 27 \bmod 11 = 6, \text{Hash}(3) = 3 \bmod 11 = 3.$$

对于产生的冲突,哈希函数可以采用线性探测法解决冲突,哈希地址和关键字的对应关系如表 3-2 所示。

表 3-2 哈希地址和关键字的对应关系

哈希地址	0	1	2	3	4	5	6	7	8
关键字	33	34	13	47	12	38	27	3	52

3.3.2 排序

1. 直接插入排序

在插入第 i 个记录时, $R_1, R_2, \dots, R_{i-1}$ 均已排好序, 这时将第 i 个记录依次与 $R_{i-1}, \dots, R_2, R_1$ 进行比较, 找到合适的位置插入, 插入位置及之后的记录依次向后移动。

例如: 43 55 70 30

结果: 30 43 55 70

直接插入排序在最好情况下的时间复杂度为 $O(n)$, 在最坏情况下的时间复杂度为 $O(n^2)$ 。

注: 时间复杂度的概念请参考第 8 小时内容。

2. 冒泡排序

通过相邻元素 (i 与 $i-1$) 之间的比较和交换, 将排序码较小的元素逐渐从底层移向顶层。整个过程像水底的气泡逐渐向上冒, 由此而得名冒泡排序。冒泡排序的时间复杂度为 $O(n^2)$ 。

例如: 43 55 70 30

排序过程如下:

```

43  55  70  30
      └───┘
43  55  30  70
      └───┘
43  30  55  70
└───┘
30  43  55  70

```

3. 简单选择排序

每一趟从待排序的数据元素中选出最小的元素, 顺序放在待排序数列的最前面, 直到全部待排序的数据元素全部排完。简单选择排序的时间复杂度为 $O(n^2)$ 。

例如: 43 55 70 30

第一趟, 最小值 30 与第一个元素交换: 30 55 70 45

第二趟, 最小值 45 与第二个元素交换: 30 45 70 55

第三趟, 最小值 55 与第三个元素交换: 30 45 55 70

4. 希尔排序

先将整个待排元素序列分成若干个子序列 (由相隔某个“增量”的元素组成) 后分别进行直接

插入排序，然后依次缩减增量再进行排序，待整个序列中的元素基本有序（增量足够小）时，再对全体元素进行一次直接插入排序。因为直接插入排序在元素基本有序的情况下（接近最好情况），效率是很高的。希尔排序的时间复杂度为 $O(n^{1.3})$ 。

先取一个小于 n 的整数 d_1 作为第一个增量，把文件的全部记录分成 d_1 个组，即将所有距离为 d_1 倍数序号的记录放在同一个组中，在组内进行直接插入排序；然后取第二个增量 $d_2 < d_1$ ，重复上述步骤，依次类推，直到所取的增量 $d_i = 1$ ，即将所有记录放在同一组进行直接插入排序。例如：

初始关键字：48 37 64 96 75 12 26 48 54 03

48 12

37 26

64 48

96 54

75 03

第一趟排序结果：12 26 48 54 03 48 37 64 96 75

12 54 37 75

26 03 64

48 48 96

第二趟排序结果：12 03 48 37 26 48 54 64 96 75

第三趟排序结果：03 12 26 37 48 48 54 64 75 96

5. 快速排序

快速排序是对冒泡排序的一种改进。基本思路是：通过一趟排序将要排序的数据分成独立的两个部分，其中一部分的所有数据都比另外一部分的所有数据要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。快速排序在最好情况下的时间复杂度为 $O(n \log_2 n)$ ；在最坏情况下，即初始序列按关键字有序或基本有序时，快速排序的时间复杂度为 $O(n^2)$ 。

例如：43 55 70 30 选择 43 为基准值， $i=0, j=3$ 。

$i=0$ $j=3$

30 55 70 43 $43 > 30$ ，两个元素交换位置，移动 i 。

$i=1$ $j=3$

30 43 70 55 $55 > 43$ ，两个元素交换位置，移动 j 。

$i=1$ $j=2$

30 43 70 55 $70 > 43$ ，不需要交换，移动 j 。

$i=1$ $j=2$

30 43 70 55 接下来对 70、55 再进行快速排序，依次类推。

$i=j=1$

6. 堆排序

对于 n 个元素的序列，当且仅当满足下列关系时称其为堆，其中， $2i$ 和 $2i+1$ 不大于 n 。

$K_i \leq K_{2i}$ 且 $K_i \leq K_{2i+1}$ 或 $K_i \geq K_{2i}$ 且 $K_i \geq K_{2i+1}$ ，其中，前者称为小顶堆，后者称为大顶堆。

堆排序的基本思路是：对于一组待排序记录的关键字，首先按堆的定义排成一个序列（建立初始堆），之后输出堆顶最大关键字（对于大顶堆而言），然后将剩余的关键字再调整成新堆，从而得到次大关键字，如此反复，直到全部关键字排成有序序列为止。堆排序（以大顶堆为例）的建立过程与调整新堆的过程如图 3-18 和图 3-19 所示。堆排序的时间复杂度为 $O(n\log_2 n)$ 。

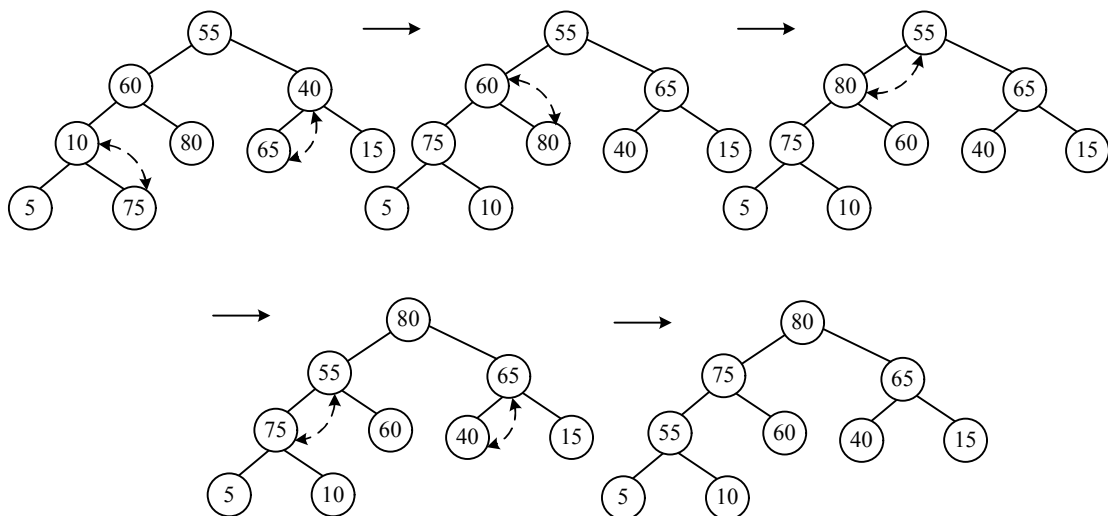


图 3-18 初始大顶堆的建立过程

7. 归并排序

归并排序过程分为三步：

- (1) 分解。将 n 个元素分成各含 $n/2$ 个元素的子序列。
- (2) 求解。用归并排序对两个子序列递归地排序。

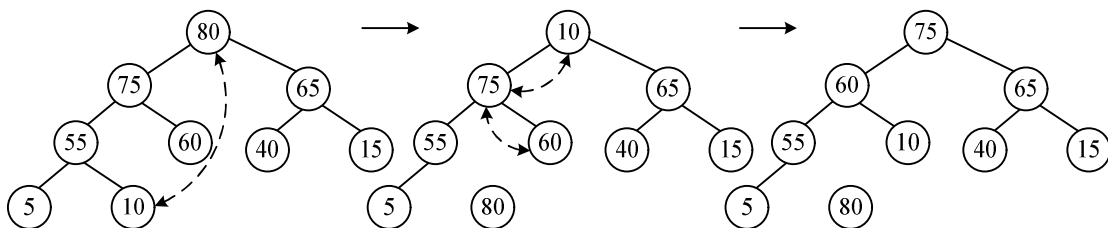


图 3-19 调整新堆的过程

- (3) 合并。合并两个已经排好序的子序列以得到排序结果。

例如：43 55 70 30

第一次排序的结果: [43 55] [30 70]

第二次排序的结果: [30 43 55 70]

归并排序的时间复杂度为 $O(n\log_2^n)$ 。

3.4 练习题

1. 对于线性表, 相比顺序存储, 采用链表存储的缺点是_____。

- A. 数据元素之间的关系需要占用存储空间, 导致存储密度不高
- B. 表中结点必须占用地址连续的存储单元, 存储密度不高
- C. 插入新元素时需要遍历整个链表, 运算的时间效率不高
- D. 删除元素时需要遍历整个链表, 运算的时间效率不高

解析:

链表存储的缺点为数据元素之间的关系需要占用存储空间, 导致存储密度不高。

答案: A

2. 快速排序算法在排序过程中, 在待排序数组中确定一个元素为基准元素, 根据基准元素把待排序数组划分成两个部分, 前面一部分元素值小于等于基准元素, 而后面一部分元素值大于基准元素, 再分别对前后两个部分进行进一步划分。根据上述描述, 快速排序算法采用了____(a)____算法设计策略。已知确定基准元素操作的时间复杂度为 $O(n)$, 则快速排序算法在最好和最坏情况下的时间复杂度为____(b)_____。

- | | | | |
|-----------------------------|----------------------|------------------------------|---------------------------|
| (a) A. 分治 | B. 动态规划 | C. 贪心 | D. 回溯 |
| (b) A. $O(n)$ 和 $O(n\lg n)$ | B. $O(n)$ 和 $O(n^2)$ | C. $O(n\lg n)$ 和 $O(n\lg n)$ | D. $O(n\lg n)$ 和 $O(n^2)$ |

解析:

快速排序采用分治法的思想。快速排序在最好情况下的时间复杂度为 $O(n\lg n)$; 在最坏情况下, 即初始序列按关键字有序或基本有序时, 时间复杂度为 $O(n^2)$ 。

答案: A、D

3. 若对线性表最常用的操作是访问任意指定序号的元素, 并在表尾加入和删除元素, 则适宜采用_____存储。

- A. 顺序表
- B. 单链表
- C. 双向链表
- D. 哈希表

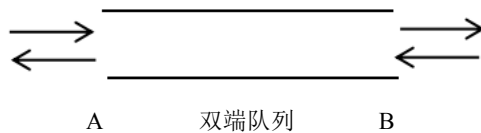
解析:

线性表的顺序存储方式可以随机存取表中元素, 同时在表尾加入和删除元素比在表头等处操作效率高。

答案: A

4. 某双端队列如下图所示, 要求元素进出队列必须在同一端口, 即从 A 端进入的元素必须从 A 端出, 从 B 端进入的元素必须从 B 端出, 则对于四个元素的序列 e_1 、 e_2 、 e_3 、 e_4 , 若要求前两个

元素 e_1 、 e_2 从 A 端按次序全部进入队列，后两个元素 e_3 、 e_4 从 B 端按次序全部进入队列，则可能得到的出队序列是_____。



A. e_1 、 e_2 、 e_3 、 e_4

B. e_2 、 e_3 、 e_4 、 e_1

C. e_3 、 e_4 、 e_1 、 e_2

D. e_4 、 e_3 、 e_2 、 e_1

解析：

根据题意，从 A 端进入的元素必须从 A 端出，从 B 端进入的元素必须从 B 端出，则 A 端的出队顺序是 e_2 、 e_1 ，B 端的出队顺序是 e_4 、 e_3 。只有 D 选项符合要求。

答案： D

第4小时

操作系统知识

4.0 章节考点分析

根据考试大纲，本小时内容比较重要，需要掌握进程管理、存储管理、设备管理、文件管理、作业管理等基础知识，上午的客观题约占7分。

本小时的内容偏重于理解与应用，本小时的架构如图4-1所示。

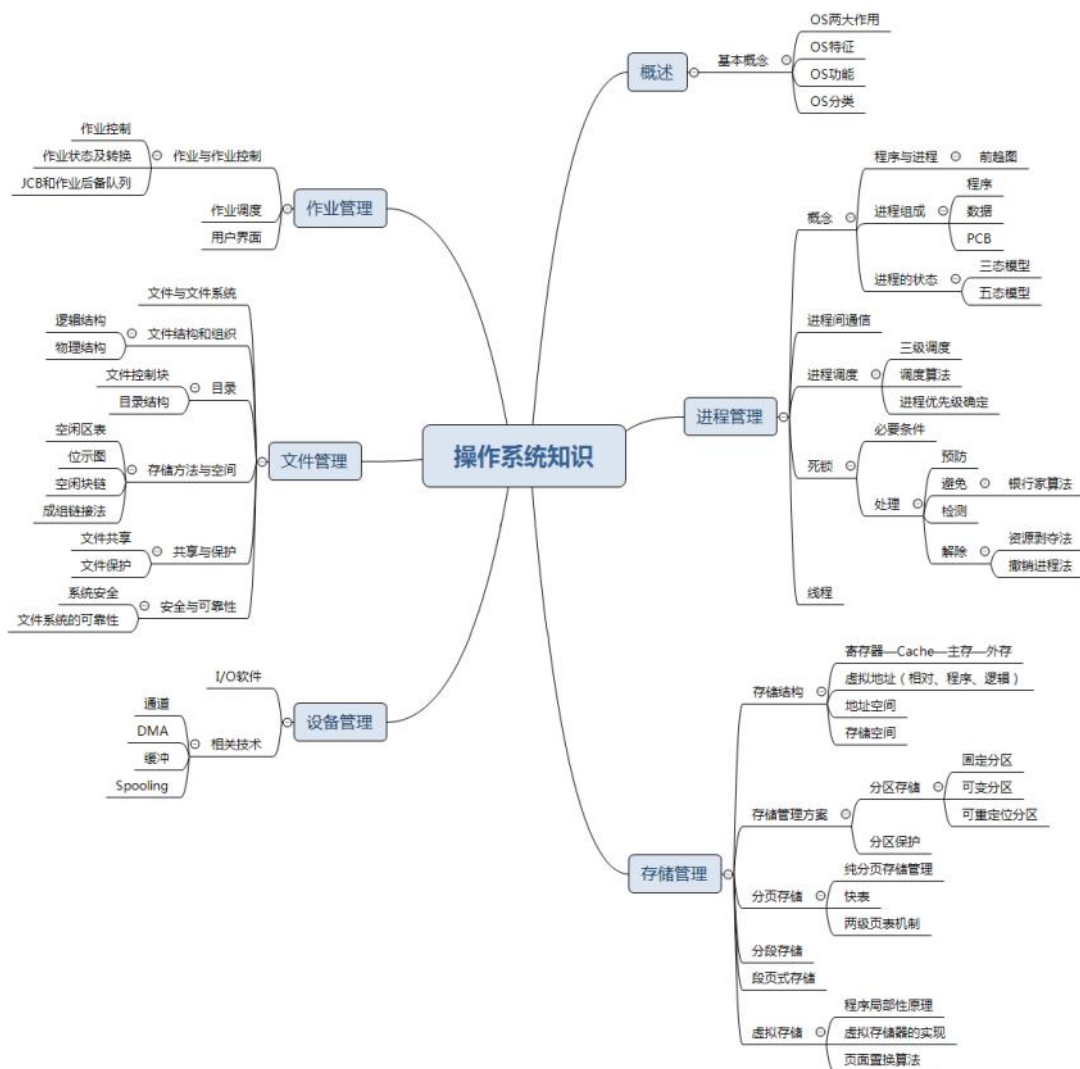


图 4-1 第4小时架构图



【导读小贴士】

操作系统（Operating System, OS）是管理和控制计算机硬件与软件资源的计算机程序，是直接接在“裸机”上运行的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行，其关系到软件基础产业，关系到国家信息化战略，不管是PC时代、移动互联网时代，还是在软件驱

动、软件定义大行其道的当下，得 OS 者得天下，失 OS 者失天下。

4.1 OS 概述

1. OS 两大作用

- (1) 通过资源管理提高计算机系统的效率。
- (2) 改善人机界面，向用户提供良好的工作环境。

2. OS 特征

并发性、共享性、虚拟性、不确定性。

3. OS 功能

进程管理、文件管理、存储管理、设备管理、作业管理。

4. OS 分类

批处理、分时、实时、网络（集中、C/S、对等）、分布式、微型、嵌入式。

4.2 进程管理

1. 进程间通信

同步与互斥、信号量、PV 操作。

2. 进程的状态

三态模型、五态模型。

3. 进程调度

- (1) 三级调度。
- (2) 调度算法：先来先服务、时间片轮转（固定、可变）、优先级调度、多级反馈调度。
- (3) 进程优先级确定。

4. 死锁

- (1) 必要条件：互斥、请求保持、不可剥夺、环路。
- (2) 处理：预防、避免（银行家算法）、检测、解除。

4.3 存储管理

1. 存储结构

- (1) 寄存器—Cache—主存—外存。
- (2) 虚拟地址：又称相对地址、程序地址、逻辑地址等。
- (3) 地址空间。
- (4) 存储空间。

2. 存储管理方案
- (1) 固定分区。

(2) 可变分区：最佳适应、最差适应、首次适应、循环首次适应。
3. 分页存储
- (1) 存分页存储管理：分页原理、地址结构、地址映射。

(2) 快表。

(3) 两级页表机制。
4. 分段存储
5. 段页式存储
6. 虚拟存储
- (1) 程序局部性原理：时间局部性、空间局部性。

(2) 虚拟存储器的实现：分页、分段、段页。

(3) 页面置换算法：最佳置换、FIFO、LRU、NRU。

4.4 设备管理

1. I/O 软件
- I/O 软件的操作过程如图 4-2 所示。

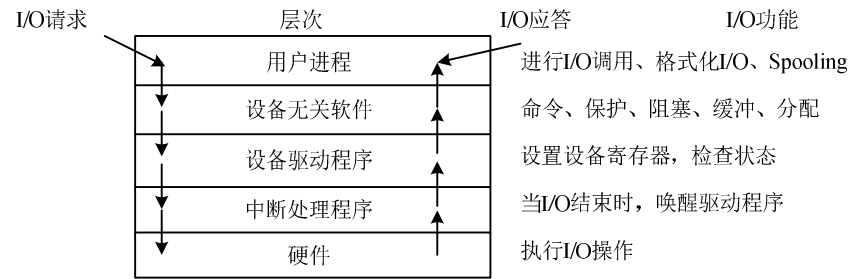


图 4-2 I/O 软件

2. 相关技术
- 通道、DMA、缓冲、Spooling。

4.5 文件管理

1. 文件与文件系统
- (1) 文件。

(2) 文件系统。

(3) 文件类型。

2. 文件结构和组织

(1) 逻辑结构。

(2) 物理结构：连续、链接、索引、索引表。

3. 目录

文件控制块、目录结构（一级、二级、多级）。

4. 存储方法与空间

(1) 空闲区表。

(2) 位示图。

(3) 空闲块链。

(4) 成组链接法。

5. 共享与保护

(1) 文件共享：硬链接、符号链接（软链接）。

(2) 文件保护：存取控制矩阵、存取控制表、用户权限表、密码。

6. 安全与可靠性

(1) 系统安全：系统级、用户级、目录级、文件级。

(2) 文件系统的可靠性：转储和恢复、日志文件、一致性。

4.6 作业管理

1. 作业与作业控制

(1) 作业控制。

(2) 作业状态及转换：提交、后备、执行、完成。

(3) JCB 和作业后备队列。

2. 作业调度

先来先服务、短作业优先、响应比高优先、优先级调度算法、均衡调度算法、调度算法均衡指标。

4.7 练习题

1. 假设系统采用 PV 操作系统实现进程同步与互斥，若有 n 个进程共享一台扫描仪，那么当信号量 S 的值为 -3 时，表示系统中有_____个进程等待使用扫描仪。

A. 0

B. $n-3$

C. 3

D. n

解析：

在 PV 操作中，信号量用于表示系统中现有资源的数量，当信号量的值为负数时，代表这类资

源系统已经分配完毕。此时，对负数取绝对值就能得到当前等待进程的数量。

答案：C

2. 假设系统中有三类互斥资源 R_1 、 R_2 和 R_3 ，可用资源分别为 10、5 和 3，在 T_0 时刻系统中有 P_1 、 P_2 、 P_3 、 P_4 、 P_5 五个进程，这些进程对资源的最大需求量和已分配资源数如表 4-1 所示，此时系统剩余的可用资源数分别为 (a)。如果进程按 (b) 序列执行，那么系统状态是安全的。

表 4-1 五个进程对资源的最大需求量和已分配资源数

进程 \ 资源	最大需求量			已分配资源数		
	R_1	R_2	R_3	R_1	R_2	R_3
P_1	5	3	1	1	1	1
P_2	3	2	0	2	1	0
P_3	6	1	1	3	1	0
P_4	3	3	2	1	1	1
P_5	2	1	1	1	1	0

(a) A. 1、1、0 B. 1、1、1 C. 2、1、0 D. 2、0、1

(b) A. $P_1-P_2-P_4-P_5-P_3$ B. $P_5-P_2-P_4-P_3-P_1$

C. $P_4-P_2-P_1-P_5-P_3$ D. $P_5-P_1-P_4-P_2-P_3$

解析：

本题考查银行家算法。首先需要求系统剩余资源，计算方法是将总资源数逐一减去已分配资源数。

R_1 剩余的可用资源数为：10-1-2-3-1-1=2；

R_2 剩余的可用资源数为：5-1-1-1-1-1=0；

R_3 剩余的可用资源数为：3-1-1=1。

接下来分析按什么样的序列执行，系统状态才是安全的，所谓系统状态安全是指不产生死锁。在进行分析时，需要先了解每个进程各类资源还需要多少个，此信息可以通过最大需求量减去已分配资源数获得，情况如表 4-2 所示。

表 4-2 五个进程的还需资源数

进程 \ 资源	最大需求量			已分配资源数			还需资源数		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_1	5	3	1	1	1	1	4	2	0
P_2	3	2	0	2	1	0	1	1	0
P_3	6	1	1	3	1	0	3	0	1
P_4	3	3	2	1	1	1	2	2	1
P_5	2	1	1	1	1	0	1	0	1

答案: D、B

A. 硬件自动完成

B. 程序员调度

C. 操作系统管理

D. 程序员与操作系统协作完成

Cache 与主存的地址映像需要专门的硬件自动完成，使用硬件来处理具有更高的转换速率。

4. 假设段页式存储管理系统中的地址结构如下图所示, 则系统中_____。

31	24	23	12	11	0
段号		页号		页内地址	

- A. 页的大小为 4K，每个段的大小均为 4096 个页，最多可有 256 个段。
B. 页的大小为 4K，每个段最大只许有 4096 个页，最多可有 256 个段。
C. 页的大小为 8K，每个段的大小均为 2048 个页，最多可有 128 个段。
D. 页的大小为 8K，每个段最大只许有 2048 个页，最多可有 128 个段。

已知页内地址是从第 0 位到第 11 位，共 12 个位，一页的大小为 $2^{12} = 4K$ 。

页号是从第 12 位到第 23 位, 共 12 个位, 一段中有 $2^{12} = 4096$ 个页。

段号是从第 24 位到第 31 位，共 8 个位，一共有 $2^8 = 256$ 个段。

答案: B

5. 某文件管理系统采用位示图(bitmap)记录磁盘的使用情况,如果系统的字长为 32 位,磁盘物理块的大小为 4MB,物理块依次编号为 0、1、2……,位示图字依次编号为 0、1、2……。16385 号物理块的使用情况在位示图中的第 (a) 个字中描述;如果磁盘的容量为 1000GB,那么,位示图需要 (b) 个字来表示。

- (a) A. -128 B. 256
C. 512 D. 1024
- (b) A. 1200 B. 3200
C. 6400 D. 80100

解析:

由于物理块是从 0 开始编号的, 所以 16385 号物理块是第 16386 块。 $16386/32 = 512.0625$, 所以 16385 号物理块的使用情况在位示图中的第 513 个字中描述。

磁盘的容量为 1000GB, 物理块的大小为 4MB, 则磁盘共有 $(1000 \times 1024/4)$ 个物理块, 一个字对应 32 个物理块, 位示图的大小为 $1000 \times 1024/(32 \times 4) = 8000$ 个字。

答案: C、D

第5小时

软件工程基础知识

5.0 章节考点分析

第5小时主要学习计算机软件、软件过程模型、软件开发方法、软件开发环境、软件项目管理、软件风险管理、软件度量管理等内容。

根据考试大纲，本小时的知识点会涉及单选题型，约占2~5分。本小时的内容偏重于概念知识，根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，考查的概念知识点多数参照教材，扩展内容较少。本小时的架构如图5-1所示。



图 5-1 第5小时架构图



【导读小贴士】

本小时涉及的内容是较为重要的知识点，作为软件设计师考试，一些选择题、案例分析题会来自本小时，建议除了在理解的基础上掌握核心知识点外，还要学会灵活应用。

5.1 软件工程概述

软件工程指的是应用计算机科学、数学及管理科学等原理，以工程化的原则和方法来解决软件问题的工程，目的是提高软件生产率、提高软件质量、降低软件成本。

5.1.1 计算机软件

计算机软件指的是计算机系统上的程序及其文档。程序是计算任务的处理对象和处理规则的描述。任何以计算机为处理工具的任务都是计算任务。

按照软件的应用领域，将计算机软件分为以下十类，包括：①系统软件；②应用软件；③工程/科学软件；④嵌入式软件；⑤产品线软件；⑥Web 应用软件（Web APP）；⑦人工智能软件；⑧开放计算；⑨网络资源；⑩开源软件。

5.1.2 软件工程基本原理

美国著名的软件工程专家 B.W.Boehm 于 1983 年提出了软件工程的七条基本原理，包括：①用分阶段的生命周期计划严格管理；②坚持进行阶段评审；③实现严格的产品控制；④采用现代的程序设计技术；⑤结果应能清楚地审查；⑥开发小组的人员应少而精；⑦承认不断改进软件工程实践的必要性。

5.1.3 软件生存周期

同任何事物一样，一个软件产品或软件系统要经历孕育、诞生、成长、成熟、衰亡等阶段，一般称为软件生存周期。软件生存周期包括以下七个方面：

（1）可行性分析与项目开发计划。这个阶段主要确定软件的开发目标及其可行性。参与该阶段的人员有用户、项目负责人、系统分析师。产生的文档有可行性分析报告、项目开发计划。

（2）需求分析。该阶段的任务不是具体的解决问题，而是要确定软件系统要做什么，确定软件系统的功能、性能、数据和界面等要求，从而确定系统的逻辑模型。参与该阶段的人员有用户、项目负责人、系统分析师。产生的文档主要是软件需求说明书。

（3）概要设计。该阶段开发人员把确定的各项功能需求转换成需要的体系结构。概要设计就是设计软件的结构，明确软件由哪些模块组成，这些模块层次结构是怎样的，调用关系是怎样的，

每个模块的功能是什么。参与该阶段的人员有系统分析师、软件设计师。产生的文档主要是概要设计说明书。

(4) 详细设计。该阶段的主要任务是对每个模块的功能进一步详细、具体的描述。参与该阶段的人员有软件设计师、程序员。产生的文档主要是详细设计文档。

(5) 编码。把每个模块的控制结构转换成计算机可接受的程序代码,即写成某种特定程序设计语言表示的源程序清单。

(6) 测试。测试是保证软件质量的重要手段。参加测试的人员通常是另一部门(或单位)的软件设计师或系统分析师。产生的文档主要是软件测试计划、测试用例、测试报告。

(7) 维护。软件维护是软件生存周期中时间最长的阶段。软件已交付且正式投入使用后,便进入维护阶段。对软件进行修改的原因包括:①运行中发现隐含的错误而需要修改;②为了适应变化的(或变化后的)工作环境而修改;③需要对软件功能进行扩充、增强而进行的修改;④为将来软件维护活动做预先准备。

5.1.4 软件过程

软件开发中遵循一系列可预测的步骤(即路线图),该路线图称为软件过程。过程是活动的集合,活动是任务的集合。

软件过程有三层含义:

(1) 个体含义:指某产品、系统在生存周期中的某一类活动的集合,如开发过程、管理过程等。

(2) 整体含义:指软件产品、系统在所有上述含义下的软件过程的总体。

(3) 工程含义:指解决软件过程的工程。

1. 能力成熟度模型(CMM)

CMM 是对软件组织进化阶段的描述,随着软件组织定义、实施、测量、控制和改进其软件过程,软件组织的能力经过这些阶段逐步提高。CMM 将软件过程的改进分为五个成熟度级别,如图 5-2 所示。

2. 能力成熟度模型集成(CMMI)

CMMI 提供了两种表示方法:阶段式模型和连续式模型。

(1) 阶段式模型。结构类似于 CMM,它关注组织的成熟度。CMMI-SE/SW/IPPD 1.1 版本中有五个成熟度等级。

- 1) 初始级:过程不可预测且缺乏控制。
- 2) 已管理级:过程为项目服务。
- 3) 已定义级:过程为组织服务。
- 4) 定量管理级:过程已度量和控制。
- 5) 优化级:集中过程改进。

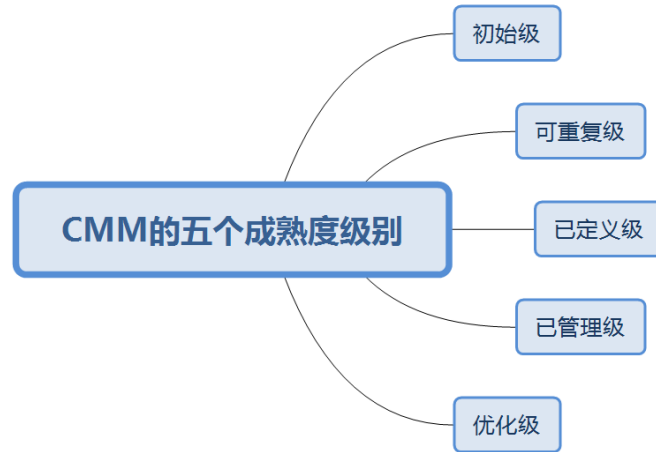


图 5-2 CMM 的五个成熟度级别

(2) 连续式模型。关注每个过程域的能力，一个组织对不同的过程域可以达到不同的过程域能力等级（简称 CL）。CMMI 中包括六个过程域能力等级，如图 5-3 所示。

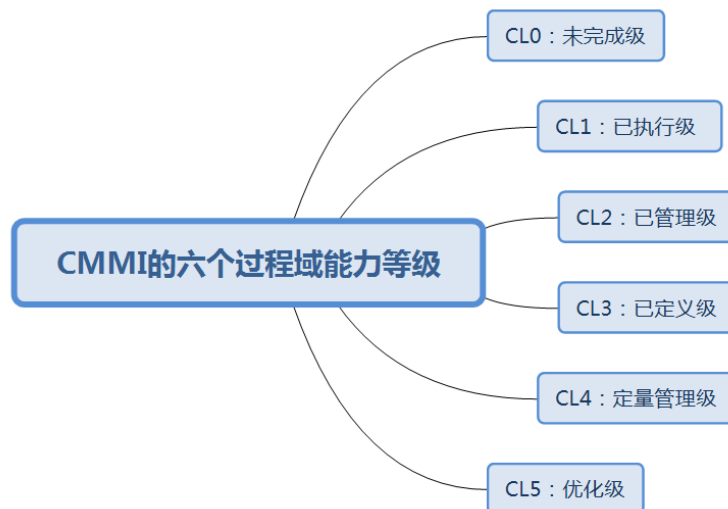


图 5-3 CMMI 的六个过程域能力等级

3. 统一过程（UP）

统一过程定义了四个技术阶段及其产品：

- (1) 起始阶段：专注于项目初创活动。
- (2) 精化阶段：在理解了最初领域范围之后，需要进行需求分析和架构演进。
- (3) 构建阶段：关注系统的构建，产生实现模型。

随着 UP 的阶段进展,每个核心工作流的工作量发生变化,四个技术阶段由主要里程碑所终止。

- ## 5.2 软件过程模型

5.2.1 瀑布模型

```

graph TD
    A[需求分析] --> B[设计]
    B --> C[编码]
    C --> D[测试]
    D --> E[运行与维护]
    B -.-> A
    C -.-> B
    D -.-> C
    E -.-> D
  
```

图 5-4 瀑布模型

瀑布模型的优点：容易理解、成本低、强调开发的阶段性早期计划及需求调查和产品测试。

在瀑布模型中,需求或设计的错误往往只有到了项目后期才能被发现,对项目风险的控制能力较弱,导致项目通常会延期,开发费用超出预算。

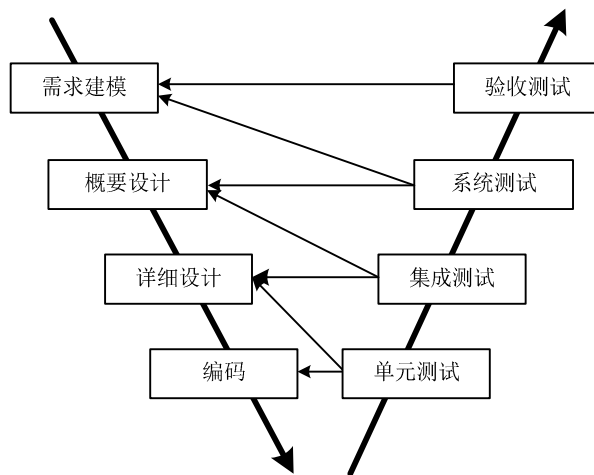


图 5-5 V 模型

5.2.2 增量模型

增量模型融合了瀑布模型的基本成分和原型实现的迭代特征，如图 5-6 所示。

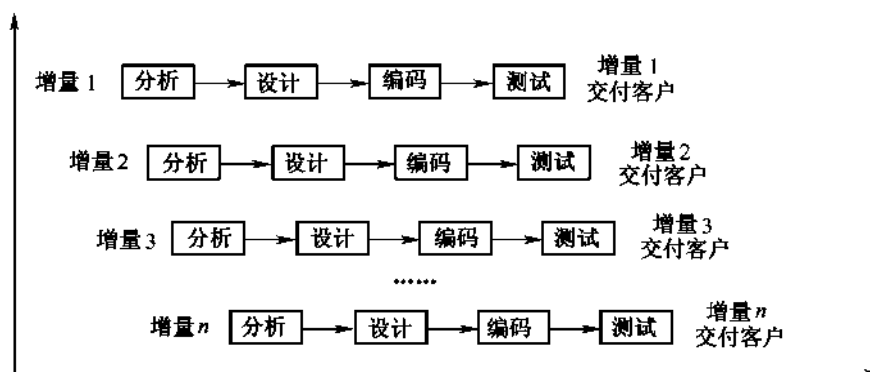


图 5-6 增量模型

增量模型的优点：可交付的第一个版本所需要的成本和时间很少，开发由增量表示的小系统所承担的风险不大，由于很快发布了第一个版本，因此可减少用户需求的变更。同时，它也具有瀑布模型所有的优点。

增量模型的缺点：若没有对用户的变更要求进行规划，那么产生的初始增量可能会造成后来增量的不稳定；若需求不像早期思考的那样稳定和完整，那么一些增量就可能需要重新开发或重新发布；管理发生的成本、进度和配置的复杂性可能会超出组织的能力。

5.2.3 演化模型

典型的演化模型有原型模型和螺旋模型。

1. 原型模型

原型模型开始于沟通，目的是定义软件的总体目标、标识需求，然后快速制定原型开发计划，确定原型的目标和范围，快速构建原型并交付用户使用，收集客户反馈意见，并在下一轮中对原型进行改进。在前一个原型需要改进（或扩展其范围）的时候，进入下一轮原型的迭代开发。原型模型如图 5-7 所示。

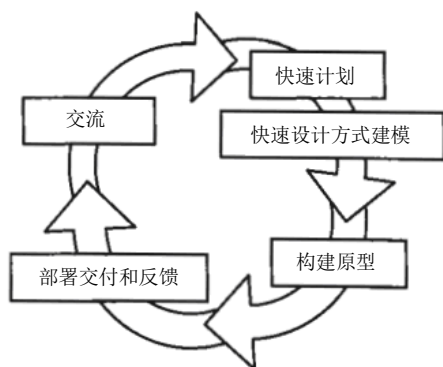


图 5-7 原型模型

原型模型又根据使用目的的不同，分为探索型原型、实验型原型和演化型原型。

2. 螺旋模型

对于一个复杂的大项目，开发一个原型往往达不到要求。螺旋模型将瀑布模型和演化模型结合起来，加入两种模型均忽略的风险分析，弥补了这两种模型的不足。螺旋模型如图 5-8 所示。

螺旋模型中的每个螺旋周期分为以下四个步骤：

- (1) 制订计划：确定软件目标，选定实施方案，明确项目开发的限制条件。
- (2) 风险分析：对所选方案进行分析，识别风险，消除风险。
- (3) 实施工程：实施软件开发，验证阶段性产品。
- (4) 用户评估：评价开发工作，提出修正建议，建立下一个周期的开发计划。

螺旋模型强调风险分析，使用户、开发人员对演化层出现的风险有所了解，从而作出反映。因此，螺旋模型适合用于庞大、复杂、高风险的系统。

5.2.4 喷泉模型

喷泉模型是以用户需求为动力、以对象为驱动力的模型。适用于面向对象的开发方法，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。喷泉模型使开发过程具有迭代性和无间隙性。喷泉模型如图 5-9 所示。



图 5-8 螺旋模型

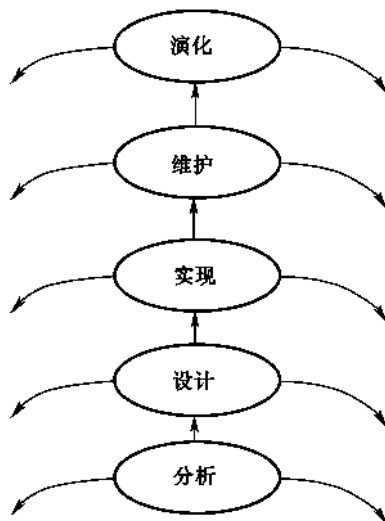
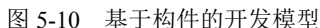


图 5-9 喷泉模型

基于构件的开发模型是指利用预先包装的构件来构造应用系统。构件可以是组织内部开发的构件，也可以是商品化成品（COTS）软件构件。一种基于构件的开发模型包括领域工程和应用系统工程。

- 基于构件的开发模型如图 5-10 所示。



形式化方法是建立在严格数学基础上的一种软件开发方法,主要活动是生成计算机软件形式化的数学规格说明。

统一过程（UP）模型是一种“用例和风险驱动，以架构为中心，迭代并增量”的开发过程，由 UML 方法和工具支持。统一过程定义了四个技术阶段及其主要工作产品：

- 64

例模型、补充需求、分析模型、整体体系结构描述、可执行的软件体系结构原型、初步设计模型、修订的风险列表、项目计划及初始用户手册。

(3) 构建阶段：关注系统的构建，产生实现模型，主要工作产品有设计模型、软件构件、集成软件增量、测试计划及步骤、测试用例及支持文档（用户手册、安装手册等）。

(4) 移交阶段：关注软件提交方面的工作，产生软件增量，主要工作产品有提交的软件增量、 β 测试报告和综合用户反馈。

统一过程的典型代表是 RUP，RUP 是 UP 的商业扩展，完全兼容 UP，比 UP 更完整、更详细。

敏捷方法的总体目标是通过“尽可能早地、持续地对有价值的软件进行交付”使客户满意。敏捷过程的典型方法有很多，每一种方法基于一套原则，这些原则实现了敏捷方法所宣称的理念，即敏捷宣言。

常用的方法：极限编程(XP)、水晶法(Crystal)、并列争球法(Scrum)、自适应软件开发(ASD)、敏捷统一过程(AUP)。

5.3 软件项目需求分析

需求分析也称为软件需求分析、系统需求分析或需求分析工程等，是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么的过程。

5.3.1 软件需求

软件需求包括以下内容：

- (1) 功能需求：考虑系统要做什么、什么时候做、如何修改或升级。
- (2) 性能需求：考虑软件开发的技术性指标，如存储容量限制、执行速度、响应时间、吞吐量等。
- (3) 用户或人的因素：考虑用户的类型。
- (4) 环境需求：考虑软件应用的环境。
- (5) 界面需求：考虑来自其他系统的输入或到其他系统的输出等。
- (6) 文档需求：考虑需要哪些文档、文档针对哪些读者。
- (7) 数据需求：考虑输入、输出格式，接收、发送数据的频率，数据的精准度、数据流量、数据保持时间。
- (8) 资源使用需求：考虑软件运行时所需要的资源。
- (9) 安全保密需求：考虑是否需要访问系统或系统信息加以控制。
- (10) 可靠性需求：考虑系统的可靠性需求、系统是否必须检测和隔离错误、出错后重启系统所允许的时间等。
- (11) 软件成本消耗或开发进度需求：考虑开发是否有规定的时间表。

(12) 其他非功能性需求：如采用某种开发模式、确定质量控制标准、里程碑和评审、验收标准等。

5.3.2 需求分析原则

需求分析过程有不同的分析方法，它们要遵循的操作原则有：①必须能表示和理解问题的信息域；②必须能定义软件将完成的任务；③必须能表示软件的行为；④必须划分描述数据、功能和行为的模型；⑤分析过程应该从要素信息移向细节信息。

5.3.3 需求工程

需求工程是一个不断反复的需求定义、文档记录、需求演进的过程，并最终在验证的基础上冻结需求。

需求工程可以细分为六个阶段：①需求获取；②需求分析与协商；③系统建模；④需求规约；⑤需求验证；⑥需求管理。

5.4 软件项目系统设计

系统设计的主要内容包括新系统总体结构设计、代码设计、输出设计、输入设计、处理过程设计、数据存储设计、用户界面设计和安全控制设计等。

常用的设计方法有：面向数据流的结构化设计方法（SD）、面向对象的分析方法（OOD）。

系统设计包括两个基本的步骤：概要设计、详细设计。

5.4.1 概要设计

概要设计主要包括：①软件系统总体结构设计；②数据结构及数据库设计；③编写概要设计文档（概要设计说明书、数据库设计说明书、用户手册及修订测试计划）；④评审。

5.4.2 详细设计

详细设计主要包括：①对每个模块进行详细设计；②对模块内部的数据结构进行设计；③对数据库进行物理设计，即确定数据库的物理结构；④其他设计（代码设计、输入/输出设计、用户界面设计）；⑤编写详细设计说明书；⑥评审。

5.5 软件项目系统测试

软件项目系统测试是对整个系统的测试，将硬件、软件、操作人员看作一个整体，检验它是否有不符合系统说明书的地方。这种测试可以发现系统分析和设计中的错误。

5.5.1 系统测试与调试

信息系统测试包括软件测试、硬件测试、网络测试。测试的目的是以最少的人力和时间发现潜在的各种错误和缺陷。

测试应遵循的基本原则：①应尽早并不断地进行测试；②测试工作应避免原先开发软件的人员或小组参与；③设计测试方案时要确定输入数据，还要根据系统功能确定预期的输出结果；④设计测试用例时要设计合理、有效的输入条件，还要包含不合理、失效的输入条件。人们在测试时通常忽略了对异常、不合理、意想不到的情况进行测试，这可能就是隐患；⑤在测试时要检查程序是否做了该做、不该做的事，多余的工作会影响程序的效率；⑥严格按照测试计划进行测试；⑦妥善保存测试计划、测试用例；⑧要精心设计测试用例。

测试的过程包括：①制定测试计划；②编制测试大纲；③根据测试大纲设计和生产测试用例；④事实测试；⑤生成测试报告。

5.5.2 传统软件的测试策略

1. 单元测试

单元测试也称模块测试，在模块编写完成且编译无误后进行，侧重于模块中的内部处理逻辑和数据结构。单元测试环境如图 5-11 所示。

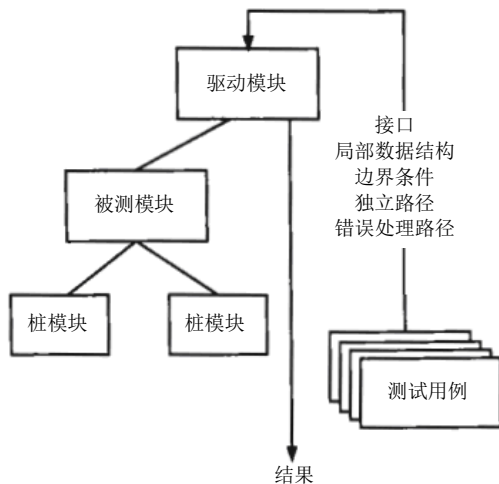


图 5-11 单元测试环境

2. 集成测试

集成测试通常有以下两种方法：

- (1) 非增量集成：分别测试各个模块，再将这些模块组合起来进行整理测试。
- (2) 增量集成：以小增量的方式逐步进行构造和测试。

常用的增量集成策略包括：自顶向下集成测试、自底向上集成测试、回归测试、冒烟测试等。

3. 确认测试

确认测试始于集成测试的结束，那时已测试完单个构件，软件已经组装成完整的软件包，而且接口错误已被发现和改正。

确认过程的一个重要成分是配置评审，主要检查软件、文档、数据是否齐全、分类有序。

4. 系统测试

系统测试是将已经确认的软件、硬件、外设、网络等其他因素结合在一起，进行各种集成测试和确认测试，主要包括恢复测试、安全性测试、压力测试、性能测试、部署测试。

5.5.3 测试方法

软件测试分为静态测试和动态测试。

(1) 静态测试：被测程序不在机器上运行，采用人工检测和计算机辅助静态分析的手段对程序进行测试，包括人工检测、计算机辅助静态分析。

(2) 动态测试：通过运行程序发现错误，一般采用黑盒测试和白盒测试。

1) 黑盒测试：也称功能测试，在不考虑软件内部结构和特性的情况下，测试软件的外部特性。

2) 白盒测试：也称结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的路径和过程进行测试，检查是否满足设计的需要。

5.5.4 调试

目前常用的调试方法有以下五种：

(1) 试探法：调试人员分析错误的症状，猜测问题所在的位置，一步步试探和分析问题所在。该方法效率低，适用于结构比较简单的程序。

(2) 回溯法：调试人员从发现错误症状的位置开始，人工沿着程序的控制流程往回追踪代码，直到找出问题根源为止。该方法适用于小型程序。

(3) 对分查找法：该方法主要用来缩小错误范围，直到把故障范围缩小到比较容易诊断为止。

(4) 归纳法：从测试所暴露的问题出发，收集所有正确、不正确的数据，并分析它们之间的关系，提出假想的错误原因，用这些数据证明或反驳，从而查出错误所在。

(5) 演绎法：根据测试结果，列出可能的错误原因，分析已有的数据，排除不可能和彼此矛盾的原因。若有多个错误同时存在，就要重新分析，提出新的假设，直到发现错误为止。

5.6 软件项目管理

5.6.1 项目管理涉及的范围

有效的软件项目管理集中在以下四点：人员（person）、产品（product）、过程（procedure）、

项目（project）。

（1）人员：参与项目的人员类型有项目管理人员、高级管理人员、开发人员、客户、最终用户。

（2）产品：开展项目计划之前，应先进行项目定义，即定义项目范围，其中包括建立产品的目的和范围、可选的解决方案、技术、管理约束等。

（3）过程：对于软件项目来说，强调的是对其进行过程控制，通常将项目分解为任务、子任务等。

（4）项目：常识的软件项目办法有明确目标及过程、保持动力、跟踪进展、作出明智的决策、进行事后分析。

5.6.2 项目估算

常用的估算方法有三种：基于已经完成的类似项目进行估算、基于分解技术进行估算、基于经验估算模型的估算。

常用的成本估算方法有：自顶向下估算方法、自底向上估算方法、差别估算方法、其他估算方法（如专家估算、类推估算、算式估算）。

5.6.3 进度管理

1. 进度管理的基本原则

（1）划分：项目必须要被划分成若干个可以管理的活动和任务。

（2）相互依赖性：划分后的各个活动之间的依赖关系必须是明确的，如有的任务必须按顺序完成，有的任务可以并发进行，有的任务只能在其他活动完成后才能开展，有的任务则可以独立进行。

（3）时间分配：必须为每个任务规定开始和结束时间。

（4）工作量确认：每个项目都有预定的人员参与，项目管理者在任何时间节点中所分配的人员数量不能超过项目团队的总人数。

（5）确定责任：为每个任务指定特定的团队成员进行负责。

（6）明确输出结果：每个任务都要有一个明确的输出结果，如一个可交付的工作产品。

（7）确定里程碑：每个任务或任务组都应该与一个项目里程碑相关联，当一个或多个工作产品经过质量评审并得到认可时，标志着一个里程碑的完成。

2. 进度安排

为了监控项目的进度计划和实际的进展情况，也表示各项任务之间进度的相互依赖关系，需要采用图示的方法。常用的方法有甘特图（Gantt Chart）和项目计划评审技术图（PERT），分别如图 5-12 和图 5-13 所示。

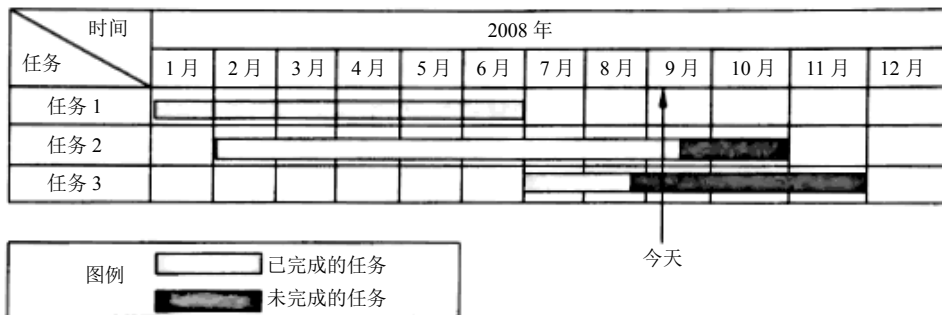


图 5-12 甘特图实例

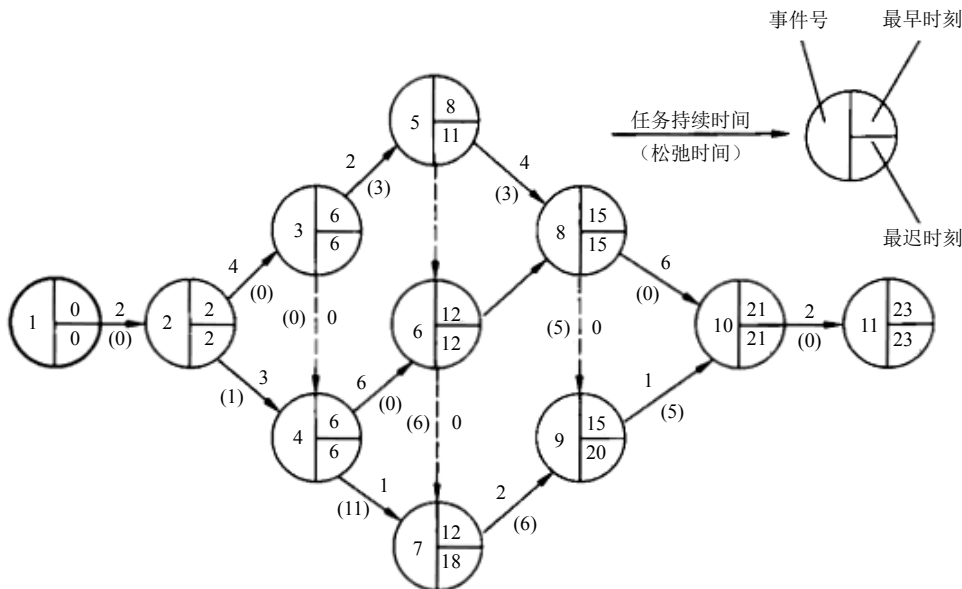


图 5-13 PERT 实例

5.6.4 软件项目组织

开发组织采用什么形式，不仅要考虑到项目的特点，还要考虑参与人员的素质。软件项目组织的原则有：①尽早落实责任；②减少交流接口；③责权均衡。

组织结构的模式，根据项目的分解和过程的分解，软件项目有以下三种组织形式：

- (1) 按项目划分的模式。将开发人员组织成项目组，项目组成员共同完成项目的任务，如项目的定义、需求分析、设计、编码、测试、评审等，甚至还包括项目的维护。
- (2) 按职能划分的模式。按软件过程中所反映的各种职能将项目参与者组织成相应的专业组，如开发组、测试组、质量保证组、维护组等。

(3) 矩阵模式：该模式是上述两种模式的组合，它既按职能来组织相应的专业组，又按项目来组织项目组。

5.6.5 软件质量管理

1. 软件质量特性

在 ISO/IEC 9126 中，软件质量模型由三个层次组成，第一层为质量特性，第二层为质量子特性，第三层为度量指标，如图 5-14 所示。

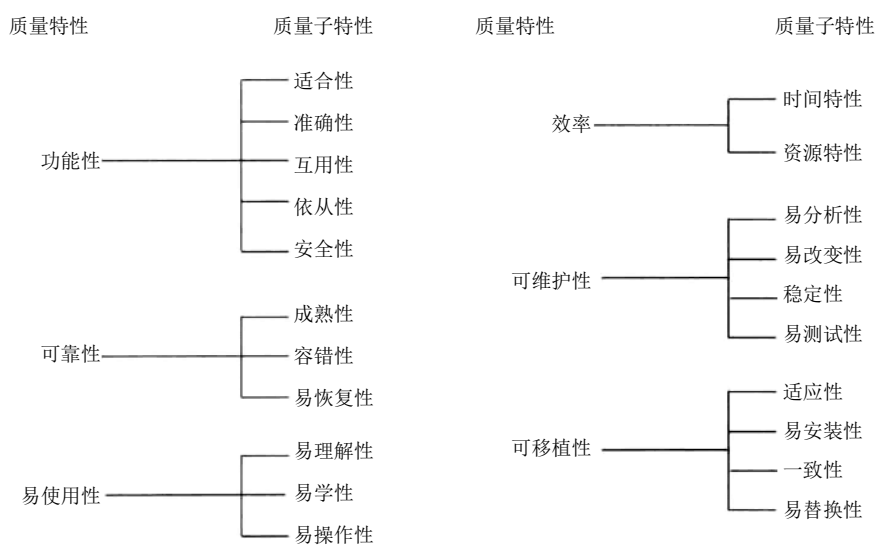


图 5-14 ISO/IEC 9126 软件质量模型

McCall 软件质量模型给出了一个三层模型框架，第一层为质量特性，第二层为评价准则，第三层为度量指标，如图 5-15 所示。

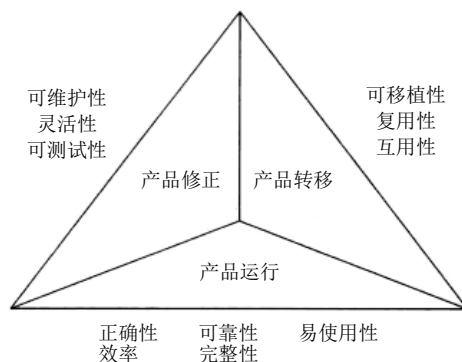


图 5-15 McCall 软件质量模型

2. 软件质量保证

在软件质量方面强调三个要点：①软件必须满足用户规定的需求；②软件应遵循规定标准所定义的一系列开发准则；③软件还应满足某些隐含的需求。

软件质量保证包括七个主要活动相关的各种任务：①应用技术方法；②进行正式的技术评审；③测试软件；④标准的实施；⑤控制变更；⑥度量；⑦记录、保存和报告。

3. 软件评审

软件评审的内容包括以下三个方面：

(1) 设计质量的评审。包括评审软件的规格说明书是否符合用户的要求；评审可靠性；评审保密措施的实施情况；评审操作特性的实施情况；评审性能；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；评审软件可测试性；评审软件可复用性。

(2) 程序质量的评审。从开发者的角度进行评审，与开发技术直接相关，着眼于软件本身的结构与运行环境的接口，以及变更带来的影响。

(3) 与运行环境的接口。运行环境包括硬件、其他软件 and 用户，主要检查项目与硬件的接口、与用户的接口。

4. 软件容错技术

提高软件质量和可靠性的技术大致分为两类：避开错误和容错技术。

实现容错的主要手段是冗余，常见的冗余技术有：

(1) 结构冗余：又分为静态冗余、动态冗余和混合冗余。

(2) 信息冗余：为检测或纠正正在运算或传输中的信息错误需外加的一部分信息。

(3) 时间冗余：以重复执行指令或程序来消除瞬时错误带来的影响。

(4) 冗余附加技术：为实现上述冗余技术所需要的资源和技术，包括程序、指令、数据、存放和调动它们的空间和通道等。

5.6.6 软件配置管理

软件配置管理（SCM）用于整个软件工程，主要目标是识别变更、控制变更、确保变更正确实现、报告有关变更。

1. 基线

基线即软件生存周期中各个开发阶段的一个特定点，作用是将各阶段的工作划分得更加正确。

2. 软件配置项

软件配置项（SCI）是配置管理的基本单位，对已经成为基线的 SCI，虽然可以修改，但必须按照一个特殊的、正式的过程进行评估，确认每一处修改。

3. 版本控制

版本控制实际上是一个动态的概念，一方面随着软件生存周期向前推进，SCI 的数量在不断增加，一些文档经过转换生成另一些文档，并产生一些信息；另一方面又随时会有新的变更出现，形成新的版本。

4. 变更控制

软件工程中的变更都会引起软件配置的变更，必须要对变更进行严格的控制和管理。配置数据库有：开发库、受控库、产品库。

5.6.7 软件风险管理

1. 软件风险

软件风险包括两个特性：不确定性和损失。不确定性是指风险可能发生也可能不发生；损失是指风险发生后会产生恶性后果。

常见的商业风险有：市场风险、策略风险、销售风险、管理风险、预算风险。

2. 风险识别

当识别出已知风险和可预测风险后，项目管理者首先要做的是尽可能回避这些风险，在必要时控制这些风险。风险因素可以定义为：性能风险、成本风险、支持风险、进度风险。

3. 风险预测

风险预测又称为风险估计，它试图从两个方面评估一个风险：风险发生的可能性或概率；风险发生后所产生的后果。

4. 风险控制

应对风险最好的办法是主动地避免风险，即在风险发生前分析引起风险的原因，然后采取措施避免风险的发生。

5.7 软件度量

软件度量用于对产品 & 开发产品的过程进行度量。软件产品、软件过程、资源都具有外部属性和内部属性。

1. 面向规模的度量

面向规模的度量主要是通过对质量和生产率的测量进行规范化得到的，而这些量都是根据开发过的软件的规模得到的。面向规模的度量公式如表 5-1 所示。

2. 面向功能的度量

面向功能的度量以功能测量数据为规范化值。应用最广泛的面向功能的度量是功能点（FP）。功能点是根据软件信息域的特性及复杂性来计算的。

表 5-1 面向规模的度量公式

度量	表示及含义
LOC 或 KLOC	代码行数或千行代码数
生产率 P	$P = LOC/E$, E 为开发的工作量（常用人数表示）
每行代码平均成本 C	$C = S/LOC$, S 为总成本

续表

度量	表示及含义
文档代码比 D	$D = Pe/KLOC$, Pe 为文档页数
代码错误率 EQR	$EQR = N/KLOC$, N 为代码中的错误数

3. 软件复杂性度量

软件复杂性度量是指理解 and 处理软件的难易程度。软件复杂性度量的参数有很多，主要包括：

①规模；②难度；③结构；④智能度。

软件复杂性包括程序复杂性和文档复杂性。

4. 程序复杂性度量

程序复杂性度量的原则包括：①程序理解的难度；②纠错、维护程序的难度；③向他人解释程序的难度；④根据设计文件编写程序的工作量；⑤执行程序时需要资源的程度。

5.8 练习题

1. 以下关于结构化开发方法的叙述中，不正确的是_____。

- 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

解析：

结构化分析与设计方法是一种面向数据流的开发方法，它以数据流为中心构建软件的分析模型和设计模型。结构化设计是将结构化分析的结构（数据流图）映射成软件的体系结构（结构图）。根据信息流的特点，可以将数据流图分为变换型数据流图和事务型数据流图，其对应的映射分别称为变换分析和事务分析。

在规模较大的实际系统中，其数据流图往往是变换型和事务型的混合结构，此时可以把变换分析和事务分析应用在同一数据流图的不同部分。

答案：D

2. 以下关于进度管理工具甘特图的叙述中，不正确的是_____。

- A. 能清晰地表达每个任务的开始时间、结束时间和持续时间
- B. 能清晰地表达任务之间的并行关系
- C. 不能清晰地确定任务之间的依赖关系
- D. 能清晰地确定影响进度的关键任务

解析：

甘特图内在思想简单，即以图示的方式通过活动列表和时间刻度形象地表示出任何特定项目的

活动顺序与持续时间。甘特图基本上是一条形图，横轴表示时间，纵轴表示活动（项目），条形表示在整个期间计划和实际活动的完成情况。它直观地表明任务计划在什么时候进行，以及实际进展与计划要求的对比。管理者由此可以便利地弄清一项任务（项目）还剩下哪些工作要做，并可以评估工作进度。

优点：能清晰地描述每个任务从何时开始、到何时结束及各个任务之间的并行性。

缺点：不能清晰地反映出每个任务之间的依赖关系，难以确定整个项目的关键所在，也不能反映计划中有潜力的部分。

答案：D

3. 项目复杂性、规模和结构的不确定性属于_____风险。

- A. 项目 B. 技术 C. 经济 D. 商业

解析：

项目风险是指可能导致项目损失的不确定性。美国项目管理大师马克思·怀德曼将其定义为某一事件发生给项目目标带来不利影响的可能性。

答案：A

4. 以下关于统一过程（UP）的叙述中，不正确的是_____。

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

解析：

统一过程（UP）的基本特征是以用例和风险驱动，以架构为中心，受控的迭代式的增量开发。

UP 定义了四个阶段：

起始阶段（Inception）：该阶段的主要目的是建立项目的范围和版本，确定项目目标的可行性和稳定性，提交结果包括需求和用例。

精化阶段（Elaboration）：该阶段的目的是对问题域进行分析，建立系统需求和架构，确定实现目标的可行性和稳定性，提交结果包括系统架构、问题领域、修改后的需求及项目开发计划等相关文档。

构建阶段（Construction）：增量式开发可以交付给用户的软件产品。

移交阶段（Transition）：该阶段的目的是将软件产品交付给用户。

答案：B

5. 以下关于文档的叙述中，不正确的是_____。

- A. 文档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，没有文档的软件不能称为软件
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

解析:

文档是软件开发使用和维护中的必备资料。文档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。文档不仅包括软件开发过程中产生的文档,还包括硬件采购和网络设计中形成的文档;不仅包括有一定格式要求的规范文档,也包括系统建设过程中的各种往来文件、会议纪要、会计单据等资料形成的不规范文档。

答案: A

6. 面向对象_____选择合适的面向对象程序设计语言,将程序组织为相互协作的对象集合,每个对象表示某个类的实例,类通过继承等关系进行组织。

- A. 分析 B. 设计 C. 程序设计 D. 测试

解析:

本题考查面向对象的知识。

面向对象程序设计(OOP)是一种实现方法,在这种方法中,程序被组织成许多相互协作的对象,每个对象代表某个类的一个实例,类则属于一个通过继承关系形成的层次结构。

面向对象分析(OOA)是一种分析方法,利用从问题域的词汇表中找到的类和对象来分析需求,重点是找到和描述问题域的对象或概念,然后构建真实世界的模型,利用面向对象的观点来看世界。

面向对象设计(OOD)是一种设计方法,包括面向对象分解的过程和一种表示法,这种表示法用于展现被设计系统的逻辑模型(类和对象结构)、物理模型(模块和处理架构)、静态模型和动态模型。

答案: C

7. 采用面向对象的方法进行软件开发,在分析阶段,架构师主要关注系统的_____。

- A. 技术 B. 部署 C. 实现 D. 行为

解析:

采用面向对象的方法进行软件开发,在分析阶段,架构师主要关注系统的行为,即系统应该做什么。

答案: D

8. 某企业财务系统的需求中,属于功能需求的是_____。

- 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

解析:

功能需求即软件必须完成哪些事,必须实现哪些功能。选项 B、C、D 属于性能需求。

答案: A

9. 配置管理贯穿软件开发的整个过程。以下内容中, 不属于配置管理的是_____。

- A. 版本控制 B. 风险管理 C. 变更管理 D. 配置状态报告

解析:

配置管理是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程, 确保软件开发者在软件生命周期中的各个阶段都能得到精确的产品配置。

配置管理的活动包括制定配置管理计划、配置库管理、配置控制、配置状态报告、配置审计、发布管理和交付。

风险管理是与配置管理并列的项目管理过程。

答案: B

10. 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。

- A. 改正性 B. 适应性 C. 改善性 D. 预防性

解析:

软件的维护内容主要有以下四个方面:

(1) 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) 适应性维护是指使应用软件适应信息技术变化和管理需求变化而进行的修改。

(3) 完善性维护是指为了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) 预防性维护是指为了改进应用软件的可靠性和可维护性, 为了适应未来的软件、硬件环境的变化, 应主动增加预防性的新功能, 以使应用系统可以适应各类变化而不被淘汰。

答案: C

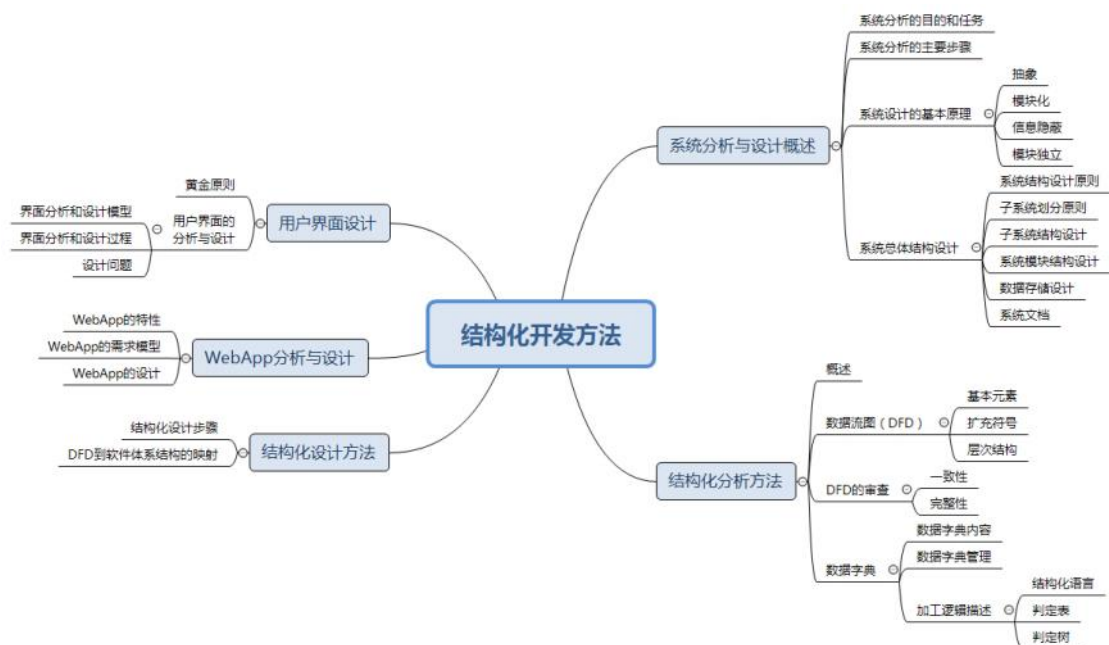
第6小时

结构化开发方法

6.0 章节考点分析

根据考试大纲，本小时为重点考核内容，需要掌握结构化分析方法、结构化设计方法，上午的客观题约占2分，下午的案例题至少一题。

本小时的内容偏重于基本概念的理解和分析，本小时的架构如图6-1所示。



【导读小贴士】

结构化方法包括结构化分析（SA）、结构化设计（SD）、结构化程序设计（SPD），它是一种面向数据流的开发方法。自顶向下、逐层分解是结构化方法的指导思想，功能的分解与抽象是结构化方法的基本原则。

6.1 系统分析与设计概述

1. 系统分析的目的和任务

系统分析报告（系统方案说明书）。

2. 系统分析的主要步骤

如图 6-2 所示，可以将系统分析阶段的主要工作分为以下五步：

- （1）对当前系统进行详细的调查，收集数据。
- （2）建立当前系统的逻辑模型。
- （3）对现状进行分析，提出改进意见和新系统应达到的目标。

- (4) 建立新系统的逻辑模型。
- (5) 编写系统方案说明书。

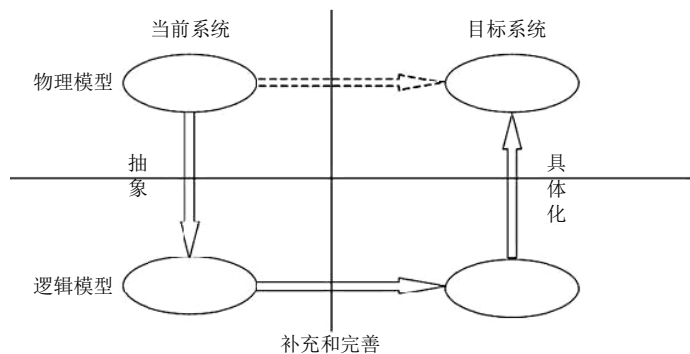


图 6-2 系统分析的过程

3. 系统设计的基本原理

- (1) 抽象。
- (2) 模块化。
- (3) 信息隐蔽。
- (4) 模块独立：低耦合、高内聚。系统的耦合类型如表 6-1 所示，内聚类型如表 6-2 所示。

表 6-1 系统的耦合类型

耦合类型	描述
无直接耦合	两模块无直接关系，分属不同模块的控制与调用，独立性最高
数据耦合	模块之间通过值传递完成调用关系
标记耦合	模块之间传递的是数据结构
控制耦合	模块之间传递的是控制变量
外部耦合	模块之间通过外部环境联结
公共耦合	访问同一个公共数据环境（如全局数据结构、共享通信、公共内存）
内容耦合	直接使用另一个模块的内部数据，或者非正常入口转入另一个模块

表 6-2 系统的内聚类型

内聚类型	描述
功能内聚	完成单一功能，各部分协同工作，缺一不可，是最强的内聚
顺序内聚	模块内的处理元素都密切相关且按顺序执行
通信内聚	模块内的所有处理元素集中在一个数据结构的区域上
过程内聚	模块内按指定的过程完成多个任务

续表

内聚类型	描述
时间内聚	模块内的组合动作需要同时执行
逻辑内聚	模块内通过参数确定完成哪一个逻辑上相似的功能
偶然内聚	模块内的处理元素之间没有任何联系，是最弱的内聚

4. 系统结构设计原则
- (1) 分解—协调。

(2) 自顶向下。

(3) 信息隐蔽、抽象。

(4) 一致性原则。

(5) 明确性原则。

(6) 模块间松耦合、模块内高内聚。

(7) 模块的扇入系数和扇出系数要合理。

(8) 模块规模适当。
5. 子系统划分原则
- (1) 子系统相对独立。

(2) 子系统间的依赖小。

(3) 数据冗余小。

(4) 考虑扩展性。

(5) 便于系统分阶段实现。

(6) 考虑到各资源的利用情况。
6. 子系统结构设计
- (1) 各个子系统划分成多个模块。

(2) 子系统、模块之前的数据及调用关系。

(3) 评价并改进模块结构的质量。

(4) 从数据流图导出模块结构图。
7. 系统模块结构设计
- (1) 模块四要素：输入输出、处理功能、内部数据、程序代码。

(2) 模块结构图。
8. 数据存储设计
- (1) 数据资源分布。

(2) 数据安全保密：八个等级（0~7 级），四种方式（只读、只写、删除、修改）。
9. 系统文档

1. 结构化分析 (SA) 概述
抽象 (自底向上)、分解 (自顶向下)。
2. 数据流图 (DFD)
DFD 的常见错误如图 6-3 所示。



- (1) DFD 的基本图形元素：外部实体、加工、数据存储、数据流。
 - (2) 扩充符号：*（与）、+（或）、 $\oplus$ （互斥）。
 - (3) 层次结构：顶层图、0 层图、底层图、父图、子图。
- ### 3. DFD 的审查
- (1) 一致性：父图与子图平衡、数据守恒、具备数据存储、输出不能与输入同名。
 - (2) 完整性：奇迹（无入有出）、黑洞（有入无出）、灰洞（无法出）。
- ### 4. 数据字典（DD）
- 数据流、数据项、数据存储、基本加工。

1. 结构化设计

结构化设计（SD）是一种面向数据流的设计方法，与 SA 衔接，基本思想是将系统设计成相互独立、功能单一的模块组成的结构。

2. 结构化设计步骤

- (1) 建立初始结构图。
- (2) 对结构图进行改进。
- (3) 书写设计文档。
- (4) 设计评审。

3. 数据流图到软件体系结构的映射

- (1) 信息流的类型：交换流（主加工）、事物流。
- (2) 变换分析。

6.4 WebApp 分析与设计

1. WebApp 的特性

网络密集型、并发性、负载量无法预知、性能、可用性、数据驱动。

2. WebApp 的需求模型

内容模型、交互模型、功能模型、导航模型、配置模型。

3. WebApp 的设计

架构设计、构件设计、内容设计、导航设计。

6.5 用户界面设计

1. Theo Mandel 黄金原则

- (1) 用户操纵控制：不强迫、交互灵活、中断和撤销、定制、内部隔离、直接交互。
- (2) 减少用户的记忆负担。
- (3) 保持界面一致。

2. UI 分析与设计

- (1) UI 分析和设计模型：设计模型、用户模型、系统感觉、系统映像。
- (2) UI 分析和设计过程：界面分析及建模、界面设计、界面构造、界面确认。
- (3) UI 设计问题：系统响应、帮助设施、错误信息处理、菜单和命令标记。

6.6 练习题

1. 以下关于数据流图中基本加工的叙述，不正确的是_____。
 - 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

解析:

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。

答案: C

2. 在规划分模块时, 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。

- 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, 使该判定处于较高层次

解析:

一个模块的作用范围(又称影响范围)是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是指模块本身及其所有下属模块(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, 而且判定所在的模块应和其影响的模块在层次上尽量靠近。如果在设计过程中, 发现模块的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。

答案: D

第二篇

进阶篇

第7小时

面向对象技术

7.0 章节考点分析

第7小时主要学习面向对象、UML、设计模式等内容。

根据考试大纲，本小时的知识点会涉及单选题型和案例分析题型，上午的客观题占11分，下午的案例题至少一题。本小时的内容偏重于概念知识，根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，考查概念的知识点多数参照教材，扩展内容较少。本小时的架构如图7-1所示。

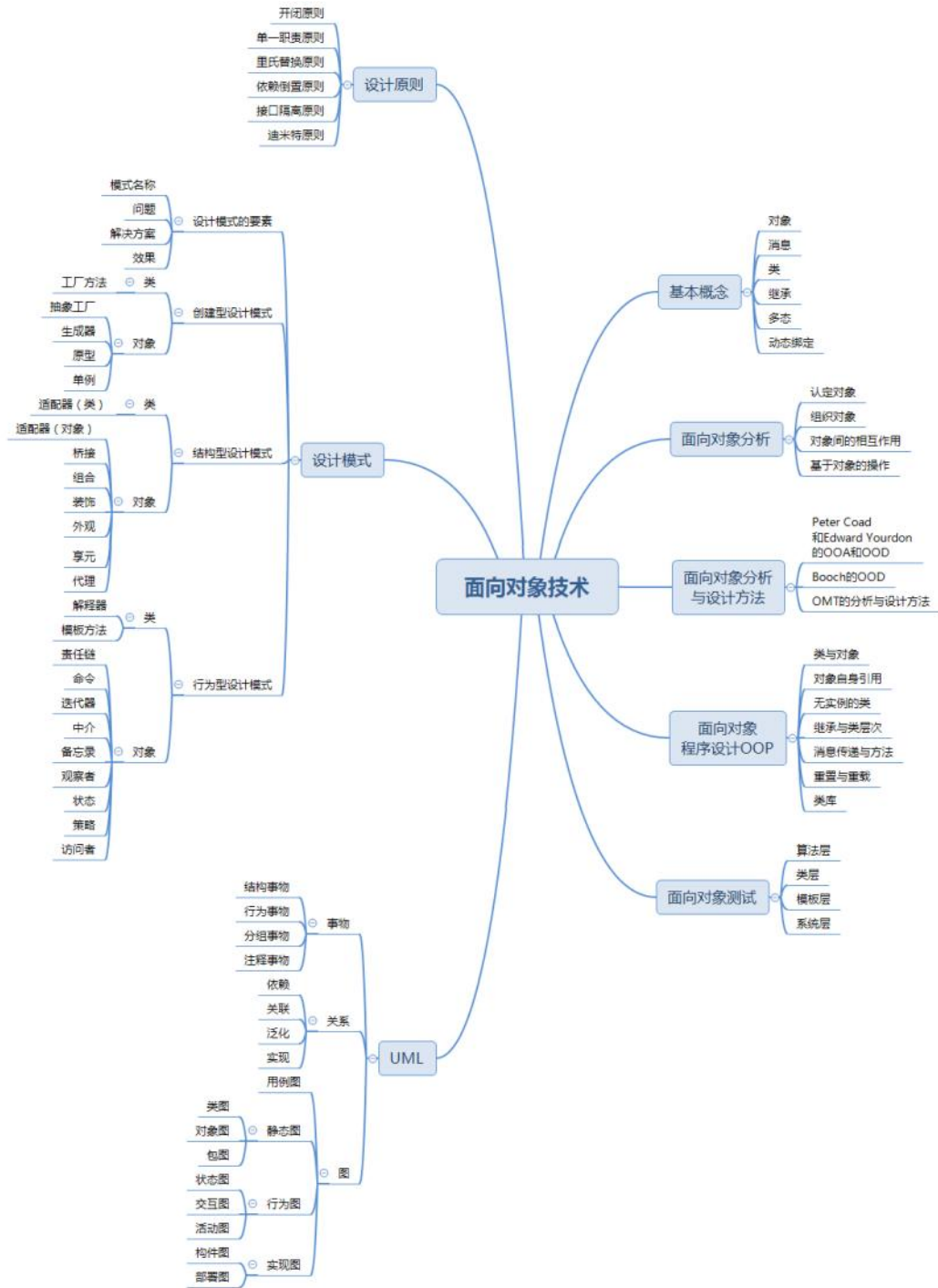


图 7-1 第 7 小时架构图



【导读小贴士】

对于软件设计师而言，面向对象技术是最基础的知识点，建议除了理解的基础上掌握核心知识点外，还需要灵活应用。

7.1 面向对象基础

7.1.1 面向对象基础的概念

面向对象 = 对象(object)+ 类(classification)+ 继承(inheritance)+ 消息通信(communication with messages)。

1. 对象

在面向对象的系统中，对象是基本运行时的实体，即包括数据（属性），也包括作用于数据的操作（行为）。

一个对象把属性和行为封装为一个整体。封装是一种信息隐藏技术，目的是使对象的使用者和生产者分离，使对象的定义和实现分开。从程序设计者来看，对象是一个程序模块；从用户来看，对象为他们提供了所希望的行为。

对象内进行的操作通常称为方法。一个对象一般由对象名（标识）、属性和方法三个部分组成。

2. 消息

对象之间进行通信的一种构造称为消息。当一个消息发送给某个对象时，包含要求接收对象去执行某些活动的信息，接收到信息的对象经过解释，然后予以响应，这种通信机制称为消息传递。发送消息的对象不需要知道接收消息的对象如何对请求予以响应。

3. 类

类是在对象上的抽象，对象是类的具体化，是类的实例(instance)。在分析和设计时，通常把注意力集中在类上，而不是具体的对象。也不必逐个定义每个对象，只需对类作出定义，而对类的属性进行不同的赋值即可得到该类的对象实例。

4. 继承

继承是父类与子类之间共享数据和方法的机制。这是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上进行，并把这个类所定义的内容作为自己的内容，并加入若干新的内容。父类 A 与它的子类 B 的继承关系如图 7-2 所示。

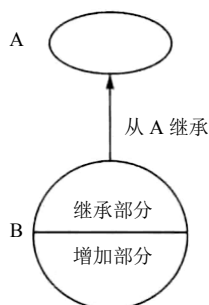


图 7-2 类的继承关系示意图

5. 多态

在收到消息时，对象要予以响应。不同的对象收到同一信息可以产生不同的结果，这个现象称为多态（polymorphism）。Cardelli 和 Wegner 将其分为通用的（参数的、包含的）和特定的（过载的、强制的）。

6. 动态绑定

绑定是一个把过程调用和响应调用所需要执行的代码加以结合的过程。在一般的程序设计语言中，绑定是在编译时进行的，称为静态绑定。动态绑定则是在运行时进化的，因此，一个给定的过程调用和代码的结合直到调用发生时才进行。

7.1.2 面向对象分析

面向对象分析（OOA）的目的是获得对应用问题的理解。理解的目的是确定系统的功能和性能要求。面向对象分析技术可以将系统的行为和信息间的关系表示为迭代构造特征。面向对象分析包括五个活动：①认定对象；②组织对象；③对象间的相互作用；④定义对象的操作；⑤定义对象的内部信息。

7.1.3 面向对象设计

面向对象设计（OOD）的含义是设计分析模型和实现相应的源代码，在目标代码环境中，这种源代码可以被执行。通常情况下，由概念模型生成的分析模型被装入到相应的执行环境中时，还需要被修改。

对象标识期间的目标是分析对象，设计过程也是发现对象的过程，称为再处理。

面向对象是一种程序设计风格，不只是一种具有构造继承性、封装性和多态的程序设计语言族的命名。

7.1.4 面向对象程序设计

面向对象程序设计（OOP）的实质是选用一种面向对象程序设计语言（OOPPL），采用对象、类及其相关概念所进行的程序设计。它的关键在于加入了类和继承性，从而进一步提高了抽象程度。

1. 类

通常在介绍 OOP 的书籍中总是先引入对象的概念，然后从对对象进行抽象的角度来引入类的概念。但当设计和实现一个面向对象的程序时，首先接触到的不是对象，而是类和类层次结构。

2. 继承和类层次结构

孤立的类只能描述实体集合的特征一致性，而客观世界中实体集合的划分通常还要考虑实体特征方面有关联的相似性。在 OOP 中使用继承机制解决这一问题；在 OOPL 中，继承一般通过定义类之间的关系来体现。

在面向对象系统中，子类和父类之间的继承关系构成了这个系统的类层次结构，可以用树（对应于单继承）或格（对应于多继承）这样的图来描述。

与一般数据类型的实例化过程相比，类的实例化过程是一种实例的合成过程，而不仅仅是根据单个类型进行的空间分配、初始化和绑定。

3. 对象、消息传递和方法

对象是类的实例。对象的表示在形式上与一般数据类型十分相似，但是它们之间存在着本质区别：对象之间通过消息传递的方式进行通信。

信息传递源是一种与通信有关的概念，OOP 使得对象具有交互能力的主要模型就是消息传递模型。对象被看成用传递消息的方式互相联系的通信实体，它们既可以接收，也可以拒绝外界发来的消息。一般情况下，对象接收它能识别的消息，拒绝它识别不了的消息。

4. 对象自身引用

对象自身引用（self-reference）是 OOPL 中的一种特有结构。这种结构在不同的 OOPL 中有不同的名称，在 C++ 和 Java 中称为 this，在 Smalltalk-80、Object-C 和其他一些 OOPL 中则称为 self。

5. 重载

重载或覆盖（overriding）是在子类中重新定义父类中已经定义的方法，其基本思想是通过一种动态绑定机制的支持，使子类在继承父类接口定义的前提下用适合自己要求的实现去置换父类中的相应实现。

6. 类属

类属是程序设计语言中普遍注重的一种参数多态机制，在 OOPL 中也不例外。值得注意的是，在 C++ 语言中，类属有专门的术语 template。

7. 无实例的类

前面我们讲到，类是对象的模板，对象是类的实例。那么，是否每个类都至少有一个实例？如果在类之间没有定义继承关系，回答是肯定的。这是因为若存在没有实例的类，那么这样的类对程序的行为没有任何贡献，因而是冗余的；相反，如果存在继承关系，那么的确有可能在类层次结构的较高层次上看到始终没有实例的类。

7.1.5 面向对象测试

面向对象测试可以分为四个层次：

(1) 算法层。用于测试类中定义的每个方法，基本上相当于传统软件测试中的单元测试。

(2) 类层。用于测试封装在同一个类中的所有方法与属性之间的相互作用。在面向对象软件中，类是基本模块，因此可以认为这是面向对象测试中所特有的模块测试。

(3) 模板层。用于测试一组协同工作的类之间的相互作用。大体上相当于传统软件测试中的集成测试，但是也有面向对象软件的特点，如对象之间通过发送消息相互作用。

(4) 系统层。把各个子系统组装成完整的面向对象软件系统，在组装过程中同时进行测试。

软件工程中传统的测试用例设计技术，如逻辑覆盖、等价类划分和边界值分析等方法，仍然可以作为测试类中每个方法的主要技术。面向对象测试的主要目标也是用尽可能低的测试成本和尽可能少的测试用例，发现尽可能多的错误。但面向对象程序中特有的封装、继承和多态等机制，也给面向对象测试带来一些新的特点，增加了测试和调试的难度。

7.2 UML

7.2.1 事物

UML 中有四种事物，如图 7-3 所示。

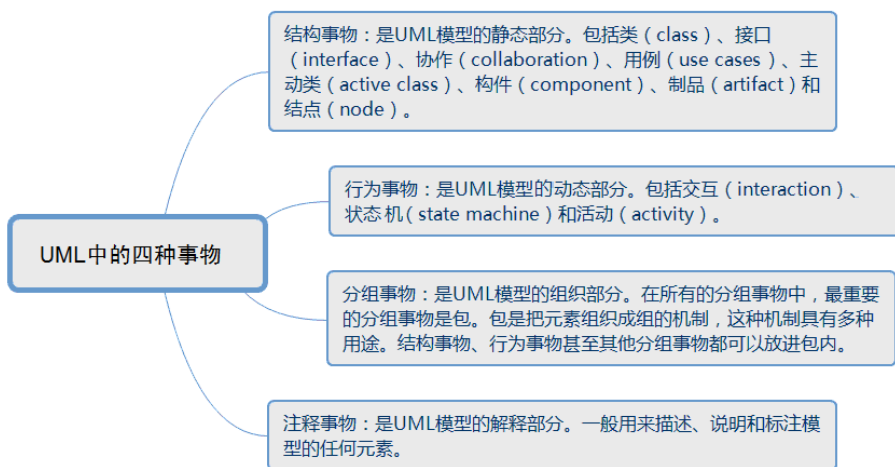


图 7-3 UML 中的四种事物

7.2.2 关系

UML 中有四种关系：依赖、关联、泛化和实现，如图 7-4 所示。

这四种关系是 UML 模型中可以包含的基本关系事物。它们也有变体，例如，依赖的变体有精华、跟踪、包含和延伸。聚焦和组合的关系示意图如图 7-5 所示。

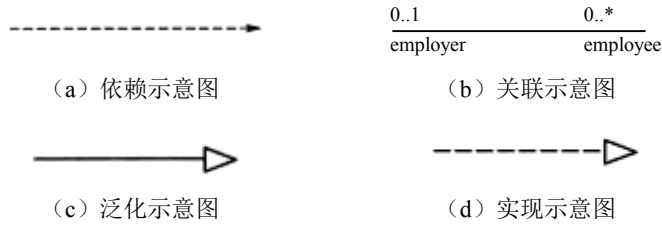


图 7-4 UML 中的四种关系



图 7-5 聚焦和组合关系

7.2.3 UML 中的图

1. 类

类图展现了一组对象、接口、协作和它们之间的关系。类图中包含的内容如图 7-6 所示。

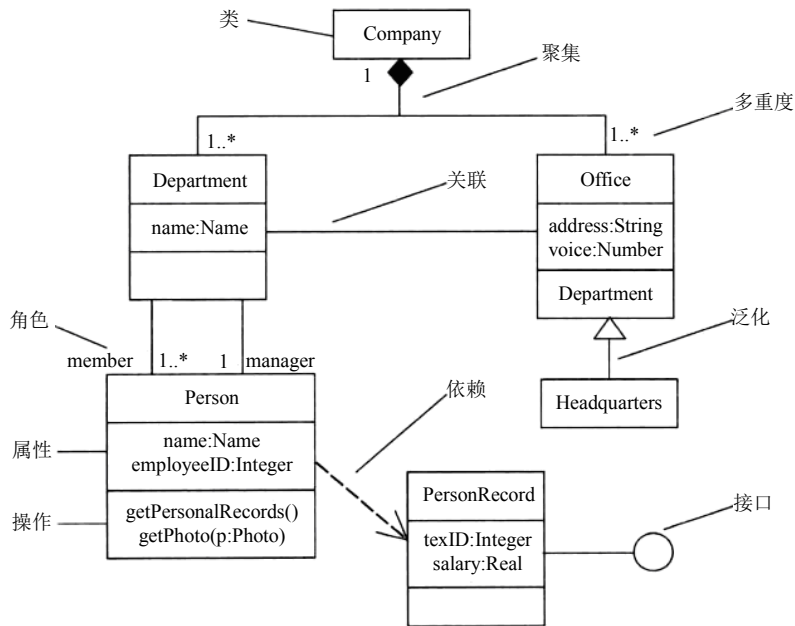


图 7-6 类图

2. 对象图

对象图展现了某一时刻一组对象和它们之间的关系。对象图一般包括对象和链，如图 7-7 所示。

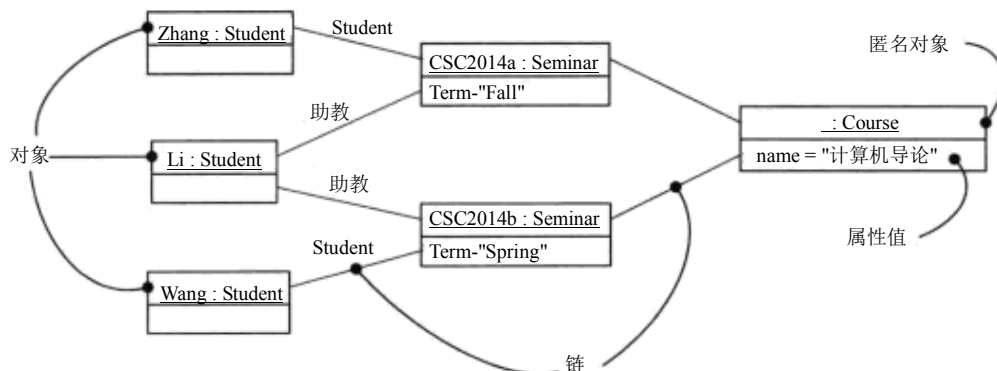


图 7-7 对象图

3. 用例图

用例图展现了一组用例、参与者和它们之间的关系，如图 7-8 所示。

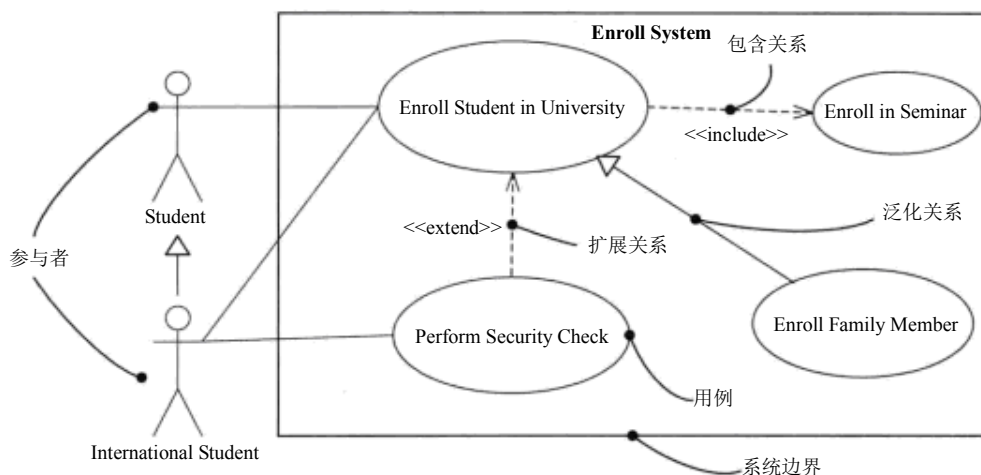


图 7-8 用例图

4. 交互图

交互图用于对系统的动态方面进行建模。一张交互图表现的是一个交互程序由一组对象和它们之间的关系组成，包含它们之间可能传递的消息。

交互图可以分为序列图、通信图、交互概览图和时序图。

(1) 序列图。序列图是场景的图形化表示，描述了以时间顺序组织的对象之间的交互活动，强调的是信息时间顺序的交互图。有两个不同于通信图的特征：序列图有对象生命线；序列图有控制焦点。序列图如图 7-9 所示。

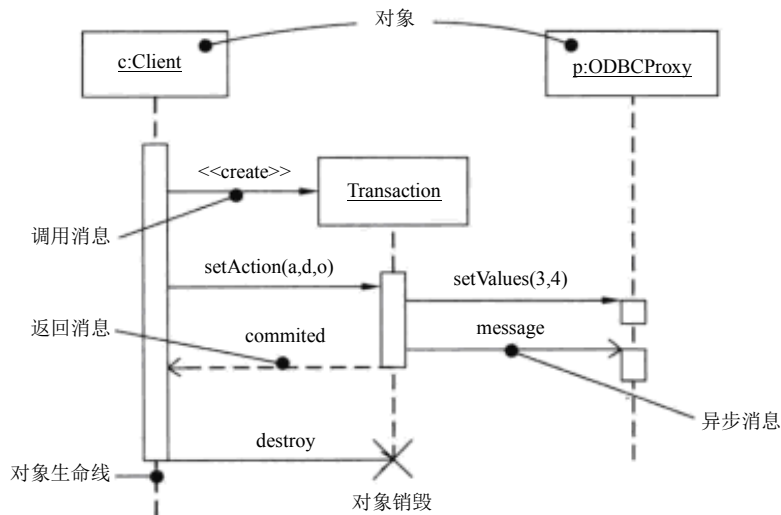


图 7-9 序列图

(2) 通信图。通信图强调收发信息的对象的结构组织，在早期版本中也被称为协作图，强调参加交互的对象的结构组织。通信图有两个不同于序列图的特征：通信图有路径；通信图有顺序号。通信图如图 7-10 所示。

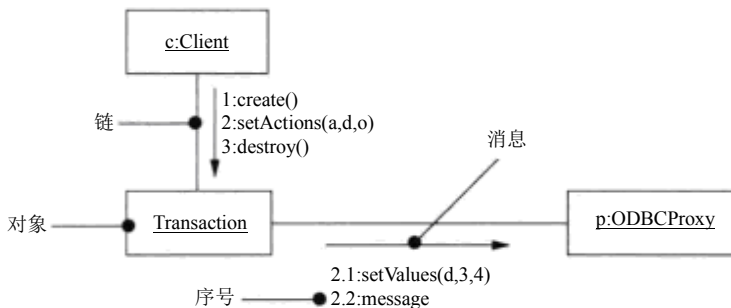


图 7-10 通信图

(3) 交互概览图。交互概览图是 UML 2.0 新增的交互图之一，它是活动图的变体，描述业务过程中的控制流概览、软件过程中的详细逻辑概览及多个图进行拼接，抽象掉信息和生命线。它使用了活动图的表示法，如图 7-11 所示。

(4) 时序图。另一种新增的、特别适合实时嵌入式系统建模的交互图称为时序图。其关注沿着线性时间轴、生命线内部和生命线之间的条件改变。它描述对象的状态随着时间改变的情况，很像示波器，如图 7-12 所示。

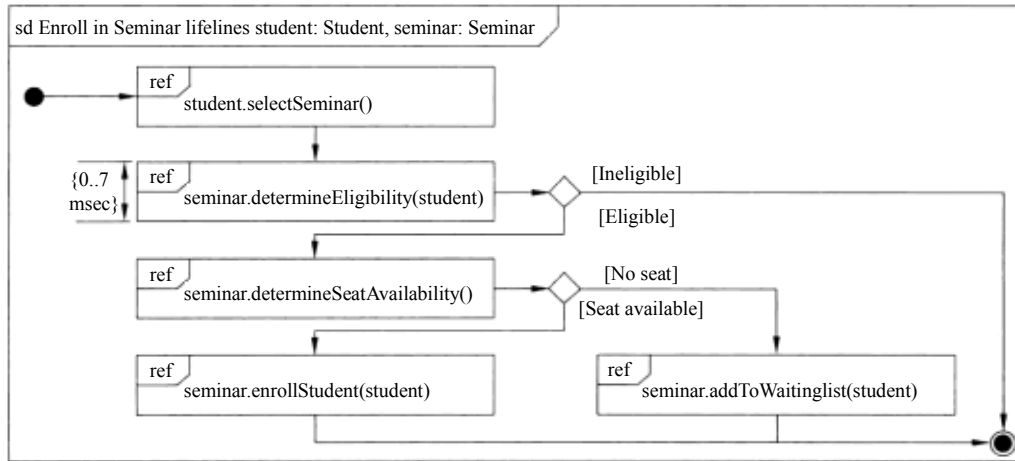


图 7-11 交互概览图

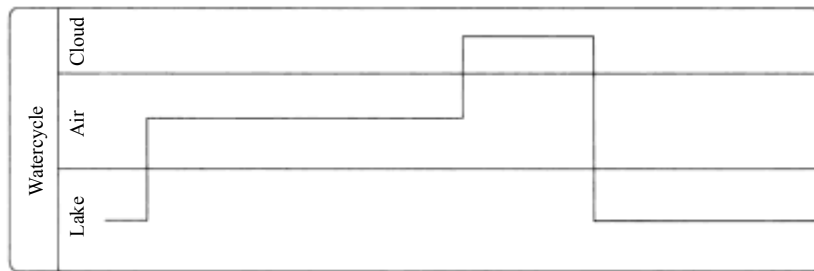


图 7-12 时序图

5. 状态图

状态图通常包括简单状态和组合状态、转换（事件和动作），如图 7-13 所示。

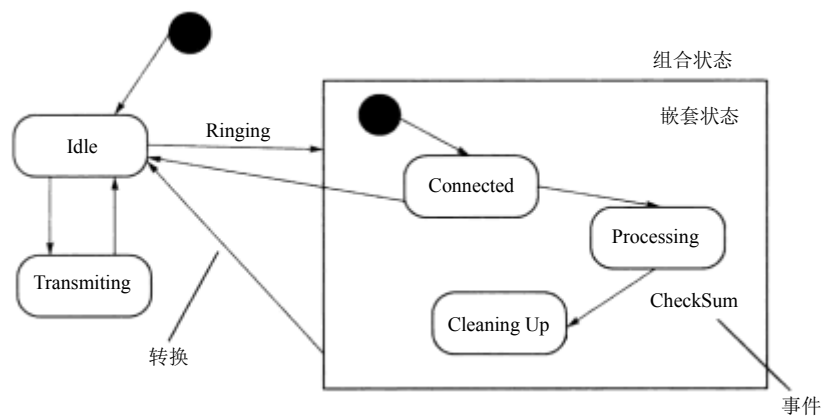


图 7-13 状态图

6. 活动图

活动图专注系统的动态视图，它对于系统的功能建模特别重要，并强调对象间的控制流程。活动图一般包括活动状态和动作状态、转换和对象，如图 7-14 所示。

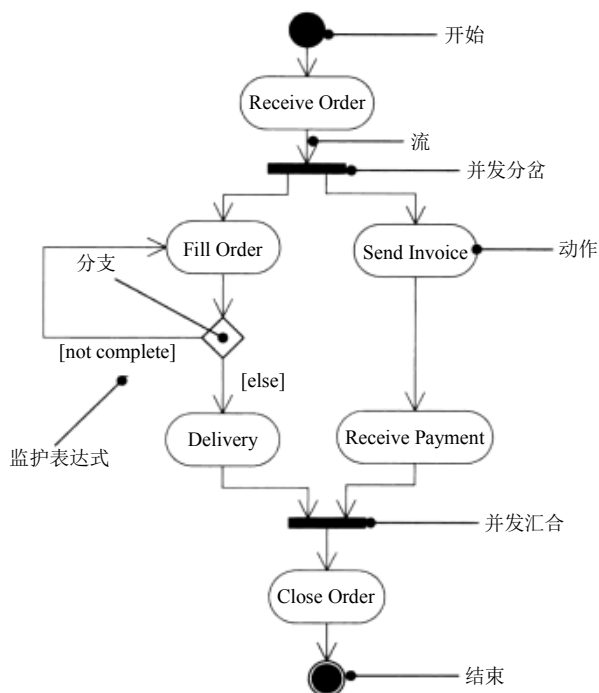


图 7-14 活动图

7. 构件图

构件图展现了一组构件之间的组织和依赖关系，专注于系统静态实现视图，如图 7-15 所示。

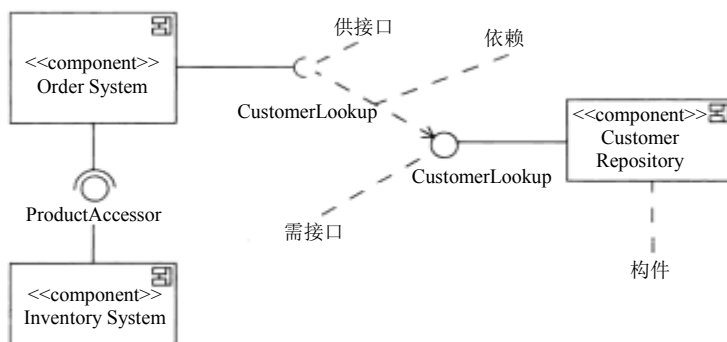


图 7-15 构件图

8. 组合结构图

组合结构图用于描述一个分类器（如类、构件等）的内部结构，分类器与系统中其他组成部分之间的交互端口，展示一组相互协作的实例如何完成特定的任务，描述设计、架构模式或策略。

9. 部署图

部署图是用来对面向对象系统的物理方面建模的方法，展现了运行时处理结点及其中构件（制品）的配置。一般对系统的静态部署视图进行建模，与构件图相关，如图 7-16 所示。

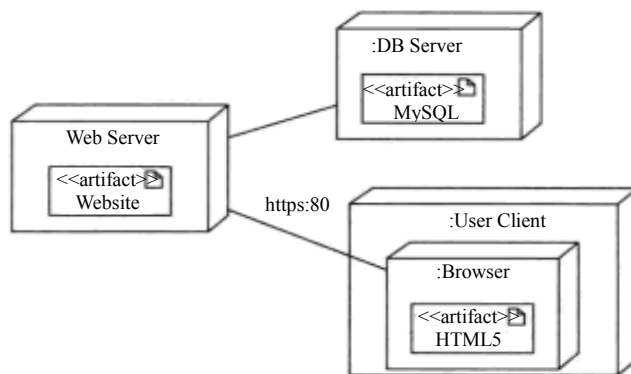


图 7-16 部署图

10. 包图

包图用于把模型本身组织成层次结构的通用机制，不能执行，展现由模型本身分解而成的组织单元及之间的依赖关系，如图 7-17 所示。

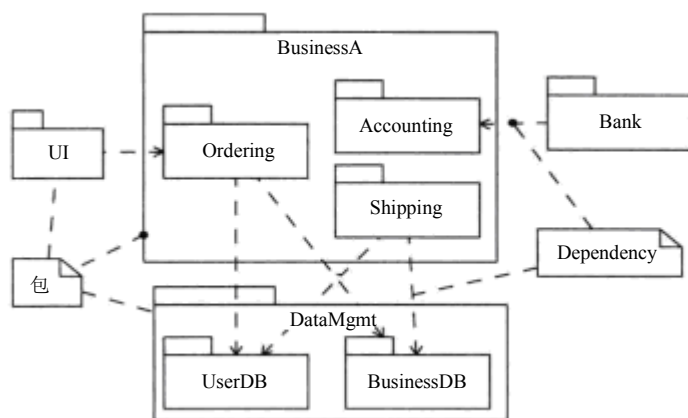


图 7-17 包图

7.3 设计模式

7.3.1 设计模式的要素

设计模式的核心在于提供了相关问题的解决方案,使人们可以更加简单方便地复用成功的设计和体系结构。

设计模式的四个基本要素为:①模式名称;②问题;③解决方案;④效果。

7.3.2 创建型设计模式

创建型设计模式抽象了实例化过程,它们帮助一个系统独立于如何创建、组合和表示它的那些对象。一个类的创建型模式使用继承改变被实例化的类,而一个对象创建型模式将实例化委托给另一个对象。

创建型设计模式的基本要素为:①抽象工厂 (Abstract Factory);②生成器 (Builder);③工厂方法 (Factory Method);④原型 (Prototype);⑤单例 (Singleton)。

7.3.3 结构型设计模式

结构型设计模式的基本要素为:①适配器 (Adapter);②桥接 (Bridge);③组合 (Composite);④装饰 (Decorator);⑤外观 (Facade);⑥享元 (Flyweight);⑦代理 (Proxy)。

7.3.4 行为设计模式

行为设计模式涉及算法和对象间职责的分配。行为设计模式不仅描述对象或类的模式,还描述了它们之间的通信模式。这些模式刻画了在运行时难以跟踪的、复杂的控制流。它们将用户的注意力从控制流转移到对象间的联系方式上来。

行为设计模式的基本要素为:①责任链 (Chain of Responsibility);②命令 (Command);③解释器 (Interpreter);④迭代器 (Iterator);⑤中介者 (Mediator);⑥备忘录 (Memento);⑦观察者 (Observer);⑧状态 (State);⑨策略 (Strategy);⑩模板方法 (Template Method);⑪访问者 (Visitor)。

7.4 练习题

1. 面向对象技术中,组合关系表示_____。

- A. 包与其模型元素的关系
- B. 用例之间的一种关系
- C. 类与其对象的关系
- D. 整体与其部分之间的一种关系

解析:

在面向对象技术中,组合关系表示整体与其部分之间的一种关系。包用于将关系紧密的模型元

素组织在一起，提供一个命名空间，以提供访问控制。用例之间有包含（include）、扩展（extend）和泛化（generalization）三种相关性的关系。类是在对象之上的抽象，对象是类的具体化，对定义好的类的属性进行不同的赋值就可以得到该类的对象实例。

答案：D

2. 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能进行，这种绑定称为_____。

- A. 静态绑定 B. 动态绑定 C. 过载绑定 D. 强制绑定

解析：

动态绑定（Dynamic Binding）是面向对象程序设计语言中的一种机制，绑定是在运行时进行的，即一个给定的过程调用和响应调用需执行的代码的结合直到调用发生时才进行。这种机制实现了方法的定义与具体的对象无关，而对方法的调用则可以关联到具体的对象。

答案：B

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

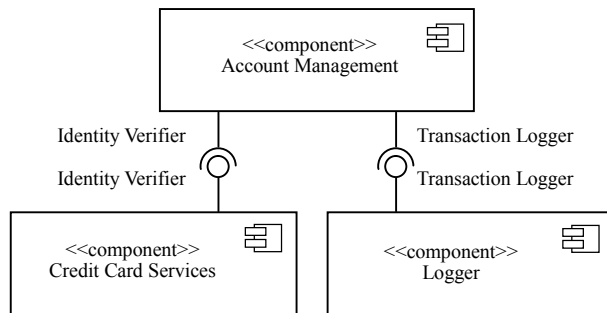
- A. 精化 B. 设计 C. 重构 D. 抽象

解析：

本题考查采用敏捷方法进行软件开发。在敏捷方法中，重构是一种重新组织技术，重新审视需求和设计，重新明确地描述它们以符合新的和现有的需求，可以简化构件的设计而无须改变其功能或行为。

答案：C

4. 下图属于 UML 中的__(a)___，其中，Account Management 需要__(b)___。



- (a) A. 组件图 B. 部署图 C. 类图 D. 对象图

- (b) A. 实现 Identity Verifier 接口并被 Credit Card Services 调用
B. 调用 Credit Card Services 实现的 Identity Verifier 接口
C. 实现 Identity Verifier 接口并被 Logger 调用
D. 调用 Logger 实现的 Identity Verifier 接口

解析:

本题考查 UML 语言基础知识。

UML 语言是标准的建模语言,通过图形化的方式展现系统的模型。本题是 UML 中的组件图,其表示的含义是组件 Account Management 需要调用 Credit Card Services 组件和 Logger 组件分别实现的 Identity Verifier 接口和 Transaction Logger 接口。

答案: A、B

5. 当不适合采用生成子类的方法对已有的类进行扩充时,可以采用 (a) 设计模式动态地给一个对象添加一些额外的职责;当应用程序由于使用大量的对象造成很大的存储开销时,可以采用 (b) 设计模式运用共享技术来有效地支持大量细粒度的对象;当想使用一个已经存在的类,但其接口不符合需求时,可以采用 (c) 设计模式将该类的接口转换成想要的接口。

- | | |
|---------------------|-------------------|
| (a) A. 命令 (Command) | B. 适配器 (Adapter) |
| C. 装饰 (Decorator) | D. 享元 (Flyweight) |
| (b) A. 命令 (Command) | B. 适配器 (Adapter) |
| C. 装饰 (Decorator) | D. 享元 (Flyweight) |
| (c) A. 命令 (Command) | B. 适配器 (Adapter) |
| C. 装饰 (Decorator) | D. 享元 (Flyweight) |

解析:

装饰模式主要的目的是在无法生成子类的情况下,给一个对象动态地增加新的职责;享元设计模式是共享大量细粒度的对象;适配器设计模式则是将已有的接口转换为系统需要的接口形式。

答案: C、D、B

6. 下面关于面向对象分析与面向对象设计的说法中,不正确的是_____。

- A. 面向对象分析侧重于理解问题
- B. 面向对象设计侧重于理解解决方案
- C. 面向对象分析描述软件要做什么
- D. 面向对象设计一般不关注技术和实现层面的细节

解析:

本题考查面向对象分析与设计的基本概念。

面向对象分析主要强调理解问题是什么,不考虑问题的解决方案,因此 A、C 选项是正确的;面向对象设计侧重于问题的解决方案,并且需要考虑实现中的细节问题,因此 D 选项的说法是不正确的。

答案: D

7. 采用面向对象开发方法时,对象是系统运行时的基本实体。以下关于对象的叙述中,正确的是_____。

- 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

解析:

本题考查面向对象的基本知识。采用面向对象开发方法时,对象是系统运行时的基本实体。它既包括数据(属性),也包括作用于数据的操作(行为)。一个对象通常可由对象名、属性和操作三个部分组成。

答案: D

8. 一个类是 (a)。在定义类时,将属性声明为 `private` 的目的是 (b)。

- | | |
|-----------------------|-------------------|
| (a) A. 一组对象的封装 | B. 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 |
| (b) 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 |

解析:

本题考查面向对象的基本知识。在面向对象技术中,将一组大体上相似的对象定义为一个类。把一组对象的共同特征加以抽象并存储在一个类中,是面向对象技术中的一个重要特点。一个所包含的方法和数据描述一组对象的共同行为和属性。在类定义时,属性声明 `private` 的目的是实现数据隐藏,以免意外更改。

答案: D、A

第8小时

算法设计与分析

8.0 章节考点分析

根据考试大纲，本小时的知识点会涉及考试中的各种题型，约占4分。本小时的内容偏重于理论，根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，考查题型虽不多，但需要理解各种算法的基本思想、实例及各算法的区别和应用场景，考题常与数据结构相结合。本小时的架构如图8-1所示。

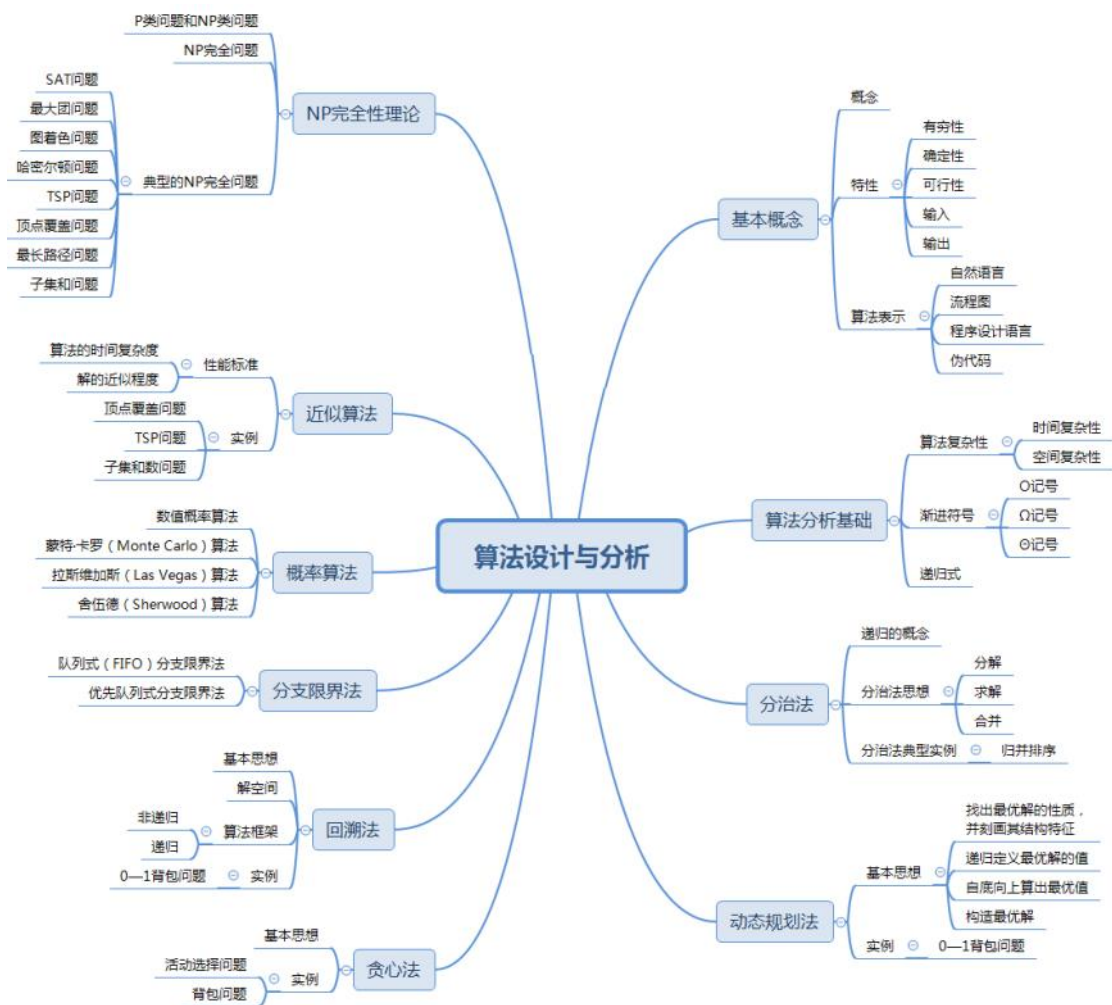


图 8-1 第 8 小时架构图



【导读小贴士】

算法被公认为是计算机科学的基石，算法理论研究的是算法的设计技术和分析技术。

8.1 算法的基本概念

1. 基本概念

算法 (Algorithm) 是对特定问题求解步骤的一种描述, 它是指令的有限序列, 其中每一条指令表示一个或多个操作。

2. 算法的特性

有穷性、确定性、可行性、输入、输出。

3. “好” 算法的要求

正确性、健壮性、高效性。

4. 算法的表示

算法表示的优缺点如表 8-1 所示。

表 8-1 算法表示的优缺点

算法表示	优点	缺点
自然语言	容易理解	二义性, 冗长
流程图	直观易懂	严密性不如程序设计语言, 灵活性不如自然语言
程序设计语言	计算机直接运行	抽象性差, 拘泥细节, 忽视正确性, 需要掌握编程
伪代码	简明扼要	

8.2 算法的分析基础

1. 算法复杂性

时间复杂性、空间复杂性。

2. 渐进符号

(1) O 记号: 给出一个函数的渐进上界。给定一个函数 $g(n)$, $O(g(n))$ 表示为一个函数集合 $\{f(n): \text{存在正常数 } c \text{ 和 } n_0, \text{ 使得对所有的 } n \geq n_0, \text{ 有 } 0 \leq f(n) \leq cg(n)\}$, 如图 8-2 (a) 所示。

(2) Ω 记号: 给出一个函数的渐进下界。给定一个函数 $g(n)$, $\Omega(g(n))$ 表示为一个函数集合 $\{f(n): \text{存在正常数 } c \text{ 和 } n_0, \text{ 使得对所有的 } n \geq n_0, \text{ 有 } 0 \leq cg(n) \leq f(n)\}$, 如图 8-2 (b) 所示。

(3) Θ 记号: 给出一个函数的渐进上界和下界, 即渐进确界。给定一个函数 $g(n)$, $\Theta(g(n))$ 表示为一个函数集合 $\{f(n): \text{存在正常数 } C_1, C_2 \text{ 和 } n_0, \text{ 使得对所有的 } n \geq n_0, \text{ 有 } 0 \leq C_1g(n) \leq f(n) \leq C_2g(n)\}$ 。

由上述定义可知, $f(n) = \Theta(g(n))$, 当且仅当 $f(n) = O(g(n))$ 和 $f(n) = \Omega(g(n))$, 如图 8-2 (c) 所示。

3. 递归式

(1) 展开法。

(2) 代换法。

(3) 递归树法。

(4) 主方法。

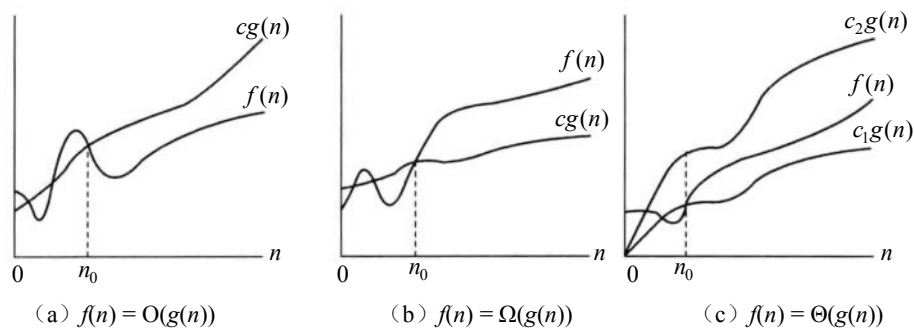


图 8-2 渐进符号

8.3 分治法

1. 递归的概念

2. 分治法思想

分解、求解、合并。

3. 分治法实例

归并排序。

8.4 动态规划法

1. 基本思想

(1) 找出最优解的性质，并刻画其结构特征。

(2) 递归定义最优解的值。

(3) 自底向上算出最优解。

(4) 构造最优解。

2. 实例

0-1 背包问题。

8.5 贪心法

1. 基本思想

2. 实例

活动选择问题、背包问题。

8.6 回溯法

1. 解空间

2. 基本思想

3. 算法框架

非递归、递归。

4. 实例

0-1 背包问题。

8.7 分支限界法

1. 队列式 (FIFO) 分支限界法

2. 优先队列式分支限界法

8.8 概率算法

1. 数值概率算法

2. 蒙特·卡罗 (Monte Carlo) 算法

3. 拉斯维加斯 (Las Vegas) 算法

4. 舍伍德 (Sherwood) 算法

8.9 近似算法

1. 性能标准

算法的时间复杂度、解的近似程度。

2. 实例

定点覆盖问题、TSP 问题、子集和数问题。

8.10 NP 完全性理论

1. P 类问题和 NP 类问题

2. NP 完全问题

3. 典型的 NP 完全问题

8.11 练习题

1. 在求解某问题时, 经过分析发现该问题具有最优子结构性质, 求解过程中子问题被重复求解, 则采用 (a) 算法设计策略; 以深度优先的方法搜索解空间, 则采用 (b) 算法设计策略。

- (a) A. 分治 B. 动态规则 C. 贪心 D. 回溯
(b) A. 动态规划 B. 贪心 C. 回溯 D. 分治限界

解析:

动态规划算法总体思想: 如果能够保存已解决的子问题的答案, 而在需要时再找出已求得的答案, 就可以避免大量的重复计算, 从而得到多项式时间算法。从根开始计算, 直到找到位于某个节点的解, 回溯法(深度优先搜索)作为最基本的搜索算法, 其采用了一种“一直向下走, 走不通就掉头”的思想(体会“回溯”二字), 相当于采用了先根遍历的方法来构造搜索树。

答案: B、C

2. 考虑下述背包问题的实例。有 5 件物品, 背包容量为 100, 每件物品的价值和重量如下表所示, 已经按照物品的单位重量价值从大到小进行了排序。根据物品单位重量价值大优先的策略装入背包中, 则采用了 (a) 设计策略。考虑 0-1 背包问题(每件物品或者全部装入背包, 或者不全部装入背包)和部分背包问题(物品可以部分装入背包), 求解该实例得到的最大价值分别为 (b)。

物品编号	价值(元)	重量(千克)
1	50	5
2	200	25
3	180	30
4	225	45
5	200	50

- (a) A. 分治 B. 贪心 C. 动态规划 D. 回溯
(b) A. 605 和 630 B. 605 和 605 C. 430 和 630 D. 630 和 430

解析:

本题考查贪心算法和背包问题的知识点。

贪心算法(又称贪婪算法)是指在对问题求解时, 总是做出当前看来最好的选择。也就是说, 不从整体最优上加以考虑, 所做出的仅是在某种意义上的局部最优解。贪心算法不是让所有问题都能得到整体最优解, 但对范围相当广泛的许多问题能得到整体最优解或整体最优解的近似解。

采用 0-1 背包考虑问题时, 只能放入 1、2、3 号物品, 故总价值为 430; 采用部分背包可以

将物品拆分,故放入 1、2、3 号物品后还可以将 4 号物品部分地装入,使得背包容量尽量满,故总容量为 630。

答案: B、C

3. 给定 n 个整数构成的数组 $A=\{a_1, a_2, \dots, a_n\}$ 和整数 x , 判断 A 中是否存在两个元素 a_i 和 a_j , 使得 $a_i + a_j = x$ 。为了解问题, 首先用归并排序算法对数组 A 进行从大到小的排序, 然后判断是否存在 $a_i + a_j = x$, 具体的方法如下列伪代码所示。则求解该问题时排序算法应用了 (a) 算法设计策略, 整个算法的时间复杂度为 (b)。

```
i=1; j=n
While i < j
    If  $a_i + a_j = x$  return true
    Else if  $a_i + a_j > x$ 
        j--;
    Else
        i++;
Return false;
```

- | | | | |
|---------------|-----------------|-------------|------------------|
| (a) A. 分治 | B. 贪心 | C. 动态规划 | D. 回溯 |
| (b) A. $O(n)$ | B. $O(n \lg n)$ | C. $O(n^2)$ | D. $O(n \lg 2n)$ |

解析:

分治算法的基本思想是将一个规模为 n 的问题分解为 k 个规模较小的子问题, 这些子问题相互独立且与原问题性质相同。求出子问题的解, 就可以得到原问题的解。

答案: A、B

第9小时

数据库技术基础

9.0 章节考点分析

根据考试大纲，本小时内容比较重要，需要掌握数据库模型、数据模型、ER图、规范化、数据库的集合运算和关系运算、数据库语言 SQL，了解数据库管理系统、数据库的控制功能、数据仓库、分布式数据库基础知识，上午客观题约占6分，下午案例题至少一题。

本小时的内容偏重于理解与应用，本小时的架构如图9-1所示。

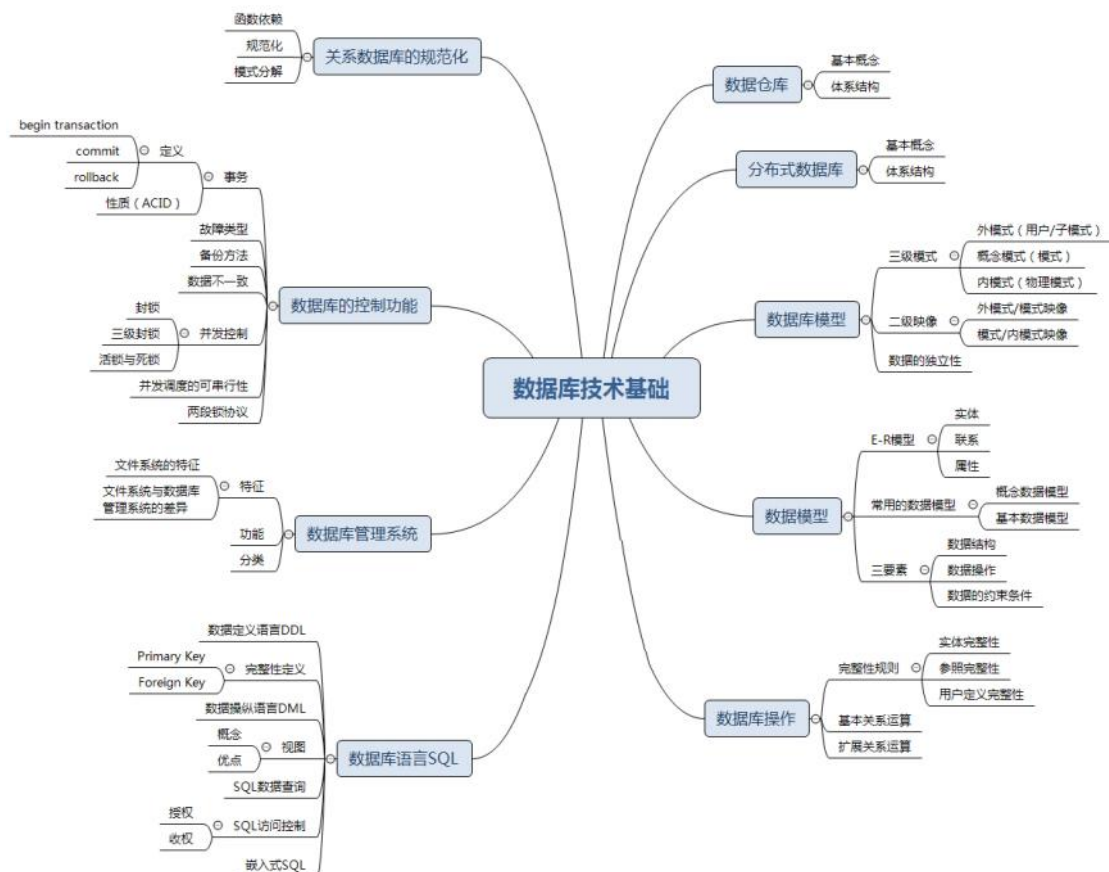


图 9-1 第 9 小时架构图



【导读小贴士】

数据库技术是每个程序员应掌握的基础知识和基本技能，本小时内容虽然多，但如果有数据库的实践经验，学起来并不难。如果没有这方面的经验，就必须把各种概念、计算彻底搞懂。

9.1 数据库模型

1. 三级模式

概念模式（模式）、外模式（用户/子模式）、内模式（物理模式）。

2. 二级映射

模式/内模式映像、外模式/模式映像。

3. 数据的独立性

数据的逻辑独立性、数据的物理独立性。

三级模式和二级映射如图 9-2 所示。

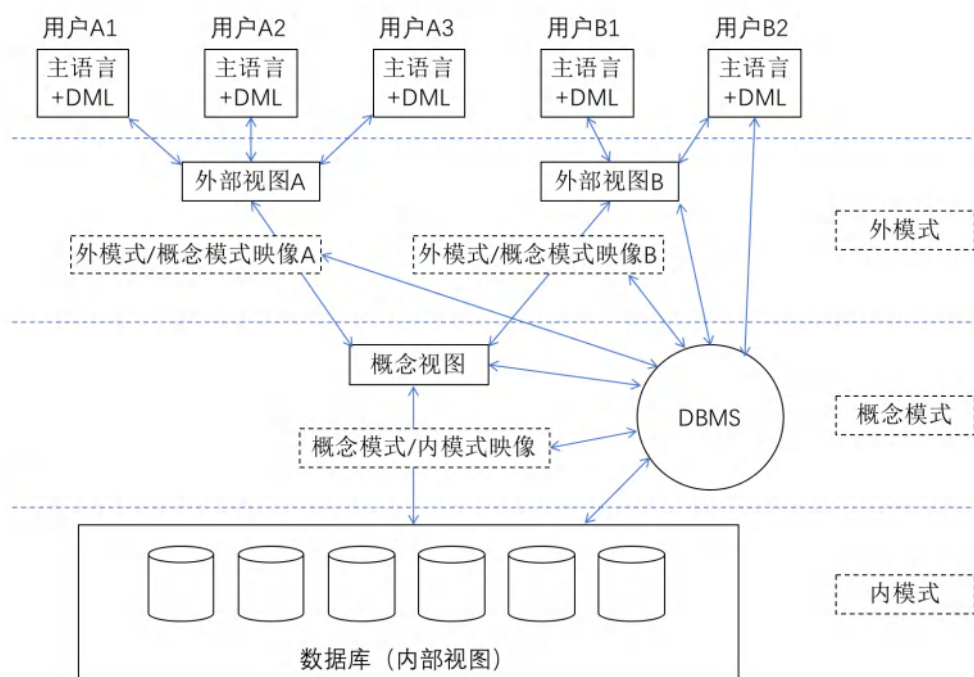


图 9-2 三级模式和二级映射

9.2 数据模型

1. 常用的数据模型

(1) 概念数据模型：也称信息模型，以用户视角对数据和信息建模，是对现实世界信息的抽象，如 E-R 模型。

(2) 基本数据模型：以计算机视角对数据建模，是对现实世界数据特征的抽象，用 DBMS 来实现，有层次模型、网状模型、关系模型和面向对象模型等。

2. 数据模型三要素

数据结构、数据操作、数据的约束条件。

3. E-R 模型的概念

用 E-R 图来描述现实世界的概念模型，E-R 图中的主要构件如表 9-1 所示。

- (1) 实体：用矩形表示，每个实体由一组属性表示，包括主键、候选键、外键。
- (2) 联系：用菱形表示，分为一对一 (1:1)、一对多 (1:n)、多对多 (m:n)。
- (3) 属性：用椭圆表示，是实体某方面的特性。E-R 模型中的属性分为：①简单和复合属性；②单值和多值属性；③null 属性；④派生属性。

表 9-1 E-R 图中的主要构件

构件	说明
矩形 	表示实体集
双边矩形 	表示弱实体集
菱形 	表示联系集
双边菱形 	表示弱实体集对应的标识性联系
椭圆 	表示属性
线段 	将属性与相关的实体集连接，或者将实体集与联系集相连
双椭圆 	表示多值属性
虚椭圆 	表示派生属性
双线 	表示一个实体全部参与到联系集中

9.3 数据库操作

1. 关系模型的完整性规则

- (1) 实体完整性：主属性不能为空值。
- (2) 参照完整性：外键的值必须存在。
- (3) 用户定义完整性：满足用户要求。

2. 基本关系运算

关系代数运算符如表 9-2 所示。

- (1) 并。
- (2) 差。

- (3) 笛卡尔积：相乘。
- (4) 投影 π ：选列。
- (5) 选择 σ ：选行。

表 9-2 关系代数运算符

运算符		含义	运算符		含义
集合运算符	$\cup$	并	比较运算符	$>$	大于
	$-$	差		$\geq$	大于等于
	$\cap$	交		$<$	小于
	$\times$	笛卡尔积		$\leq$	小于等于
				$=$	等于
				$\neq$	不等于
关系运算符	σ	选择	逻辑运算符	$\neg$	非
	π	投影		$\wedge$	与
	$\bowtie$	自然连接		$\vee$	或
	$\div$	除			

3. 扩展关系运算

- (1) 交。
- (2) 自然连接：符号为 $\bowtie$ ，去除相同属性的组，而且水平与垂直方向的分量均相等，示例如图 9-3 所示。

A	B	C
a	b	c
b	a	d
c	d	e
d	f	g

(a) 关系 R

A	C	D
a	c	d
d	f	g
b	d	g

(b) 关系 S

A	B	C	D
a	b	c	d
b	a	d	g

(c) $R \bowtie S$

图 9-3 自然连接

- (3) 除：符号为 $\div$ ，示例如图 9-4 所示。
- (4) 外连接：左外连接、右外连接、全外连接，示例如图 9-5 所示。

A	B	C	D
a	b	c	d
a	b	e	f
a	b	h	k
b	d	e	f
b	d	d	l
c	k	c	d
c	k	e	f

(a) 关系 R

C	D
c	d
e	f

(b) 关系 S

A	B
a	b
c	k

(c) $R \div S$

图 9-4 除

A	B	C
a	b	c
b	a	d
c	d	e
d	f	g

(a) 关系 R

B	C	D
b	c	d
d	e	g
f	d	g
d	e	c

(b) 关系 S

A	B	C	D
a	b	c	d
c	d	e	g
c	d	e	c
b	a	d	null
d	f	g	null

(c) 左外连接 $R \bowtie S$

A	B	C	D
a	b	c	d
c	d	e	g
c	d	e	c
null	f	d	g

(d) 右外连接 $R \bowtie S$

A	B	C	D
a	b	c	d
c	d	e	g
c	d	e	c
b	a	d	null
d	f	g	null
null	f	d	g

(e) 全外连接 $R \bowtie S$

图 9-5 外连接

9.4 数据库语言 SQL

- 1. 数据定义
CREATE、DROP、ALTER。
- 2. 数据操纵
INSERT、UPDATE、DELETE。

3. 数据查询

SELECT、简单查询、连接查询、子查询与聚集函数、分组查询、更名运算、字符串操作。

4. 数据控制

GRANT、REVOKE。

关系数据库的三级模式结构如图 9-6 所示。

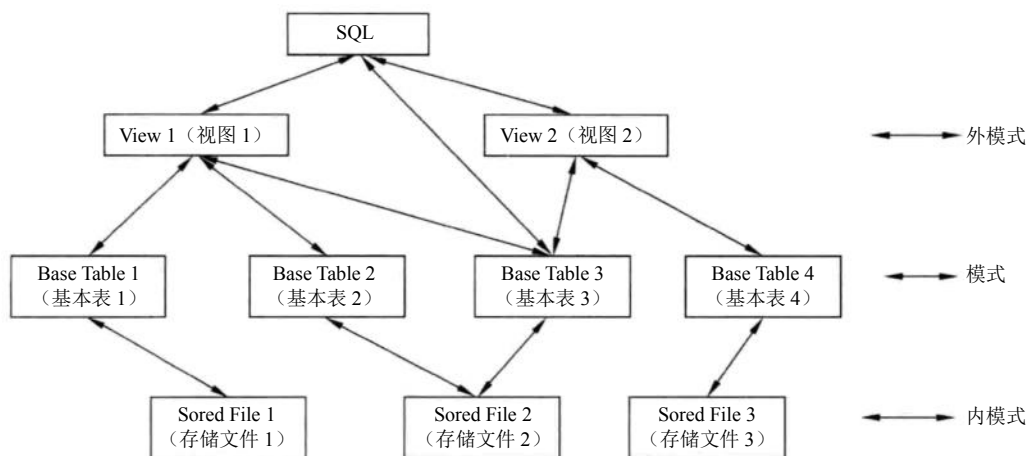


图 9-6 关系数据库的三级模式结构

9.5 数据库管理系统

1. 数据库管理系统

数据库管理系统（DBMS）是操纵和管理数据库的软件，用于建立、使用和维护数据库。

2. DBMS 的功能

- (1) 数据定义。
- (2) 数据库的操作。
- (3) 数据库的运行管理。
- (4) 数据的组织、存储和管理。
- (5) 数据库的建立和维护。
- (6) 其他功能，如通信、数据转换等。

3. DBMS 的特征

- (1) 数据的结构化且统一管理。
- (2) 有较高的数据独立性。
- (3) 数据的控制功能，对数据库中数据的安全性、完整性、并发和恢复的控制。

4. DBMS 的分类

- (1) 关系数据库系统 (RDBS)。
- (2) 面向对象的数据库系统 (OODBS)。
- (3) 对象关系数据库系统 (ORDBS)。

9.6 关系数据库的规范化

1. 函数依赖

函数依赖、平凡依赖、非平凡依赖、完全依赖、部分依赖、传递依赖、Armstrong 公理系统 (A1 自反律、A2 增广率、A3 传递率; 合并规则、伪传递率、分解规则)。

2. 规范化

- (1) 1NF: 每个分量 (属性) 不可分割。
- (2) 2NF: 满足 1NF, 而且消除非主属性对候选键的部分依赖。
- (3) 3NF: 满足 2NF, 而且消除非主属性对候选键的传递依赖。

3. 模式分解

- (1) 分解。
- (2) 无损连接。
- (3) 保存函数依赖。

9.7 数据库的控制功能

1. 事务

事务是指作为单个逻辑工作单元执行的一系列操作, 要么完全执行, 要么完全不执行。

2. 事务的性质 (ACID)

- (1) 原子性 (Atomicity): 要么都做, 要么都不做。
- (2) 一致性 (Consistency): 事务完成时, 必须保证所有数据都保持一致的状态。
- (3) 隔离性 (Isolation): 事务之间相互隔离。
- (4) 持久性 (Durability): 事务完成后, 更新操作永久有效。

3. 数据库的故障类型

- (1) 事务内部故障: 如运算溢出、并发事务发生死锁等。
- (2) 系统故障: 也称为软故障, 是指造成系统停运的事件, 如 CPU 故障、OS 故障、突然停电等。
- (3) 介质故障: 也称为硬故障, 如磁盘损坏等。
- (4) 计算机病毒。

4. 备份方法
- (1) 静态、动态转储。
 - (2) 海量、增量转储。
 - (3) 日志文件。

5. 数据不一致
- (1) 丢失修改。
 - (2) 不可重复读。
 - (3) 读脏数据。

数据不一致的三种情况如表 9-3 所示。

6. 封锁
- (1) 排他锁 (X/写锁)。

表 9-3 数据不一致的三种情况

时间	丢失修改		不可重复读		读脏数据	
	T ₁	T ₂	T ₁	T ₂	T ₁	T ₂
t ₁	read(A)[16]		read(A)[50]		read(C)[100]	
t ₂		read(A)[16]	read(B)[100]		C=C*2[200]	
t ₃	A=A-1		C=A+B[150]		write(C)[200]	
t ₄		A=A-1		read(B)[100]		read(C)[200]
t ₅	write(A)[15]			B=B*2[200]		...
t ₆				write(B)[200]		...
t ₇		write(A)[15]			ROLLBACK	
t ₈			read(A)[50]		(C=100)	
t ₉			read(B)[200]			
t ₁₀			C=A+B[250]			
t ₁₁			(验算不对)			

- (2) 共享锁 (S/读锁)。

7. 三级封锁协议

- (1) 一级：修改数据前先加 X 锁，事务结束后释放，可解决丢失修改问题。
- (2) 二级：在一级基础上，读数据之前加 S 锁，读完后释放即可解决读脏数据问题。
- (3) 三级：在一级基础上，读数据之前加 S 锁，直到事务结束后释放 S 锁，即可解决丢失修改、读脏数据、不可重复读三个数据不一致的问题。

8. 活锁与死锁

9. 并发调度的可串行性
10. 两段锁协议

9.8 数据仓库和分布式数据库基础知识

1. 数据仓库

数据仓库是一个面向主题的、集成的、相对稳定的、反映历史变化的数据集合，用于支持管理决策，数据仓库的体系结构如图 9-7 所示。

2. ETL

- (1) 抽取 (Extraction)。
- (2) 转换 (Transformation)。
- (3) 加载 (Loading)。

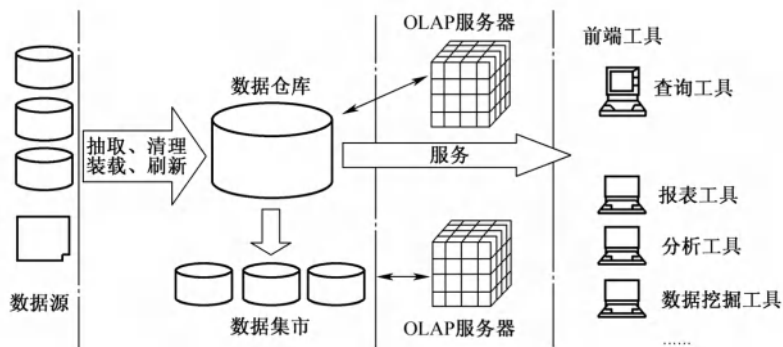


图 9-7 数据仓库的体系结构

3. 数据仓库的分类

- (1) 企业仓库。
- (2) 数据集市。
- (3) 虚拟仓库。

4. 分布式数据库

通常使用较小的计算机系统，每台计算机可以单独放在一个地方，每台计算机中都可能存有 DBMS 的一份完整拷贝副本或部分拷贝副本，并具有自己局部的数据库，位于不同地点的许多计算机通过网络互相连接，共同组成一个完整的、全局的逻辑上集中、物理上分布的大型数据库，分布式数据库的体系结构如图 9-8 所示。

5. 分布式数据库的特点

- (1) 高可扩展性。
- (2) 高并发性。

(3) 高可用性。

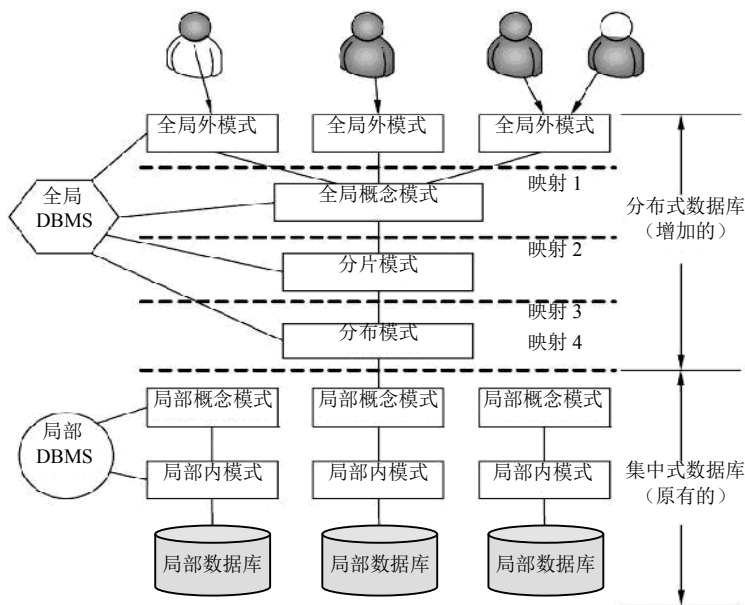


图 9-8 分布式数据库的体系结构

9.9 练习题

- 在数据库系统中，视图是一个_____。
 - 真实存在的表，并保存了待查询的数据
 - 虚拟表，查询时只能从一个基本表中导出
 - 真实存在的表，只有部分数据来源于基本表
 - 虚拟表，查询时可以从一个或多个基本表或视图中导出

解析：

视图是从一个或几个基本表（或视图）导出的虚拟表，其内容由查询定义。同真实的表一样，视图包含一系列带有名称的列和行数据。但视图并不在数据库中以存储的数据值集形式存在，行和列数据来自自定义视图的查询所引用的表，并且在引用视图时动态生成。

答案：D

2. 给定关系模式 $R(U, F)$ ，其中，属性集 $U = \{A, B, C, D, E, G\}$ ，函数依赖集 $F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D, AE \rightarrow G\}$ 。若将 R 分解为两个子模式_____，则分解后的关系模式保持函数依赖。

- $R_1(A, B, C)$ 和 $R_2(D, E, G)$
- $R_1(A, B, C, D)$ 和 $R_2(A, E, G)$
- $R_1(B, C, D)$ 和 $R_2(A, E, G)$
- $R_1(B, C, D, E)$ 和 $R_2(A, E, G)$

解析:

本题考查数据库中的函数依赖。函数依赖的定义为: 设 $R(U)$ 是属性集 $U=\{A_1, A_2, \dots, A_n\}$ 上的关系模式, X 和 Y 是 U 的子集。若对 $R(U)$ 的任一具体关系 r 中的任意两个元组 t_1 和 t_2 , 只要 $t_1[X]=t_2[X]$, 就有 $t_1[Y]=t_2[Y]$, 则称“ X 函数确定 Y ”或“ Y 函数依赖于 X ”(Functional Dependence), 记作 $X \rightarrow Y$ 。只有 B 选项中的两个关系式满足定义。

答案: B

3. 假设学生 Students 和教师 Teachers 的关系模式如下所示:

Students (学号, 姓名, 性别, 类别, 身份证号)

Teachers (教师号, 姓名, 性别, 身份证号, 工资)

其中, 学生关系中的类别分为“本科生”和“研究生”两类。

(1) 查询在读研究生的教师的平均工资、最高与最低工资的差值的 SQL 语句如下:

SELECT (a)

FROM Students, Teachers

WHERE (b);

- (a) A. AVG (工资) AS 平均工资, MAX (工资) - MIN (工资) AS 差值
 B. 平均工资 AS AVG (工资), 差值 AS MAX (工资) - MIN (工资)
 C. AVG (工资) ANY 平均工资, MAX (工资) - MIN (工资) ANY 差值
 D. 平均工资 ANY AVG (工资), 差值 ANY MAX (工资) - MIN (工资)
- (b) A. Students.身份证号=Teachers.身份证号
 B. Students.类别='研究生'
 C. Students.身份证号=Teachers.身份证号 AND Students.类别='研究生'
 D. Students.身份证号=Teachers.身份证号 OR Students.类别='研究生'

(2) 查询既是女教师, 又是研究生且工资大于等于 3500 元的身份证号和姓名的 SQL 语句如下:

(SELECT 身份证号, 姓名

FROM Students

WHERE (c))

INTERSECT

(SELECT 身份证号, 姓名

FROM Teachers

WHERE (d))

- (c) A. 工资 >= 3500
 B. 工资 >= '3500'
 C. 性别=女 AND 类别=研究生
 D. 性别='女' AND 类别='研究生'
- (d) A. 工资 >= 3500
 B. 工资 >= '3500'
 C. 性别=女 AND 类别=研究生
 D. 性别='女' AND 类别='研究生'

解析:

查询在读研究生的教师的平均工资，最高与最低工资的差值的 SQL 语句如下:

```
SELECT AVG(工资) AS 平均工资, MAX(工资) - MIN(工资) AS 差值
```

```
FROM Teachers, Students
```

```
WHERE Students.身份证号=Teachers.身份证号
```

```
AND Students.类别='研究生'
```

查询既是女教师又是研究生且工资大于等于 3500 元的身份证号和姓名的 SQL 语句如下:

```
(SELECT 身份证号,姓名
```

```
FROM Students
```

```
WHERE 性别='女' AND 类别='研究生')
```

```
INTERSECT
```

```
(SELECT 身份证号,姓名
```

```
FROM Teachers
```

```
WHERE 工资>=3500)
```

答案: A、C、D、A

第10小时

网络与信息安全基础知识

10.0 章节考点分析

第10小时主要学习计算机网络基础知识、计算机网络分类、网络设备、网络标准、网络协议、TCP/IP协议族、网络安全、网络信息安全、防火墙技术、入侵检测等内容。

根据考试大纲，本小时的知识点会涉及单选题型和案例分析题型，约占3~7分。本小时的内容偏重于概念知识，根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，考查概念的知识点多数参照教材，扩展内容较少。本小时的架构如图10-1所示。

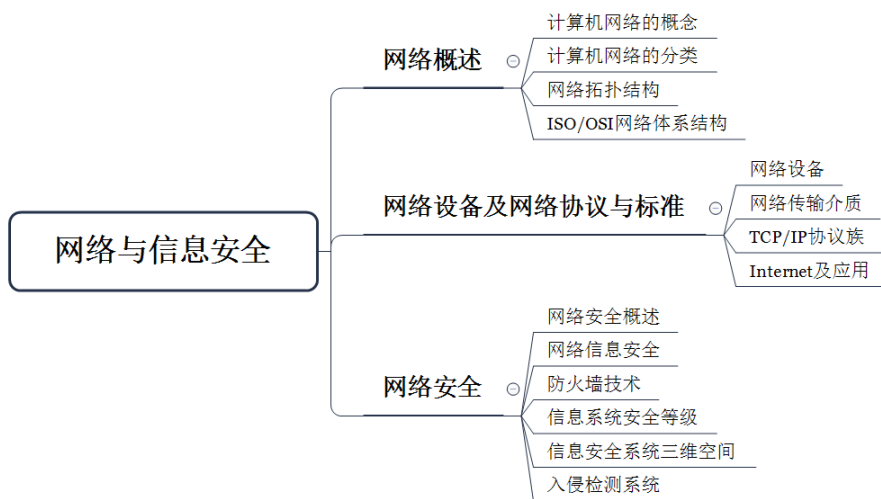


图 10-1 第 10 小时架构图



【导读小贴士】

网络与信息安全基础知识是最基础的知识点，在软件设计师考试中，一些选择题、案例分析题都会来自本小时，建议除了理解的基础上掌握核心知识点外，还可以灵活应用。

10.1 网络概述

计算机网络是计算机技术与通信技术相结合的产物，它实现了远程通信、远程信息处理和资源共享。

10.1.1 计算机网络的概念

计算机网络的功能：数据通信、资源共享、负载均衡、高可靠性。
计算机网络按照数据通信和数据处理的功能分为两层：内层通信子网和外层资源子网。

10.1.2 计算机网络的分类

这里主要介绍局域网、城域网、广域网，如表 10-1 所示。

表 10-1 各类网络的特征参数

网络分类	缩写	分布距离	计算机分布范围	传输速率范围
局域网	LAN	10m 左右	房间	4Mbps~1Gbps
		100m 左右	楼宇	
		1000m 左右	校园	
城域网	MAN	10km	城市	50kbps~100Mbps
广域网	WAN	100km 以上	国家或全球	9.6kbps~45Mbps

10.1.3 网络拓扑结构

常见的网络拓扑结构包括总线型结构、星型结构、环型结构、树型结构、分布式结构，如图 10-2 所示。

10.1.4 ISO/OSI 网络体系结构

ISO/OSI 的参考模型一共有七层，如图 10-3 所示。

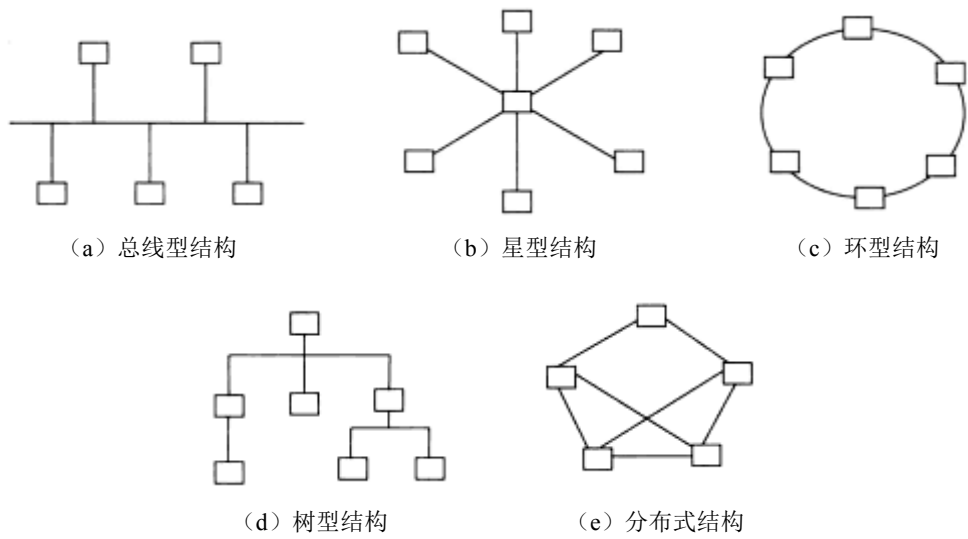


图 10-2 常见的网络拓扑结构

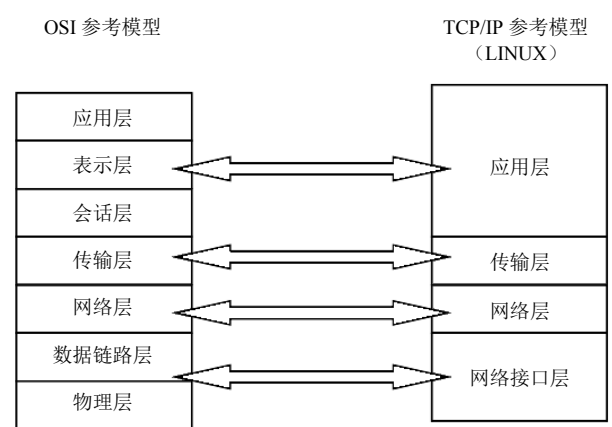


图 10-3 OSI 参考模型

由低层到高层分别为：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层（速记词：巫术忘传会飘鹰）。

10.2 网络设备及网络协议与标准

构建一个实际的网络需要网络的传输介质、网络互连设备作为支持，这里主要介绍构件网络的传输介质和互连设备。

10.2.1 网络设备

网络设备如图 10-4 所示。

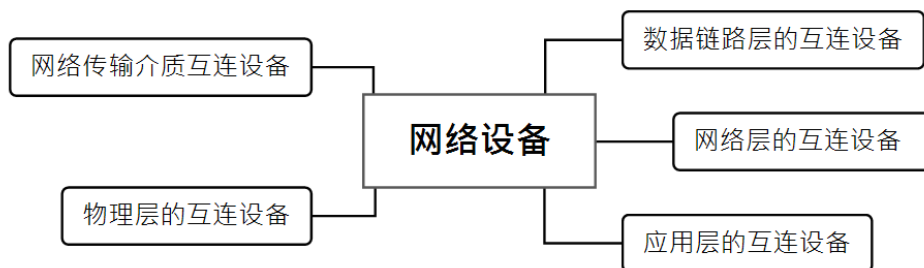


图 10-4 网络设备

10.2.2 网络传输介质

网络传输介质如图 10-5 所示。

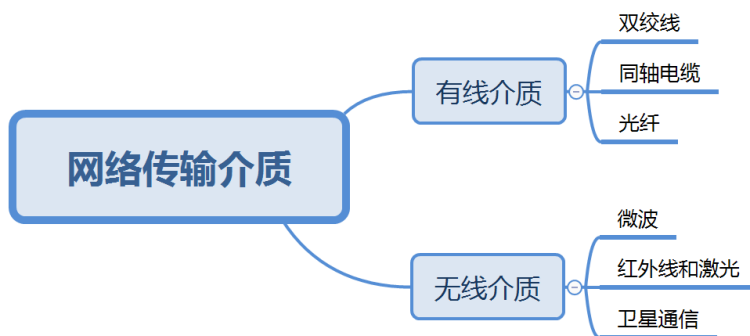


图 10-5 网络传输介质

10.2.3 TCP/IP 协议族

1. TCP/IP 分层模型

协议是对数据在计算机或设备之间传输时的表示方法进行定义和描述的标准。协议规定了进行传输、检测错误及传送确认信息等内容。TCP/IP 是个协议族，它包含了多种协议。ISO/OSI 模型、TCP/IP 的分层模型及协议的对比如图 10-6 所示。

2. 网络接口层协议

TCP/IP 协议不包含具体的物理层和数据链路层，只定义了网络接口层作为物理层与网络层的接口规范。这个物理层可以是广域网，也可以是局域网。网络接口层处在 TCP/IP 协议的最底层，主要负责管理为物理网络准备数据所需的全部服务程序和功能。

ISO/OSI 模型		TCP/IP 协议				TCP/IP 模型		
应用层	文件传输 协议 (FTP)	远程登录 协议 (Telnet)	电子邮件 协议 (SMTP)	网络文件 服务协议 (NFS)	网络管理 协议 (SNMP)	应用层		
表示层								
会话层								
传输层	TCP				UDP			传输层
网络层	IP		ICMP		ARP RARP			网际层
数据链路层	Ethernet IEEE 802.3	FDDI	Token-Ring/ IEEE 802.5	ARCnet	PPP/SLIP	网络接口层		
物理层						硬件层		

图 10-6 ISO/OSI 模型与 TCP/IP 模型的对比

3. 网际层协议 IP

网际层是整个 TCP/IP 协议族的重点，在网际层定义的协议除了 IP 外，还有 ICMP、ARP 和 RARP 等几个重要的协议。

IP 的主要功能包括将上层数据（如 TCP、UDP 数据）或同层的其他数据（如 ICMP 数据）封装到 IP 数据中；将 IP 数据报送到最终目的地；为了使数据能够在链路层上进行传输，对数据进行分段；确定数据报到达其他网络中的目的地的路径。

4. ARP 和 RARP

地址解析协议 ARP 的作用是将 IP 地址转换为物理地址，反地址解析协议 RARP 的作用是将物理地址转换为 IP 地址。

5. Internet 控制信息协议 ICMP

Internet 控制信息协议（ICMP）是网际层的另一个比较重要的协议。由于 IP 是一种尽力传送的通信协议，即传送的数据报可能会丢失、重复、延迟或乱序，因此 IP 需要一种避免差错并在发生差错时报告的机制。ICMP 就是一个专门用于发送差错报文的协议。ICMP 定义了五种差错报文（源抑制、超时、目的不可达、重定向和要求分段）和四种信息报文（回应请求、回应应答、地址屏蔽码请求和地址屏蔽码应答）。

6. 传输控制协议 TCP

传输控制协议 TCP 是 TCP/IP 协议族中最重要的协议之一，利用 TCP 在源主机和目的机之间建立和关闭连接操作时，均需要通过三次握手来确认建立和关闭是否成功，如图 10-7 所示。

7. 用户数据报协议 UDP

用户数据报协议 UDP 是一种不可靠、无连接的协议。UDP 协议软件的主要作用是将 UDP 消息展示给应用层，并不负责重新发送丢失或出错的数据消息，不对接收到的无序 IP 数据报重新排序，不消除重复的 IP 数据报等。

8. 应用层协议

应用层协议包括 NFS、Telnet、SMTP、DNS、SNMP、FTP 等。

源 10 小时

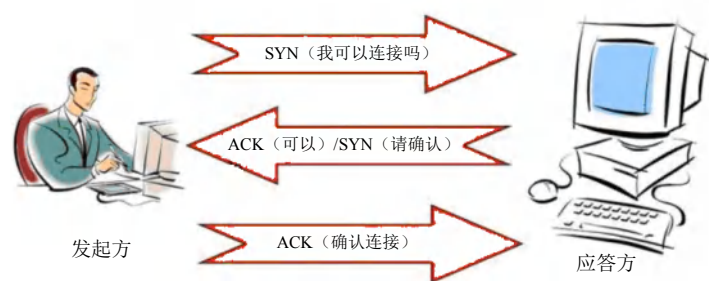


图 10-7 TCP 建立连接的“三次握手”过程

10.2.4 Internet 及应用

一个完整、通用的层次型主机域名由四个部分组成：计算机主机名、本地名、组名、最高层域名。从右到左，子域名分别表示不同的国家或地区的名称、组织类型、组织名称、分组织名称和计算机名称等。域名地址的最后一部分子域名称为高层域名，它大致分成两类：一类是组织性顶级域名，另一类是地理性顶级域名。

例如：`www.dzkjdx.edu.cn` `cn` 是地理性顶级域名，表示“中国”。
`www.263.net` `net` 是组织性顶级域名，表示“网络技术组织机构”。

Internet 中的地址可以分为五类：A 类、B 类、C 类、D 类和 E 类，各类地址分配方案如图 10-8 所示。

	0 1	7 8	16	24	31
A 类:	0	网络地址	主机地址 (24)		
	0 1 2		15 16	24	31
B 类:	10	网络地址 (14)		主机地址 (16)	
	0 1 2 3			24	31
C 类:	110	网络地址 (21)			主机地址
	0 1 2 3 4			24	31
D 类:	1110	组播、路由器修改 (28)			
	0 1 2 3 4			24	31
E 类:	1111	试验用地址 (28)			

图 10-8 各类地址分配方案

10.3 网络安全

10.3.1 网络安全概述

计算机网络安全是指计算机、网络系统的软件或硬件及系统中的数据受到保护，不因偶然的或恶意的原因遭到破坏、更改、泄露，确保系统能连续、可靠地运行，使网络服务不中断。网络安全从本质上讲就是网络上的信息安全。

凡是涉及网络上信息的保密性、完整性、可用性、真实性和可控性的相关技术和理论，都是网络安全所要研究的领域。

网络安全涉及的主要内容如图 10-9 所示。

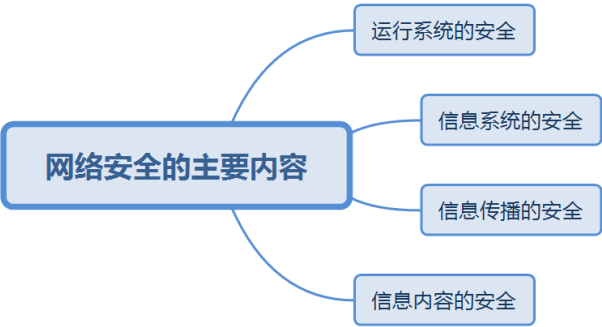


图 10-9 网络安全涉及的主要内容

信息系统安全的基本需求：保密性、完整性、可用性、可控性、可核查性。

网络安全威胁主要来自五类：物理威胁、网络攻击、身份鉴别、编程威胁、系统漏洞。

10.3.2 网络信息安全

1. 信息存储安全

信息存储安全的主要内容如图 10-10 所示。

2. 信息传输安全

信息传输加密的主要措施有链路加密、结点加密、端一端加密。

10.3.3 防火墙技术

1. 防火墙的分类

防火墙技术经历了包过滤、应用代理网关和状态监测三个发展阶段。

第 10 小时



图 10-10 信息存储安全

(1) 包过滤防火墙。包过滤防火墙工作在网络层，对数据包的源及目的 IP 具有识别和控制作用，对于传输层，也只能识别数据报是 TCP 还是 UDP 及所用的端口信息。包过滤防火墙的处理速度较快，也易于配置。

包过滤防火墙的优点是防火墙对每条传入和传出网络的包实行低水平控制：①每个 IP 包的字段都被检查，如源地址、目的地址、协议和端口等；②可以识别和丢弃带欺骗性的源 IP 地址的包；③两个网络之间访问的唯一来源；④通常被包含在路由器数据报中，所以不必额外的系统来处理这个特征。

(2) 应用代理网关防火墙。应用代理网关防火墙的优点是可以检查应用层、传输层和网络层的协议特征，对数据包的检测能力较强，其缺点是难于配置、处理速度非常慢。

(3) 状态监测防火墙。状态监测防火墙结合了代理防火墙的安全性和包过滤防火墙的高速度等优点，在不损失安全性的技术上将代理防火墙的性能提高了 10 倍。

2. 典型防火墙的体系结构

典型防火墙的体系结构包括过滤路由器、双宿主主机、屏蔽主机网关和被屏蔽子网等类型。

(1) 包过滤路由器。又称为屏蔽路由器，一般作用在网络层，对进出内部网络的所有信息进行分析，并按照一定的安全策略（过滤规则）对进出内部网络的信息进行限制。包过滤路由器防火墙示意图如图 10-11 所示。

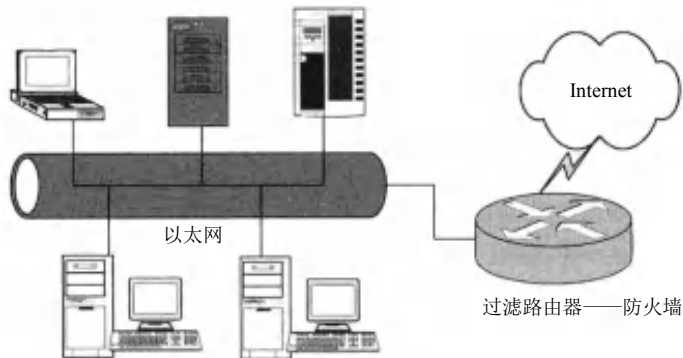


图 10-11 包过滤路由器防火墙示意图

采用包过滤路由器防火墙的优点在于速度快、实现方便。缺点是安全性能差，不同操作系统环境下 TCP 和 UDP 端口号所代表的应用服务协议类型有所不同，故兼容性差，没有或只有较少的日志记录能力。

(2) 双宿主主机。双宿主主机结构是围绕着至少具有两个网络接口的双宿主主机构成的，每一个接口都连接在物理和逻辑上分离的不同网段，代理服务器软件在双宿主主机上运行，如图 10-12 所示。

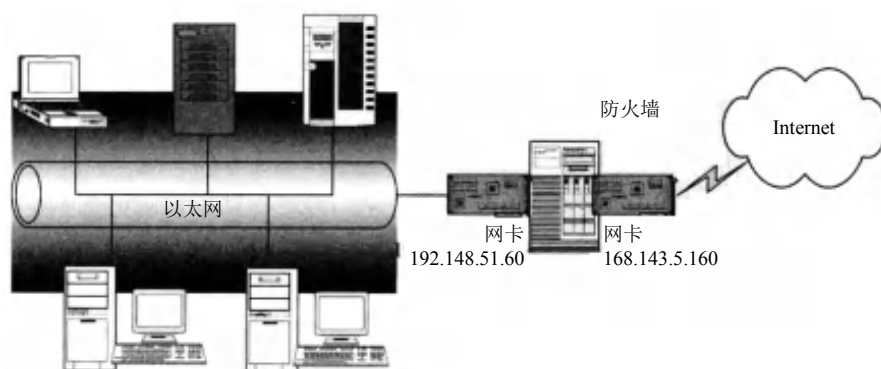


图 10-12 双宿主主机防火墙示意图

(3) 屏蔽主机网关。屏蔽主机网关防火墙的优点是安全等级较高，可以提供公开的信息服务的服务器，可以放置在由包过滤器路由器和堡垒主机共用的网段上，如图 10-13 所示。

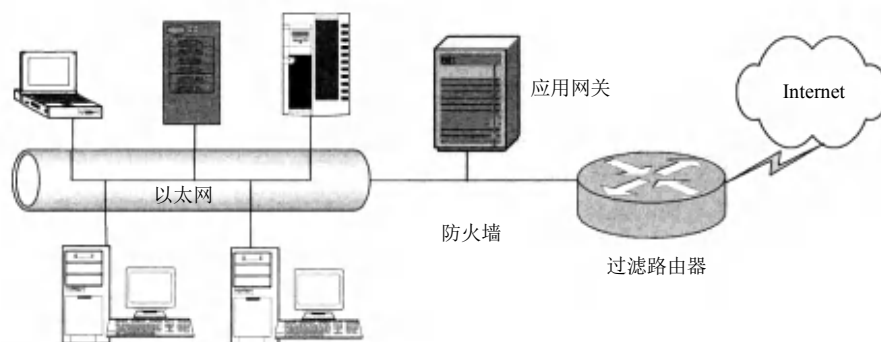


图 10-13 屏蔽主机网关防火墙示意图

(4) 被屏蔽子网。被屏蔽子网由两个包过滤路由器和一个应用网关组成，如图 10-14 所示。

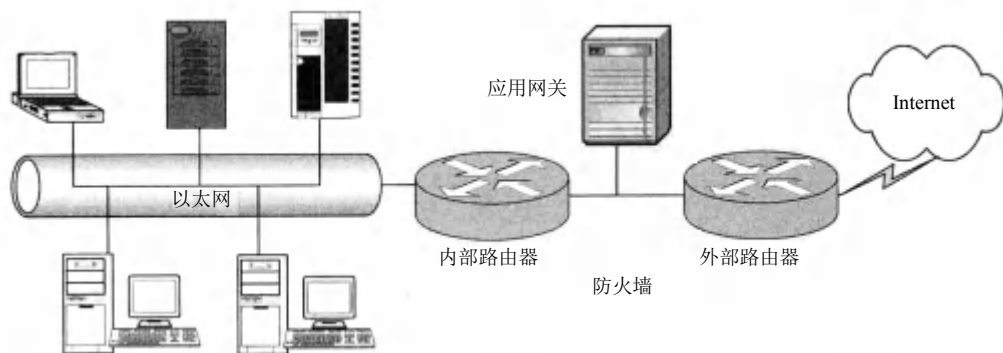


图 10-14 被屏蔽子网防火墙示意图

被屏蔽子网防火墙系统的优点如下：

- 1) 入侵者必须突破三个不同的设备（外部路由器、堡垒主机、内部路由器）才能侵袭内部网络。
- 2) 由于外部路由器只能向 Internet 通告 DMZ 网络的存在，Internet 上的系统不需要有路由器和内部网络相对。
- 3) 由于内部路由器只能向内部网络通告 DMZ 网络的存在，内部网络上的系统不能直接通往 Internet，保证了内部网络上的用户必须通往驻留在堡垒主机上的代理服务器才能访问 Internet。
- 4) 包过滤路由器直接将数据引向 DMZ 网络上所指定的系统，消除了堡垒主机双宿的必要。
- 5) 内部路由器在作为内部网络和 Internet 之间最后的防火墙系统时，能够支持比双宿堡垒主机更大的数据报吞吐量。
- 6) 由于 DMZ 网络是一个与内部网络不同的网络，NAT 可以安装在堡垒主机上，从而避免在内部网络上重新编制或重新划分子网。

10.3.4 信息系统安全等级

信息系统安全等级保护详见表 10-2。（速记词：泳洗俺解放，即俺是一个游泳教练，教了一天的游泳课很累了，下班回到家洗个澡，俺就解放了。）

表 10-2 信息系统安全等级保护

等级保护		应用对象
用户自主保护级	泳	用于普通互联网用户
系统审计保护级	洗	用于互联网、国际网进行商务活动的、需要保密的非重要单位
安全标记保护级	俺	用于地方国家机关、金融单位、邮电通信、能源与水源供给部门、交通运输、大型工商与信息技术企业、重点工程建设等单位
结构化保护级	解	用于中央级国家机关、广播电视部门、重要物资储备单位、社会应急服务部门、尖端科技企业集团、国家重点科研单位、国防建设等部门
访问验证保护级	放	用于国防关键部门、依法需要对计算机信息系统实施特殊隔离的单位

10.3.5 信息安全系统三维空间

信息安全系统三维空间：X、Y、Z 轴形成的信息安全系统的三维空间（即信息系统的安全空间）。

X 轴：安全机制（安全防范体系、授权和审计安全、管理安全、运行安全、应用安全、通信安全、数据安全、平台安全、基础设施安全）。

Y 轴：OSI。

Z 轴：安全服务（实体认证、访问控制、数据保密、数据完整、数据源认证、禁止否认、犯罪证据提供等服务）。

10.3.6 入侵检测系统

入侵检测是指对计算机、网络上的恶意行为进行识别、响应的处理过程。入侵检测系统（IDS）不仅检测外部入侵，也检测内部未授权（违规行为）的活动。对数据的分析是入侵检测的核心。

IDS 是防火墙之后的又一道防线，可以发现防火墙没有发现的入侵行为，如果防火墙是大厦的门禁系统，IDS 就是监控系统。

防火墙允许内部的一些主机可以被外部访问，但 IDS 没有这些功能，只是监视、分析用户和系统活动。IDS 是一个独立的系统，不需要其他安全审计系统的配合。

入侵检测系统的主要作用为：

- （1）实时检测：实时地监视、分析网络中所有的数据报文；发现并实时处理所捕获的数据报文。
- （2）安全审计：对系统记录的网络事件进行统计分析；发现异常现象；得出系统的安全状态，找出所需要的证据。
- （3）主动响应：主动切断连接或与防火墙的联动，调用其他程序处理。
- （4）工作原理：实时监控网络数据，与已知的攻击手段进行匹配，从而发现网络或系统中是否有违反安全策略的行为和遭遇袭击的迹象。
- （5）优点：能及时获知网络安全状况，借助分析发现安全隐患或攻击信息，便于及时采取措施。
- （6）缺点：难以及时阻断危险行为。

10.4 练习题

1. ARP 协议属于 （a） 协议，它的作用是 （b）。

- （a）A. 物理层 B. 数据链路层 C. 网络层 D. 传输层
- （b）A. 实现 MAC 地址与主机名之间的映射 B. 实现 IP 地址与 MAC 地址之间的变换
- C. 实现 IP 地址与端口号之间的映射 D. 实现应用进程与物理地址之间的变换

解析：

ARP 是网络层协议，它的作用是实现 IP 地址与 MAC 地址之间的变换。IP 地址是分配给主机的逻辑地址，在互联网中表示唯一的主机。另外，每个主机还有一个物理地址，通常用网卡地址（MAC 地址）来表示主机的物理地址。

物理地址和逻辑地址的区别可以从两个角度来看：从网络互联的角度看，逻辑地址在整个互联网络中有效，而物理地址只是在子网内部有效；从网络协议分层的角度看，逻辑地址由 Internet 层使用，而物理地址由子网访问子层（具体地说，就是数据链路层）使用。

由于有两种主机地址，因而需要一种映像关系把这两种地址对应起来。在 Internet 中用地址分解协议（Address Resolution Protocol，ARP）来实现逻辑地址到物理地址的映像。ARP 分组的格式如表 10-3 所示。

表 10-3 ARP 分组的格式

硬件类型		协议类型
硬件地址长度	协议地址长度	操作类型
发送结点硬件地址		
发送结点协议地址		
目标结点硬件地址		
目标结点协议地址		

各字段的含义解释如下：

- 硬件类型：网络接口硬件的类型，以太网中此值为 1。
- 协议类型：发送方使用的协议，0800H 表示 IP 协议。
- 硬件地址长度：以太网中，地址长度为 6 字节。
- 协议地址长度：IP 协议中，地址长度为 4 字节。
- 操作类型：1—ARP 请求，2—ARP 响应，3—RARP 请求，4—RARP 响应。

通常 Internet 应用程序把要发送的报文交给 IP 协议，IP 当然知道接收方的逻辑地址（否则就不能通信了），但不一定知道接收方的物理地址。在把 IP 分组向下传送给本地数据链路实体之前可以用两种方法得到目标物理地址：

（1）查询本地内存中的 ARP 地址映像表，其逻辑结构如表 10-4 所示。可以看出，这是 IP 地址和以太网地址的对照表。

（2）如果在 ARP 表中查不到，就广播一个 ARP 请求分组，这种分组经过路由器进一步转发，可以到达所有连网的主机。它的含义是“如果你的 IP 地址是这个分组中的目标结点协议地址，请回答你的物理地址是什么”。收到该分组的主机一方面可以用分组中的两个源地址更新自己的 ARP 地址映像表，另一方面用自己的 IP 地址与目标结点协议地址字段比较。若相符，则发回一个 ARP 响应分组，向发送方报告自己的硬件地址；若不相符，则不予回答。

表 10-4 IP 地址和以太网地址的对照表

IP 地址	以太网地址
130.130.87.1	08 00 39 00 29 D4
129.129.52.3	08 00 5A 21 17 22
192.192.30.5	08 00 10 99 A1 44

答案：C、B

2. 如果使用大量的连接请求攻击计算机，使得所有可用的系统资源都被消耗殆尽，最终计算机无法再处理合法用户的请求，这种手段属于_____攻击。

- A. 拒绝服务 B. 口令入侵 C. 网络监听 D. IP 欺骗

解析：

网络攻击的主要手段包括口令入侵、放置特洛伊木马程序、拒绝服务（DOS）攻击、端口扫描、网络监听、欺骗攻击和电子邮件攻击等。

口令入侵是指使用某些合法用户的账号和口令登录到目的主机，然后再实施攻击活动。

特洛伊木马（Trojans）程序常被伪装成工具程序或游戏，一旦用户打开了带有特洛伊木马程序的邮件附件，或者从网上直接下载，或者执行了这些程序之后，当用户连接到互联网上时，这个程序就会向黑客通知用户的 IP 地址及被预先设定的端口。

拒绝服务（DOS）攻击目的是使计算机或网络无法提供正常的服务。最常见的拒绝服务攻击有网络带宽攻击和连通性攻击。带宽攻击是指以极大的通信量冲击网络，使得所有可用网络资源都被消耗殆尽，最后导致合法的用户请求无法通过。连通性攻击是指用大量的连接请求冲击计算机，使得所有可用的操作系统资源都被消耗殆尽，最终计算机无法再处理合法用户的请求。

端口扫描就是利用 Socket 编程与目标主机的某些端口建立 TCP 连接、进行传输协议的验证等，从而可知目标主机的扫描端口是否处于激活状态、主机提供了哪些服务、提供的服务中是否含有某些缺陷等。

网络监听是主机的一种工作模式，在这种模式下，主机可以接收到本网段在同一条物理通道上传输的所有信息。使用网络监听工具可以轻而易举地截取包括口令和账号在内的信息资料。

欺骗攻击是攻击者创造一个易于误解的上下文环境，以诱使受攻击者进入且作出缺乏安全考虑的决策。IP 欺骗是欺骗攻击的一种，IP 欺骗实现的过程是：使得被信任的主机丧失工作能力，同时采样目标主机发出的 TCP 序列号，猜测出它的数据序列号。然后伪装成被信任的主机，同时建立起与目标主机基于地址验证的应用连接。如果成功，黑客可以使用一种简单的命令放置一个系统后门，以进行非授权操作。

答案：A

3. ARP 攻击造成网络无法跨网段通信的原因是_____。

- A. 发送大量 ARP 报文造成网络拥塞
B. 伪造网关 ARP 报文使得数据包无法发送到网关

第 10 小时

- C. ARP 攻击破坏了网络的物理连通性
- D. ARP 攻击破坏了网关设备

解析:

ARP 攻击 (ARP 欺骗) 是欺骗攻击的一种, 通过伪造 IP 地址和 MAC 地址, 能够在网络中产生大量的 ARP 通信量使网络阻塞。如果伪造网关的 IP 地址和 MAC 地址, 则所有发往网关的 IP 包将因为 MAC 地址错误而无法到达网关 (ARP 攻击一般会将 MAC 地址改为发起 ARP 攻击的主机地址), 造成无法跨网段通信。

处理 ARP 攻击的方法是先断开 ARP 攻击主机的网络连接, 然后用 “arp-d” 命令清除受攻击影响的 ARP 缓存。

答案: B

4. 下列选项中, 防范网络监听最有效的方法是_____。
- A. 安装防火墙
 - B. 采用无线网络传输
 - C. 数据加密
 - D. 漏洞扫描

解析:

网络监听是主机的一种工作模式, 在这种模式下, 主机可以接收到本网段在同一条物理通道上传输的所有信息。使用网络监听工具可以轻而易举地截取包括口令和账号在内的信息资料。采用数据加密的方式保护包括口令和账号在内的信息资料, 使得即使网络监听获取密文后也无法解密成明文, 是对付网络监听的有效手段。

答案: C

5. 网络安全体系设计可以从物理线路安全、网络安全、系统安全、应用安全等方面来进行, 其中, 数据库容灾属于_____。
- A. 物理线路安全和网络安全
 - B. 应用安全和网络安全
 - C. 系统安全和网络安全
 - D. 系统安全和应用安全

解析:

网络安全体系设计是逻辑设计工作的重要内容之一, 数据库容灾属于系统安全和应用安全考虑范畴。

答案: D

6. 下面关于漏洞扫描系统的叙述, 错误的是_____。
- A. 漏洞扫描系统是一种自动检测目标主机安全弱点的程序
 - B. 黑客利用漏洞扫描系统可以发现目标主机的安全漏洞
 - C. 漏洞扫描系统可以用于发现网络入侵者
 - D. 漏洞扫描系统的实现依赖于系统漏洞库的完善

解析:

漏洞扫描系统是一种自动检测目标主机安全弱点的程序, 漏洞扫描系统的原理是根据系统漏洞库对系统可能存在的漏洞进行一一验证。黑客利用漏洞扫描系统可以发现目标主机的安全漏洞, 从

而有针对性地对系统发起攻击；系统管理员利用漏洞扫描系统可以查找系统中存在的漏洞进行修补从而提高系统的可靠性。漏洞扫描系统不能用于发现网络入侵者，用于检测网络入侵者的系统称为入侵检测系统。

答案：C

7. 网络安全包含了网络信息的可用性、保密性、完整性和网络通信对象的真实性。其中，数字签名是对_____的保护。

- A. 可用性 B. 保密性 C. 连通性 D. 真实性

解析：

数字签名（Digital Signature）技术是不对称加密算法的典型应用。数字签名的应用过程是：数据源发送方使用自己的私钥对数据校验和或其他与数据内容有关的变量进行加密处理，完成对数据的合法“签名”；数据接收方则利用对方的公钥来解读收到的“数字签名”，并将解读结果用于对数据完整性的检验，以确认签名的合法性。数字签名技术是在网络系统虚拟环境中确认身份的重要技术，完全可以代替现实过程中的“亲笔签字”，在技术和法律上有保证，可见数字签名是对签名真实性的保护。

答案：D

8. 计算机感染特洛伊木马后的典型现象是_____。

- A. 程序异常退出 B. 有未知程序试图建立网络连接
C. 邮箱被垃圾邮件填满 D. Windows 系统黑屏

解析：

特洛伊木马是一种通过网络传播的病毒，分为客户端和服务端两部分。服务端于被感染的计算机，特洛伊木马服务端运行后会试图建立网络连接，所以计算机感染特洛伊木马后的典型现象是有未知程序试图建立网络连接。

答案：B

第11小时

标准化和软件知识产权基础知识

11.0 章节考点分析

第 11 小时主要学习的是标准化和软件知识产权基础知识，包括标准化基础知识和知识产权基础知识。本小时的内容偏重于管理方面的知识，属于基础概念，考查的知识点多数参照教材。本小时的架构如图 11-1 所示。

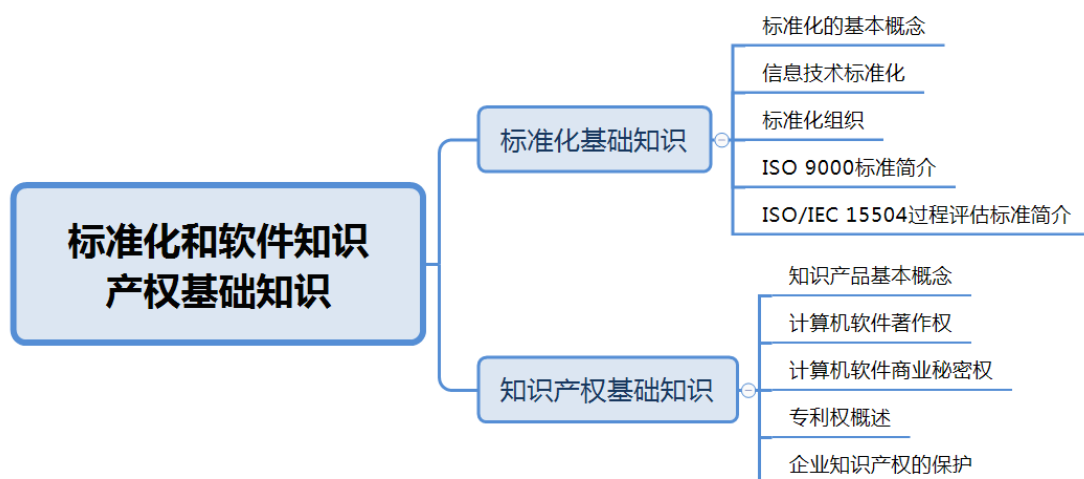


图 11-1 第 11 小时架构图



【导读小贴士】

标准化和软件知识产权基础知识均属于基本常识，教材中涉及的知识量较少，可结合实际经验理解并掌握。

11.1 标准化基础知识

1. 标准化的基本概念

- (1) 标准是规定，标准化是过程。
- (2) 标准的对象：具体对象研究制定标准的具体事物，总体对象研究全体构成整体的共同属性、本质和普遍规律。
- (3) 标准化过程模式如表 11-1 所示。

表 11-1 标准化过程模式

步骤	包括的活动	要点
标准的制定/产生	调查研究、制订计划（立项）、起草标准、征求意见、审查、批准发布	我国依据 ISO 和 IEC 的导则性文件制定了标准过程的规范，即标准的标准
标准的实施	没有统一的规定，一般包括标准的宣传、贯彻执行和监督检查	《标准化法》《国家标准管理办法》规定了我国标准化工作的方针、政策、任务和标准化体制等，是我国推行标准化、实施标准化管理和监督的重要依据
标准的更新	标准复审、标准确认和标准修订	

- (4) 标准的分类如表 11-2 所示。

表 11-2 标准的分类

分类维度	分类	内容
根据适用范围分类	国际标准（IS）	国际标准化组织（ISO）、国际电工委员会（IEC）制定的标准，以及 ISO 出版的《国际标准题内关键词索引》所收录的其他国际组织制定的标准
	国家标准（NB）	中国（GB）：中华人民共和国国家技术监督局 美国（ANSI）：美国国家标准协会 英国（BS）：英国标准学会 日本（JIS）：日本工业标准调查会
	区域标准	按地理、经济或政治划分的某一区域的标准化团体：太平洋地区标准会议（PASC）

续表

分类维度	分类	内容
		欧洲标准化委员会（CEN） 亚洲标准咨询委员会（ASAC） 非洲地区标准化组织（ARSO）
	行业标准	行业机构、学术团体或国防机构 美国电气和电子工程师学会标准（IEEE） 中华人民共和国国家军用标准（GJB） 美国国防部标准（DOD-STD） 美国军用标准（MIL-S）
	企业（机构）标准	供公司内部使用
	项目（课题）标准	计算机集成制造系统（CIMS）的软件工程规范
根据标准性质分类	技术标准	按其标准化对象特征和作用划分 按其标准化对象在生产流程中的作用划分 按标准的强制程度划分 按标准在企业中的适用范围划分
	管理标准	可以分为技术管理标准、生产组织标准、经济管理标准、行政管理标准、业务管理标准和工作标准等
	工作标准	分为工作程序标准和管理工作标准
根据标准化的对象和作用分类	基础标准、产品标准、方法标准、安全标准、卫生标准、环境保护标准和服务标准等	
根据法律的约束性分类	强制标准	任何更改或降低产生的不良后果或事故，可以由司法机构追究刑事责任，我国强制性标准与 WTO/TBT 规定衔接，其范围限制在国家安全、防止欺诈行为、保护人身健康与安全、保护动植物的生命和健康及保护环境方面
	推荐标准	属于指导性标准，鼓励采用

《中华人民共和国标准化法》将我国标准划分为国家标准、行业标准、地方标准和企业标准。

（5）标准的代号和编号如表 11-3 所示。

表 11-3 标准的代号和编号

标准	编号规则
国际标准	ISO+标准号+[杠+分标准号]+冒号+发布年号，方括号中的内容可有可无
国家标准	国家标准的代号+标准发布顺序号-标准发布年代号（4 位） 国家标准的代号：强制性国家标准 GB，推荐性国家标准 GB/T
行业标准	行业标准的代号（强制或推荐）+标准发布顺序号-标准发布年代号（4 位） 已正式公布的行业标准代号：QJ（航天）、SJ（电子）、JB（机械）、JR（金融系统）
地方标准	DB+行政区域代码（前两位）/（强制或推荐）+地方标准发布顺序号-标准发布年代号（4 位）
企业标准	Q/企业代号+标准发布顺序号-标准发布年代号（4 位）

(6) 采用国际标准和国外先进标准的原则：①我国国民经济发展的需要；②先使用、再验证、后修补；③促进产品质量水平的提高；④紧密结合我国的实际条件，符合政策法规要求；⑤基础标准及方法论标准先行采用；⑥应符合《技术引进和设备进口标准化审查管理办法（试行）》；⑦国际标准缺失或不满足，采用国外先进标准。

(7) 国际标准和国外先进标准的采用程度如表 11-4 所示。

表 11-4 国际标准和国外先进标准不同类型的采用程度

类型	内容	表示方法
等同采用	国家标准和国际标准之间的差异仅限于没有编辑性修改（不改变技术内容）	GB XXXX-XXXX（idt ISO XXXX-XXXX）
等效采用	技术上存在较小偏差，编辑和编写方法不完全相同	GB XXXX-XXXX（eqv ISO XXXX-XXXX）
非等效采用	技术上存在重大偏差	GB XXXX-XXXX（neq ISO XXXX-XXXX）

2. 信息技术标准化

(1) 信息技术标准化包括信息编码标准化（ASCII）、汉字编码标准化（字母+数字+符号）、软件工程标准化。

(2) 汉字编码的方法包括数字编码（如电报码、四角号码）、拼音编码和字形编码。

(3) 我国在汉字编码标准化方面取得的突出成就就是信息交换用汉字编码字符集国家标准的制定。

(4) 汉字信息处理标准化包括汉字编码标准化、汉字键盘输入标准化、汉字文字识别输入和语音识别输入标准化、汉字输出字体和质量标准化、汉字属性和汉语词语标准化等。

(5) 软件工程标准化主要包括过程标准、产品标准、专业标准、记法标准、开发规范、文件规范、维护规范和质量规范等。

(6) 计算机与信息处理标准化技术委员会下设 13 个分会，其中程序设计语言分技术委员会和软件工程技术委员会与软件相关。

3. 标准化组织

(1) 世界上两个最大、最具权威的国际标准化组织分别为 ISO 和 IEC。

(2) 国际标准化组织（ISO）具有非政府性，世界最大，由各国标准化团体组成，官方语言为英文、法文、俄文，会址设在日内瓦，最高的权力机构是成员全体大会。

(3) 国际标准化组织（ISO）成员分为正式成员和通信成员。正式成员是由各国最具代表性的标准化机构代表其国家和地区参加，而且每个国家有且只有一个组织；通信成员只由不具备标准化机构且一般不参与 ISO 技术工作的成员国参与。

(4) 国际电工委员会（IEC）具有非政府性，是最早的世界范围的资源认证机构，宗旨是促进国家或国际间的贸易自由，会址设在日内瓦，最高的权力机构是理事会。

（5）区域标准化组织，参与组织的机构有政府性，也有非政府性，维护该地区国家的利益，如国家电信联盟（ITU）。

4. ISO 9000 标注简介

（1）ISO 9000 是一系列标准的统称，该标准族由 ISO 900X 构成，标准名称及重点如表 11-5 所示。

（2）ISO 9000：2000 族标准。

表 11-5 标准名称及重点

类型	标准名称	重点
核心标准	《质量管理体系 基础和术语》 ISO 9000：2000	明确了一个组织在实施质量管理中必须遵循的八项质量管理原则
	《质量管理体系 要求》 ISO 9001：2000	该标准是用于第三方认证的唯一质量管理体系要求标准，分为五大模块
	《质量管理体系 业绩改进指南》 ISO 9004：2000	当管理者希望超越 ISO 9001 标准的最低要求，追求业绩改进时使用
	《质量管理体系和环境管理体系审核指南》 ISO 9011：2001	对审核机构提出要求

（3）ISO 9000 的 2000 系列标准的八项质量管理原则：以顾客为中心、领导作用、全员参与、过程方法、管理的系统方法、持续改进、基于事实的决策方法、互利的供方关系。

5. ISO/IEC 15504 过程评估标准简介

（1）ISO/IEC 15504 提供了一个软件过程评估的框架，可以用于任何软件活动。

（2）ISO/IEC 15504 包含的主要内容：概念和绪论指南、过程和过程能力参考模型、实施评估、评估实施指南、评估模型和标志指南、评估师能力指南、过程改进应用指南、确定供方能力应用指南、词汇九个部分。

11.2 知识产权基础知识

1. 知识产权基本概念

（1）知识产权是指民事权利主体（公民、法人）基于创造性的智力成果。

（2）根据国际公约，知识产权的保护对象包括：①文学、艺术和科学作品；②表演艺术家的表演，以及唱片和广播节目；③人类一切活动领域的发明；④科学发现；⑤工业品外观设计；⑥商标、服务标记、商业名称和标志；⑦制止不正当竞争；⑧在工业、科学、文学艺术领域内由于智力创造活动而产生的一切其他权利。

（3）根据世贸协议，知识产权还包括“未披露过的信息专有权”，即商业秘密。

(4) 知识产权的分类如图 11-2 所示。

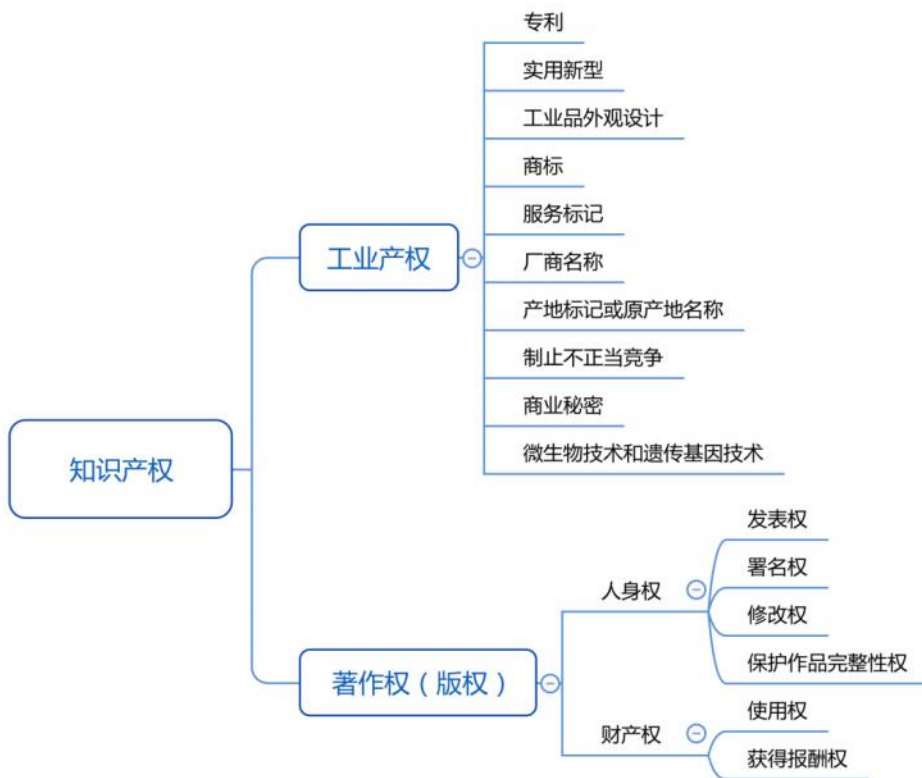


图 11-2 知识产权的分类

(5) 计算机软件和实用艺术品受著作权保护的同时，权利人还可以通过申请发明专利和外观设计专利获得专利权（工业产权）。

(6) 知识产权的特点：无形性（智力成果）、双重性（多权并存）、确认性（依法审查）、独占性（权利人的相对性）、地域性、时间性。

2. 计算机软件著作权

(1) 获取软件著作权主体资格的途径如下：

- 1) 公民：独立开发、委托开发、转让、合作开发、继承（署名权除外）。
- 2) 法人：组织并提供创作条件；委托、转让等合同关系；主体变更。
- 3) 其他组织。

(2) 计算机软件著作权的客体：计算机程序及其有关文档。

(3) 计算机软件受著作权法保护的的条件：

- 1) 独立创作：非复制和抄袭。
- 2) 可被感知：创作思想在固定载体中的实际表达。

3) 逻辑合理。

(4) 计算机软件著作权的权利如图 11-3 所示。



图 11-3 计算机软件著作权的权利

(5) 软件合法持有人的权利：安装、备份和改进。

(6) 计算机软件著作权的行使如图 11-4 所示。

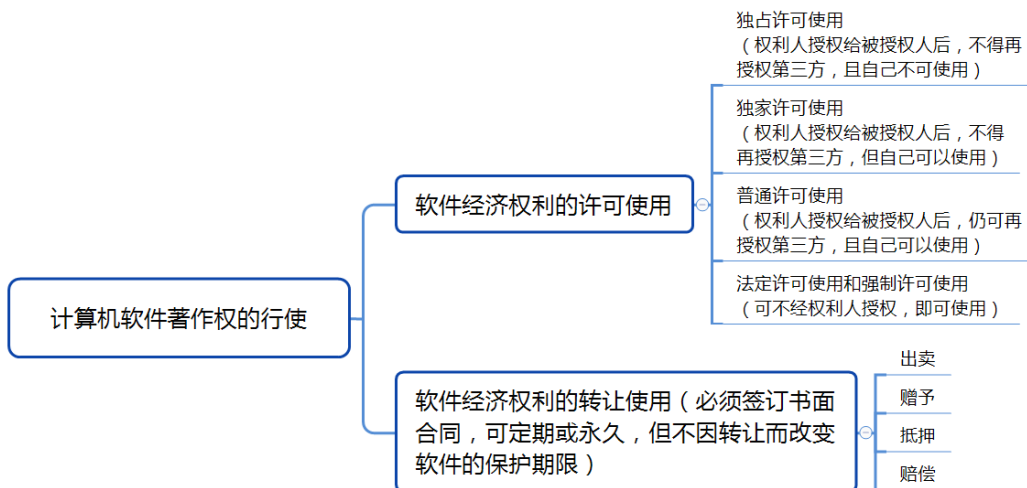


图 11-4 计算机软件著作权的行使

(7) 计算机软件著作权自软件开发完成之日起产生，保护期为 50 年，期满后除身份权外，其

余权利终止，进入公有领域可被无偿使用。

(8) 同时满足下列三个条件时，属于非职务软件作品，雇员拥有软件著作权。

- 1) 不是本职工作的成果。
- 2) 与本职工作无直接关系。
- 3) 开发成果未使用单位的物质技术条件。

(9) 合作开发无书面协议时，当软件可分割，各自享用该部分软件著作权；软件不可分割，共同享有著作权。任何一方不得单独行使软件的转让权。

(10) 委托开发或接受任务开发无书面协议，软件著作权归受托人（实际开发人）所有。

(11) 计算机软件著作权侵权行为的认定：

- 1) 未经著作权人同意而发表或登记其软件作品。
- 2) 将他人软件当作自己的作品发表或登记。
- 3) 未经合作者同意将共同开发的软件当作自己的作品发表或登记。
- 4) 在他人开发的软件上署名或更改他人署名。
- 5) 未经著作权人或其合法受让者许可，修改、翻译、复制或部分复制、向公众发行或出租、办理权利许可或转让或通过网络传播其作品。

(12) 以学习和研究软件内含的设计思想和原理为由，通过安装、显示、传输或存储软件的方式使用软件时，可以不经许可，不支付报酬。

(13) 计算机软件著作权区别于其他著作权的特点：技术性、依赖性、多样性和运行性。

(14) 计算机软件著作权侵权的法律责任包括民事责任、行政责任和刑事责任（销售复制品）。

3. 计算机软件的商业秘密

- (1) 商业秘密的构成条件：①未公开的；②具有经济效益的；③采取了保密措施的。
- (2) 侵犯计算机软件商业秘密的表现形式：非法获得、非法使用、违约和明知故犯。

4. 专利权概述

- (1) 我国专利法主要保护的对象：发明、实用新型和外观设计。
- (2) 授予专利权的条件：新颖性、创造性和实用性。
- (3) 专利申请日（关键日）是专利局或专利局指定的专利申请受理代办处收到完整专利申请文件的日期，邮寄以寄出的邮戳日为准。
- (4) 专利权的保护期限：发明专利权自申请日起 20 年，实用新型和外观设计自申请日起 10 年。

5. 企业知识产权的保护

(1) 企业保护软件知识产权成果的途径：①明确软件知识产权的归属；②及时对软件技术秘密采取保护措施；③依靠专利保护新技术和新产品；④软件产品进入市场之前的商标权和商业秘密保护权；⑤软件产品进入市场之前申请登记软件著作权。

11.3 练习题

1. 我国著作权法中，_____是指同一概念。

- A. 出版权和版权
- B. 著作权和版权
- C. 作者权和专有权
- D. 发行权和版权

解析：

著作权又称版权，前者属于中国大陆法系，后者源于英美法系。我国不排斥英美版权法模式，并在《著作权法》中对上述两种说法进行了释义。

答案：B

2. 由我国工业和信息化部批准发布，在信息化行业范围内统一使用的标准，称为_____。

- A. 地方标准
- B. 部门标准
- C. 行业标准
- D. 企业标准

解析：

《中华人民共和国标准化法》将我国标准划分为国家标准、行业标准、地方标准和企业标准，B 选项是法律法规体系分类维度的混淆。

答案：C

第12小时

软件系统分析与设计

12.0 章节考点分析

第12小时主要学习的是软件系统分析与设计基础知识，包括当前软件设计开发中的主流技术和工程应用背景，以基本概念和理论知识为主。本小时偏重于技术方面的知识，属于需要理解和掌握的内容，但考查的知识点仍以教材内容为主。本小时的架构如图12-1所示。

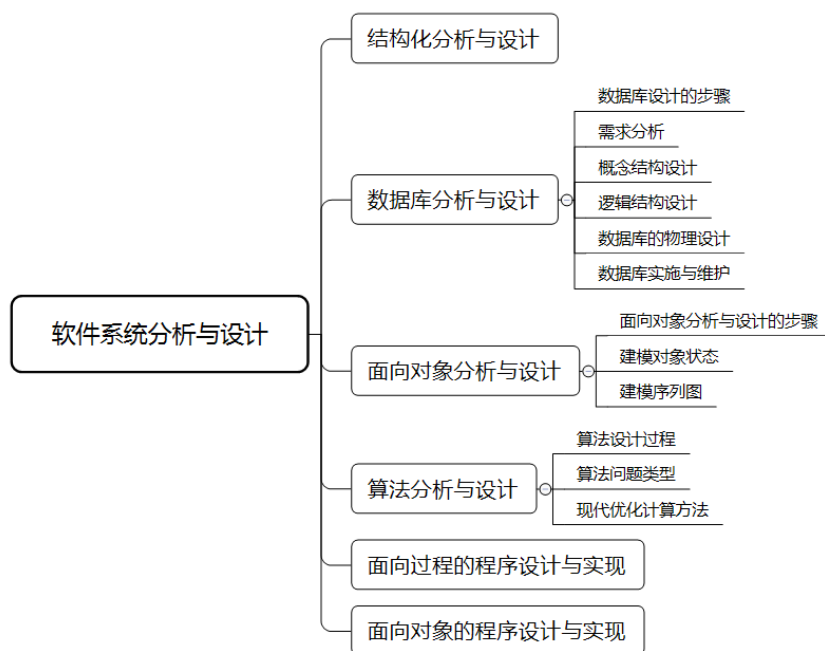


图 12-1 第 12 小时架构图



【导读小贴士】

软件系统分析与设计基础知识属于软件设计的基本知识,由于算法的内容涉及较多,要求考生对 C 语言或 Java 语言有一定的掌握,不排除下午简答题出现与教材类似的案例,要求考生进行图的设计或编程。本小时仅对书内的基础概念进行归纳总结,考生在复习考试时,建议结合书中本小时的通篇案例(图书馆管理系统)进行理解与掌握。

12.1 结构化分析与设计

(1) 结构化分析的对象包括数据(实体对象的属性和关系)和处理(对信息的加工和处理)。

(2) 数据流图(DFD)是面向数据流建模的工具。

(3) 进行结构化分析的步骤:①确定系统边界,绘制系统环境图;②绘制各层数据流图(自顶向下);③定义数据字典;④定义加工(处理)说明;⑤将图、字典和加工组成分析模型。

(4) 系统总体设计模型反映模块间的调用关系,可以采用层次图、HIPO 图和机构图进行表达。

(5) 数据流图分为变换型数据流图和事务型数据流图。事务型数据流图的处理为条件判断式,根据不同输入数据的类型对应不同的处理动作。

(6) 详细设计可以采用程序流程图、N-S 图、PAS 图和 PDL 语言等工具进行表达。

(7) 结构化分析的最终结果:数据流图、数据字典和加工处理说明。

12.2 数据库分析与设计

1. 数据库设计的步骤

数据库应用系统的生命周期分为六个阶段:数据库规划、需求描述与分析、数据库设计与应用程序设计、实现、测试、运行与维护。其中:

(1) 需求描述与分析是以用户的角度进行的。

(2) 应用程序设计包括事务设计和用户界面设计。

(3) DDL(数据定义语言)用于建立数据库。

2. 数据库设计的四个主要阶段

用户需求分析、概念结构设计、逻辑结构设计、物理结构设计。

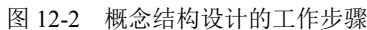
(1) 用户需求分析。在需求调研的过程中,需要获得用户对系统的要求包括信息要求(保存哪些数据)、处理要求(实现什么功能)、系统要求(安全要求等)。

数据字典是各类数据描述的集合,描述的是元数据(描述数据的数据),而不是数据本身。数

需求分析阶段的成果是系统需求说明书。

- 1) 概念结构设计的目标是产生反映系统信息需求的数据库概念结构,即概念模型。
- 2) 概念结构设计的策略通常有以下四种:自顶向下、自底向上、逐步扩张和混合策略。

概念结构设计的工作步骤如图 12-2 所示。



(3) 逻辑结构设计。逻辑结构设计包括模式初始设计、子模式设计、应用程序设计、模式评价、模式求精。逻辑结构设计即是在概念结构设计的基础上进行数据模型设计,可以分为层次模型、网状模型和关系模型。逻辑结构设计的工作步骤如图 12-3 所示。

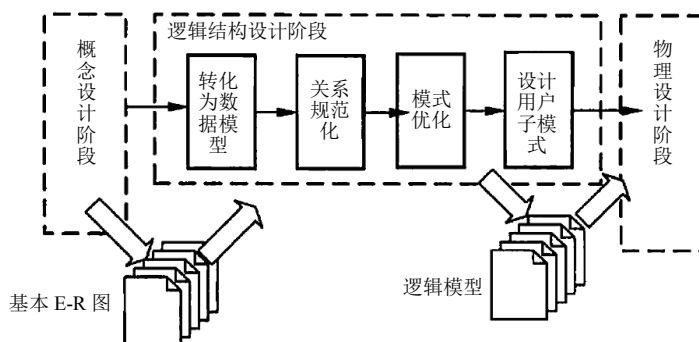


图 12-3 逻辑结构设计的工作步骤

E-R 图的三种联系：一对一、一对多、多对多。

关系模型规范化的步骤：确定依赖关系、确定范式、关系模型分解、关系模型的评价及修正。

(4) 数据库的物理设计。数据库物理设计的工作步骤如图 12-4 所示。

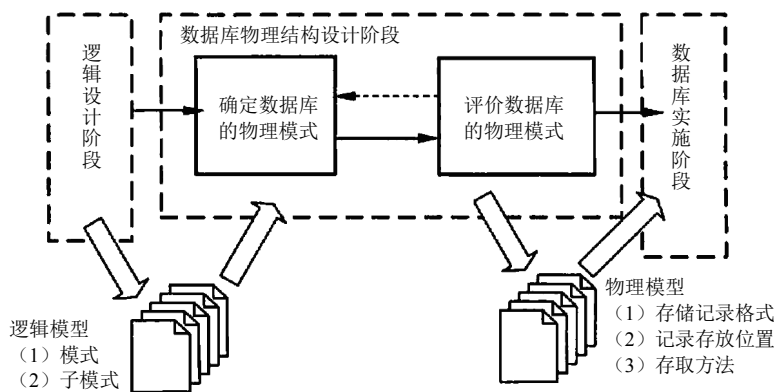


图 12-4 数据库物理设计的工作步骤

物理设计包括确定数据分布、存储结构和访问方式。综合考虑提高数据的访问速度，本阶段需要确定数据库文件的索引字段和索引类型。数据库的物理结构主要由存储记录格式、记录在物理设备上的安排及访问路径（存取方法）构成。

建立聚簇（按属性或属性组）索引的原则如下：

- 1) 聚簇码值相对稳定。
- 2) 表主要用于查询，而且主要用于聚簇码的访问和连接。
- 3) 对应每个聚簇码值的平均元组，既不能太多也不能太少。

确定索引的一般顺序：①确定关系的存储结构：无序或有序（聚簇）存放；②确定不宜建立索引的属性或表；③确定宜建立索引的属性。

3. 数据库的实施与维护

(1) 数据库实施阶段的工作步骤如图 12-5 所示。

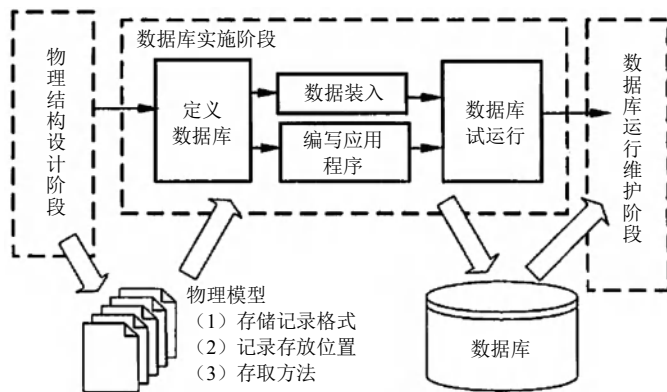


图 12-5 数据库实施阶段的工作步骤

- (2) 在 Oracle 系统中，数据库逻辑结果的描述包括表空间、段、范围和数据块。
- (3) 数据的完整性是指有效性、正确性和一致性。
- (4) 系统可对用户的数据操作进行两方面的控制：用户授权（身份认证）、不同用户不同权限。
- (5) 数据加载前，必须对数据进行整理，建立校验规范和程序，排除不合格数据。
- (6) 数据库维护工作的主要内容包括对数据库的监测和性能改善、故障恢复、数据库的重组（不改变数据库逻辑和物理结构，类似于碎片整理，清理指针及废弃空间）和重构（谨慎执行）。

12.3 面向对象分析与设计

1. 面向对象分析与设计的步骤

面向对象分析包括四个活动：建模系统功能、定义领域模型、定义交互行为和状态、定义设计类图。

通过用例建模系统功能的步骤：①确定参与者；②确定需求用例；③构造用例模型；④记录需求用例描述。

2. 建模对象状态

建模状态图应遵循的指导原则如下：①状态名称简单但具有描述性；②避免黑洞（有进无出）；③避免奇迹（有出无进）；④符合状态需对子状态集进行建模；⑤为复杂的实体创建分层的状态图。

3. 建模序列图

系统序列图描述决策和系统在用例场景下的交互。

12.4 算法分析与设计

1. 算法设计过程

算法设计的主要步骤如图 12-6 所示。

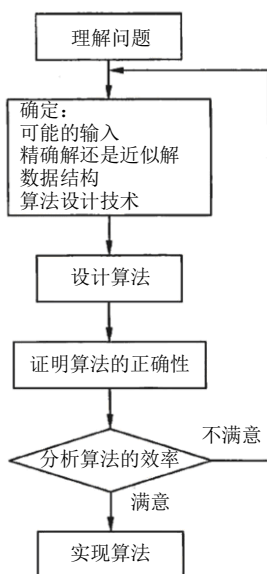


图 12-6 算法设计的主要步骤

2. 算法问题类型

算法问题类型如表 12-2 所示。

表 12-2 算法问题类型

问题	要点
查找	没有一种算法对任何情况都适用
排序	排序的一个主要目的是进行快速查找
图	举例：TSP 问题、最大团/独立集问题、图着色问题、哈密尔顿回路问题、顶点覆盖问题和最长路径问题
组合	一般是最优化问题，举例：0-1 背包问题、TSP 问题和整数线性规划
几何	举例：最近点对问题、凸包问题

3. 现代优化计算方法

现代优化计算方法如表 12-3 所示。

表 12-3 现代优化计算方法

算法	要点
禁忌搜索算法	全局逐步寻优算法
模拟退火算法	全局最优算法
遗传算法	选择、较差和搬移构成了遗传算法的基本操作
蚁群优化算法	分布式的智能模拟算法，间接异步互联
人工神经网络	主要用于网络连接拓扑结构、神经元的特征和学习规则，模型包括前向网络、反馈网络（无向）

12.5 面向过程的程序设计 with 实现

- （1）变量的本质是程序中用来存放数据的一段存储空间，通常其对应的存储空间为内存区域。
- （2）变量具有类型、值、地址、作用域和生存期等属性。
- （3）存储变量地址的变量称为指针变量。
- （4）当一个指针变量指向仍为指针变量时，称为多级变量，级数越多，数据访问效率越低。

12.6 面向对象的程序设计 with 实现

- 当设计与实现面向对象的程序时，需要注意以下四点：
- （1）能够根据设计模式的名称画出其对应的类图。
 - （2）理解类图中每一个类的作用与功能。
 - （3）能够将现实问题中所描述的各种职责映射到类图中具体的类。
 - （4）能够使用一种面向对象语言实现设计。

12.7 练习题

1. 在采用结构化开发方法进行软件开发时，设计阶段的接口设计主要依据需求分析阶段的 (a)。接口设计的任务主要是 (b)。
- (a) A. 数据流图 B. E-R 图 C. 状态—迁移图 D. 加工规格说明
- (b) A. 定义软件的主要结构元素及其之间的关系
- B. 确定软件涉及的文件系统的结构及数据库的表结构
- C. 描述软件与外部环境之间的交互关系，软件内模块之间的调用关系
- D. 确定软件各个模块内部的算法和数据结构

第 12 小时

解析:

接口设计的主要依据是数据流图。接口设计的任务主要是描述软件与外部环境之间的交互关系,以及软件内模块之间的调用关系。定义软件的主要结构元素及其之间的关系是架构阶段的任务;确定软件涉及的文件系统的结构及数据库的表结构是数据存储设计阶段的任务;确定软件各个模块内部的算法和数据结构是详细设计阶段的任务。

答案: A、C

2. 在设计软件的模块结构时, _____ 不能改进设计质量。

- 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

解析:

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以下原则:

(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的页数在 1~2 页,或者实现代码的行数在 50~200 行,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该模块直接调用的上级模块的个数;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,中间扇出比较少,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。宽度表示软件结构中同一个层次上的模块总数的最大值,一般来说,宽度越大的系统越复杂,对宽度影响最大的因素是模块的扇出。在设计系统时,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。

答案: D

3. 模块 A、B、C 有相同的程序块,块内的语句之间没有任何联系。现把该程序块取出来,形成新的模块 D,则模块 D 的内聚类型为 (a) 内聚。以下关于该内聚类型的叙述中,不正确的是 (b)。

- (a) A. 巧合
- B. 逻辑
- C. 时间
- D. 过程
- (b) 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

解析:

内聚是指模块内部各元素之间联系的紧密程度。模块的内聚类型分为七种,根据内聚度从低到

高的排序如下：

- (1) 偶然内聚。指一个模块内的各个处理元素之间没有任何联系。
- (2) 逻辑内聚。指模块内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。
- (3) 时间内聚。指把需要同时执行的动作组合在一起。
- (4) 通信内聚。指模块内所有的处理元素都在同一个数据结构上操作，或者各个处理元素使用相同的输入数据或产生相同的输出数据。
- (5) 顺序内聚。指一个模块中各个处理元素都密切相关于同一功能且必须按照顺序执行，前一功能元素的输出是下一功能元素的输入。
- (6) 功能内聚。这是最强的内聚。指模块内所有元素共同完成一个功能，缺一不可。耦合性和内聚性是模块独立性的两个定性标准，将软件系统划分模块时，尽量做到高内聚、低耦合，提高模块的独立性。

答案：A、D

第13小时

新技术

13.0 章节考点分析

第13小时主要学习大数据、云计算、物联网、移动互联网、工业互联网、人工智能、区块链等内容。

根据考试大纲，本小时的知识点会涉及单选题型，约占2~5分。本小时的内容偏重于概念知识，根据以往全国计算机技术与软件专业技术资格（水平）考试的出题规律而言，考查概念的知识点多数参照教材，扩展内容较少。本小时的架构如图13-1所示。



图 13-1 第 13 小时架构图



【导读小贴士】

本小时涉及的内容是较为重要的知识点，在软件设计师考试中，一些选择题、案例分析题会来自本小时，建议除了理解的基础上掌握核心知识点外，还可以灵活应用。

13.1 大数据

1. 大数据的 5V 特点

大量（volume）、高速（velocity）、多样（variety）、价值（value）、真实性（veracity）（速记词：价多真高大）。

2. 大数据的五个环节

大数据需要经过五个环节：①数据准备；②存储管理；③计算处理；④数据分析；⑤知识展现。大数据技术框架如图 13-2 所示。

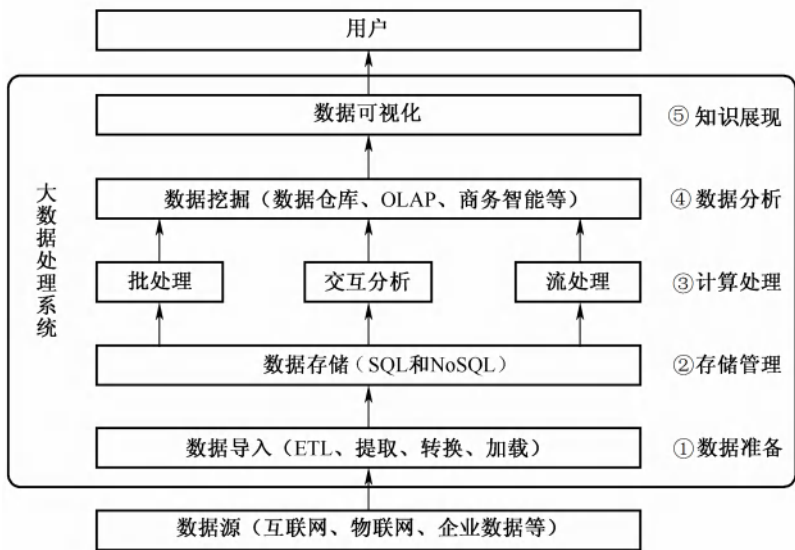


图 13-2 大数据技术框架

3. 大数据的关键技术

- （1）HDFS。能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。
- （2）HBase。不同于一般的关系数据库，是非结构化数据存储的数据库。

- (3) MapReduce。一种编程模型，主要思想为 Map（映射）和 Reduce（归约）。
- (4) Chukwa。用于监控大型分布式系统的数据收集系统。
- 开源大数据技术框架如图 13-3 所示。

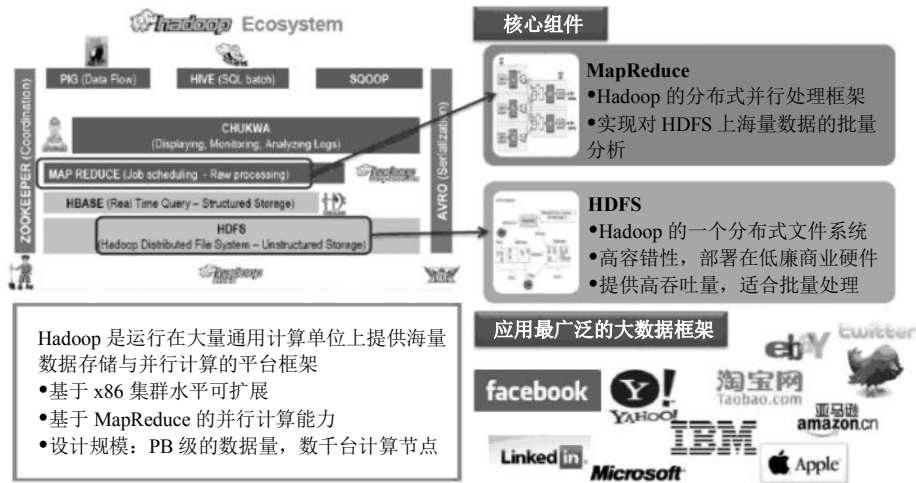


图 13-3 开源大数据技术框架

13.2 云计算

1. 云计算服务类型

云计算服务类型包括 IaaS（基础设施即服务）、PaaS（平台即服务）、SaaS（软件即服务）。

2. 云计算技术架构

云计算技术架构有四层，分别为设施层、资源层、资源控制层、服务层。云计算技术架构如图 13-4 所示。

13.3 物联网

1. 物联网架构

- (1) 感知层：负责信息采集和物物之间的信息传输。
- (2) 网络层：是物联网三层中标准化程度最高、产业化能力最强、最成熟的部分。
- (3) 应用层：实现应用。

物联网架构如图 13-5 所示。

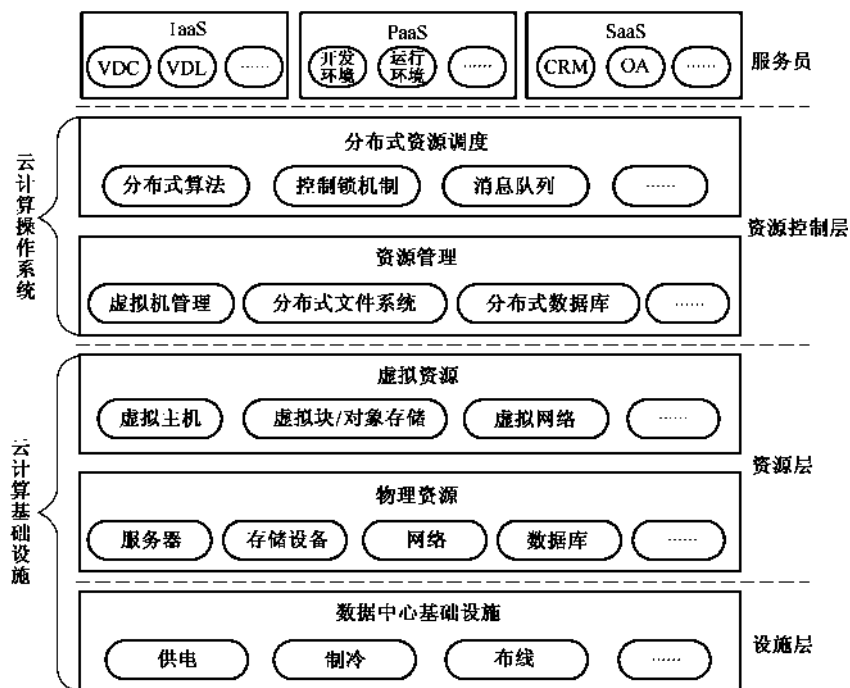


图 13-4 云计算技术架构

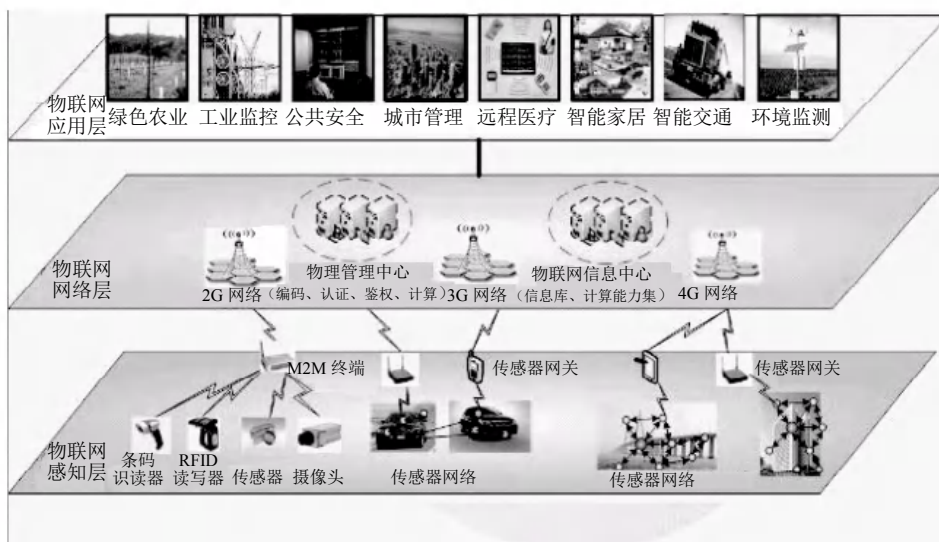


图 13-5 物联网架构

2. 物联网的关键技术

感知层作为物联网架构的基础层面，主要技术包括：产品和传感器（条码、RFID、传感器等）自动化识别技术、无线传输技术（WLAN、Bluetooth、ZigBee、UWB）、自组织组网技术、中间件技术。

13.4 移动互联网

1. 移动互联网

移动互联网等于移动通信网络加互联网的内容和应用。它不仅是互联网的延伸，而且是互联网的发展方向。

2. 移动互联网的特征

移动互联网不仅具有传统互联网应用的简单复制和移植，还具有以下新特征：接入移动性、时间碎片性、生活相关性、终端多样性。

3. 移动互联网的关键技术

（1）架构技术 SOA（Service Oriented Architect）。面向服务的架构，不涉及底层编程接口和通信模型，Web Service 是目前实现 SOA 的主要技术。

（2）页面展示技术 Web 2.0。严格来说不是一种技术，而是互联网思维模式。Web 1.0 与 Web 2.0 的区别如表 13-1 所示。

表 13-1 Web 1.0 与 Web 2.0 的区别

项目	Web 1.0	Web 2.0
页面风格	结构复杂，页面繁冗	页面简洁，风格流畅
个性化程度	垂直化，大众化	个性化，突出自我品牌
用户体验程度	低参与度，被动接受	高参与度，互动接受
通信程度	信息闭塞，知识程度低	信息灵通，知识程度高
感性程度	追求物质性价值	追求精神性价值
功能性	实用追求功能性利益	体验追求情感性利益

（3）页面展示技术 HTML 5。在原有 HTML 的基础上扩展了 API，最大的优势是在网页上直接调试和修改。

（4）主流开发平台 Android。特点是入门容易，因为 Android 的中间层多以 Java 实现，指令相对较少、开发相对简单，而且开发社群活跃，开发资源丰富。

（5）主流开发平台 iOS。一个非开源的操作系统，开发人员必须加入苹果开发者计划，需要付款以获得苹果的批准，开发语言是 Objective-C、C 和 C++，开发难度大于 Android。

（6）主流开发平台 Windows Phone。微软的一款手机操作系统，开发技术有 C、C++、C#等。

13.5 工业互联网

1. 工业互联网的定义

工业互联网是全球工业系统与高级计算、分析、感应技术和互联网连接融合的结果。它通过智能机器间的连接并最终将人机连接，结合软件和大数据分析，重构全球工业，激发生产力，让世界更美好、更快速、更安全、更清洁、更经济。

2. 工业互联网的实质

首先是全面互联，在全面互联的基础上，通过数据流动和分析，形成智能化变革，形成新的模式和新的业态。互联是基础，工业互联网使工业系统的各种元素互联起来，无论是机器、人还是系统。

互联解决了通信的基本，更重要的是数据端到端的流动和跨系统的流动，在数据流动技术上充分分析、建模。伯特认为智能化生产、网络化协同、个性化定制、服务化延伸是在互联的基础上，通过数据流动和分析，形成新的模式和新的业态。

这是工业互联网的基理，比现在的互联网更强调数据，更强调充分的连接，更强调数据的流动和集成及分析和建模，这和互联网是有所不同的。工业互联网的本质是要有数据的流动和分析。

工业互联网生态系统的持续拓展基于 Predix 和 Predix.io 两大基础。

13.6 人工智能

1. 定义

人工智能 (Artificial Intelligence, AI) 是研究使计算机来模拟人的某些思维过程和智能行为 (如学习、推理、思考、规划等) 的学科，主要包括计算机实现智能的原理、制造类似于人脑智能的计算机、使计算机能实现更高层次的应用。

人工智能涉及计算机科学、心理学、哲学和语言学等学科。可以说几乎涵盖了自然科学和社会科学的所有学科，其范围已远远超出了计算机科学的范畴，人工智能与思维科学的关系是实践和理论的关系，人工智能处于思维科学的技术应用层次，是它的一个应用分支。

从思维的观点看，人工智能不仅限于逻辑思维，还要考虑形象思维、灵感思维才能促进人工智能的突破性发展。数学常被认为是多种学科的基础科学，可以进入语言、思维领域，不仅在标准逻辑、模糊数学等范围发挥作用，还可以进入人工智能学科。人工智能学科也必须借用数学工具，互相促进彼此更快地发展。

2. 应用

人工智能的应用包括机器视觉、指纹识别、人脸识别、视网膜识别、虹膜识别、掌纹识别、专家系统、自动规划、智能搜索、定理证明、博弈、自动程序设计、智能控制、机器人学、语言和图像理解、遗传编程等。

3. 新一代人工智能的发展规划

2017 年 7 月 20 日, 国务院印发了新一代人工智能发展规划的通知(国发〔2017〕35 号)。主要内容如下:

(1) 战略态势。人工智能发展进入新阶段。经过 60 多年的演进, 特别是在移动互联网、大数据、超级计算、传感网、脑科学等新理论、新技术和经济社会发展强烈需求的共同驱动下, 人工智能加速发展, 呈现出深度学习、跨界融合、人机协同、群智开放、自主操控等新特征。大数据驱动知识学习、跨媒体协同处理、人机协同增强智能、群体集成智能、自主智能系统成为人工智能的发展重点, 受脑科学研究成果启发的类脑智能蓄势待发, 芯片化、硬件化、平台化趋势更加明显, 人工智能发展进入新阶段。当前, 新一代人工智能相关学科的发展、理论建模、技术创新、软硬件升级等整体推进, 正在引发链式突破, 推动经济社会各领域从数字化、网络化向智能化加速跃升。

(2) 新一代人工智能发展规划的基本原则:

1) 科技引领。把握世界人工智能的发展趋势, 突出研发部署前瞻性, 在重点前沿领域探索布局、长期支持, 力争在理论、方法、工具、系统等方面取得变革性、颠覆性突破, 全面增强人工智能原始创新能力, 加速构筑先发优势, 实现高端引领发展。

2) 系统布局。根据基础研究、技术研发、产业发展和行业应用的不同特点, 制定有针对性的系统发展策略。充分发挥社会主义制度集中力量办大事的优势, 推进项目、基地、人才统筹布局, 已部署的重大项目与新任务有机衔接, 当前急需与长远发展梯次接续, 创新能力建设、体制机制改革和政策环境营造协同发力。

3) 市场主导。遵循市场规律, 坚持应用导向, 突出企业在技术路线选择和行业产品标准制定中的主体作用, 加快人工智能科技成果的商业化应用, 形成竞争优势。把握好政府和市场分工, 更好地发挥政府在规划引导、政策支持、安全防范、市场监管、环境营造、伦理法规制定等方面的重要作用。

4) 开源开放。倡导开源共享理念, 促进“产学研用”各创新主体共创共享。遵循经济建设和国防建设协调发展的规律, 促进军民科技成果双向转化应用、军民创新资源共建共享, 形成全要素、多领域、高效益的军民深度融合发展新格局。积极参与人工智能全球研发和治理, 在全球范围内优化配置、创新资源。

(3) 战略目标。新一代人工智能的发展按照以下三步走:

第一步, 到 2020 年人工智能总体技术和应用与世界先进水平同步, 人工智能产业成为新的经济增长点, 人工智能技术应用成为改善民生的新途径, 有力支撑我国进入创新型国家行列和实现全面建成小康社会的奋斗目标。

新一代人工智能理论和技术取得重要进展。大数据智能、跨媒体智能、群体智能、混合增强智能、自主智能系统等基础理论和核心技术实现重要进展, 人工智能模型方法、核心器件、高端设备和基础软件等方面取得标志性成果。

人工智能产业的竞争力进入国际第一方阵。初步建成人工智能技术标准、服务体系和产业链, 培育若干全球领先的人工智能骨干企业, 人工智能核心产业规模超过 1500 亿元, 带动相关产

业规模超过 1 万亿元。

人工智能发展环境进一步优化。在重点领域全面展开创新应用，聚集起一批高水平的人才队伍和创新团队，部分领域的人工智能伦理规范和政策法规初步建立。

第二步，到 2025 年人工智能基础理论实现重大突破，部分技术与应用达到世界领先水平，人工智能成为带动我国产业升级和经济转型的主要动力，智能社会建设取得积极进展。

新一代人工智能理论与技术体系初步建立，具有自主学习能力的人工智能取得突破，在多领域取得引领性研究成果。

人工智能产业进入全球价值链高端。新一代人工智能在智能制造、智能医疗、智慧城市、智能农业、国防建设等领域得到广泛应用，人工智能核心产业的规模超过 4000 亿元，带动相关产业规模超过 5 万亿元。

初步建立人工智能法律法规、伦理规范和政策体系，形成人工智能安全评估和管控能力。

第三步，到 2030 年人工智能理论、技术与应用总体达到世界领先水平，成为世界主要人工智能的创新中心，智能经济和智能社会取得明显成效，为跻身创新型国家前列和经济强国奠定了重要基础。

形成较为成熟的新一代人工智能理论与技术体系。在类脑智能、自主智能、混合智能和群体智能等领域取得重大突破，在国际人工智能研究领域具有重要影响，占据人工智能科技制高点。

人工智能产业的竞争力达到国际领先水平。极大地拓展了人工智能在生产生活、社会治理、国防建设等各方面应用的广度和深度，形成涵盖核心技术、关键系统、支撑平台和智能应用的完备产业链和高端产业群，人工智能核心产业规模超过 1 万亿元，带动相关产业规模超过 10 万亿元。

形成一批全球领先的人工智能科技创新和人才培养基地，建成更加完善的人工智能法律法规、伦理规范和政策体系。

(4) 总体部署。发展人工智能是一项事关全局的复杂系统工程，要按照“构建一个体系，把握双重属性，坚持三位一体，强化四大支撑”进行布局，形成人工智能健康持续发展的战略路径。

构建开放协同的人工智能科技创新体系。针对原创性理论基础薄弱、重大产品和系统缺失等重点难点问题，建立新一代人工智能基础理论和关键共性技术体系，布局建设重大科技创新基地，壮大人工智能高端人才队伍，促进创新主体协同互动，形成人工智能持续的创新能力。

把握人工智能技术属性和社会属性高度融合的特征。既要加大人工智能研发和应用的力度，最大程度发挥人工智能的潜力，又要预判人工智能的挑战，协调产业政策、创新政策与社会政策，实现激励发展与合理规划协调，最大限度防范风险。

坚持人工智能研发攻关、产品应用和产业培育“三位一体”推进。适应人工智能发展的特点和趋势，强化创新链和产业链深度融合、技术供给和市场需求互动演进，以技术突破推动领域应用和产业升级，以应用示范推动技术和系统优化。在当前大规模推动技术应用和产业发展的同时，加强面向中长期的研发布局和攻关，实现滚动发展和持续提升，确保理论上走在前面、技术上占领制高点、应用上安全可控。

全面支撑科技、经济、社会发展和国家安全。以人工智能技术突破带动国家创新能力全面提升，

引领建设世界科技强国进程；通过壮大智能产业、培育智能经济，为我国未来十几年乃至几十年经济繁荣创造一个新的增长周期；以建设智能社会促进民生福祉的改善，落实以人民为中心的发展思想；以人工智能提升国防实力，保障和维护国家安全。

13.7 区块链

1. 定义

区块链（Blockchain）是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。所谓共识机制是区块链系统中实现不同节点之间建立信任、获取权益的数学算法。

区块链是比特币的一个重要概念，它本质上是一个去中心化的数据库，同时作为比特币的底层技术。区块链是一串使用密码学方法相关联产生的数据块，每一个数据块中包含了一次比特币网络交易的信息，用于验证其信息的有效性（防伪）并生成下一个区块。

2. 架构

一般说来，区块链系统由数据层、网络层、共识层、激励层、合约层和应用层组成。数据层封装了底层数据区块及相关的数据加密和时间戳等技术；网络层包括了分布式组网机制、数据传播机制和数据验证机制等；共识层主要封装网络节点的各类共识算法；激励层将经济因素集成到区块链技术体系中来，主要包括经济激励的发行机制和分配机制等；合约层主要封装各类脚本、算法和智能合约，是区块链可编程特性的基础；应用层则封装了区块链的各种应用场景和案例。

该模型中，基于时间戳的链式区块结构、分布式节点的共识机制、基于共识算力的经济激励及灵活可编程的智能合约是区块链技术最具代表性的创新点。

13.8 练习题

1. 信息化的层次从小到大有五层，其中产品信息化最小，它的下一层是_____。

- A. 国民经济信息化 B. 产业信息化 C. 企业信息化 D. 社会生活信息化

解析：

信息化的层次从小到大有五层，分别为产品信息化、企业信息化、产业信息化、国民经济信息化、社会生活信息化。

答案：C

2. 关于两化融合，以下说法错误的是_____。

- A. 是指信息化与制造业发展战略的融合。
B. 是指信息资源与材料、能源等工业资源的融合。
C. 是指虚拟经济与工业实体经济的融合。
D. 是指信息技术与工业技术、IT 设备与工业装备的融合。

解析：

两化融合的含义有以下四条：

- (1) 信息化与工业化发展战略的融合。
- (2) 信息资源与材料、能源等工业资源的融合。
- (3) 虚拟经济与工业实体经济的融合。
- (4) 信息技术与工业技术、IT 设备与工业装备的融合。

答案：A

3. 以下_____不是电子商务的类型。

- A. B2B B. G2C C. C2C D. O2O

解析：

电子政务主要包括四个方面：G2G、G2B、G2C、G2E；电子商务的类型主要包括：B2B、B2C、C2C、O2O。

答案：B

4. 目前企业竞争中的“大”吃“小”正在转向为_____。

- A. “快”吃“慢” B. “强”吃“弱” C. “紧”吃“松” D. “灵”吃“笨”

解析：

目前企业竞争中的“大”吃“小”正在转向为“快”吃“慢”。

答案：A

5. 随着互联网的普及，电子商务已经进入到人们的日常生活，下列_____业务全部属于电子商务的范畴。

- ①网上客服 ②电视购物 ③网上营销 ④电话交易 ⑤商场广播 ⑥网上调查
- A. ①②③④⑤⑥ B. ①②③④⑤ C. ①②③④⑥ D. ①③④⑤⑥

解析：

全部都属于电子商务类型。

答案：A

6. 我国在“十三五”规划纲要中指出要加快信息网络新技术的开发应用，以拓展新兴产业发展空间，纲要中提出将培育的新一代信息技术产业创业重点中不包括_____。

- A. 人工智能 B. 移动智能终端
- C. 第四代移动通信 D. 先进传感器

解析：

我国在“十三五”规划纲要中，将培育人工智能、移动智能终端、第五代移动通信（5G）、先进传感器等作为新一代信息技术产业创新重点发展，拓展新兴产业发展空间。

答案：C

7. 智能具有感知、记忆、自适应等特点，能够存储感知到的外部信息及由思维产生的知识，同时能够利用已有的知识对信息进行分析、计算、比较、判断、联想和决策，这属于智能的_____能力。

- A. 感知 B. 记忆和思维 C. 学习和自适应 D. 行为决策

解析:

智能的特点: 一是具有感知能力, 即具有能够感知外部世界、获取外部信息的能力, 这是产生智能活动的前提条件和必要条件; 二是具有记忆和思维能力, 即能够存储感知到的外部信息及由思维产生的知识, 同时能够利用已有的知识对信息进行分析、计算、比较、判断、联想、决策; 三是具有学习能力和自适应能力, 即通过与环境的相互作用不断学习和积累知识, 使自己能够适应环境的变化; 四是具有行为决策能力, 即对外界的刺激作出反应, 形成决策并传达相应的信息。

答案: B

8. 人工智能是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。今年在技术上取得长足的进步, 其主要的研究方向不包括_____。

- A. 人机对弈 B. 人脸识别 C. 自动驾驶 D. 3D 打印

解析:

人工智能是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

3D 打印技术出现在 20 世纪 90 年代中期, 实际上是利用光固化和纸层叠等技术的最新快速成型装置。它与普通打印的工作原理基本相同, 打印机内装有液体或粉末等“打印材料”, 与电脑连接后, 通过电脑控制把“打印材料”一层层叠加起来, 最终把计算机上的蓝图变成实物。

答案: D

9. 区块链 2.0 技术架构自上而下分为数据层、网络层、共识层、激励层、智能合约层, 数据传播机制、数据验证机制属于其中的_____。

- A. 数据层 B. 网络层 C. 共识层 D. 激励层

解析:

区块链 2.0 技术架构及每层包括的内容如图 13-6 所示。

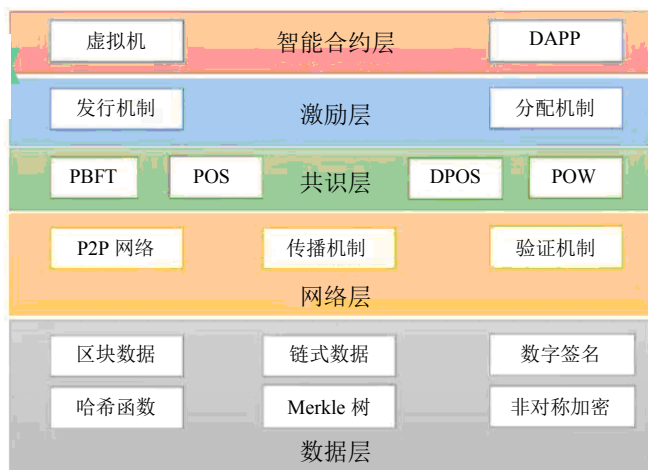


图 13-6 区块链 2.0 技术架构

答案：B

10. 区块链是_____、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。

- A. 数据仓库
- B. 中心化数据库
- C. 非链式数据结构
- D. 分布式数据存储

解析：

区块链（Blockchain）是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。所谓共识机制是区块链系统中实现不同节点之间建立信任、获取权益的数学算法。

区块链是比特币的一个重要概念，它本质上是一个去中心化的数据库，同时作为比特币的底层技术。区块链是一串使用密码学方法相关联产生的数据块，每一个数据块中包含了一次比特币网络交易的信息，用于验证其信息的有效性（防伪）并生成下一个区块。

答案：D

11. 区块链是一种按照时间顺序将数据区块按顺序相连的方式组合成的一种链式数据结构，也是一种以密码学的方式保证不可篡改和不可伪造的分布式账本。主要解决交易的信任和安全问题，最初是作为_____的底层技术出现的。

- A. 电子商务
- B. 证券交易
- C. 比特币
- D. 物联网

解析：

区块链是比特币的一个重要概念，在《2014—2016 年全球比特币发展研究报告》中提到，区块链是比特币的底层技术和基础架构。

答案：C

第14小时

专业英语

14.0 章节考点分析

第14小时主要学习专业英语知识。根据考试大纲，上午的单选题会有5道英文选择题，主要涉及软件工程中一些概念性的知识点，占5分。这部分知识是一些考生的难点所在，本小时总结出一些软考中常考的英文知识点供广大考生参考。



【导读小贴士】

软件设计师考试的英文试题的出题范围基本局限于软件工程方面与项目管理的基础性、概念性的知识，大多属于名词解释范畴。如果考生具有一定的英文水平，同时对基本概念掌握得比较牢固，这部分的分值不难拿到。

14.1 软件文档

Why Have Formal Documents?

Finally, writing the decisions down is essential. Only when one writes do the gaps appear and the inconsistencies protrude. The act of writing turns out to require hundreds of mini-decisions, and it is the existence of these that distinguishes clear, exact policies from fuzzy ones.

Second, the documents will communicate the decisions to others. The manager will be continually amazed that policies he took for common knowledge are totally unknown by some member of his team. Since his fundamental job is to keep everybody going in the same direction, his chief daily task will be communication, not decision-making, and his documents will immensely lighten this load.

Finally, a manager's documents give him a data base and checklist. By reviewing them periodically he sees where he is, and he sees what changes of emphasis or shifts in direction are needed.

The task of the manager is to develop a plan and then to realize it. But only the written plan is precise and communicable. Such a plan consists of documents on what, when, how much, where, and who. This small set of critical documents decides much of the manager's work. If their comprehensive and critical nature is recognized in the beginning, the manager can approach them as friendly tools rather than annoying busywork. He will set his direction much more crisply and quickly by doing so.

为什么要有正式的文档？

首先，把决策写下来是很有必要的，只有写出来才能显示出矛盾和有差距的地方。写文档的过程中需要很多小的决策，能够把清楚的、准确的策略和模糊不清的策略区分开来。

其次，文档是与他人交流的工具。项目经理会惊讶地发现，对于他采取的一些常识性的措施，团队里的成员居然并不了解。既然他的基本工作是使每个人在相同的方向上前进，他的主要工作就是交流，而不是制定决策，文档能很好地减轻他的负担。

最后，项目经理的文档给他提供了一个数据库和检验表。通过定期地回顾，他能知道自己所处的位置，并看到工作重心的转移与工作方向是否需要变动。

项目经理的任务是制定计划，并实现计划。但是只有书面计划是精确的和可以沟通的。计划中包括了时间、地点、人物、事件、资金。这些少量的关键文档决定了项目经理的工作。如果一开始就认识到文档的普遍性和重要性，那么就可以将文档作为工具更好地利用起来，而不会让它成为令人厌烦的繁重任务。根据文档开展工作，项目经理能更清晰和快速地设定自己的方向。

14.2 非功能需求

The architecture design specifies the overall architecture and the placement of software and hardware that will be used. Architecture design is a very complex process that is often left to experienced architecture designers and consultants. The first step is to refine the nonfunctional requirements into more detailed requirements that are then employed to help select the architecture to be used and the software components to be placed on each device. In a client-based architecture, one also has to decide whether to use a two-tier, three-tier, or n-tier architecture. Then the requirements and the architecture design are used to develop the hardware and software specification. There are four primary types of nonfunctional requirements that can be important in designing the architecture. A operational requirements specify the

operating environment(s) in which the system must perform and how those may change over time. Performance requirements focus on the nonfunctional requirements issues such as response time, capacity, and reliability. Security requirements are the abilities to protect the information system from disruption and data loss, whether caused by an intentional act. Cultural and political requirements are specific to the countries in which the system will be used.

架构设计指定了将要使用的软件和硬件的总体架构和布局。架构设计是一个非常复杂的过程, 往往留给经验丰富的架构设计师和顾问。第一步是将非功能需求细化为更详细的要求, 然后用于帮助选择要使用的体系结构及要放置在每个设备上的软件组件。

在基于客户端的架构中, 还必须决定是使用两层、三层, 还是 n 层架构, 然后使用需求和体系结构设计来开发硬件和软件规范。有四种主要的非功能需求类型可能在设计架构时非常重要。操作要求指定系统必须执行的操作环境及这些操作环境如何随时间变化。性能要求侧重于非功能性需求问题, 如响应时间、容量和可靠性。安全要求是指是否有能力保护信息系统免受故意行为造成的破坏和数据丢失。文化和政治要求明确了特定系统将被使用的国家。

14.3 开发方法

Extreme Programming (XP) is a discipline of software development with values of simplicity, communication, feedback and courage. Successful software development is a team effort—not just the development team, but the larger team consisting of customer, management and developers. XP is a simple process that brings these people together and helps them to succeed together. XP is aimed primarily at object-oriented projects using teams of a dozen or fewer programmers in one location. The principles of XP apply to any moderately-sized project that needs to deliver quality software rapidly and flexibly.

An XP project needs a full-time customer to provide guidance. Customers, programmers, managers, are all working together to build the system that's needed. Customers—those who have software that needs to be developed—will learn simple, effective ways to communicate what they need, to be sure that they are getting what they need, and to steer the project to success.

极限编程是一个专业软件开发方法, 它包含简捷、沟通、反馈和勇气四大价值观。成功的软件开发是一个团队努力的结果, 而这个团队不只是开发团队, 而是一个由客户、管理和开发人员等更多人组成的团队。极限编程是一个简单的过程, 将一些人聚集在一起, 这样做更容易完成项目目标。极限编程主要是针对十几个或更少人的项目团队进行面向对象开发工作。原则上极限编程适用于任何需要迅速和灵活提供高质量的轻量级项目。

一个极限编程项目需要客户的全程指导, 客户、程序员和项目经理都是必备人员。客户是指那

些有开发需求的人，我们需要和他们建立有效的沟通方式来确定他们的需求，引导项目走向成功。

14.4 新技术

At a basic level, cloud computing is simply a means of delivering IT resources as services. Almost all IT resources can be delivered as a cloud service: applications, compute power, storage capacity, networking, Programming tools, even communication services and collaboration tools.

Cloud computing began as large-scale Internet service providers such as Google, Amazon, and others built out their infrastructure. An architecture emerged: massively scaled, horizontally distributed system resources, abstracted as virtual IT services and managed as continuously configured, pooled resources. In this architecture, the data is mostly resident on servers "somewhere on the Internet" and the application runs on both the "cloud servers" and the user's browser.

Both clouds and grids are built to scale horizontally very efficiently. Both are built to withstand failures of individual elements or nodes. Both are charged on a per-use basis. But while grids typically process batch jobs, with a defined start and end point, cloud services can be continuous. What's more, clouds expand the types of resources available—file storage, databases, and Web services—and extend the applicability to Web and enterprise applications.

基本上，云计算只是将 IT 资源作为服务来提供。几乎所有的 IT 资源都可以作为一个云服务来提供，包括应用程序、计算能力、存储容量、网络、编程工具，甚至是通信服务和协作工具。

大规模提供云计算服务的互联网服务提供商有谷歌、亚马逊和其他一些提供商。云架构的特点是：系统不断扩大，系统资源水平分布，抽象的虚拟服务和管理，不断配置，汇集资源。在这种体系结构中，数据主要存放在处于“互联网某处”的服务器上，应用程序运行在“云服务器”与客户端浏览器上。

云网格要满足分布式需求，要建立能够抵御单个主机或节点失败的云网格，并且客户使用云服务要支付费用。当用网格来处理一批日常工作时，需要定义一个明确的开始点和结束点，云服务可以是连续的。但更重要的是，云服务可以扩展可用的文件存储、数据库、网络服务来适用于网络和企业。

14.5 练习题

The purpose of the systems analysis phase is to build a logical model of the new system. The first step is (a), where you investigate business processes and document what the new system must do to satisfy users. This step continues the investigation that began during the (b). You use the fact-finding results to build business models, data and process models, and object models. The deliverable for the

systems analysis phase is the (c), which describes management and user requirements, costs and benefits, and outlines alternative development strategies.

The purpose of the systems design phase is to create a physical model that will satisfy all documented requirements for the system. During the systems design phase, you need to determine the (d), which programmers will use to transform the logical design into program modules and code. The deliverable for this phase is the (e), which is presented to management and users for review and approval.

- | | |
|--|-------------------------------------|
| (a) A. system logical modeling | B. use case modeling |
| C. requirements modeling | D. application modeling |
| (b) A. systems planning phase | B. systems modeling phase |
| C. systems analysis phase | D. systems design phase |
| (c) A. system charter | B. system scope definition |
| C. system blueprint | D. system requirements document |
| (d) A. application architecture | B. system data model |
| C. system process model | D. implement environment |
| (e) A. system architecture description | B. system design specification |
| C. system technique architecture | D. physical deployment architecture |

翻译:

系统分析阶段的目的是建立新系统的逻辑模型, 第一步就是进行 (a), 调查业务流程并记录新系统必须做哪些以满足用户需求。这一步是 (b) 的延续, 可以使用调查结果构建业务模型、数据和流程模型及对象模型。系统分析阶段的可交付成果是 (c), 它描述了管理和用户要求、成本和收益, 还有可替代的方案。

系统设计阶段的目的是创建一个能够满足系统所有需求文件要求的物理模型。在系统设计阶段, 需要确定 (d), 程序员通过系统过程模型把逻辑设计转换为程序模块和代码。该阶段的可交付成果是 (e), 提交给管理层和用户审查和批准。

- | | |
|-----------------|--------------|
| (a) A. 系统逻辑建模 | B. 使用用例模型 |
| C. 需求建模 | D. 应用建模 |
| (b) A. 系统计划阶段 | B. 系统建模阶段 |
| C. 系统分析阶段 | D. 系统设计阶段 |
| (c) A. 系统章程 | B. 系统范围定义 |
| C. 系统蓝图 | D. 系统需求文件 |
| (d) A. 应用程序架构描述 | B. 系统数据模型 |
| C. 系统过程模型 | D. 实施环境 |
| (e) A. 应用程序架构 | B. 系统设计规格说明书 |
| C. 系统技术架构 | D. 系统部署架构 |

答案: A、A、B、C、B

第三篇

真题篇

第15小时

2016 年 5 月考试真题（上午基础知识）

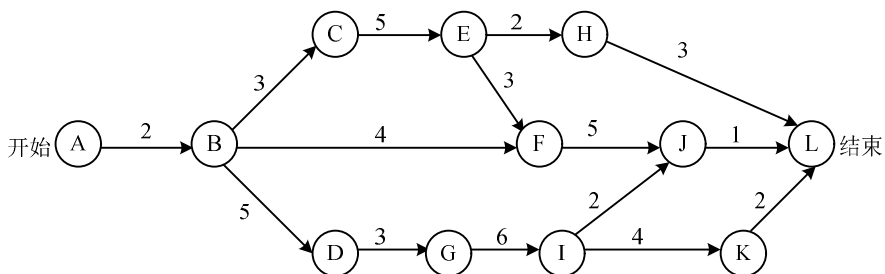


【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

- VLIW 是 (1) 的简称。
 - (1) A. 复杂指令系统计算机 B. 超大规模集成电路
 - C. 单指令流多数据流 D. 超长指令字
- 主存与 Cache 的地址映射方式中，(2) 方式可以实现主存任意一块装入 Cache 中任意位置，只有装满才需要替换。
 - (2) A. 全相联 B. 直接映射 C. 组相联 D. 串并联
- 如果“2X”的补码是“90H”，那么 X 的真值是 (3)。
 - (3) A. 72 B. -56 C. 56 D. 111
- 移位指令中的 (4) 指令的操作结果相当于对操作数进行乘 2 操作。
 - (4) A. 算术左移 B. 逻辑右移 C. 算术右移 D. 带进位循环左移

- 内存按字节编址, 从 A1000H 到 B13FFH 的区域的存储容量为 (5) kB。
(5) A. 32 B. 34 C. 65 D. 67
- 以下关于总线的叙述中, 不正确的是 (6)。
(6) A. 并行总线适合近距离高速数据传输
 B. 串行总线适合长距离数据传输
 C. 单总线结构在一个总线上适应不同种类的设备, 设计简单且性能很高
 D. 专用总线在设计上可以与连接设备实现最佳匹配
- 以下关于网络层次与主要设备对应关系的叙述中, 配对正确的是 (7)。
(7) A. 网络层——集线器 B. 数据链路层——网桥
 C. 传输层——路由器 D. 会话层——防火墙
- 传输经过 SSL 加密的网页所采用的协议是 (8)。
(8) A. HTTP B. HTTPS C. S-HTTP D. HTTP-S
- 为了攻击远程主机, 通常利用 (9) 技术检测远程主机状态。
(9) A. 病毒查杀 B. 端口扫描 C. QQ 聊天 D. 身份认证
- 某软件公司参与开发管理系统软件的程序员张某, 辞职到另一公司任职, 于是该项目负责人将该管理系统软件上开发者的署名更改为李某 (接张某工作)。该项目负责人的行为 (10)。
(10) A. 侵犯了张某开发者的身份权 (署名权)
 B. 不构成侵权, 因为程序员张某不是软件著作权人
 C. 只是行使管理者的权利, 不构成侵权
 D. 不构成侵权, 因为程序员张某现已不是项目组成员
- 美国某公司与中国某企业谈技术合作, 合同约定使用一项美国专利 (获得批准并在有效期内), 该项技术未在中国和其他国家申请专利。依照该专利生产的产品 (11) 需要向美国公司支付这件美国专利的许可使用费。
(11) A. 在中国销售, 中国企业 B. 如果返销美国, 中国企业不
 C. 在其他国家销售, 中国企业 D. 在中国销售, 中国企业不
- 以下媒体文件格式中, (12) 是视频文件格式。
(12) A. WAV B. BMP C. MP3 D. MOV
- 以下软件产品中, 属于图像编辑处理工具的软件是 (13)。
(13) A. Powerpoint B. Photoshop C. Premiere D. Acrobat
- 使用 150DPI 的扫描分辨率扫描一幅 3×4 英寸的彩色照片, 得到原始的 24 位真彩色图像的数据量是 (14) Byte。
(14) A. 1800 B. 90000 C. 270000 D. 810000
- 某软件项目的活动图如下图所示, 其中顶点表示项目里程碑, 连接顶点的边表示包含的活动, 边上的数字表示活动的持续时间 (天), 则完成该项目的最少时间为 (15) 天。活动 BD 最多可以晚开始 (16) 天而不会影响整个项目的进度。



(15) A. 15 B. 21 C. 22 D. 24

(16) A. 0 B. 2 C. 3 D. 5

- 在结构化分析中,用数据流图描述 (17)。当采用数据流图对一个图书馆管理系统进行分析时, (18) 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

- 软件开发过程中,需求分析阶段的输出不包括 (19)。

(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

- 以下关于高级程序设计语言实现编译和解释方式的叙述中,正确的是 (20)。

(20) A. 编译程序不参与用户程序的运行控制,而解释程序则参与
 B. 编译程序可以用高级语言编写,而解释程序只能用汇编语言编写
 C. 编译方式处理源程序时不进行优化,而解释方式则进行优化
 D. 编译方式不生成源程序的目标程序,而解释方式则生成

- 以下关于脚本语言的叙述中,正确的是 (21)。

(21) A. 脚本语言是通用的程序设计语言
 B. 脚本语言更适合应用在系统级程序开发中
 C. 脚本语言主要采用解释方式实现
 D. 脚本语言中不能定义函数和调用函数

- 将高级语言源程序先转化为一种中间代码是现代编译器常见的处理方式。常用的中间代码有后缀式、 (22)、树等。

(22) A. 前缀码 B. 三地址码 C. 符号表 D. 补码和移码

- 当用户通过键盘或鼠标进入某应用系统时,通常最先获得键盘或鼠标输入信息的是 (23) 程序。

(23) A. 命令解释 B. 中断处理 C. 用户登录 D. 系统调用

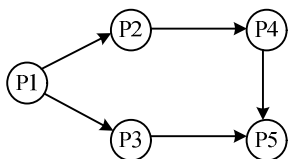
- 在 Windows 操作系统中，当用户双击“IMG_20160122_103.jpg”文件名时，系统会自动通过建立的 (24) 来决定使用什么程序打开该图像文件。

(24) A. 文件 B. 文件关联 C. 文件目录 D. 临时文件

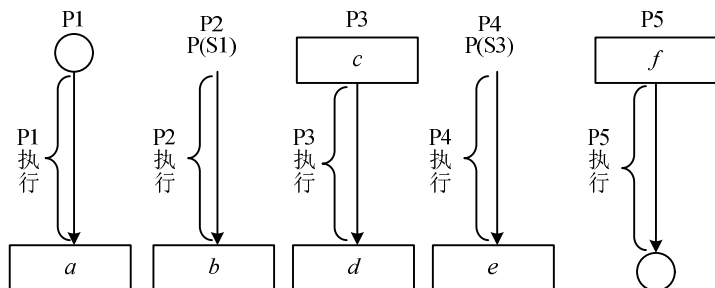
- 某磁盘有 100 个磁道，磁头从一个磁道移至另一个磁道需要 6ms。文件在磁盘上非连续存放，逻辑上相邻数据块的平均距离为 10 个磁道，每块的旋转延迟时间及传输时间分别为 100ms 和 20ms，则读取一个 100 块的文件需要 (25) ms。

(25) A. 12060 B. 12600 C. 18000 D. 186000

- 进程 P1、P2、P3、P4 和 P5 的前趋图如下图所示：



若用 PV 操作控制进程 P1、P2、P3、P4 和 P5 并发执行的过程，则需要设置五个信号 S1、S2、S3、S4 和 S5，而且信号量 S1~S5 的初值都等于零。下图中 a 和 b 处应分别填写 (26)；c 和 d 处应分别填写 (27)；e 和 f 处应分别填写 (28)。

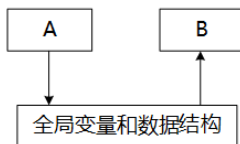


(26) A. V(S1)P(S2)和 V(S3) B. P(S1)V(S2)和 V(S3)
C. V(S1)V(S2)和 V(S3) D. P(S1)P(S2)和 V(S3)

(27) A. P(S2)和 P(S4) B. P(S2)和 V(S4)
C. V(S2)和 P(S4) D. V(S2)和 V(S4)

(28) A. P(S4)和 V(S4)V(S5) B. V(S5)和 P(S4)P(S5)
C. V(S3)和 V(S4)V(S5) D. P(S3)和 P(S4)V(P5)

- 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 (29) 耦合。



- (29) A. 公共 B. 控制 C. 标记 D. 数据
- 以下关于增量开发模型的叙述中, 不正确的是 (30) 。

(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
 - 在设计软件的模块结构时, (31) 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
 - 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 (32) , (33) 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
 - 如图 1 所示为一个模块层次结构的例子, 如图 2 所示为对其进行集成测试的顺序, 则此测试采用了 (34) 测试策略。该测试策略的优点不包括 (35) 。

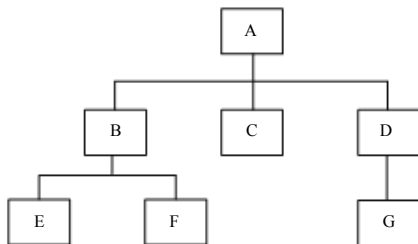


图 1

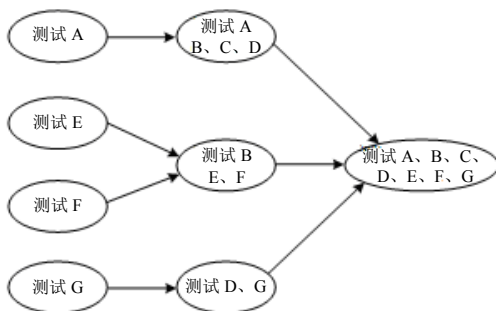
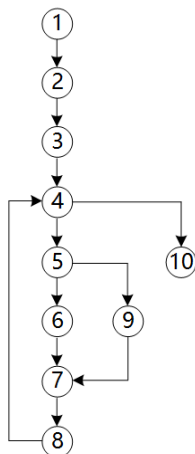


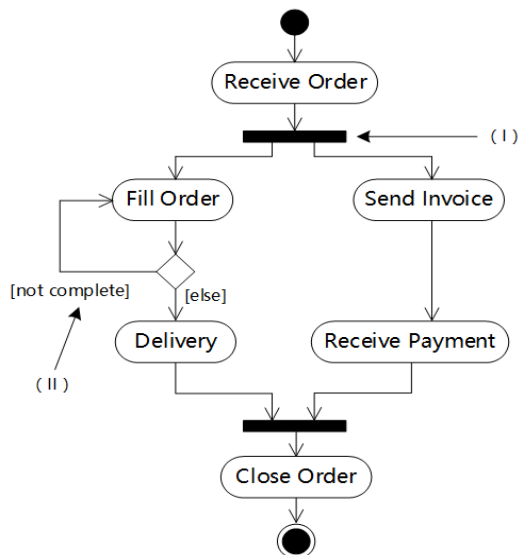
图 2

- (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

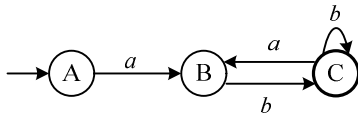
- B. 较早地验证了底层模块
- C. 测试的并行程度较高
- D. 较少的驱动模块和桩模块编写工作量
- 采用 McCabe 度量法计算下图所示程序的环路复杂性为 (36)。



- (36) A. 1 B. 2 C. 3 D. 4
- 在面向对象方法中, (37) 是父类和子类之间共享数据和方法的机制。子类在原有父类接口的基础上, 用适合于自己要求的实现去置换父类中的相应实现称为 (38)。
 - (37) A. 封装 B. 继承 C. 覆盖 D. 多态
 - (38) A. 封装 B. 继承 C. 覆盖 D. 多态
 - 在 UML 用例图中, 参与者表示 (39)。
 - (39) A. 人、硬件或其他系统可以扮演的角色
 - B. 可以完成多种动作的相同用户
 - C. 不管角色的实际物理用户
 - D. 带接口的物理系统或硬件设计
 - UML 中关联是一个结构关系, 描述了一组链两个类之间 (40) 关联。
 - (40) A. 不能有多
 - B. 可以有多个由不同角色标识的
 - C. 可以有任意多个
 - D. 的多个关联必须聚合成一个
 - 如下所示的 UML 图是 (41), 图中 (I) 表示 (42), (II) 表示 (43)。
 - (41) A. 序列图 B. 状态图 C. 通信图 D. 活动图
 - (42) A. 合并分叉 B. 分支 C. 合并汇合 D. 流
 - (43) A. 分支条件 B. 监护表达式 C. 动作名 D. 流名称



- 为图形用户界面（GUI）组件定义不同平台的并行类层次结构，适合采用 （44） 模式。
 （44） A. 享元（Flyweight） B. 抽象工厂（Abstract Factory）
 C. 外观（Facade） D. 装饰器（Decorator）
- （45） 设计模式将一个请求封装为一个对象，从而使得可以用不同的请求对客户进行参数化，对请求排队或记录请求日志，以及支持可撤销的操作。
 （45） A. 命令（Command） B. 责任链（Chain of Responsibility）
 C. 观察者（Observer） D. 策略（Strategy）
- （46） 设计模式最适合用于发布/订阅消息模型，即当订阅者注册一个主题后，此主题有新消息到来时订阅者就会收到通知。
 （46） A. 适配器（Adapter） B. 通知（Notifier）
 C. 观察者（Observer） D. 状态（State）
- 因使用大量的对象而造成很大的存储开销时，适合采用 （47） 模式进行对象共享，以减少对象数量从而达到较少的内存占用并提升性能。
 （47） A. 组合（Composite） B. 享元（Flyweight）
 C. 迭代器（Iterator） D. 备忘（Memento）
- 移进-归约分析法是编译程序（或解释程序）对高级语言源程序进行语法分析的一种方法，属于 （48） 的语法分析方法。
 （48） A. 自顶向下（或自上而下） B. 自底向上（或自下而上）
 C. 自左向右 D. 自右向左
- 某确定的有限自动机（DFA）的状态转换图如下图所示（A 是初态，C 是终态），则该 DFA 能识别 （49）。



(49) A. aabb B. abab C. baba D. abba

- 函数 $\text{main}()$ 、 $\text{f}()$ 的定义如下所示, 调用函数 $\text{f}()$ 时, 第一个参数采用传值 (call by value) 方式, 第二个参数采用传引用 (call by reference) 方式, main 函数中 “ $\text{print}(x)$ ” 执行后输出的值为 (50)。

$\text{main}()$

```
int x=1;
f(5,x);
print(x);
```

$\text{f}(\text{int } x, \text{int } \&a)$

```
x=2*x+1;
a=a+x;
return;
```

(50) A. 1 B. 6 C. 11 D. 12

- 数据的物理独立性和逻辑独立性分别是通过修改 (51) 来完成的。

(51) A. 外模式与内模式之间的映像、模式与内模式之间的映像
 B. 外模式与内模式之间的映像、外模式与模式之间的映像
 C. 外模式与模式之间的映像、模式与内模式之间的映像
 D. 模式与内模式之间的映像、外模式与模式之间的映像
- 关系规范化在数据库设计的 (52) 阶段进行。

(52) A. 需求分析 B. 概念设计 C. 逻辑设计 D. 物理设计
- 若给定的关系模式为 R , $U=\{A,B,C\}$, $F=\{AB \rightarrow C, C \rightarrow B\}$, 则关系 R (53)。

(53) A. 有 2 个候选关键字 AC 和 BC , 并且有 3 个主属性
 B. 有 2 个候选关键字 AC 和 AB , 并且有 3 个主属性
 C. 只有一个候选关键字 AC , 并且有 1 个非主属性和 2 个主属性
 D. 只有一个候选关键字 AB , 并且有 1 个非主属性和 2 个主属性
- 某公司数据库中的元件关系模式为 P (元件号, 元件名称, 供应商, 供应商所在地, 库存量), 函数依赖集 F 表示为: $F=\{\text{元件号} \rightarrow \text{元件名称}, (\text{元件号}, \text{供应商}) \rightarrow \text{库存量}, \text{供应商} \rightarrow \text{供应商所在地}\}$, 元件关系的主键为 (54), 该关系存在冗余、插入异常和删除异常等问题。为了解决这一问题需要将元件关系分解为 (55), 分解后的关系模式可以达到 (56)。

(54) A. 元件号, 元件名称 B. 元件号, 供应商
 C. 元件号, 供应商所在地 D. 供应商, 供应商所在地

(55) A. 元件 1 (元件号, 元件名称, 库存量)、元件 2 (供应商, 供应商所在地)
 B. 元件 1 (元件号, 元件名称)、元件 2 (供应商, 供应商所在地, 库存量)
 C. 元件 1 (元件号, 元件名称)、元件 2 (元件号, 供应商, 库存量)、元件 3 (供应商, 供应商所在地)

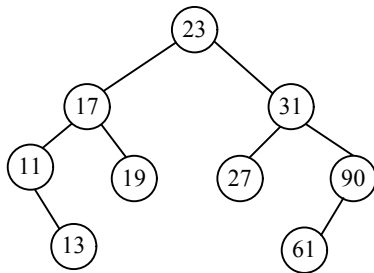
D. 元件 1 (元件号, 元件名称)、元件 2 (元件号, 库存量)、元件 3 (供应商, 供应商所在地)、元件 4 (供应商所在地, 库存量)

(56) A. 1NF B. 2NF C. 3NF D. 4NF

- 若元素以 a 、 b 、 c 、 d 、 e 的顺序进入一个初始为空的栈中, 每个元素进栈、出栈各 1 次, 要求出栈的第一个元素为 d , 则合法的出栈序列共有 (57) 种。

(57) A. 4 B. 5 C. 6 D. 24

- 设有二叉排序树 (或二叉查找树) 如下图所示, 建立该二叉树的关键码序列不可能是 (58)。



(58) A. 23 31 17 19 11 27 13 90 61 B. 23 17 19 31 27 90 61 11 13
C. 23 17 27 19 31 13 11 90 61 D. 23 31 90 61 27 17 19 11 13

- 若一棵二叉树的高度 (即层数) 为 h , 则该二叉树 (59)。

(59) A. 有 2^h 个节点 B. 有 2^h-1 个节点
C. 最少有 2^{h-1} 个节点 D. 最多有 2^h-1 个节点

- 在 13 个元素构成的有序表 $A[1..13]$ 中进行折半查找 (或二分查找, 向下取整)。那么以下叙述中, 错误的是 (60)。

(60) A. 无论要查找哪个元素, 都是先与 $A[7]$ 进行比较
B. 若要查找的元素等于 $A[9]$, 则需要分别与 $A[7]$ 、 $A[11]$ 、 $A[9]$ 进行比较
C. 无论要查找的元素是否在 $A[]$ 中, 最多与表中的 4 个元素进行比较
D. 若待查找的元素不在 $A[]$ 中, 最少需要与表中的 3 个元素进行比较

- 以下关于图的遍历的叙述中, 正确的是 (61)。

(61) A. 图的遍历是从给定的源点出发对每一个顶点仅访问一次的过程
B. 图的深度优先遍历方法不适用于无向图
C. 使用队列对图进行广度优先遍历
D. 图中有回路时则无法进行遍历

- 考虑一个背包问题, 共有 $n = 5$ 个物品, 背包容量为 $W = 10$, 物品的重量和价值分别为 $w = \{2, 2, 6, 5, 4\}$, $v = \{6, 3, 5, 4, 6\}$, 求背包问题的最大装包价值。若此为 0-1 背包问题, 分析该问题具有最优子结构, 定义递归式如下:

$$c(i, j) = \begin{cases} 0 & \text{若 } i = 0 \text{ 或 } j = 0 \\ c(i-1, j) & \text{若 } w[i] > j \\ \max\{c(i-1, j), c(i-1, j-w(i))\} & \text{其他} \end{cases}$$

其中, $c(i, j)$ 表示 i 个物品、容量为 j 的 0-1 背包问题的最大装包价值, 最终要求解 $c(n, W)$ 。

采用自底向上的动态规划方法求解, 得到最大装包价值为 (62), 算法的时间复杂度为 (63)。若此为部分背包问题, 首先采用归并排序算法, 根据物品的单位重量价值从大到小排序, 然后依次将物品放入背包直至所有物品放入背包中或背包再无容量, 则得到的最大装包价值为 (64), 算法的时间复杂度为 (65)。

- (62) A. 11 B. 14 C. 15 D. 16.67
 (63) A. $\Theta(nW)$ B. $\Theta(n \lg n)$ C. $\Theta(n^2)$ D. $\Theta(n \lg n W)$
 (64) A. 11 B. 14 C. 15 D. 16.67
 (65) A. $\Theta(nW)$ B. $\Theta(n \lg n)$ C. $\Theta(n^2)$ D. $\Theta(n \lg n W)$

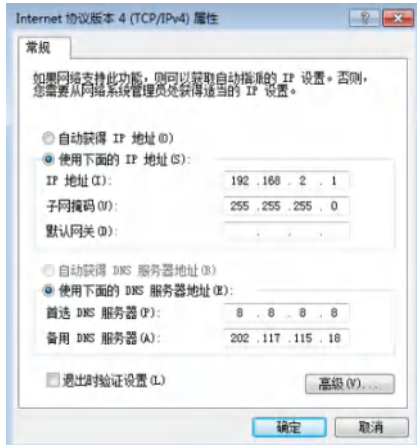
- 默认情况下, FTP 服务器的控制端口为 (66), 上传文件时的端口为 (67)。

- (66) A. 大于 1024 的端口 B. 20 C. 80 D. 21
 (67) A. 大于 1024 的端口 B. 20 C. 80 D. 21

- 使用 ping 命令可以进行网络检测, 在进行一系列检测时, 按照由近及远的原则, 首先执行的是 (68)。

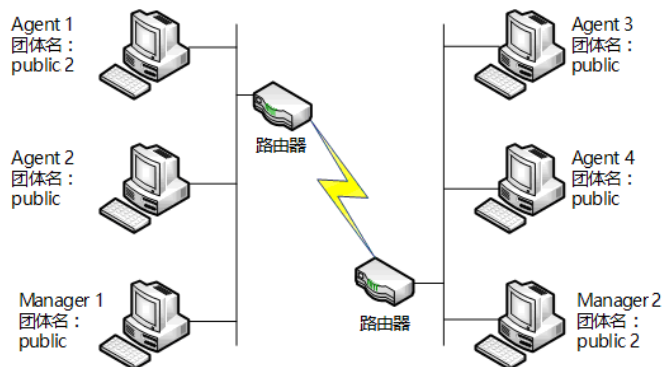
- (68) A. ping 默认网关 B. ping 本地 IP
 C. ping 127.0.0.1 D. ping 远程主机

- 某 PC 的 Internet 协议属性参数如下图所示, 默认网关的 IP 地址是 (69)。



- (69) A. 8.8.8.8 B. 202.117.115.3
 C. 192.168.2.254 D. 202.117.115.18

- 在下图的 SNMP 配置中, 能够响应 Manager 2 的 getRequest 请求的是 (70)。



(70) A. Agent 1 B. Agent 2 C. Agent 3 D. Agent 4

- In the fields of physical security and information security, access control is the selective restriction of access to a place or other resource. The act of accessing may mean consuming, entering, or using. Permission to access a resource is called authorization (授权).

An access control mechanism (71) between a user (or a process executing on behalf of a user) and system resources, such as applications, operating systems, firewalls, routers, files, and databases. The system must first authenticate (验证) a user seeking access. Typically the authentication function determines whether the user is (72) to access the system at all. Then the access control function determines if the specific requested access by this user is permitted. A security administrator maintains an authorization database that specifies what type of access to which resources is allowed for this user. The access control function consults this database to determine whether to (73) access. An auditing function monitors and keeps a record of user accesses to system resources.

In practice, a number of (74) may cooperatively share the access control function. All operating systems have at least a rudimentary (基本的), and in many cases a quite robust, access control component. Add-on security packages can add to the (75) access control capabilities of the OS. Particular applications or utilities, such as a database management system, also incorporate access control functions. External devices, such as firewalls, can also provide access control services.

- (71) A. cooperates B. coordinates C. connects D. mediates
 (72) A. denied B. permitted C. prohibited D. rejected
 (73) A. open B. monitor C. grant D. seek
 (74) A. components B. users C. mechanisms D. algorithms
 (75) A. remote B. native C. controlled D. automated

第16小时

2016年5月考试真题（下午案例分析）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

试题一（15分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某会议中心提供举办会议的场地设施和各种设备，供公司与各类组织机构租用。场地包括一个大型报告厅、一个小型报告厅和诸多会议室。这些报告厅和会议室可提供的设备有投影仪、白板、视频播放/回放设备、计算机等。为了加强管理，该中心欲开发一个会议预订系统，系统的主要功能如下。

（1）检查可用性。客户提交预订请求后，检查预订表，判定所申请的场地是否在申请日期内可用；如果不可用，返回不可用信息。

（2）临时预订。会议中心管理员收到客户预定请求的通知之后，提交确认。系统生成新的临时预订存入预订表，并对新客户创建一条客户信息记录加以保存。根据客户记录给客户发送临时预订确认信息和支付定金要求。

（3）分配设施与设备。根据临时预订或变更预定的设备和设施需求，分配所需设备（均能满足用户要求）和设施，更新相应的表和预订表。

（4）确认预订。管理员收到客户支付定金的通知后，检查确认，更新预订表，根据客户记录给客户发送预订确认信息。

（5）变更预订。客户还可以在支付余款前提交变更预订请求，检查变更预订请求的可用性，如果可用，分配设施和设备；如果不可用，返回不可用信息。管理员确认变更后，根据客户记录给客户发送确认信息。

（6）要求付款。管理员从预订表中查询距预订的会议时间两周内的预定，根据客户记录给满足条件的客户发送支付余款要求。

（7）支付余款。管理员收到客户支付余款的通知后，检查确认，更新预订表中的已支付余款信息。

现在采用结构化方法对会议预定系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和如图 1-2 所示的 0 层数据流图（不完整）。

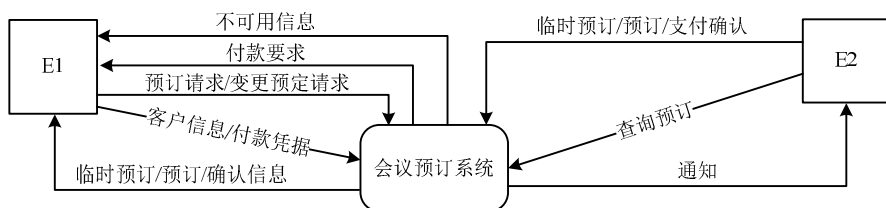


图 1-1 上下文数据流图

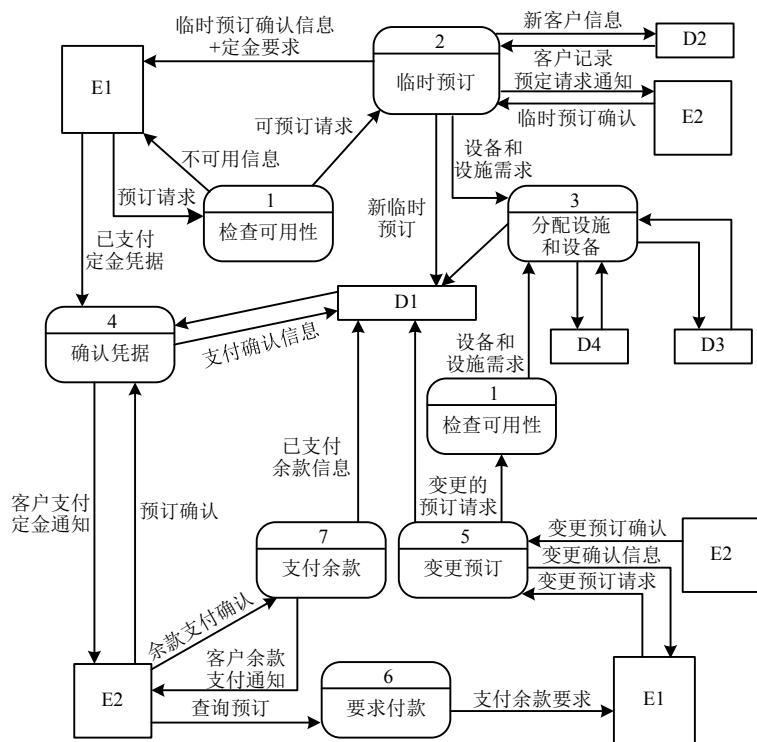


图 1-2 0 层数据流图

【问题 1】(2 分)

使用说明中的词语, 给出图 1-1 中的实体 E1 和 E2 的名称。

【问题 2】(4 分)

使用说明中的词语, 给出图 1-2 中的数据存储 D1~D4 的名称。

【问题 3】(6 分)

根据说明和图中术语, 补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】(3 分)

如果发送给客户的确认信息是通过 Email 系统向客户信息中的电子邮件地址进行发送的, 那么需要对图 1-1 和图 1-2 进行哪些修改? 用 150 字以内的文字加以说明。

试题二 (15 分)

阅读下列说明和图, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某销售公司当前的销售业务为商城实体店销售。现该公司拟开展网络销售业务, 需要开发一个信息化管理系统。请根据公司现有业务及需求完成该系统的数据库设计。

【需求描述】

(1) 记录公司所有员工的信息。员工信息包括工号、身份证号、姓名、性别、出生日期和电话, 并只登记一部电话。

(2) 记录所有商品的信息。商品信息包括商品名称、生产厂家、销售价格和商品介绍。系统内部用商品条码唯一地区别每种商品。

(3) 记录所有顾客的信息。顾客信息包括顾客姓名、身份证号、登录名、登录密码和电话号码。一位顾客只能提供一个电话号码。系统自动生成唯一的顾客编号。

(4) 顾客登录系统之后, 在网上商城购买商品。顾客可将选购的商品置入虚拟的购物车内, 购物车可以长期存放顾客选购的所有商品。顾客可在购物车内选择商品、修改商品数量后生成网购订单。订单生成后, 由顾客选择系统提供的第三方支付平台进行电子支付, 支付成功后系统需要记录唯一的支付凭证编号, 然后由商城根据订单进行线下配送。

(5) 所有的配送商品均由仓库统一出库。为方便顾客, 允许每位顾客在系统中提供多组收货地址、收货人及联系电话。一份订单所含的多个商品可能由多名分拣员根据商品所在仓库信息从仓库中进行分拣操作, 分拣后的商品交由配送员根据配送单上的收货地址进行配送。

(6) 新设计的系统要求记录实体店的每笔销售信息, 包括营业员、顾客、所售商品及其数量。

【概念模型设计】

根据需求阶段收集的信息, 设计的实体联系图 (不完整) 如图 2-1 所示。

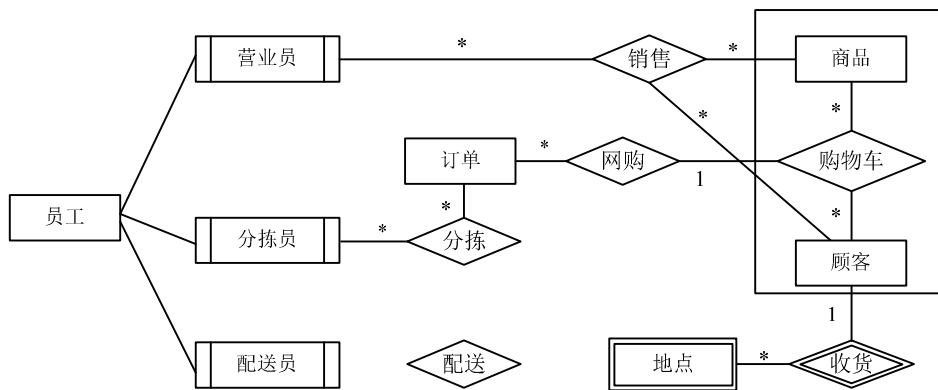


图 2-1 实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

员工（工号，身份证号，姓名，性别，出生日期，电话）

商品（商品条码，商品名称，生产厂家，销售价格，商品介绍， (a) ）

顾客（顾客编号，姓名，身份证号，登录名，登录密码，电话）

收货地点（收货 ID，顾客编号，收货地址，收货人，联系电话）

购物车（顾客编号，商品条码，商品数量）

订单（订单 ID，顾客编号，商品条码，商品数量， (b) ）

分拣（分拣 ID，分拣员工号， (c) ，分拣时间）

配送（配送 ID，分拣 ID，配送员工号，收货 ID，配送时间，签收时间，签收快照）

销售（销售 ID，营业员工号，顾客编号，商品条码，商品数量）

【问题 1】（4 分）

补充图 2-1 中的“配送”联系所关联的对象及联系类型。

【问题 2】（6 分）

补充逻辑结构设计中的（a）、（b）、（c）三处空缺。

【问题 3】（5 分）

对于实体店销售，若要增加送货上门服务，由营业员在系统中下订单，与网购的订单进行后续的统一管理。请根据该需求，对图 2-1 进行补充，并修改订单关系模式。

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某软件公司欲设计实现一个虚拟世界仿真系统。系统中的虚拟世界用于模拟现实世界中的不同环境（由用户设置并创建），用户通过操作仿真系统中的 1~2 个机器人来探索虚拟世界。机器人维

护着两个变量 b1 和 b2，用来保存从虚拟世界中读取的字符。

该系统的主要功能描述如下：

(1) 机器人探索虚拟世界 (Run Robots)。用户使用编辑器 (Editor) 编写文件以设置想要模拟的环境，将文件导入系统 (Load File) 从而在仿真系统中建立虚拟世界 (Setup World)。机器人在虚拟世界中的行为也在文件中进行定义，建立机器人的探索行为程序 (Setup Program)。机器人在虚拟世界中探索时有以下两种运行模式：

1) 自动控制 (Run)：事先编排好机器人的动作序列或指令 (Instruction)，执行指令，使机器人可以连续动作。若干条指令构成机器人的指令集 (Instruction Set)。

2) 单步控制 (Step)：自动控制方式的一种特殊形式，只执行指定指令中的一个动作。

(2) 手动控制机器人 (Manipulate Robots)。选定一个机器人后，可以采用手动方式控制它。手动控制有以下四种方式：

1) Move：机器人朝着正前方移动一个交叉点。

2) Left：机器人原地沿逆时针方向旋转 90 度。

3) Read：机器人读取其所在位置的字符，并将这个字符的值赋给 b1；如果这个位置上没有字符，则不改变 b1 的当前值。

4) Write：将 b1 中的字符写入机器人当前所在的位置，如果这个位置上已经有字符，该字符的值将会被 b1 的值替代。如果这时 b1 没有值，即在执行 Write 动作之前没有执行过任何 Read 动作，那么需要提示用户相应的错误信息 (Show Errors)。

手动控制与单步控制的区别在于，单步控制时执行的是指令中的动作，只有一种控制方式，即执行下一个动作；而手动控制有四种动作。

现在采用面向对象方法设计并实现该仿真系统，得到如图 3-1 所示的用例图和如图 3-2 所示的初始类图。图 3-2 中的类 “Interpreter” 和 “Parser” 用于解析描述虚拟世界的文件和机器人行为文件中的指令集。

【问题 1】(6 分)

根据说明中的描述，给出图 3-1 中 U1~U6 所对应的用例名。

【问题 2】(4 分)

图 3-1 中用例 U1~U6 分别与哪个 (哪些) 用例之间有关系，是何种关系？

【问题 3】(5 分)

根据说明中的描述，给出图 3-2 中 C1~C5 所对应的类名。

试题四 (15 分)

阅读下列说明和 C 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

在一块电路板的上下两端分别有 n 个接线柱。根据电路设计，用 $(i, \pi(i))$ 表示将上端接线柱 i 与下端接线柱 $\pi(i)$ 相连，称其为该电路板上的第 i 条连线。如图 4-1 所示的 $\pi(i)$ 排列为 {8,7,4,2,5,1,9,

3,10,6}。对于任何 $1 \leq i < j \leq n$, 第 i 条连线和第 j 条连线相交的充要条件是 $\pi(i) > \pi(j)$ 。

在制作电路板时, 要求将这 n 条连线分布到若干绝缘层上, 在同一层上的连线不相交。现在要确定将哪些连线安排在一层上, 使得该层上有尽可能多的连线, 即确定连线集 $\text{Nets} = \{(i, \pi(i)), 1 \leq i \leq n\}$ 的最大不相交子集。

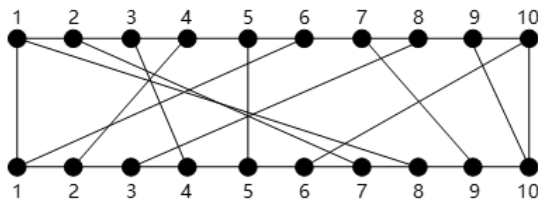


图 4-1 电路布线示意图

【分析问题】

记 $N(i, j) = \{t | (t, \pi(t)) \in \text{Nets}, t \leq i, \pi(t) \leq j\}$, $N(i, j)$ 的最大不相交子集为 $\text{MNS}(i, j)$, $\text{size}(i, j) = |\text{MNS}(i, j)|$ 。

经分析, 该问题具有最优子结构性质。对规模为 n 的电路布线问题, 可以构造如下递归式:

- $$(1) \text{ 当 } i=1 \text{ 时, } \text{size}(1, j) = \begin{cases} 0 & j < \pi(1) \\ 1 & \text{其他情况} \end{cases}$$
- $$(2) \text{ 当 } i > 1 \text{ 时, } \text{size}(i, j) = \begin{cases} \text{size}(i-1, j) & j < \pi(i) \\ \max\{\text{size}(i-1, j), \text{size}(i-1, \pi(i)-1) + 1\} & \text{其他情况} \end{cases}$$

【C 代码】

下面是算法的 C 语言实现。

(1) 变量说明。

$\text{size}[i][j]$: 上下端分别有 i 个和 j 个接线柱的电路板的第一层最大不相交连接数。

$\text{pi}[i]$: $\pi(i)$, 下标从 1 开始。

(2) C 程序。

```
#include "stdlib.h"
#include <stdio.h>
#define N 10 /*问题规模*/
int m=0; /*记录最大连接集合中的接线柱*/
Void maxNum(int pi[], int size[N+1][N+1], int n) /*求最大不相交连接数*/
{
    int i, j;
    for(j=0; j < pi[1]; j++) size[1][j] = 0; /*当 j < pi(1)时*/
    for(j=pi[1]; j <= n; j++) (1); /*当 j >= pi(1)时*/
    for(i=2; i <= n; i++) {
        for(j=0; j < pi[i]; j++) (2); /*当 j < pi[i]时*/
        for(j=pi[i]; j <= n; j++) { /*当 j >= pi[i]时, 考虑两种情况*/
            size[i][j] = size[i-1][j] >= size[i-1][pi[i]-1] + 1 ? size[i-1][j] : size[i-1][pi[i]-1] + 1;
        }
    }
    /*最大连接数*/
    size[n][n] = size[n-1][n] >= size[n-1][pi[n]-1] + 1 ? size[n-1][n] : size[n-1][pi[n]-1] + 1;
}
```

```

}
/*构造最大不相交连接集合，net[i]表示最大不相交子集中第 i 条连线的上端接线柱的序号*/
void constructSet (int pi[],int size[N+1][N+1],int n,int net[n]) {
    int i,j=n;
    m=0;
    for(i=n; i>1; i--) {          /*从后往前*/
        if(size[i][j]!=size[i-1][j]){ /*(i,pi[i])是最大不相交子集的一条连线*/
            (3); /*将 i 记录到数组 net 中，连接线数自增 1*/
            j= pi[i]-1; /*更新扩展连线柱区间*/
        }
    }
    if(j>=pi[1]) net[m++]=1; /*当 i=1 时*/
}

```

【问题 1】(6 分)

根据以上说明和 C 代码，填充 C 代码中的 (1) ~ (3)。

【问题 2】(6 分)

根据题干说明和以上 C 代码，算法采用了 (4) 算法设计策略。

函数 maxNum 和 constructSet 的时间复杂度分别为 (5) 和 (6) (用 O 表示)。

【问题 3】(3 分)

若连接排列为 {8,7,4,2,5,1,9,3,10,6}，即如图 4-1 所示，则最大不相交连接数为 (7)，包含的连线为 (8) (用 $(i, \pi(i))$ 的形式给出)。

试题五 (15 分)

阅读下列说明和 C++ 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某软件系统中，已设计并实现了用于显示地址信息的类 Address，如图 5-1 所示。现要求提供基于 Dutch 语言的地址信息显示接口，为了实现该要求并考虑到以后可能还会出现新的语言接口，决定采用适配器 (Adapter) 模式实现该要求。

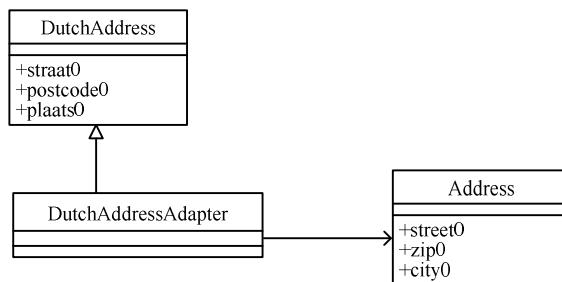


图 5-1 适配器模式类图

【C++代码】

```
#include <iostream>
using namespace std;

class Address {
public:
    void stree()    { /*实现代码省略*/ }
    void zip()      { /*实现代码省略*/ }
    void city()     { /*实现代码省略*/ }
    //其他成员省略
};

class DutchAddress {
public:
    virtual void straat()=0;
    virtual void postcode()=0;
    virtual void plaats()=0;
    //其他成员省略
};

class DutchAddressAdapter: public DutchAddress {
private:
    ____ (1) ____;
public:
    DutchAddressAdapter(Address *addr) {
        address=addr;
    }
    void straat() {
        ____ (2) ____;
    }
    void postcode(){
        ____ (3) ____;
    }
    void plaat(){
        ____ (4) ____;
    }
    //其他成员省略
};

void testDutch(DutchAddress *addr){
    addr->straat();
    addr->postcode();
    addr->plaats();
}
```

```

    }

    int main(){
        Address*addr = new Address();
        (5);
        cout<< "\n The DutchAddress\n"<< endl;
        testDutch(addrAdapter);
        return 0;
    }

```

【问题】(15 分)

阅读上述说明和 C++ 代码，将应填入的字句写在答题纸的对应栏内。

试题六 (15 分)

阅读下列说明和 Java 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某软件系统中，已设计并实现了用于显示地址信息的类 Address，如图 6-1 所示。现要求提供基于 Dutch 语言的地址信息显示接口，为了实现该要求并考虑到以后可能还会出现新的语言接口，决定采用适配器（Adapter）模式实现该要求。

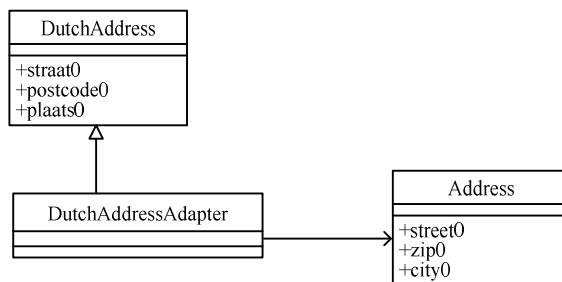


图 6-1 适配器模式类图

【Java 代码】

```

import java.util.*;

Class Address{
    public void street()    { //实现代码省略 }
    public void zip()      { //实现代码省略 }
    public void city()     { //实现代码省略 }
    // 其他成员省略
}

class DutchAddress{
    public void straat()    { //实现代码省略 }
    public void postcode()  { //实现代码省略 }
}

```

```
    public void plaats()          { //实现代码省略 }  
    //其他成员省略  
}
```

```
class DutchAddressAdapter extends DutchAddress {  
    private____(1)____;  
  
    public DutchAddressAdapter (Address addr){  
        address = addr;  
    }  
  
    public void straat() {  
        _____(2)____;  
    }  
  
    public void postcode() {  
        _____(3)____;  
    }  
  
    public void plaats(){  
        _____(4)____;  
    }  
    //其他成员省略  
}
```

```
class Test {  
    public static void main(String[] args) {  
        Address addr = new Address();  
        _____(5)____;  
        System.out.println("\n The DutchAddress\n");  
        testDutch(addrAdapter);  
    }  
  
    Static void testDutch(DutchAddress addr){  
        addr.straat();  
        addr.postcode();  
        addr.plaats();  
    }  
}
```

【问题】（15 分）

阅读上述说明和 Java 代码，将应填入的字句写在答题纸的对应栏内。

第17小时

2016年5月真题解析

上午基础知识

(1) 试题分析: VLIW (Very Long Instruction Word) 是指超长指令字, 一种非常长的指令组合, 它把许多条指令连在一起, 增加了运算的速度。

答案: (1) D

(2) 试题分析:

全相联地址映射: 主存的任意一块可以映射到 Cache 中的任意一块。

直接相联映射: 主存中的一块只能映射到 Cache 的一个特定的块中。

组相联映射: 各区中的某一块只能存入缓存的同组号的空间内, 但组内各块地址之间则可以任意存放。即从主存的组到 Cache 的组之间采用直接映射方式, 在两个对应的组内部采用全相联映射方式。

答案: (2) A

(3) 试题分析: 首先判断符号位 (最高位) 为 0, 表示该数为正数, 正数的原码、反码、补码不变; 符号位为 1, 表示该数为负数, 负数的补码为其原码, 符号位不变, 数值部分按位取反, 然后整个数加 1。90H = 1001 0000, 补码为 1001 0000, 原码为 1111 0000, 其中最高位代表符号位, 1110000 代表数值, 即 -112, $2x = -112$, 所以 $x = -56$ 。

答案: (3) B

(4) 试题分析: 移位运算符就是在二进制的基础上对数字进行平移。按照平移的方向和填充数字的规则分为三种: << (左移)、>> (带符号右移) 和 >>> (无符号右移)。在数字没有溢出的前提下, 对于正数和负数, 左移一位都相当于乘以 2 的一次方, 左移 n 位就相当于乘以 2 的 n 次方。

答案: (4) A

(5) 试题分析: $(B13FFH-A1000H)/1024 = 65KB$ 。

答案: (5) C

(6) 试题分析: 在单总线结构中, CPU 与主存之间、CPU 与 I/O 设备之间、I/O 设备与主存之间、各种设备之间都通过系统总线交换信息。单总线结构的优点是控制简单方便, 扩充方便。但由于所有设备部件均挂在单一总线上, 使这种结构只能分时工作, 即同一时刻只能在两个设备之间传送数据, 这就使系统总体数据传输的效率和速度受到限制, 这是单总线结构的主要缺点。

答案: (6) C

(7) 试题分析:

层次	名称	主要功能	主要设备及协议
7	应用层	实现具体的应用功能	POP3、FTP、HTTP、Telnet、SMTP、DHCP、TFTP、SNMP、DNS
6	表示层	数据的格式与表达、加密、压缩	
5	会话层	建立、管理和终止会话	
4	传输层	端到端的连接(端口)	TCP、UDP
3	网络层	分组传输和路由选择(IP)	三层交换机、路由器 ARP、RARP、IP、ICMP、IGMP
2	数据链路层	传送以帧为单位的信息(MAC)	网桥、交换机、网卡 PPTP、L2TP、SLIP、PPP
1	物理层	二进制传输(0/1)	中继器、集线器

答案: (7) B

(8) 试题分析: HTTPS (Hyper Text Transfer Protocol over Secure Socket Layer) 是以安全为目标的 HTTP 通道。即在 HTTP 下加入 SSL 层, HTTPS 的安全基础是 SSL, 因此加密的详细内容就需要 SSL。

答案: (8) B

(9) 试题分析: 端口扫描器通过选用远程 TCP/IP 不同端口的服务, 并记录目标给予的回答。通过这种方法, 可以搜集到很多关于目标主机的各种有用的信息。

答案: (9) B

(10) 试题分析: 我国《著作权法》第 9 条和《计算机软件保护条例》第 8 条的规定, 软件著作权人享有发表权和开发者身份权, 这两项权利与著作权人的人身是不可分离的主体。开发者的身份权不随软件开发者的消亡而丧失, 而且无时间限制。依据《计算机软件保护条例》第 23 条第 4 款的规定, 该项目负责人的行为侵犯了张某的开发者身份权及署名权。

答案: (10) A

(11) 试题分析: 在美国申请了专利, 在中国没有申请, 在中国不享有专利权。

答案: (11) D

(12) 试题分析: WAV 为微软公司 (Microsoft) 开发的一种声音文件格式, 它符合 RIFF (Resource Interchange File Format) 文件规范, 用于保存 Windows 平台的音频信息资源, 被 Windows 平台及其应用程序广泛地支持, 该格式也支持 MSADPCM、CCITT A LAW 等多种压缩运算法, 支持多种音频数字、取样频率和声道, 标准格式化的 WAV 文件和 CD 格式一样, 也是 44.1K 的取样频率, 16 位量化数字, 因此在声音文件质量和 CD 相差无几。

BMP (Bitmap) 是 Windows 操作系统中的标准图像文件格式, 可以分成两类: 设备相关位图 (DDB) 和设备无关位图 (DIB), 它采用位映射存储格式, 除了图像深度可选以外, 不采用其他任何压缩, 因此, BMP 文件所占用的空间很大。

MP3 是一种音频压缩技术, 其全称是动态影像专家压缩标准音频层面三 (Moving Picture Experts Group Audio Layer III), 简称为 MP3。

MOV 即 QuickTime 影片格式, 它是 Apple 公司开发的一种音频、视频文件格式, 用于存储常用的数字媒体类型。

答案: (12) D

(13) 试题分析: Microsoft Office PowerPoint 是微软公司的演示文稿软件; Adobe Photoshop 简称 PS, 是由 Adobe Systems 开发和发行的图像处理软件; Adobe Acrobat 是由 Adobe 公司开发的一款 PDF (Portable Document Format, 便携式文档格式) 编辑软件, 借助它可以以 PDF 格式制作和保存文档。

答案: (13) B

(14) 试题分析: DPI 表示单位 “像素/英寸”, 即指每英寸长度内所含有的像素是多少。24 位真彩色图像的每个像素使用三个字节表示 RGB 的分量值。一幅不经压缩的图像数据量的计算公式为: 字节数 = 图像水平分辨率 × 图像垂直分辨率 × 颜色深度 (位数) / 8。

根据题意, $3 \times 150 \times 4 \times 150 \times 24 / 8 = 810000$ 。

答案: (14) D

(15)、(16) 试题分析: 先找出关键路径为 ABDGIKL, 其长度为 22, 所以最短工期为 22 天。BD 是关键路径上的活动, 其总时差为 0, 不能耽搁, 所以 BD 最多延误 0 天不会影响总工期。

答案: (15) C (16) A

(17)、(18) 试题分析: 在结构化分析中, 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)。外部实体是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。

答案: (17) B (18) A

(19) 试题分析: A、B、C 均为需求分析阶段确定。

答案: (19) D

(20) 试题分析: 编译程序的功能是把用高级语言书写的源程序翻译成与之等价的目标程序。编译过程可以划分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成六个阶段。目标程序可以独立于源程序运行。

解释程序是一种语言处理程序,在词法、语法和语义分析方面与编译程序的工作原理基本相同,但在运行用户程序时,它是直接执行源程序或源程序的内部形式(中间代码)。因此,解释程序并不产生目标程序,这是它和编译程序的主要区别。

答案:(20) A

(21) 试题分析:脚本语言是为了缩短传统的编写—编译—链接—运行过程而创建的计算机编程语言。此命名起源于一个脚本“screenplay”,每次运行都会使对话框逐字重复。早期的脚本语言经常被称为批处理语言或工作控制语言,一个脚本通常用于解释运行而非编译。

答案:(21) C

(22) 试题分析:常用的中间代码的表达形式有语法树、后缀式、三地址代码。

答案:(22) B

(23) 试题分析:首先获得信息的是解释程序。

答案:(23) A

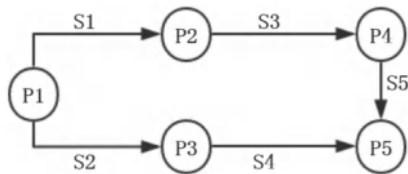
(24) 试题分析:当用户双击一个文件名时,Windows 系统通过建立的文件关联来决定使用什么程序打开该文件。

答案:(24) B

(25) 试题分析: $(6\text{ms} \times 10 \text{ 个磁道} + 100\text{ms} + 20\text{ms}) \times 100 \text{ 块} = 18000$

答案:(25) C

(26)、(27)、(28) 试题分析:先将信号量标于箭线之上,如下图所示。



再以此原则进行 PV 操作填充:

(1) 若从 P 进程节点引出某些信号量,则在 P 进程末尾对这些信号量执行 V 操作。例如, P1 引出了信号量 S1 与 S2,则 P1 末尾有 V(S1)V(S2)。

(2) 若有信号量指向某进程 P,则在 P 进程的开始位置有这些信号量的 P 操作。例如, S1 进程指向 P2,所以 P2 开始位置有 P(S1)。

答案:(26) C (27) B (28) B

(29) 试题分析:

公共耦合:通过一个公共数据环境相互作用的那些模块间的耦合。

控制耦合:如果一个模块通过传送开关、标志、名字等控制信息,明显地控制另一模块的功能,就是控制耦合。

标记耦合:一组模块通过参数表传递记录信息,就是标记耦合。这个记录是某一数据结构的子结构,而不是简单变量。

数据耦合：一个模块访问另一个模块时，彼此之间是通过简单的数据参数（不是控制参数、公共数据结构或外部变量）来交换输入、输出信息的。

答案：(29) A

(30) 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。

答案：(30) D

(31) 试题分析：模块设计讲究高内聚、低耦合，而非功能越单纯越好。

答案：(31) D

(32)、(33) 试题分析：仓库风格的优点包括：①解决问题的多方法性；②具有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：测试困难、不能保证有好的求解方案、效率低、开发成本高、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。

答案：(32) D (33) D

(34)、(35) 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，因此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，其效率较高，运用一定的技巧，减少了桩模块和驱动模块的开发。本题缺乏严谨性。

答案：(34) C (35) B

(36) 试题分析：环形复杂性计算公式为 $V(G) = E - N + 2$ ，其中， E 是流图中边（Edge）的条数， N 是节点（Node）的个数。所以， $V(G) = E - N + 2 = 11 - 10 + 2 = 3$ 。

答案：(36) C

(37)、(38) 试题分析：继承是父类和子类之间共享数据和方法的机制；覆盖是子类的方法覆盖了基类的方法，以实现不同的功能，或者对父类的功能进行扩充。

答案：(37) B (38) C

(39) 试题分析：在 UML 用例图中，参与者是指存在于系统外部并直接与系统进行交互的人、系统、子系统或类的外部实体的抽象。

答案：(39) A

(40) 试题分析：两个类之间可以由不同角色标识存在多个关联。

答案：(40) B

(41)、(42)、(43) 试题分析：该图为活动图，注意在状态图中，每条箭线上都会有事件，而活动图则不一定有。图中 I 所代表的是同步条，也称分支，代表从此处开始有多个并行分支；而 II 所对应的是分支条件。

答案: (41) D (42) A (43) B

(44) 试题分析: 抽象工厂模式可以向客户端提供一个接口, 使客户端在不指定产品的具体情况下, 创建多个产品族中的产品对象。

外观模式: 定义了一个高层接口, 为子系统中的一个组接口提供一个一致的界面, 从而简化子系统的使用。

答案: (44) C

(45) 试题分析: 命令模式将一个请求封装为一个对象, 从而可以用不同的请求对客户进行参数化, 将请求排队或记录请求日志, 支持可撤销的操作。

答案: (45) A

(46) 试题分析: 观察者模式 (Observer) 是定义对象间的一种一对多的依赖关系, 当一个对象的状态发生改变时, 所有依赖于它的对象都得到通知并自动更新。

答案: (46) C

(47) 试题分析: 享元模式的特点是复用内存中已存在的对象, 减少系统创建对象的实例。组合模式 (Composite Pattern) 有时候又叫作部分—整体模式, 它在树型结构的问题中, 模糊了简单元素和复杂元素的概念, 客户程序可以向处理简单元素一样来处理复杂元素, 从而使得客户程序与复杂元素的内部结构解耦。

答案: (47) B

(48) 试题分析: 先分析词, 即词法分析; 而分析词的组合, 即语法分析, 移进—递归是自下而上的语法分析方法。

答案: (48) B

(49) 试题分析: abab 能被顺序执行。

答案: (49) B

(50) 试题分析: 考察地址与传值, 在主函数中调用 $f(5, x)$ 之后:

$f()$ 函数中的 $x=5$, $a=1$ 。

$x=2 \times x+1$, 则 $x=11$ 。

$a=a+x$, 则 $a=12$ 。由于 a 是以传址的方式传入的参数, 所以主函数中的 x 与其值相同, 也为 12。打印结果应为 12。

答案: (50) D

(51) 试题分析: 物理独立性是指内模式发生变化, 只需要调整模式与内模式之间的映像, 而不用修改应用程序; 逻辑独立性是指模式发生变化, 只需要调整外模式与模式之间的映像, 而不用修改应用程序。

答案: (51) D

(52) 试题分析: 数据库设计中规范化是在逻辑设计阶段进行的一项工作, 该工作负责把关系模式规范化, 以减少冗余, 以及一定程度上消除修改异常, 插入异常及删除异常。

答案: (52) C

(53) 试题分析: 将本题关系模式 R 的函数依赖关系表达为: C 依赖于 A 和 B , B 依赖于 C 。因此 A 的入度为 0, 所以它必然为候选关键字的一部分。

通过 A 与 B 组合或 A 与 C 组合, 均能遍历全图, 所以候选关系字有 AB 和 AC , 因此 A 、 B 、 C 均是主属性。

答案: (53) B

(54)、(55)、(56) 试题分析:

(54) 选 B, 元件号和供应商的组合键能推导出该关系模式的所有属性, 因此为主键。

(55) 选 C, 因为关系模式 P 存在冗余、插入异常和删除异常等问题。为了解决这一问题需要将元件关系分解。 A 选项、 B 选项和 D 选项是有损连接的, 而且不保持函数依赖性, 故分解是错误的。例如, 分解为 A 选项、 B 选项和 D 选项后, 用户无法查询某元件由哪些供应商供应, 原因是分解有损连接且不保持函数依赖。

元件关系存在非主属性对码的部分函数依赖: (元件号, 供应商) 供应商所在地, 但是供应商 $\rightarrow$ 供应商所在地, 故原关系模式元件是非 2NF 的。分解后的关系模式元件 1、元件 2 和元件 3 消除了非主属性对码的部分函数依赖, 同时不存在传递依赖, 故达到 3NF。

答案: (54) B (55) C (56) C

(57) 试题分析: 以 a 、 b 、 c 、 d 、 e 的顺序入栈, 还要求第一个出栈的是 d , 所以只能先 $abcd$ 入栈, 然后 d 出栈, 这样栈里面还有 abc 三个元素, e 还没有入栈, e 可以有四个时机入栈, 就是四种合法的出栈顺序。①在栈里面有 abc 的时候入栈, 合法的出栈顺序是 $decba$; ②在栈里面的 c 出栈后 e 再入栈, 合法的出栈顺序是 $dceba$; ③在栈里面的 bc 出栈后 e 再入栈, 合法的出栈顺序是 $dcbea$; ④在栈里面的 abc 都出栈后 e 再入栈, 合法的出栈顺序是 $dcbae$ 。

所以, 合法的出栈顺序总共是四种。

答案: (57) A

(58) 试题分析: 31 是 27 的父亲节点, 31 必须在 27 前面。

答案: (58) C

(59) 试题分析: 本题考察二叉树节点数的计算。一棵高度为 h 的二叉树, 节点数最多时即为满二叉树。而高度为 h 的满二叉树有 2^h-1 个节点, 所以一棵二叉树的高度 (即层数) 为 h , 则它最多有 2^h-1 个节点。

答案: (59) D

(60) 试题分析: 考察数据结构的折半查找算法。 B 选项错误之处在于, 要查找 $a[9]$ 元素, 第一次比较的是 $A[7]$ (下标计算方法为: $(1+13)/2=7$), 第二次比较的是 $A[10]$ (下标计算方法为: $(8+13)/2=10$)。

答案: (60) B

(61) 试题分析: 使用队列对图进行广度优先遍历。

答案: (61) C

(62) ~ (65) 试题分析: 这是典型的 0-1 背包问题。动态规划算法中, 自底向上 (递推):

从小范围递推计算到大范围, 可以看到装第一个和第五个物品的价值是最高的, 这时候 $V=12$, 占了 6 的重量了, 只能装物品 2, 价值 15。第二个问题是部分背包, 部分背包的时候计算每个物品的单位重量价值多少, 单位重量 $v=\{3\ 1.5\ 5/6\ 0.8\ 1.5\}$, 可以看到 1、2、5 的单位价值最高, 选择 1、2、5 后背包重量还只有 8, 还有两个重量可以选择, 选择 3 得 $5/3$ 的价值, 就是 1.67, 所以第三问为 16.67; 复杂度没有进行指数级别的运算, 问题 1 只需要找 n 个物品与价值 W 相乘, 问题 3 计算单位物品价值然后考虑背包的大小就可以了。

答案: (62) C (63) A (64) D (65) B

(66)、(67) 试题分析: 众所周知, FTP 协议的端口号一个是 20、一个是 21, 其中, 20 为数据口, 21 为控制口, 但是由于 FTP 工作方式的不同, 所以端口号不会总是 20。模式分为主动模式和被动模式, 主动模式是从服务器端向客户端发起连接, 被动模式是客户端向服务器端发起连接, 二者的共同点是都使用 21 端口进行用户验证和管理, 差别在于传送数据的方式不同, 主动模式的 FTP 服务器数据端口固定在 20, 而被动模式则在 1025~65535 中随机产生。

答案: (66) D (67) A

(68) 试题分析: 检查错误时, 使用由近及远的原则意味着先要确认本机协议的栈有没有问题, 所以可以用 ping127.0.0.1 来检查本机的 TCP/IP 协议。

答案: (68) C

(69) 试题分析: 仅有 192.168.2.254 与当前主机在同一个网段, 所以仅有该地址能充当网关的角色。

答案: (69) C

(70) 试题分析: 在 SNMP 协议中, 团体名相当于一个组, 在进行管理时, 是以团体名为单位进行管理的, 其作用域也在相同的团体名之内, 选项中只有 A 和 Manager 2 是 public 2, 在同一个基本作用域。

答案: (70) A

(71)~(75) 试题分析:

翻译:

在物理安全和信息安全领域, 访问控制是对一个地方或其他资源的访问限制。访问的行为可能意味着消费、输入或使用。允许访问资源被称为授权。

访问控制机制 (71) 用户 (代表用户执行的进程) 和系统资源 (如应用程序、操作系统、防火墙、路由器、文件和数据库) 之间的连接。系统必须首先对用户进行身份验证。通常, 身份验证功能决定用户是否 (72) 访问系统。然后, 访问控制功能决定是否允许该用户指定的访问权限。安全管理员维护一个授权数据库, 该数据库指定允许该用户访问哪些资源的类型。访问控制函数咨询此数据库以确定是否 (73) 访问权。审计功能监视并保存用户访问系统资源的记录。

在实践中, 一些 (74) 可以合作共享访问控制函数。所有操作系统都至少有一个基本的, 而且在许多情况下是一个相当健壮的访问控制组件。附加安全包可以添加到操作系统的 (75) 访问控制功能。特定的应用程序或实用程序, 如数据库管理系统, 也包括访问控制功能。外部设备, 如

防火墙，也可以提供访问控制服务。

- | | | | |
|-------------|--------|---------|--------|
| (71) A. 合作 | B. 协调 | C. 连接 | D. 调解 |
| (72) A. 拒绝 | B. 允许 | C. 禁止 | D. 排斥 |
| (73) A. 开放 | B. 监视 | C. 授予 | D. 寻找 |
| (74) A. 组件 | B. 用户 | C. 机制 | D. 算法 |
| (75) A. 遥远的 | B. 原始的 | C. 被控制的 | D. 自动的 |

答案：(71) C (72) B (73) C (74) B (75) D

下午案例分析

试题一（15 分）

【问题 1】（2 分）

答案：E1：客户；E2：管理员。

【问题 2】（4 分）

答案：D1：预定表；D2：客户信息记录表；D3：设施表；D4：设备表。

【问题 3】（6 分）

答案：数据流名称：预订确认信息；起点：4（确认预定）；终点：E1。

数据流名称：客户信息；起点：E1；终点：2（临时预定）。

数据流名称：客户记录；起点：D2；终点：4（确认预定）。

数据流名称：客户记录；起点：D2；终点：5（变更预定）。

数据流名称：预定信息；起点：D1；终点：6（要求付款）。

数据流名称：客户记录；起点：D2；终点：6（要求付款）。

【问题 4】（3 分）

答案：在图 1-1 中增加外部实体“第三方 Email 系统”，将临时预订/预订/变更确认信息终点均修改至“第三方 Email 系统”。

在图 1-2 中增加外部实体“第三方 Email 系统”，增加加工“发送邮件”，将临时预订或变更预订的确认信息终点均修改至“发送邮件”加工，并增加从 D2 到“发送邮件”加工的数据流“电子邮件地址”，再从发送邮件加工引出数据流，临时预订或变更预订确认信息，终点为第三方 Email 系统。

试题二（15 分）

【问题 1】（4 分）

答案：配送员一配送一地点，多对多联系。

【问题 2】（6 分）

答案：(a) 商品数量，仓库信息

(b) 支付凭证编号

(c) 商品条码, 订单 ID

【问题 3】(5 分)

答案: 在营业员与订单之间增加联系“送货上门”, 并将联系记录于订单关系中, 即在订单关系中增加“营业员 ID”属性作为外键。

试题三 (15 分)

【问题 1】(6 分)

答案: U1、U2、U3、U4、U5、U6: Run、Step、Write、Move、Left、Read。
用例图和初始类图分别如图 3-1 和图 3-2 所示。

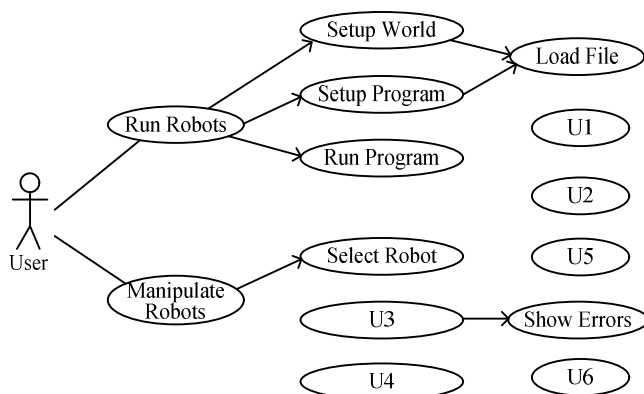


图 3-1 用例图

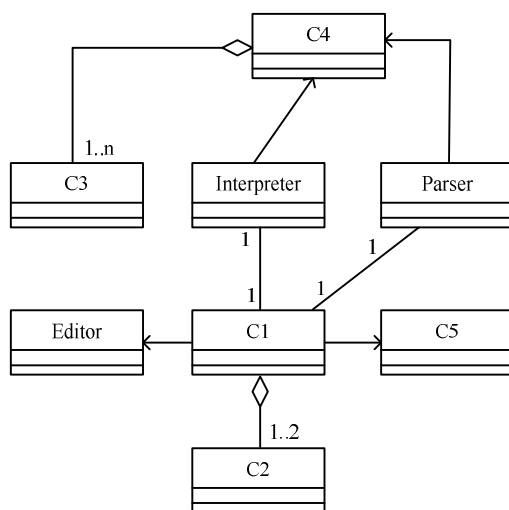


图 3-2 初始类图

【问题 2】(4 分)

答案: U1、U2 和 Run Program 有泛化关系; U3、U4、U5、U6 和 Select Robot 有扩展关系。

【问题 3】(5 分)

答案:

C1: 文件

C2: 机器人在虚拟世界的行为

C3: Instruction

C4: InstructionSet

C5: 仿真系统

试题四 (15 分)

【问题 1】(6 分)

答案:

(1) $\text{size}[i][j]=1$

(2) $\text{size}[i][j]=\text{size}[i-1][j]$

(3) $\text{net}[m++] = i$

【问题 2】(6 分)

答案:

(4) 动态规划算法

(5) $O(n^2)$

(6) $O(n)$

【问题 3】(3 分)

答案:

(7) 4

(8) $(9, \pi(9), (7, \pi(7)), (5, \pi(5)), (3, \pi(3)))$

试题五 (15 分)

答案:

(1) $\text{Address} * \text{address}$

(2) $\text{address} \rightarrow \text{street}()$

(3) $\text{address} \rightarrow \text{zip}()$

- (4) address->city()
- (5) DutchAddress*addr=new DutchAddressAdaptor(addr)

试题六 (15 分)

答案:

- (1) Address address
- (2) address.street()
- (3) address.zip()
- (4) address.city()
- (5) DutchAddress addrAdapter=new DutchAddressAdaptor(addr)

2016 年 11 月考试真题（上午基础知识）



- 在程序运行过程中，CPU 需要将指令从内存中取出并加以分析和执行。CPU 依据（1）来区分在内存中以二进制编码形式存放的指令和数据。

- 计算机在一个指令周期的过程中，为了从内存读取指令操作码，首先要将(2)的内容送到地址总线上。

- 设 16 位浮点数，其中阶符 1 位、阶码值 6 位、数符 1 位、尾数 8 位。若阶码用移码表示，尾数用补码表示，则该浮点数所能表示的数值范围是 (3) 。

- (3) A. $-2^{64} \sim (1-2^{-8})2^{64}$
 B. $-2^{63} \sim (1-2^{-8})2^{63}$

$$C. -2^{64} \sim (1-2-(1-2^{-8})2^{64} \sim (1-2^{-8})2^{64}$$

$$D. -(1-2^{-8})2^{63} \sim (1-2^{-8})2^{63}$$

- 已知数据信息为 16 位, 最少应附加 (4) 位校验位, 以实现海明码纠错。
(4) A. 3 B. 4 C. 5 D. 6
- 将一条指令的执行过程分解为取指、分析和执行三步, 按照流水方式执行, 若取指时间 $t_{\text{取指}}=4\Delta t$ 、分析时间 $t_{\text{分析}}=2\Delta t$ 、执行时间 $t_{\text{执行}}=3\Delta t$, 则执行完 100 条指令需要的时间为 (5) Δt 。
(5) A. 200 B. 300 C. 400 D. 405
- 以下关于 Cache 与主存间地址映射的叙述中, 正确的是 (6)。
(6) A. 操作系统负责管理 Cache 与主存之间的地址映射
B. 程序员需要通过编程来处理 Cache 与主存之间的地址映射
C. 应用软件对 Cache 与主存之间的地址映射进行调度
D. 由硬件自动完成 Cache 与主存之间的地址映射
- 可用于数字签名的算法是 (7)。
(7) A. RSA B. IDEA C. RC4 D. MD5
- (8) 不是数字签名的作用。
(8) A. 接收者可验证消息来源的真实性 B. 发送者无法否认发送过该消息
C. 接收者无法伪造或篡改消息 D. 可验证接收者合法性
- 在网络设计和实施过程中要采取多种安全措施, 其中 (9) 是针对系统安全需求的措施。
(9) A. 设备防雷击 B. 入侵检测
C. 漏洞发现与补丁管理 D. 流量控制
- (10) 的保护期限是可以延长的。
(10) A. 专利权 B. 商标权 C. 著作权 D. 商业秘密权
- 甲公司软件设计师完成了一项涉及计算机程序的发明。之后, 乙公司软件设计师也完成了与甲公司软件设计师相同的涉及计算机程序的发明。甲、乙公司于同一天向专利局申请发明专利。此情形下, (11) 是专利权申请人。
(11) A. 甲公司 B. 甲、乙两公司
C. 乙公司 D. 由甲、乙公司协商确定的公司
- 甲、乙两厂生产的产品类似, 而且产品都使用 B 商标。两厂于同一天向商标局申请商标注册, 而且申请注册前两厂均未使用 B 商标。此情形下, (12) 能核准注册。
(12) A. 甲厂 B. 由甲、乙两厂抽签确定的厂
C. 乙厂 D. 甲、乙两厂
- 在 FM 方式的数字音乐合成器中, 改变数字载波频率可以改变乐音的 (13), 改变它的信号幅度可以改变乐音的 (14)。
(13) A. 音调 B. 音色 C. 音高 D. 音质
(14) A. 音调 B. 音域 C. 音高 D. 带宽

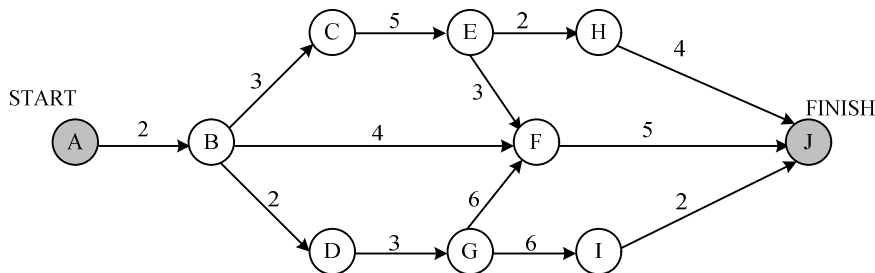
- 结构化开发方法中，__(15)__ 主要包含对数据结构和算法的设计。

(15) A. 体系结构设计 B. 数据设计
C. 接口设计 D. 过程设计

- 在敏捷过程的开发方法中，__(16)__ 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

- 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间（天），则完成该项目的最少时间为__(17)__天。活动 BC 和 BF 最多可以晚开始__(18)__天而不会影响整个项目的进度。



(17) A. 11 B. 15 C. 16 D. 18

(18) A. 0 和 7 B. 0 和 11 C. 2 和 7 D. 2 和 11

- 成本估算时，__(19)__ 方法以规模作为成本的主要因素，考虑多个成本驱动因子。该方法包括三个阶段模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。

(19) A. 专家估算 B. Wolverton C. COCOMO D. COCOMO II

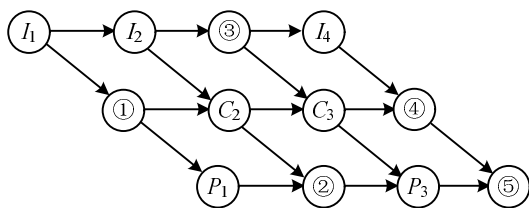
- 逻辑表达式求值时常采用短路计算方式。“&&”“||”“!”分别表示逻辑与、或、非运算，“&&”“||”为左结合，“!”为右结合，优先级从高到低为“!”“&&”“||”。对逻辑表达式“ $x \&\&(y \parallel !z)$ ”进行短路计算方式求值时，__(20)__。

(20) A. x 为真，则整个表达式的值即为真，不需要计算 y 和 z 的值
B. x 为假，则整个表达式的值即为假，不需要计算 y 和 z 的值
C. x 为真，再根据 z 的值决定是否需要计算 y 的值
D. x 为假，再根据 y 的值决定是否需要计算 z 的值

- 常用的函数参数传递方式有传值与传引用两种，__(21)__。

(21) A. 在传值方式下，形参与实参之间互相传值
B. 在传值方式下，实参不能是变量
C. 在传引用方式下，修改形参实质上改变了实参的值

- D. 在传引用方式下, 实参可以是任意的变量和表达式
- 二维数组 $a[1 \dots N, 1 \dots M]$ 可以按行存储或按列存储。对于数组元素 $a[i, j] (1 \leq i, j \leq N)$, 当 (22) 时, 在按行和按列两种存储方式下, 其偏移量相同。
(22) A. $i \neq j$ B. $i = j$ C. $i > j$ D. $i < j$
 - 实时操作系统主要用于有实时要求的过程控制等领域。实时系统对于来自外部的事件必须在 (23)。
(23) A. 一个时间片内进行处理
B. 一个周转时间内进行处理
C. 一个机器周期内进行处理
D. 被控对象规定的时间内作出及时响应并对其进行处理
 - 假设某计算机系统中只有一个 CPU、一台输入设备和一台输出设备, 若系统中有四个作业 T_1 、 T_2 、 T_3 和 T_4 , 系统采用优先级调度, 而且 T_1 的优先级 $> T_2$ 的优先级 $> T_3$ 的优先级 $> T_4$ 的优先级。每个作业 T_i 具有三个程序段: 输入 I_i 、计算 C_i 和输出 $P_i (i = 1, 2, 3, 4)$, 其执行顺序为 $I_i \rightarrow C_i \rightarrow P_i$ 。这四个作业的各个程序段并发执行的前驱图如下所示。图中①、②分别为 (24), ③、④、⑤分别为 (25)。



- (24) A. I_2, P_2 B. I_2, C_2 C. C_1, P_2 D. C_1, P_3
(25) A. C_2, C_4, P_4 B. I_2, I_3, C_4 C. I_3, P_3, P_4 D. I_3, C_4, P_4
- 假设段页式存储管理系统中的地址结构如下图所示, 则系统 (26)。

31	24	23	13	12	0
段 号		页 号		页内地址	

- (26) A. 最多可有 256 个段, 每个段的大小均为 2048 个页, 页的大小为 8K
B. 最多可有 256 个段, 每个段最大允许有 2048 个页, 页的大小为 8K
C. 最多可有 512 个段, 每个段的大小均为 1024 个页, 页的大小为 4K
D. 最多可有 512 个段, 每个段最大允许有 1024 个页, 页的大小为 4K
- 假设系统中有 n 个进程共享三台扫描仪, 并采用 PV 操作实现进程同步与互斥。若系统信号量 S 的当前值为 -1, 进程 P_1 、 P_2 又分别执行了一次 $P(S)$ 操作, 那么信号量 S 的值应为 (27)。
(27) A. 3 B. -3 C. 1 D. -1
 - 某字长为 32 位的计算机的文件管理系统采用位示图 (bitmap) 记录磁盘的使用情况。若磁盘

的容量为 300GB，物理块的大小为 1MB，那么位示图的大小为 (28) 个字。

(28) A. 1200 B. 3200 C. 6400 D. 9600

- 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，对产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程和销售过程可能会发生变化。该软件的开发最不宜采用 (29) 模型，主要是因为这种模型 (30)。

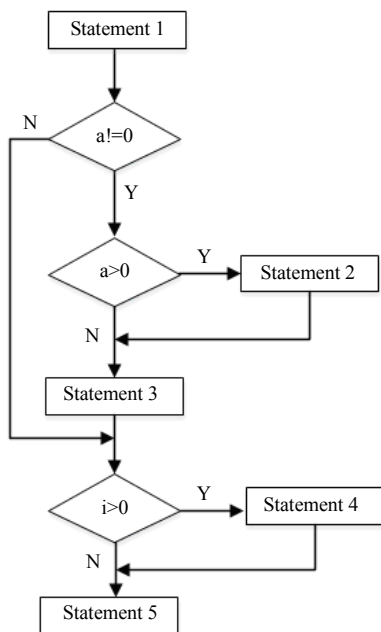
(29) A. 瀑布 B. 原型 C. 增量 D. 喷泉

(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

- (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

- 对下图所示的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 (32) 个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为 (33)。



(32) A. 3 B. 4 C. 6 D. 8

(33) A. 1 B. 2 C. 3 D. 4

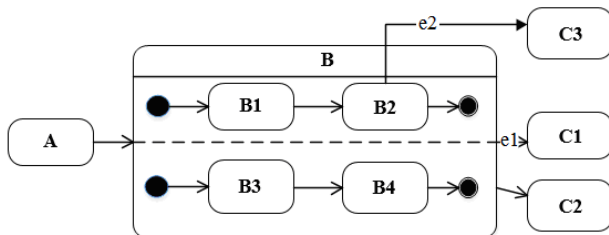
- 计算机系统的 (34) 可以用 $MTBF/(1+MTBF)$ 来度量，其中，MTBF 为平均失效间隔时间。

(34) A. 可靠性 B. 可用性 C. 可维护性 D. 健壮性

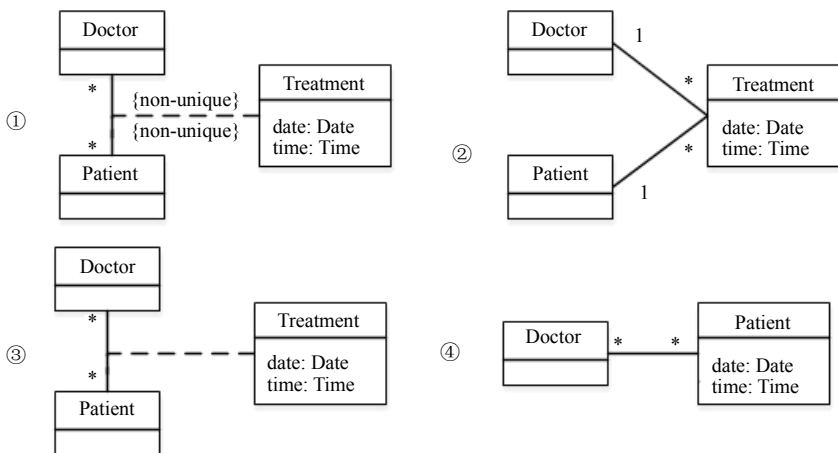
- 以下关于软件测试的叙述中，不正确的是 (35)。

(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
- 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 (36) 内聚。
 - (36) A. 逻辑 B. 过程 C. 通信 D. 内容
- 在面向对象方法中, 不同对象收到同一消息可以产生完全不同的结果, 这一现象称为 (37)。在使用时, 用户可以发送一个通用的消息, 而实现的细节则由接收对象自行决定。
 - (37) A. 接口 B. 继承 C. 覆盖 D. 多态
- 在面向对象方法中, 支持多态的是 (38)。
 - (38) A. 静态分配 B. 动态分配 C. 静态类型 D. 动态绑定
- 面向对象分析的目的是获得对应用问题的理解, 其主要活动不包括 (39)。
 - (39) A. 认定并组织对象 B. 描述对象间的相互作用
 - C. 面向对象程序设计 D. 确定基于对象的操作
- 如下所示的 UML 状态图中, (40) 时, 不一定会离开状态 B 。



- (40) A. 状态 B 中的两个结束状态均达到
- B. 在当前状态为 $B2$ 时, 事件 $e2$ 发生
- C. 事件 $e2$ 发生
- D. 事件 $e1$ 发生
- 以下关于 UML 状态图中转换 (Transition) 的叙述中, 不正确的是 (41)。
 - (41) A. 活动可以在转换时执行也可以在状态内执行
 - B. 监护条件只有在相应的事件发生时才进行检查
 - C. 一个转换可以有事件触发器、监护条件和一个状态
 - D. 事件触发转换
- 下图中①②③④所示是 UML (42)。现有场景: 一名医生 (Doctor) 可以治疗多位病人 (Patient), 一位病人可以由多名医生治疗, 一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时, 需要存储治疗 (Treatment) 的日期和时间。下图中的 (43) 是描述此场景的模型。



- (42) A. 用例图 B. 对象图 C. 类图 D. 协作图
(43) A. ① B. ② C. ③ D. ④

- (44) 模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下 (45) 情况适合选用该模式。

- ①一个客户需要使用一组相关对象
②一个对象的改变需要改变其他对象
③需要使用一个算法的不同变体
④许多相关的类仅仅是行为有异

- (44) A. 命令 (Command) B. 责任链 (Chain of Responsibility)
C. 观察者 (Observer) D. 策略 (Strategy)

- (45) A. ①② B. ②③ C. ③④ D. ①④

- (46) 模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创建不同的表示。以下 (47) 情况适合选用该模式。

- ①抽象复杂对象的构建步骤
②基于构建过程的具体实现构建复杂对象的不同表示
③一个类仅有一个实例
④一个类的实例只能有几个不同状态组合中的一种

- (46) A. 生成器 (Builder) B. 工厂方法 (Factory Method)
C. 原型 (Prototype) D. 单例 (Singleton)

- (47) A. ①② B. ②③ C. ③④ D. ①④

- 由字符 a 、 b 构成的字符串中，若每个 a 后至少跟一个 b ，则该字符串集合可用正规式表示为 (48)。

- (48) A. $(b|ab)^*$ B. $(ab^*)^*$ C. $(a^*b^*)^*$ D. $(a|b)^*$

- 乔姆斯基 (Chomsky) 将文法分为四种类型，程序设计语言的大多数语法现象可用其中的 (49)

描述。

(49) A. 上下文有关文法

B. 上下文无关文法

C. 正规文法

D. 短语结构文法

- 运行下面的 C 程序代码段, 会出现 (50) 错误。

```
int k=0;
for(;k<100;);
{k++;}
```

(50) A. 变量未定义

B. 静态语义

C. 语法

D. 动态语义

- 在数据库系统中, 一般由 DBA 使用 DBMS 提供的授权功能为不同用户授权, 其主要目的是保证数据库的 (51)。

(51) A. 正确性

B. 安全性

C. 一致性

D. 完整性

- 给定关系模式 $R(U, F)$, 其中, U 为关系模式 R 中的属性集, F 是 U 上的一组函数依赖。假设 $U = \{A_1, A_2, A_3, A_4\}$, $F = \{A_1 \rightarrow A_2, A_1 A_2 \rightarrow A_3, A_1 \rightarrow A_4, A_2 \rightarrow A_4\}$, 那么关系 R 的主键应为 (52)。
函数依赖集 F 中的 (53) 是冗余的。

(52) A. A_1

B. $A_1 A_2$

C. $A_1 A_3$

D. $A_1 A_2 A_3$

(53) A. $A_1 \rightarrow A_2$

B. $A_1 A_2 \rightarrow A_3$

C. $A_1 \rightarrow A_4$

D. $A_2 \rightarrow A_4$

- 给定关系 $R(A, B, C, D)$ 和关系 $S(A, C, E, F)$, 对其进行自然连接运算 $R \bowtie S$ 后的属性列为 (54) 个, 与 $\sigma_{R.B > S.E}(R \bowtie S)$ 等价的关系代数表达式为 (55)。

(54) A. 4

B. 5

C. 6

D. 8

(55) A. $\sigma_{2>7}(R \times S)$

B. $\pi_{1,2,3,4,7,8}(\sigma_{1=5^{\wedge}2>7^{\wedge}3=6}(R \times S))$

C. $\sigma_{2>7}(R \times S)$

D. $\pi_{1,2,3,4,7,8}(\sigma_{1=5^{\wedge}2>7^{\wedge}3=6}(R \times S))$

- 下列查询 $B = \text{“大数据”}$ 且 $F = \text{“开发平台”}$, 结果集属性列为 A, B, C, F 的关系代数表达式中, 查询效率最高的是 (56)。

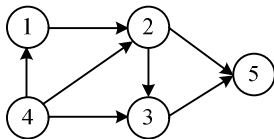
(56) A. $\pi_{1,2,3,8}(\sigma_{2=\text{“大数据”} \wedge 1=5^{\wedge}3=6^{\wedge}8=\text{“开发平台”}}(R \times S))$

B. $\pi_{1,2,3,8}(\sigma_{1=5^{\wedge}3=6^{\wedge}8=\text{“开发平台”}}(\sigma_{2=\text{“大数据”}}(R \times S)))$

C. $\pi_{1,2,3,8}(\sigma_{2=\text{“大数据”} \wedge 1=5^{\wedge}3=6}(R \times \sigma_{4=\text{“开发平台”}}(S)))$

D. $\pi_{1,2,3,8}(\sigma_{1=5^{\wedge}3=6}(\sigma_{2=\text{“大数据”}}(R) \times \sigma_{4=\text{“开发平台”}}(S)))$

- 拓扑序列是有向无环图中所有顶点的一个线性序列, 若有向图中存在弧 $\langle v, w \rangle$ 或存在从顶点 v 到 w 的路径, 则在该有向图的任一拓扑序列中, v 一定在 w 之前。下面有向图的拓扑序列是 (57)。



(57) A. 41235

B. 43125

C. 42135

D. 41325

- 设有一个包含 n 个元素的有序线性表。在等概率情况下删除其中的一个元素, 若采用顺序存

储结构, 则平均需要移动 (58) 个元素; 若采用单链表存储, 则平均需要移动 (59) 个元素。

- (58) A. 1 B. $(n-1)/2$ C. $\log n$ D. n
 (59) A. 0 B. 1 C. $(n-1)/2$ D. $n/2$

- 具有三个节点的二叉树有 (60) 种形态。

- (60) A. 2 B. 3 C. 5 D. 7

- 以下关于二叉排序树 (或二叉查找树、二叉搜索树) 的叙述中, 正确的是 (61)。

- (61) A. 对二叉排序树进行先序、中序和后序遍历, 都可以得到节点关键字的有序序列
 B. 含有 n 个节点的二叉排序树的高度为 $\lfloor \log_2 n \rfloor + 1$
 C. 从根到任意一个叶子节点的路径上, 节点的关键字呈现有序排列的特点
 D. 从左到右排列同层次的节点, 其关键字呈现有序排列的特点

- 下表为某文件中字符的出现频率, 采用霍夫曼编码对下列字符编码, 则字符序列 “bee” 的编码为 (62); 编码 110001001101 对应的字符序列为 (63)。

字符	a	b	c	d	e	f
频率 (%)	45	13	12	16	9	5

- (62) A. 10111011101 B. 10111001100 C. 001100100 D. 110011011

- (63) A. bad B. bee C. face D. bace

- 两个矩阵 $A_{m \times n}$ 和 $B_{n \times p}$ 相乘, 用基本的方法进行, 则需要的乘法次数为 $m \times n \times p$ 。多个矩阵相乘满足结合律, 不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定 $M_i, M_{(i+1)}, \dots, M_j$ 多个矩阵连乘的最优顺序, 即所需要的乘法次数最少。最少的乘法次数用 $m[i, j]$ 表示, 其递归式定义为:

$$m[i, j] = \begin{cases} 0 & i \geq j \\ \min_{i \leq k \leq j} \{m[i, k] + m[k+1, j] + p_{i-1} p_k p_j\} & i < j \end{cases}$$

其中, i, j 和 k 为矩阵下标, 矩阵序列中 M_i 的维度为 $(p_{i-1}) \times p_i$ 。采用自底向上的方法实现该算法来确定 n 个矩阵相乘的顺序, 其时间复杂度为 (64)。若四个矩阵 M_1, M_2, M_3, M_4 相乘的维度序列为 2、6、3、10、3, 采用上述算法求解, 则乘法次数为 (65)。

- (64) A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n^3 \lg n)$

- (65) A. 156 B. 144 C. 180 D. 360

- 以下协议中属于应用层协议的是 (66), 该协议的报文封装在 (67)。

- (66) A. SNMP B. ARP C. ICMP D. X.25

- (67) A. TCP B. IP C. UDP D. ICMP

- 某公司内部使用 wb.xyz.com.cn 作为访问某服务器的地址, 其中 wb 是 (68)。

- (68) A. 主机名 B. 协议名 C. 目录名 D. 文件名

- 如果路由器收到了多个路由协议转发的关于某个目标的多条路由, 那么决定采用哪条路由的策略是 (69)。

(69) A. 选择与自己路由协议相同的 B. 选择路由费用最小的
C. 比较各个路由的管理距离 D. 比较各个路由协议的版本

- 与地址 220.112.179.92 匹配的路由表的表项是 (70)。

(70) A. 220.112.145.32/22 B. 220.112.145.64/22
C. 220.112.147.64/22 D. 220.112.177.64/22

- Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a (71), open or closed. In this respect software systems differ profoundly from computers, buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more (72) than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some (73) fashion, and the complexity of the whole increases much more than linearly.

The complexity of software is a(an) (74) property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence. Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities (75) in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

(71) A. task B. job C. subroutine D. program
(72) A. states B. parts C. conditions D. expressions
(73) A. linear B. nonlinear C. parallel D. additive
(74) A. surface B. outside C. exterior D. essential
(75) A. fixed B. included C. ignored D. stabilize

第19小时

2016年11月考试真题（下午案例分析）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

试题一（15分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某证券交易所为了方便提供证券交易服务，欲开发一个证券交易平台，该平台的主要功能如下：

- （1）开户。根据客户服务助理提交的开户信息进行开户，并将客户信息存入客户记录中，账户信息（余额等）存入账户记录中。
- （2）存款。客户可以向其账户中存款，根据存款金额修改账户余额。
- （3）取款。客户可以从其账户中取款，根据取款金额修改账户余额。
- （4）证券交易。客户和经纪人都可以进行证券交易（客户通过在线方式，经纪人通过电话），将交易信息存入交易记录中。
- （5）检查交易。平台从交易记录中读取交易信息，将交易明细返回给客户。

现在采用结构化方法对该证券交易平台进行分析与设计，得到如图 1-1 所示的上下文数据流图和如图 1-2 所示的 0 层数据流图。

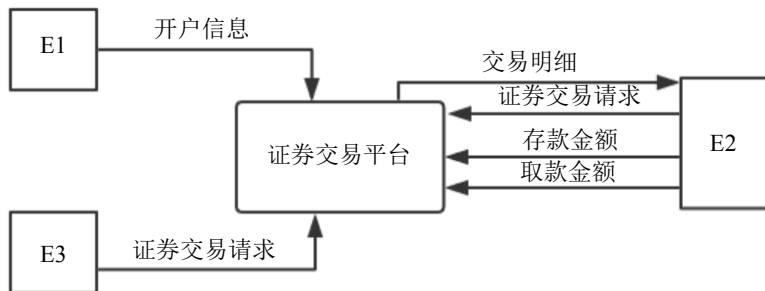


图 1-1 上下文数据流图

【问题 1】（3 分）

使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

【问题 2】（3 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1~D3 的名称。

【问题 3】（4 分）

根据说明和图中的术语，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】（5 分）

实际的证券交易通常是在证券交易中心完成的，因此，该平台的“证券交易”功能需将交易信息传递给证券交易中心。针对这个功能需求，需要对图 1-1 和图 1-2 进行哪些修改，请用 200 字以内的文字加以说明。

试题二（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某宾馆为了有效地管理客房资源，满足不同客户的需求，拟构建一套宾馆信息管理系统，以方便宾馆管理及客房预订等业务活动。

【需求分析结果】

该系统的部分功能及初步需求分析的结果如下：

（1）宾馆有多个部门，部门信息包括部门号、部门名称、电话、经理。每个部门可以有多名员工，每名员工只属于一个部门；每个部门只有一名经理，负责管理本部门。

（2）员工信息包括员工号、姓名、岗位、电话、工资，其中，员工号唯一标识员工关系中的一个元组，岗位有经理、业务员。

（3）客房信息包括客房号（如 1301、1302 等）、客房类型、收费标准、入住状态（已入住或未入住），其中客房号唯一标识客房关系中的一个元组，不同客房类型具有不同的收费标准。

(4) 客户信息包括客户号、单位名称、联系人、联系电话、联系地址, 其中客户号唯一标识客户关系中的一个元组。

(5) 客户预订客房时, 需要填写预订申请。预订申请信息包括申请号、客户号、入住时间、入住天数、客房类型、客房数量, 其中, 一个申请号唯一标识预订申请中的一个元组; 一位客户可以有多个预订申请, 但一个预订申请对应唯一的一位客户。

(6) 当客户入住时, 业务员根据客户的预订申请负责安排入住客房事宜。安排信息包括客房号、姓名、性别、身份证号、入住时间、天数、电话, 其中客房号、身份证号和入住时间唯一标识一次安排。一名业务员可以安排多个预订申请, 一个预订申请只由一名业务员安排, 而且可以安排多间同类型的客房。

【概念模型设计】

根据需求阶段收集的信息, 设计的实体联系图如图 2-1 所示。

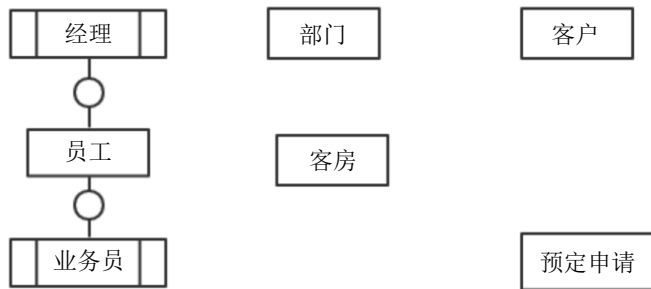


图 2-1 实体联系图

【关系模式设计】

部门 (部门号, 部门名称, 经理, 电话)

员工 (员工号, (a), 姓名, 岗位, 电话, 工资)

客户 ((b), 联系人, 联系电话, 联系地址)

客房 (客房号, 客房类型, 收费标准, 入住状态)

预订申请 ((c), 入住时间, 天数, 客房类型, 客房数量)

安排 (申请号, 客房号, 姓名, 性别, (c), 天数, 电话, 业务员)

【问题 1】(4 分)

根据问题描述, 补充四个联系, 完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替, 联系类型为 1:1、1:n 和 m:n (或 1:1、1:* 和 *:*)。

【问题 2】(8 分)

根据题意, 将关系模式中的 (a) ~ (d) 补充完整, 并填入答题纸对应的位置上。

【问题 3】(3 分)

“客房”关系模式是否存在规范性问题, 请用 100 字以内的文字解释你的观点 (若存在问题,

应说明如何修改“客房”关系模式)。

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某种出售罐装饮料的自动售货机（Vending Machine）的工作过程描述如下：

（1）顾客选择所需购买的饮料及数量。

（2）顾客从投币口向自动售货机中投入硬币（该自动售货机只接收硬币）。硬币器收集投入的硬币并计算其对应的价值。如果所投入的硬币足够购买所需数量的这种饮料且饮料数量足够，则推出饮料，计算找零，顾客取走饮料和找回的硬币；如果投入的硬币不够或所选购的饮料数量不足，则提示用户继续投入硬币或重新选择饮料及数量。

（3）一次购买结束之后，将硬币器中的硬币移走（清空硬币器），等待下一次交易。自动售货机还设有一个退币按钮，用于退还顾客所投入的硬币。已经成功购买饮料的钱是不会被退回的。

现在采用面向对象方法分析和设计该自动售货机的软件系统，得到如图 3-1 所示的用例图，其中，“购买饮料”的用例规约描述如下。

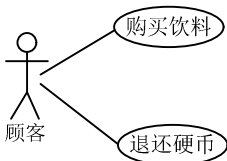


图 3-1 用例图

参与者：顾客。

主要事件流：

- （1）顾客选择需要购买的饮料和数量，投入硬币。
- （2）自动售货机检查顾客是否投入足够的硬币。
- （3）自动售货机检查饮料储存仓中所选购的饮料是否足够。
- （4）自动售货机推出饮料。
- （5）自动售货机返回找零。

各选事件流：①若投入的硬币不足，则给出提示并退回到 1；②若所选购的饮料数量不足，则给出提示并退回到 1。

根据用例“购买饮料”得到自动售货机的四个状态：空闲状态、准备服务状态、可购买状态和饮料出售状态，对应的状态图如图 3-2 所示。所设计的类图如图 3-3 所示。

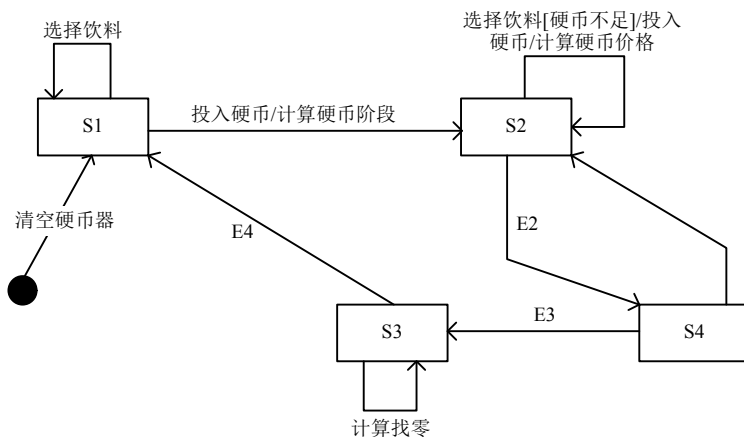


图 3-2 状态图

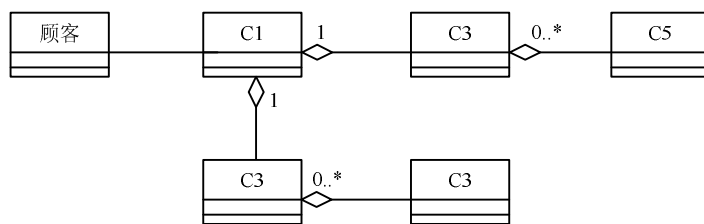


图 3-3 类图

【问题 1】(6 分)

根据说明中的描述，使用说明中的术语，给出图 3-2 中 S1~S4 所对应的状态名。

【问题 2】(4 分)

根据说明中的描述，使用说明中的术语，给出图 3-2 中 E1~E4 所对应的事件名。

【问题 3】(5 分)

根据说明中的描述，使用说明中的术语，给出图 3-3 中 C1~C5 所对应的类名。

试题四 (15 分)

阅读下列说明和 C 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

模式匹配是指给定主串 t 和子串 s ，在主串 t 中寻找子串 s 的过程，其中 s 称为模式。如果匹配成功，返回 s 在 t 中的位置，否则返回 -1。

KMP 算法用 next 数组对匹配过程进行了优化。KMP 算法的伪代码描述如下：

- (1) 在串 t 和串 s 中，分别将起始下标设为 $i=j=0$ 。
- (2) 如果串 t 和串 s 都含有字符，则循环执行下列操作：

- 1) 如果 $j = -1$ 或 $t[i] = s[j]$, 则将 i 和 j 分别加 1, 继续比较 t 和 s 的下一个字符。
 - 2) 否则, 将 j 向右滑动到 $\text{next}[j]$ 的位置, 即 $j = \text{next}[j]$ 。
 - (3) 如果 s 中所有字符均已比较完毕, 则返回匹配的起始位置 (从 1 开始); 否则返回 -1。
- 其中, next 数组根据子串 s 求解。求解 next 数组的代码已由 get_next 函数给出。

【C 代码】

(1) 常量和变量说明。

t 、 s : 长度为 l_t 和 l_s 的字符串。

next : next 数组, 长度为 l_s 。

(2) C 程序。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/*求 next[] 的值*/
void get_next( int *next, char *s, int ls) {
    int i=0, j=-1;
    next[0]=-1;          /*初始化 next[0]*/
    while(i < ls){        /*还有字符*/
        if(j==-1 || s[i]==s[j]){ /*匹配*/
            j++;
            i++;
            if(s[i]==s[j])
                next[i]=next[j];
            else
                next[i]=j;
        }
        else
            j=next[j];
    }
}

int kmp(int*next, char*t, char*s, int lt, int ls)
{
    int i=0, j=0;
    while (i < lt && (1) ){
        if(j==-1 || (2) ){
            i++;
            j++;
        } else
            (3);
    }
    if(j >= ls)
        return (4);
    else
        return -1;
}
```

【问题 1】(8 分)

根据题干说明, 填充 C 代码中的 (1) ~ (4)。

【问题 2】(2 分)

根据题干说明和 C 代码, 分析出 KMP 算法的时间复杂度为 (5) (主串和子串的长度分别为 l_t 和 l_s , 用符号 O 表示)。

【问题 3】(5 分)

根据 C 代码, 字符串 BBABBCAC 的 next 数组元素值为 (6) (直接写素值, 之间用逗号隔开)。若主串为 AABBCBBABBCACCD, 子串为 BBABBCAC, 则函数 KMP 的返回值为 (7)。

试题五 (15 分)

阅读下列说明和 C++ 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某发票(Invoice)由抬头(Head)部分、正文部分和脚注(Foot)部分构成。现采用装饰(Decorator)模式实现打印发票的功能, 得到如图 5-1 所示的类图。

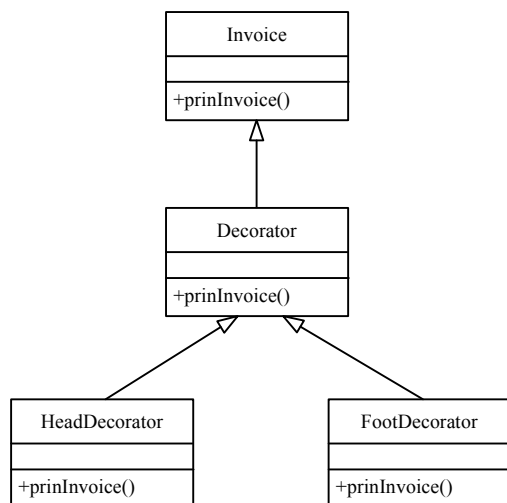


图 5-1 类图

【C++代码】

```

#include <iostream>
using namespace std;
class Invoice{
public:
    (1) {
        cout<<"This is the content of the invoice!"<<endl;
    }
};
  
```

```

class Decorator: public Invoice{
    Invoice*ticket;
public:
    Decorator(Invoice*t)    { ticket = t; }
    void printInvoice(){
        if(ticket!=NULL)
            (2);
    }
};

class HeadDecorator: public Decorator{
public:
    HeadDecorator(Invoice*t): Decorator(t) { }
    void printInvoice() {
        cout<< "This is the header of the invoice!"<< endl;
        (3);
    }
};

class FootDecorator: public Decorator{
public:
    FootDecorator(Invoice*t): Decorator(t) { }
    void printInvoice(){
        (4);
        cout<< "This is the footnote of the invoice!"<< endl;
    }
};

int main(void) {
    Invoice t;
    FootDecorator f(&t);
    HeadDecorator h(&f);
    h.printInvoice();
    cout<<"-----"<<endl;
    FootDecorator a(NULL);
    HeadDecorator b((5));
    b.printInvoice();
    return 0;
}

```

程序的输出结果为:

```

This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
-----
This is the header of the invoice!
This is the footnote of the invoice!

```

【问题】(15 分)

阅读上述说明和 C++ 代码, 将应填入的字句写在答题纸的对应栏内。

试题六 (15 分)

阅读下列说明和 Java 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某发票(Invoice)由抬头(Head)部分、正文部分和脚注(Foot)部分构成。现采用装饰(Decorator)模式实现打印发票的功能, 得到如图 6-1 所示的类图。

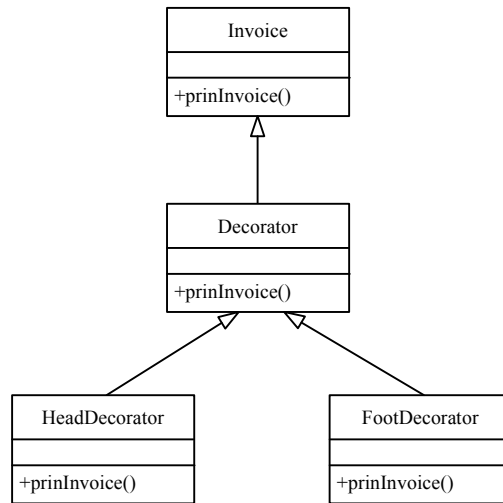


图 6-1 类图

【Java 代码】

```
class Invoice{
    public void printInvoice(){
        system.out.println ("This is the content of the invoice!");
    }
}
class Decorator extends Invoice{
    protected Invoice ticket;
    public Decorator(Invoice t){
        ticket=t;
    }
    public void printInvoice(){
        if(ticket!= null)
            (1) ;
    }
}
class HeadDecorator extends Decorator{
    public HeadDecorator(Invoice t){
        super(t);
    }
}
```

```
public void printInvoice(){
    system.out.println( "This is the header of the invoice! ");
    _____(2)_____;
}
}
class FootDecorator extends Decorator{
    public FootDecorator(Invoice t){
        super(t);
    }
    public void printInvoice(){
        _____(3)_____;
        system.out.println("This is the footnote of the invoice!");
    }
}
}
Class test {
    public static void main(String[] args){
        Invoice t=new Invoice();
        Invoice ticket;
        ticket= _____(4)_____;
        ticket.printInvoice();
        System.out.println("-----");
        ticket=_____ (5) _____;
        ticket.printInvoice();
    }
}
}
```

程序的输出结果为：

```
This is the header of the invoice!
This is the content of the invoice!
This is the footnote of the invoice!
-----
This is the header of the invoice!
This is the footnote of the invoice!
```

【问题】（15 分）

阅读上述说明和 Java 代码，将应填入的字句写在答题纸的对应栏内。

第20小时

2016年11月真题解析

上午基础知识

(1) 试题分析：指令和数据均存放在内存中，通常由 PC（程序计数器）提供存储单元地址取出的是指令，由指令地址码部分提供存储单元地址取出的是数据。因此通过不同的寻址方式来区别指令和数据。

答案：(1) B

(2) 试题分析：PC（程序计数器）是用于存放下一条指令所在单元的地址。当执行一条指令时，处理器首先需要从 PC 中取出指令在内存中的地址，通过地址总线寻址获取。

答案：(2) C

(3) 试题分析：尾数的位数越多，表示的精度越高；阶码的位数越多，表示的数值范围越大。因此，浮点数可以表示的范围为：最大正数为 $(1-2^{-M}) \times 2^{(2^R-1)}$ ，最小负数为 $-1 \times 2^{(2^R-1)}$ ，其中， M 表示尾数位数， R 表示阶码位数。

答案：(3) B

(4) 试题分析：海明码的构造方法是：在数据位之间插入 k 个校验位，通过扩大码距来实现检错和纠错。设数据位是 n 位，校验位是 k 位，则 n 和 k 必须满足关系 $2^k \geq k+n+1$ ，当 $n=16$ 时， k 取 5。

答案：(4) C

(5) 试题分析：计算公式：第一条指令执行的时间 + (指令数 - 1) × 各指令段执行时间中最大的执行时间。因此， $4\Delta t + 3\Delta t + 2\Delta t + (100-1) \times 4\Delta t = 405\Delta t$ 。

答案：(5) D

(6) 试题分析：在程序的执行过程中，Cache 与主存的地址映射是由硬件自动完成的。

答案: (6) D

(7) 试题分析: IDEA 算法和 RC4 算法都是对称加密算法, 只能用来进行数据加密; MD5 算法是消息摘要算法, 只能用来生成消息摘要, 无法进行数字签名; RSA 算法是典型的非对称加密算法, 主要具有数字签名和验签的功能。

答案: (7) A

(8) 试题分析: 数字签名是信息的发送者才能产生的一段数字串, 别人无法伪造, 这段数字串同时也是对信息的发送者发送信息真实性的一个有效证明。不能验证接收者的合法性。

答案: (8) D

(9) 试题分析: 在网络设计和实施过程中要采取多种安全措施, 其中漏洞发现与补丁管理是针对系统安全需求的措施, 其他几项都不属于。

答案: (9) C

(10) 试题分析: 专利权可以分为发明专利(保护期限 20 年)、新型实用设计专利(保护期限 10 年)、外观设计专利(保护期限 10 年), 专利期满后专利权终止, 因此专利期限是不可延长的。

商标权(有效期限 10 年)到期后可以无限续期, 每次续期有效期还是 10 年, 有效期满未续期的会被注销商标。

著作权保护期限要分开说: 作者的署名权、修改权和保护作品完整权的保护期限是没有限制的; 作品的发表权、财产权的保护权是作者的终身和死后 50 年, 该权利不可延长。

商业秘密权在法律上没有规定, 只要商业秘密未泄露出去则一直受法律保护。

答案: (10) B

(11) 试题分析: 在审查过程中, 对于不同的申请人同日(指申请日, 有优先权的指优先权日)就同样的发明创造分别提出专利申请, 并且这两件申请符合授予专利权的其他条件, 应当根据专利法实施细则第四十一条第一款的规定, 通知申请人自行协商确定申请人。同一天申请的情况处理方式如下: ①两个申请人作为一件申请的共同申请人; ②其中一方放弃权利并从另一方得到适当的补偿。如果双方协商不成, 则两件申请都不授予专利权。

答案: (11) D

(12) 试题分析: 按照商标法第 29 条和实施条例 19 条规定, 同一天申请的, 初步审定并公告使用在先的, 驳回其他人的申请; 均未使用或无法证明的, 各自协商; 不愿协商或协商不成的, 抽签决定; 不抽签的, 视为放弃。

答案: (12) B

(13)、(14) 试题分析: 声音的三个主观属性, 即音量(响度)、音调、音色(音品)。

音调主要由声音的频率决定。

音色是声音的特色, 根据不同的音色, 即使在同一音高和同一声音强度的情况下, 也能区分出是不同乐器或人声发出的。

音高即音的高度, 是人耳对声音调子高低的主观感觉。主要取决于频率的高低与响度的大小。

答案: (13) A (14) C

(15) 试题分析:

体系结构设计: 主要是指要开发的系统中包含哪些部件, 这些部件与部件之间的关系就是体系结构的设计。

数据设计: 也称为数据库设计, 主要包含数据库的设计和这个数据所包含的核心表的设计。

接口设计: 用于子系统和模块之间或内部系统和外部系统之间的各种交互, 如功能描述、输入输出的定义、错误处理的设计。

过程设计: 也称为模块详细设计, 主要是详细模块的实现算法, 以及模块所使用的数据结构。

答案: (15) D

(16) 试题分析: 以上四种方法都属于敏捷开发方法。

极限编程 XP: 近似螺旋的开发方法, 把整个开发过程分解为相对比较小而简单的周期, 通过积极的沟通反馈, 开发人员和客户都比较清楚当前的开发进度、需要解决的问题等, 根据这些实际情况去调整开发过程, 这是极限编程的思想。

并列争球法: 就是通常所说的 Scrum。Scrum 是一个增量、迭代的开发过程。在这个框架中, 整个开发过程由若干个短的迭代周期组成, 一个短的迭代周期称为一个 Sprint, 每个 Sprint 的建议长度是 2~4 周。在 Scrum 中, 使用产品 Backlog 来管理产品的需求, 产品团队总是先开发对客户具有较高价值的需求。挑选的需求在 Sprint 计划会议上经过讨论、分析和估算得到相应的任务列表, 称为 Sprint Backlog。在每个迭代结束时, Scrum 团队将递交潜在的可交付的产品增量。

答案: (16) C

(17)、(18) 试题分析: 找出关键路径并计算出关键路径、总工期、各个活动的总时差, 得到总工期为 18 天, BC 总时差为 0, BF 总时差为 7。

答案: (17) D (18) A

(19) 试题分析:

专家估算: 根据专家的行业经验和历史数据对软件开发过程的成本进行估算。

Wolverton: 也叫作 loc 方法, 通过执行的源代码行数来进行成本估算, 估算准确性低, 现在已经不用了。

COCOMO: 构造性成本模型, 是一种参数化的成本估算方法。例如, 将软件的难度、规模等作为参数进行成本估算。

COCOMO II: COCOMO II 是 COCOMO 的改进版, 把最新的软件开发方法考虑在内。COCOMO II 由以下三个不同的计算模型组成。

应用组合模型: 适用于使用现代 GUI 工具开发的项目。

早起开发模型: 适用于在软件架构确定之前对软件进行粗略的成本和事件估算, 包含了一系列新的成本和进度估算方法。基于功能点或代码行。

结构化后期模型: 是 COCOMO II 中最详细的模型, 用在整体软件架构已确定之后, 包含最新的成本估算、代码行计算方法。

答案: (19) D

(20) 试题分析: 在进行逻辑与“&&”运算时, 只有当两个操作数的值为真, 最后的结果才会为真。因此, 一旦 x 的值为假, 整个运算表达式的值则为假。

答案: (20) B

(21) 试题分析: 传值调用最显著的特征就是被调用的函数内部对形参的修改不影响实参的值。引用调用是将实参的地址传递给形参, 使得形参的地址就是实参的地址。

答案: (21) C

(22) 试题分析:

I/J	$J=1$	$J=2$	$J=3$
$I=1$	1	2	3
$I=2$	4	5	6
$I=3$	7	8	9

按行存储: 123456789; 按列存储: 147258369。可以看到当 $i=j$ 时, 其偏移量相同。

答案: (22) B

(23) 试题分析: 实时操作系统是指系统能及时响应外部事件的请求, 在规定的时间内完成对该事件的处理, 并控制所有实时任务协调一致地运行。

答案: (23) D

(24)、(25) 试题分析: 题目中一共有三个设备, 分别是一个 CPU、一台输入设备和一台输出设备。输入设备对应的程序段输入 I_i , 而 CPU 对应的程序段计算 C_i , 输出设备对应的程序段输出 P_i 。而每个作业都分为这三段, 各段之间有顺序关系。再结合图中已经给出的节点, 我们不难发现, 第一行是输入, 第二行是计算, 而第三行的节点数是输出节点。因此可以知道①、②分别为 C_1 、 P_3 , ③、④、⑤分别为 I_3 、 C_4 、 P_4 。

答案: (24) C (25) D

(26) 试题分析: 页内地址为 13 位, 页号地址为 11 位, 段号地址为 8 位。根据公式, 可以分别计算段号、页号和页内地址最大的寻址空间。存储管理系统中的地址长度均表示为最大的寻址空间。8 位段号, 11 位页号, 13 位页内地址, 所以 $2^8=256$ 段, 每段有 $2^{11}=2048$ 页, 页的大小为 $2^3=8K$ 。

答案: (26) B

(27) 试题分析: $P(S)$ 操作是申请资源, 是减量操作; $V(S)$ 操作是释放资源, 是增量操作。所以执行两次 $P(S)$ 后 S 的值为 -3。

答案: (27) B

(28) 试题分析: 位示图(bitmap)是指一位表示一个物理块的状态, 即一位表示 1MB 的使用与否, 300G 的磁盘有 $300 \times 1024 = 307200$ 个物理块, 需要 307200 位, $307200/32 = 9600$ 字 (32

位系统的字长就是 32 位)。

答案: (28) D

(29)、(30) 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。

瀑布模型不适用于需求多变或早期需求不确定的开发过程。瀑布模型难以适应变化的需求。

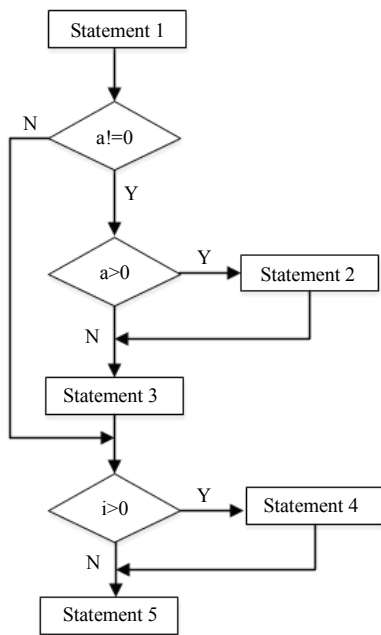
答案: (29) A (30) C

(31) 试题分析: 软件质量特性有六个: 功能性、可靠性、易用性、效率、维护性、可移植性。其中可移植性包括适应性、易安装性、共存性、易替换性、依存性。而易理解性是易用性中的一种。

答案: (31) D

(32)、(33) 试题分析: 路径覆盖是指使程序中每条路径都至少执行一次。图中每个判断都需要两个用例, 因此需要 6 个测试用例。

计算环路复杂度: $V(G) = E - N + 2$, 其中, E 是流图中边的条数, N 是节点数。



答案: (32) C (33) D

(34) 试题分析: MTBF 称为平均故障间隔时间, 用于衡量产品可靠性的一个指标。

可靠性: 在规定的时间内, 产品保持正常功能的一种能力。

答案: (34) A

(35) 试题分析: 测试用例 (Test Case) 是为某个特殊目标而编制的一组测试输入、执行条件和预期结果, 以便测试某个程序路径或核实是否满足某个特定需求。在设计测试用例时, 应该包括

合理的输入条件和不合理的输入条件。

测试是为了发现尽可能多的缺陷，不是为了说明软件中没有缺陷。成功的测试在于发现了迄今尚未发现的缺陷。

答案：(35) B

(36) 试题分析：通信内聚是指模块内各功能部分使用了相同的输入数据或产生了相同的输出数据。

1) 功能内聚：指模块内所有元素共同完成某一功能，联系紧密，缺一不可，是最强的内聚类型。

2) 顺序内聚：指一个模块中各个处理元素都密切相关于同一功能且必须顺序执行，前一功能元素的输出是下一功能元素的输入。即一个模块完成多个功能，这些模块又必须顺序执行。

3) 通信内聚：指模块内所有处理元素都在同一个数据结构上操作，或者各处理使用相同的输入数据或产生相同的输出数据。

4) 过程内聚：构件或操作的组合方式，允许在调用前面的构件或操作之后，马上调用后面的构件或操作，即使两者之间没有数据进行传递。

5) 时间内聚：把需要同时执行的动作组合在一起形成的模块为时间内聚模块，所有的动作需在同一个时间段内执行。

6) 逻辑内聚：把几种相关的功能组合在一起，每次被调用时，由传送给模块参数来确定该模块应完成哪一种功能。

7) 偶然内聚：模块内各部分之间没有联系，或者即使有联系也很松散，是内聚度最低的模块。

答案：(36) C

(37) 试题分析：多态是面向对象程序设计的一个重要特征，是指同一操作作用于不同的对象，可以有不同的解释，产生不同的执行结果。

继承是指它可以使用现有类的所有功能，并在无须重新编写原来的类的情况下对这些功能进行扩展。

答案：(37) D

(38) 试题分析：绑定是一个把过程调用和响应调用需要执行的代码加以结合的过程。在一般程序设计语言中，绑定是在编译时进行的，叫作静态绑定。动态绑定则是在运行时进行的，因此，一个给定的过程调用和代码结合是直到调用发生时才进行的。

动态绑定是和类的继承及多态相联系的。在继承关系中，子类是父类的一个特例，所以父类对象可以出现的地方，子类也可以出现。因此在运行过程中，当一个对象发送消息请求服务时，要根据接受对象的具体情况将请求的操作和实现的方法进行连接，即动态绑定。

答案：(38) D

(39) 试题分析：面向对象分析包括五个活动：认定对象、组织对象、描述对象之间的相互作用、定义对象的操作、定义对象的内部信息。

答案：(39) C

(40) 试题分析：事件 e2 发生时，如果当前状态是 B2 时，它就会离开状态 B 转入 C3；如果当

前状态不是 B2，那么 e2 发生时是不会离开状态 B 的。因此，事件 e2 发生时不一定会离开状态 B。

答案：(40) C

(41) 试题分析：转移(Transition)是两个状态之间的一种关系，表示对象将在源状态(Source State)中执行一定的动作，并在某个特定事件在发生且某个特定的警戒条件满足时进入目标状态(Target State)。转移有两个状态，原状态和目标状态。

答案：(41) C

(42)、(43) 试题分析：类图是最常用的 UML 图，显示出类、接口和它们之间的静态结构和关系，包含三个组成部分。类名、属性(Attribute)、类提供的方法。类名不能省略，其他组成部分可以省略。

对象图描述的是参与交互的各个对象在交互过程中某一时刻的状态。对象图可以被看作是类图在某一时刻的实例。①允许存在多个相同的元素数据，显然不合理；②根据试题描述，病人和医生之间应该有关联关系，显然该图不符合；③符合题目中的描述；④只表达了医生和病人的关系，没有表达存储治疗的时间和日期，缺少一个关联类，因此该图也不符合。

答案：(42) C (43) C

(44)、(45) 试题分析：

命令模式：将一个请求封装为一个对象，从而可以用不同的请求对客户进行参数化。

责任链：使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。

观察者模式：定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

策略模式：定义一系列的算法，把每一个算法封装起来，并且它们可以互相替换。本模式使得算法可以独立于使用它的客户而变化。

答案：(44) D (45) C

(46)、(47) 试题分析：

建造者模式(生成器模式)：将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。建造者模式将部件和其组装过程分开，一步一步创建一个复杂的对象。

工厂方法：也叫作虚拟构造器模式，它定义一个用于创建对象的接口，让子类决定实例化哪一个类，使一个类的实例化延迟到其子类。

原型模式：用原型实例指定创建对象的种类，并且通过复制这些原型创建新的对象。

单例模式：确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例，这个类称为单例类，它提供全局访问的方法。

答案：(46) A (47) A

(48) 试题分析：正规式中“|”表示“或”的意思，“*”表示“*”前的字符或字符串出现了 0 次或多次。

答案：(48) A

(49) 试题分析: 文法类型: 0 型、1 型、2 型、3 型。

上下文无关文法: 形式语言理论中一种重要的变化文法, 用来描述上下文无关语言, 在乔姆斯基分层中称为 2 型文法。由于程序设计语言的语法基本上都是上下文无关的文法, 因此应用十分广泛。

答案: (49) B

(50) 试题分析: 错误可以分为静态错误和动态错误两类。动态错误被称作动态语义错误, 是指源程序中的逻辑错误, 它们发生在程序运行的时候, 如死循环, 变量取值为零; 静态错误又可以分为语法错误和静态语义错误。

答案: (50) D

(51) 试题分析: DMBS 是数据库管理系统, 主要用来保证数据库的安全性和完整性。而 DBA 通过授权功能为不同用户授权, 主要的目的是保证数据的安全性。

答案: (51) B

(52)、(53) 试题分析: 依据 $U_1=\{A_1, A_2, A_3, A_4\}$, $F=\{A_1 \rightarrow A_2, A_1 A_2 \rightarrow A_3, A_1 \rightarrow A_4, A_2 \rightarrow A_4\}$, 构造出依赖关系图之后, A_1 是入度为 0 的节点且从 A_1 出发能遍历全图, 因此 A_1 为主键。 $A_1 \rightarrow A_2$, $A_2 \rightarrow A_4$, 利用传递率得到 $A_1 \rightarrow A_4$, 因此 $A_1 \rightarrow A_4$ 是冗余。

答案: (52) A (53) C

(54)、(55) 试题分析: 自然连接去掉重复属性列, 因此只剩下 R.A、B、R.C、D、E、F 这六个属性。第 (55) 题注意 B 选项与 D 选项的区别, 单引号的含义指属性的值是 7。

答案: (54) C (55) B

(56) 试题分析: π 表示关系代数投影操作, σ 表示关系代数选择操作。

避免查询一开始就进行笛卡尔积操作, 尽量对数据进行筛选过滤后再进行笛卡尔积操作, 减少参与笛卡尔积运算的数据量。

答案: (56) D

(57) 试题分析: 有向图的拓扑序列通俗一点来讲, 就是依次遍历没有前驱节点的节点。因为在某一时刻没有前驱节点的节点有可能存在多个, 所以一个图的拓扑排序可能有多个。

4 号节点没有前驱, 所以拓扑排序的第一个元素是 4。当 4 访问完了就可以访问 1, 1 号访问完了就可以访问 2, 2 号访问完了就可以访问 3 或 5。所以, 拓扑排序的结果为 41235。

答案: (57) A

(58)、(59) 试题分析: 略。

答案: (58) B (59) A

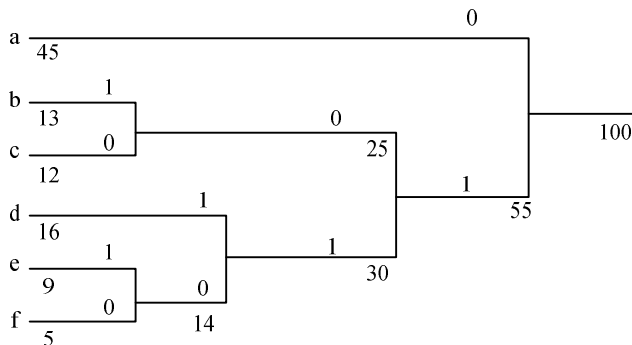
(60) 试题分析: N 个节点 ($N \geq 2$) 的二叉树有 $A[N] = \sum_{M=0}^{N-1} (A[M] \times A[N-M-1])$ 种形态。

答案: (60) C

(61) 试题分析: 二叉排序树又称为二叉查找树。或者是一棵空树, 或者是具有下列性质的二叉树: ①若左子树不空, 则左子树上所有节点的值均小于它的根节点的值; ②若右子树不为空, 则右子树上所有节点的值均大于或等于它的根节点的值; ③左、右子树分别为二叉排序树。

答案: (61) D

(62)、(63) 试题分析: 考察构造霍夫曼编码树。



如上图所示, 霍夫曼编码是依次把概率最小的两个元素分别置为 0 (更小的为 0) 和 1 (相对大的为 2), 然后从右往左写出 1/0 序列, 即 f: 1100、e: 1101、d: 111、c: 100、b: 101、a: 0。

答案: (62) A (63) C

(64)、(65) 试题分析:

方法一: 矩阵链乘法。

一个给定的矩阵序列 $A_1A_2\ldots A_n$ 计算连乘乘积有不同的结合方法, 并且在结合时, 矩阵的相对位置不能改变, 只能相邻结合。根据矩阵乘法的公式, 10×100 和 100×5 的矩阵相乘需要做 $10 \times 100 \times 5$ 次标量乘法。那么对于维数分别为 10×100 、 100×5 、 5×50 的矩阵 A 、 B 、 C , 用 $(A \times B) \times C$ 来计算需要 $10 \times 100 \times 5 + 10 \times 5 \times 50 = 7500$ 次标量乘法; 而 $A \times (B \times C)$ 则需要 $100 \times 5 \times 50 + 10 \times 100 \times 50 = 75000$ 次标量乘法。

四个矩阵分别为 2×6 、 6×3 、 3×10 、 10×3 , 先计算 $M_1 \times M_2$ 和 $M_3 \times M_4$, 计算次数分别为 $2 \times 6 \times 3 = 36$ 、 $3 \times 10 \times 3 = 90$ 。然后结果相乘, 计算次数为 $2 \times 3 \times 3 = 18$ 。

方法二:

四个矩阵分别为 2×6 、 6×3 、 3×10 、 10×3 , 先计算 $M_1 \times M_2$ 和 $M_3 \times M_4$, 计算次数分别为 $2 \times 6 \times 3 = 36$ 、 $3 \times 10 \times 3 = 90$ 。然后结果相乘, 计算次数为 $2 \times 3 \times 3 = 18$ 。

答案: (64) C (65) B

(66)、(67) 试题分析: ARP 和 ICMP 是网络层协议, X.25 是标准的接口协议, 只有 SNMP 是应用层协议。SNMP 协议的报文是封装在 UDP 协议中传送的。

答案: (66) A (67) C

(68) 试题分析: URL 由三部分组成: 协议类型、域名、路径及文件名。

答案: (68) A

(69) 试题分析: 管理距离是各种路由协议的优先权, 当多个路由协议转发的关于某个目标的多条路由, 使用管理距离选择权限最高的路由, 管理距离小的优先级最高。

路由费用是一种路由协议对于到达目标网络的各种可能路径的成本衡量。

答案: (69) C

(70) 试题分析: 因为都是/22, 因此只看前 22 位即可, 又因为选项中前两个数(16 位)一样, 因此直接对比第三个数, 将第三个数转换为二进制, 对比其前 6 位。

179→1011 0011

145→1001 0001

147→1001 0011

177→1011 0001

22 位掩码除去前两组 16 位, 还剩余 6 位, 从左向右匹配 177 一致。

答案: (70) D

(71)~(75) 试题分析:

翻译:

软件实体比其他任何人类构造都要复杂得多, 因为没有两个部分是相同的(至少在语句层之上)。如果它们是, 我们将两个相似的部分组成一个, 一个(71), 打开或关闭。在这方面, 软件系统与计算机、建筑物或汽车有很大的不同, 在那里, 重复的元素比比皆是。

数字计算机本身比人们建造的大多数东西都要复杂, 它们有很多状态。这使得构思、描述和测试很难。软件系统的(72)比计算机要多。

同样地, 一个软件实体的扩大不仅仅是重复同样的元素在更大的规模, 它必然会增加不同元素的数量。在大多数情况下, 这些元素以某种(73)的方式相互作用, 整个过程的复杂程度远远超过线性。

软件的复杂性是一个(74)属性, 而不是一个偶然的属性。因此, 对一个抽象出其复杂性的软件实体的描述往往会抽象出它的本质。数学和物理科学通过构造复杂现象的简化模型, 从模型中推导出特性, 并通过实验验证这些特性, 在三个世纪中取得了巨大的进步。这是有效的, 因为模型中(75)的复杂性并不是现象的本质属性。当复杂性是本质时, 它不会起作用。

开发软件产品的许多经典问题都源自于这种本质的复杂性, 而其非线性的增加也是如此。不仅是技术问题, 还有管理问题, 也来自复杂性。

- | | | | |
|-------------|---------|---------|--------|
| (71) A. 任务 | B. 工作 | C. 子程序 | D. 程序 |
| (72) A. 状态 | B. 部分 | C. 条件 | D. 表示 |
| (73) A. 线性 | B. 非线性的 | C. 平行的 | D. 附加的 |
| (74) A. 表面的 | B. 外部的 | C. 内部的 | D. 本质的 |
| (75) A. 固定的 | B. 包含的 | C. 被忽视的 | D. 使稳定 |

答案: (71) C (72) A (73) B (74) D (75) C

下午案例分析

试题一（15 分）

【问题 1】（3 分）

答案：E1：客户服务助理；E2：客户；E3：经纪人。

【问题 2】（3 分）

答案：D1：客户记录；D2：账户记录；D3：交易记录。

【问题 3】（4 分）

答案：

数据流名称：修改账户余额；起点：存款；终点：D2。

数据流名称：修改账户余额；起点：取款；终点：D2。

数据流名称：交易信息存入交易记录（在线）；起点：证券交易（在线）；终点：D3。

数据流名称：交易信息存入交易记录（电话）；起点：证券交易（电话）；终点：D3。

图 1-1 增加外部实体“证券交易中心”，增加“证券交易平台”到“证券交易中心”，数据流为“交易信息”。

图 1-2 增加外部实体“证券交易中心”，增加“证券交易（在线）”到“证券交易中心”，数据流为“交易信息”；增加“证券交易（电话）”到“证券交易中心”，数据流为“交易信息”。

试题二（15 分）

阅读下列说明，回答问题，将解答填入答题纸的对应栏内。

【问题 1】（4 分）

答案：

- （1）经理与部门之间存在 1:1 的联系。
- （2）部门与员工之间存在 1:n 的联系。
- （3）客户与预订申请之间存在 1:n 的联系。
- （4）业务员、客房、预订申请之间存在 1:m:n 的联系。

【问题 2】（8 分）

答案：

- （1）
 - （a）部门号。
 - （b）客户号、单位名称。
 - （c）申请号、客户号。
 - （d）身份证号、入住时间。

(2) 给出“预订申请”和“安排”关系模式的主键和外键。“预订申请”关系模式中的主键是申请号，外键是客户号；“安排”关系模式中的主键是客房号、身份证号、入住时间，外键是申请号、客房号、业务员。

【问题 3】(3 分)

答案：根据试题中的描述，客房信息中客房号是唯一标识客房关系的一个元组，即可作为唯一的主键。在客房关系模式中，不存在其他部分的依赖关系，但客户号、类型、收费标准存在传递函数依赖，所以冗余、添加异常、修改异常、删除异常均存在。

试题三 (15 分)

【问题 1】(6 分)

答案：S1：空闲；S2：准备服务；S3：饮料出售；S4：可购买。

【问题 2】(4 分)

答案：E1：饮料数量不足；E2：硬币数量足够；E3：推出饮料；E4：返回找零。

【问题 3】(5 分)

答案：C1：自动售货机；C2：硬币器；C3：饮料储存仓；C4：硬币；C5：饮料。

试题四 (15 分)

【问题 1】(8 分)

答案：

- (1) $j < ls$
- (2) $t[i] == s[j]$
- (3) `get_next(next, s, ls)`
 $j = next[j]$
- (4) $i + 1 - ls$

【问题 2】(2 分)

答案：(5) $O(ls + t)$

【问题 3】(5 分)

答案：

- (6) $[-1, -1, 1, -1, -1, 2, 0, 0]$
- (7) 6

试题五 (15 分)

答案：

- (1) `virtual void printInvoice()`
- (2) `ticket->printInvoice()`

- (3) Decorator::printInvoice()
- (4) Decorator::printInvoice()
- (5) &a

试题六 (15 分)

答案:

- (1) ticket.printInvoice()
- (2) ticket.printInvoice()
- (3) ticket.printInvoice()
- (4) new FootDecorator(new HeadDecorator(t))
- (5) new FootDecorator(new HeadDecorator(new Decorator(null)))

第21小时

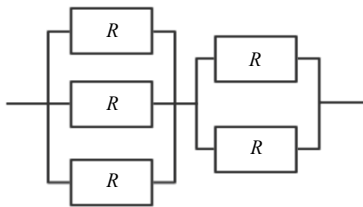
2017年5月考试真题（上午基础知识）



【导读小贴士】

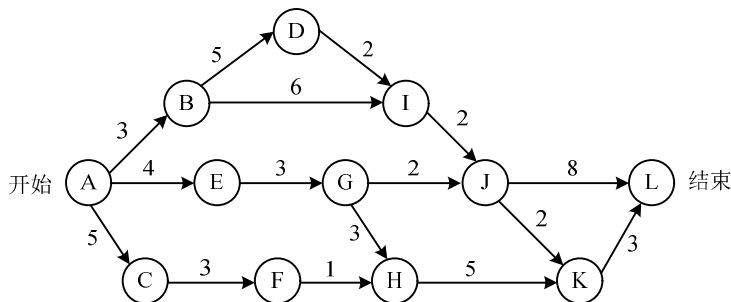
经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

- CPU 执行算术运算或逻辑运算时，常将源操作数和结果暂存在 (1) 中。
(1) A. 程序计数器 (PC) B. 累加器 (AC)
C. 指令寄存器 (IR) D. 地址寄存器 (AR)
- 要判断字长为 16 位的整数 a 的低四位是否全为 0，则 (2)。
(2) A. 将 a 与 0x000F 进行“逻辑与”运算，然后判断运算结果是否等于 0
B. 将 a 与 0x000F 进行“逻辑或”运算，然后判断运算结果是否等于 F
C. 将 a 与 0x000F 进行“逻辑异或”运算，然后判断运算结果是否等于 0
D. 将 a 与 0x000F 进行“逻辑与”运算，然后判断运算结果是否等于 F
- 计算机系统中常用的输入/输出控制方式有无条件传送、中断、程序查询和 DMA 方式等。当采用 (3) 方式时，不需要 CPU 执行程序指令来传送数据。
(3) A. 中断 B. 程序查询 C. 无条件传送 D. DMA
- 某系统由下图所示的冗余部件构成，若每个部件的千小时可靠度都为 R ，则该系统的千小时可靠度为 (4)。



- (4) A. $(1-R^3)(1-R^2)$ B. $(1-(1-R)^3)(1-(1-R)^2)$
 C. $(1-R^3)+(1-R^2)$ D. $(1-(1-R)^3)+(1-(1-R)^2)$
- 已知数据信息为 16 位, 最少应附加 (5) 位校验位, 才能实现海明码纠错。
 (5) A. 3 B. 4 C. 5 D. 6
 - 以下关于 Cache (高速缓冲存储器) 的叙述中, 不正确的是 (6)。
 (6) A. Cache 的设置扩大了主存的容量
 B. Cache 的内容是主存部分内容的拷贝
 C. Cache 的命中率并不随其容量增大而线性提高
 D. Cache 位于主存与 CPU 之间
 - HTTPS 使用 (7) 协议对报文进行封装。
 (7) A. SSH B. SSL C. SHA-1 D. SET
 - 以下加密算法中适合对大量的明文消息进行加密传输的是 (8)。
 (8) A. RSA B. SHA-1 C. MD5 D. RC5
 - 假定用户 A、B 分别在 I_1 和 I_2 两个 CA 处取得了各自的证书, 下面 (9) 是 A、B 互信的必要条件。
 (9) A. A、B 互换私钥 B. A、B 互换公钥
 C. I_1 、 I_2 互换私钥 D. I_1 、 I_2 互换公钥
 - 甲软件公司受乙企业委托安排公司软件设计师开发了信息系统管理软件, 由于在委托开发合同中未对软件著作权归属作出明确的约定, 所以该信息系统管理软件的著作权由 (10) 享有。
 (10) A. 甲 B. 乙 C. 甲与乙共同 D. 软件设计师
 - 根据我国商标法, 下列商品中必须使用注册商标的是 (11)。
 (11) A. 医疗仪器 B. 墙壁涂料 C. 无糖食品 D. 烟草制品
 - 甲、乙两人在同一天就同样的发明创造提交了专利申请, 专利局将分别向各个申请人通报有关情况, 并提出多种可能采用的解决办法。下列说法中, 不可能采用 (12)。
 (12) A. 甲、乙作为共同申请人
 B. 甲或乙一方放弃权利并从另一方得到适当的补偿
 C. 甲、乙都不授予专利权
 D. 甲、乙都授予专利权
 - 数字语音的采样频率定义为 8kHz, 这是因为 (13)。
 (13) A. 语音信号定义的频率最高值为 4kHz

- B. 语音信号定义的频率最高值为 8kHz
C. 数字语音传输线路的带宽只有 8kHz
D. 一般声卡的采样频率最高为每秒 8k 次
- 使用图像扫描仪以 300DPI 的分辨率扫描一幅 3×4 英寸的图片, 可以得到 (14) 像素的数字图像。
(14) A. 300×300 B. 300×400 C. 900×4 D. 900×1200
 - 在采用结构化开发方法进行软件开发时, 设计阶段的接口设计主要依据需求分析阶段的 (15)。接口设计的任务主要是 (16)。
(15) A. 数据流图 B. E-R 图 C. 状态—迁移图 D. 加工规格说明
(16) A. 定义软件的主要结构元素及其之间的关系
B. 确定软件涉及的文件系统的结构及数据库的表结构
C. 描述软件与外部环境之间的交互关系, 软件内模块之间的调用关系
D. 确定软件各个模块内部的算法和数据结构
 - 某软件项目的活动图如下图所示, 其中顶点表示项目里程碑, 连接顶点的边表示包含的活动, 边上的数字表示活动的持续时间 (天), 则完成该项目的最少时间为 (17) 天。活动 BD 和 HK 最早可以从第 (18) 天开始 (活动 AB、AE 和 AC 最早从第 1 天开始)。

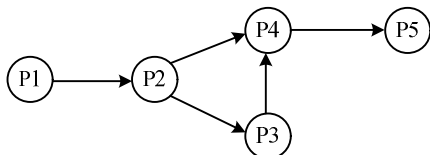


- (17) A. 17 B. 18 C. 19 D. 20
(18) A. 3 和 10 B. 4 和 11 C. 3 和 9 D. 4 和 10
- 在进行软件开发时, 采用无主程序员的开发小组, 成员之间相互平等; 而主程序员负责制的开发小组, 由一个主程序员和若干成员组成, 成员之间没有沟通。在一个由 8 名开发人员构成的小组中, 无主程序员组和主程序员组的沟通路径分别是 (19)。
(19) A. 32 和 8 B. 32 和 7 C. 28 和 8 D. 28 和 7
 - 在高级语言源程序中, 常需要用户定义的标识符为程序中的对象命名, 常见的命名对象有 (20)。
①关键字 (或保留字) ②变量 ③函数 ④数据类型 ⑤注释
(20) A. ①②③ B. ②③④ C. ①③⑤ D. ②④⑤
 - 在仅由字符 a 、 b 构成的所有字符串中, 其中以 b 结尾的字符串集合可用正规式表示为 (21)。

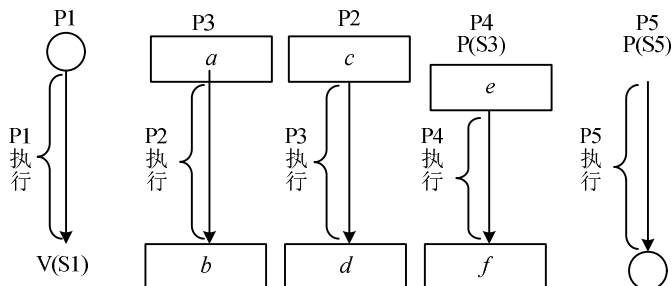
- (21) A. $(b|ab)^*b$ B. $(ab^*)^*b$ C. a^*b^*b D. $(a|b)^*b$
- 在分阶段划分的编译过程中, 判断程序语句的形式是否正确属于 (22) 阶段的工作。
 - (22) A. 词法分析 B. 语法分析 C. 语义分析 D. 代码生成
 - 某文件管理系统在磁盘上建立了位示图 (bitmap), 记录磁盘的使用情况。若计算机系统的字长为 32 位, 磁盘的容量为 300GB, 物理块的大小为 4MB, 那么位示图的大小需要 (23) 个字。
 - (23) A. 1200 B. 2400 C. 6400 D. 9600
 - 某系统中有三个并发进程竞争资源 R , 每个进程都需要 5 个 R , 那么至少有 (24) 个 R 才能保证系统不会发生死锁。
 - (24) A. 12 B. 13 C. 14 D. 15
 - 某计算机系统页面大小为 4K, 进程的页面变换表如下所示。若进程的逻辑地址为 2D16H, 该地址经过变换后, 其物理地址应为 (25)。

页号	物理块号
0	1
1	3
2	4
3	6

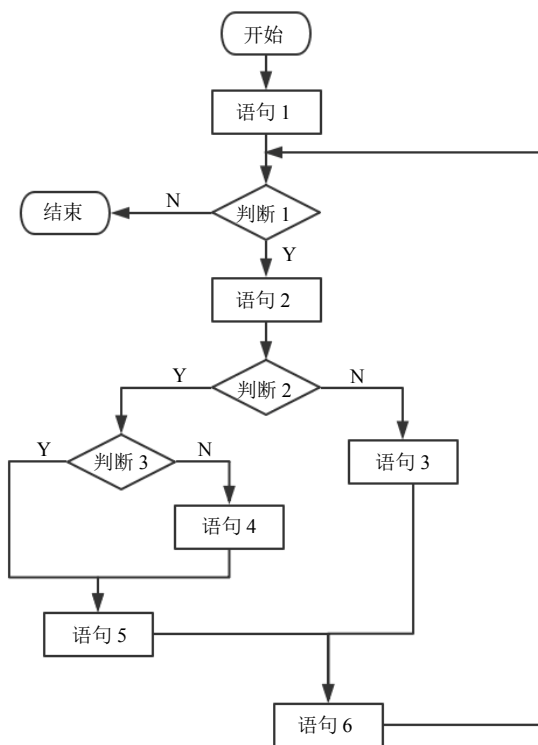
- (25) A. 2048H B. 4096H C. 4D16H D. 6D16H
- 进程 P1、P2、P3、P4、P5 的前趋图如下所示。



若用 PV 操作控制进程 P1、P2、P3、P4、P5 并发执行的过程, 需要设置五个信号量 S1、S2、S3、S4 和 S5, 而且信号量 S1~S5 的初值都等于零。如下的进程执行图中 a 和 b 处应分别填写 (26); c 和 d 处应分别填写 (27); e 和 f 处应分别填写 (28)。



- (26) A. V(S1)和 P(S2)V(S3) B. P(S1)和 V(S2)V(S3)
C. V(S1)和 V(S2)V(S3) D. P(S1)和 P(S2)V(S3)
- (27) A. P(S2)和 P(S4) B. V(S2)和 P(S4)
C. P(S2)和 V(S4) D. V(S2)和 V(S4)
- (28) A. P(S4)和 V(S5) B. V(S5)和 P(S4)
C. V(S4)和 P(S5) D. V(S4)和 V(S5)
- 以下关于螺旋模型的叙述中, 不正确的是 (29)。
(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
 - 以下关于极限编程 (XP) 中结对编程的叙述中, 不正确的是 (30)。
(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
 - 以下关于 C/S (客户机/服务器) 体系结构优点的叙述中, 不正确的是 (31)。
(31) A. 允许合理地划分三层功能, 使之在逻辑上保持相对独立性
B. 允许各层灵活地选用平台和软件
C. 各层可以选择不同的开发语言进行并行开发
D. 系统安装、修改和维护均只在服务器端进行
 - 在设计软件的模块结构时, (32) 不能改进设计质量。
(32) A. 尽量减少高扇出结构 B. 尽量减少高扇入结构
C. 将具有相似功能的模块合并 D. 完善模块的功能
 - 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。现把该程序块取出来, 形成新的模块 D, 则模块 D 的内聚类型为 (33) 内聚。以下关于该内聚类型的叙述中, 不正确的是 (34)。
(33) 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
 - 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 (35) 个测试用例。采用 McCabe 度量法计算其环路复杂度为 (36)。



(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

- 在面向对象方法中，两个及以上的类作为一个类的超类时，称为 (37)，使用它可能造成子类中存在 (38) 的成员。

(37) A. 多重继承 B. 多态 C. 封装 D. 层次继承

(38) A. 动态 B. 私有 C. 公共 D. 二义性

- 采用面向对象方法进行软件开发，在分析阶段，架构师主要关注系统的 (39)。

(39) A. 技术 B. 部署 C. 实现 D. 行为

- 在面向对象方法中，多态指的是 (40)。

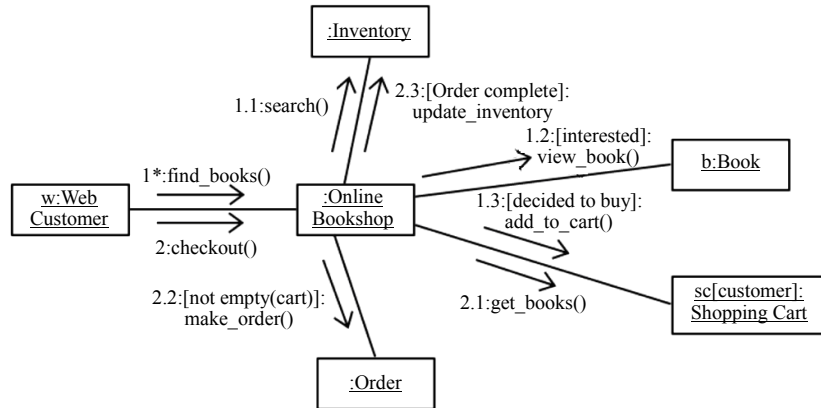
(40) A. 客户类无须知道所调用方法的特定子类的实现

B. 对象动态地修改类

C. 一个对象对应多张数据库表

D. 子类只能覆盖父类中非抽象的方法

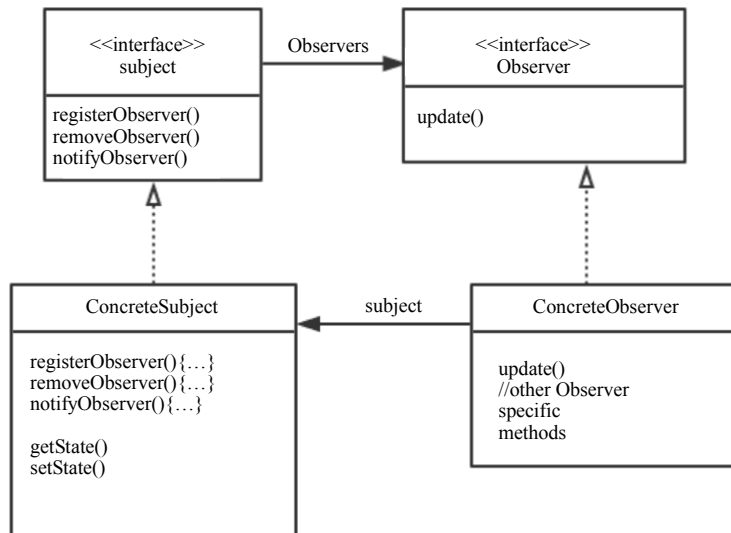
- 以下 UML 图是 (41)，图中 `Order` 和 `Book` 表示 (42)，`1*:find_books()` 和 `1.l:search()` 表示 (43)。



- (41) A. 序列图 B. 状态图 C. 通信图 D. 活动图
 (42) A. 类 B. 对象 C. 流名称 D. 消息
 (43) A. 类 B. 对象 C. 流名称 D. 消息

- 下图为观察者（Observer）模式的抽象示意图，其中（44）知道其观察者，可以有任意多个观察者观察同一个目标，提供住处和删除观察者对象的接口。此模式体现的最主要的特征是（45）。

- (44) A. Subject B. Observer C. ConcreteSubject D. ConcreteObserver
 (45) A. 类应该对扩展开放，对修改关闭 B. 使所要交互的对象尽量松耦合
 C. 组合优先于继承使用 D. 仅与直接关联类交互



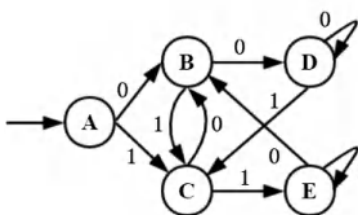
- 装饰器（Decorator）模式用于（46）；外观（Façade）模式用于（47）。

- ①将一个对象加以包装以给客户提供其希望的另外一个接口
- ②将一个对象加以包装以提供一些额外的行为
- ③将一个对象加以包装以控制对这个对象的访问
- ④将一系列对象加以包装以简化其接口

(46) A. ① B. ② C. ③ D. ④

(47) A. ① B. ② C. ③ D. ④

- 某确定的有限自动机 (DFA) 的状态转换图如下图所示 (A 是初态, D、E 是终态), 则该 DFA 能识别 (48)。



(48) A. 00110 B. 10101 C. 11100 D. 11001

- 函数 main()、f() 的定义如下所示, 调用函数 f() 时, 第一个参数采用传值 (call by value) 方式, 第二个参数采用传引用 (call by reference) 方式, main() 函数中 “print(x)” 执行后输出的值为 (49)。

main()

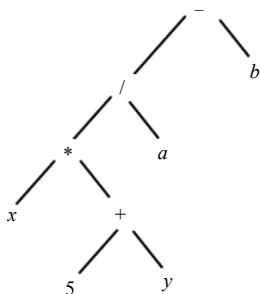
```
int x=5;
f(x+1,x);
print(x);
```

f(int x, int & a)

```
x=x*x-1;
a=x+a;
return;
```

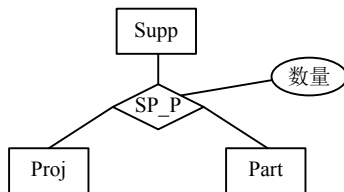
(49) A. 11 B. 40 C. 45 D. 70

- 下图为一个表达式的语法树, 该表达式的后缀形式为 (50)。



(50) A. $x5y+*a/b-$ B. $x5yab*+/-$

- C. $-/*x+5yab$ D. $x5*y+a/b-$
- 若事务 T1 对数据 D1 加了共享锁, 事务 T2、T3 分别对数据 D2、D3 加了排他锁, 则事务 T1 对数据 (51); 事务 T2 对数据 (52)。
 - (51) A. D2、D3 加排他锁都成功
 - B. D2、D3 加共享锁都成功
 - C. D2 加共享锁成功, D3 加排他锁失败
 - D. D2、D3 加排他锁和共享锁都失败
 - (52) A. D1、D3 加共享锁都失败
 - B. D1、D3 加共享锁都成功
 - C. D1 加共享锁成功, D3 加排他锁失败
 - D. D1 加排他锁成功, D3 加共享锁失败
 - 假设关系 $R<U, F>$, $U = \{A_1, A_2, A_3\}$, $F = \{A_1A_3 \rightarrow A_2, A_1A_2 \rightarrow A_3\}$, 则关系 R 的各个候选关键字中必定含有属性 (53)。
 - (53) A. A_1 B. A_2 C. A_3 D. A_2, A_3
 - 在某企业的工程项目管理系统的数据库中, 供应商关系 Supp、项目关系 Proj 和零件关系 Part 的 E-R 模型和关系模式如下所示。



Supp (供应商号, 供应商名, 地址, 电话)

Proj (项目号, 项目名, 负责人, 电话)

Part (零件号, 零件名)

其中, 每个供应商可以为多个项目供应多种零件, 每个项目可由多个供应商供应多种零件。

SP_P 需要生成一个独立的关系模式, 其联系类型为 (54)。

给定关系模式 SP_P (供应商号, 项目号, 零件号, 数量) 查询至少供应了三个项目 (包含三项) 的供应商, 输出其供应商号和供应零件数量的总和, 并按供应商号降序排列。

SELECT 供应商号, SUM (数量) FROM (55) GROUP BY 供应商号 (56) ORDER BY 供应商号 DESC;

- (54) A. $*:*:*$ B. $1:*:*$ C. $1:1:*$ D. $1:1:1$
- (55) A. Supp B. Proj C. Part D. SP_P
- (56) A. HAVING COUNT(项目号)>2
- B. WHERE COUNT(项目号)>2
- C. HAVING COUNT(DISTINCT(项目号))>2
- D. WHERE COUNT(DISTINCT(项目号))>3

- 以下关于字符串的叙述中, 正确的是 (57)。
 (57) A. 包含任意个空格字符的字符串称为空串
 B. 字符串不是线性数据结构
 C. 字符串的长度是指串中所含字符的个数
 D. 字符串的长度是指串中所含非空格字符的个数
- 已知栈 S 初始为空, 用 I 表示入栈、 O 表示出栈, 若入栈序列为 $a_1a_2a_3a_4a_5$, 则通过栈 S 得到出栈序列 $a_2a_4a_5a_3a_1$ 的合法操作序列为 (58)。
 (58) A. $IIOIIIOIOOO$ B. $IOIOIOIOIO$ C. $IOOIIIOIOIO$ D. $IIOOIOIOOO$
- 某二叉树的先序遍历序列为 $ABCDEF$, 中序遍历序列为 $BADCFE$, 则该二叉树的高度 (即层数) 为 (59)。
 (59) A. 3 B. 4 C. 5 D. 6
- 对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$, 当且仅当满足关系 $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1} \{i=1, 2, \dots, [n/2]\}$ 时称其为小根堆 (小顶堆)。以下序列中, (60) 不是小根堆。
 (60) A. 16, 25, 40, 55, 30, 50, 45 B. 16, 40, 25, 50, 45, 30, 55
 C. 16, 25, 39, 41, 45, 43, 50 D. 16, 40, 25, 53, 39, 55, 45
- 在 12 个互异元素构成的有序数组 $a[1, \dots, 12]$ 中进行二分查找 (即折半查找, 向下取整), 若待查找的元素正好等于 $a[9]$, 则在此过程中, 依次与数组中的 (61) 比较后, 查找成功结束。
 (61) A. $a[6]$ 、 $a[7]$ 、 $a[8]$ 、 $a[9]$ B. $a[6]$ 、 $a[9]$
 C. $a[6]$ 、 $a[7]$ 、 $a[9]$ D. $a[6]$ 、 $a[8]$ 、 $a[9]$
- 某汽车加工工厂有两条装配线 $L1$ 和 $L2$, 每条装配线的工位数均为 n ($S_{ij}, i=\{1, 2\}, j=\{1, 2, \dots, n\}$), 两条装配线对应的工位完成同样的加工工作, 但是所需要的时间可能不同 ($a_{ij}, i=\{1, 2\}, j=\{1, 2, \dots, n\}$)。汽车底盘开始到进入两条装配线的时间 (e_1, e_2) 和装配后到结束的时间 (X_1, X_2) 也可能不相同。从一个工位加工后传到下一个工位需要迁移时间 ($t_{ij}, i=\{1, 2\}, j=\{2, \dots, n\}$)。现在要以最快的时间完成一辆汽车的装配, 求最优的装配路线。

分析该问题, 发现问题具有最优子结构。以 $L1$ 为例, 除了第一个工位之外, 经过第 j 个工位的最短时间包含了经过 $L1$ 的第 $j-1$ 个工位的最短时间或经过 $L2$ 的第 $j-1$ 个工位的最短时间, 如式 (1) 所示。装配后到结束的最短时间包含离开 $L1$ 的最短时间或离开 $L2$ 的最短时间, 如式 (2) 所示。

$$f_{1,j} = \begin{cases} e_1 + a_{1,j} & j=1 \\ \min(f_{1,j-1} + a_{1,j} + t_{1,j-1}, f_{2,j-1} + a_{1,j} + t_{2,j-1}) & \text{其他} \end{cases} \quad (1)$$

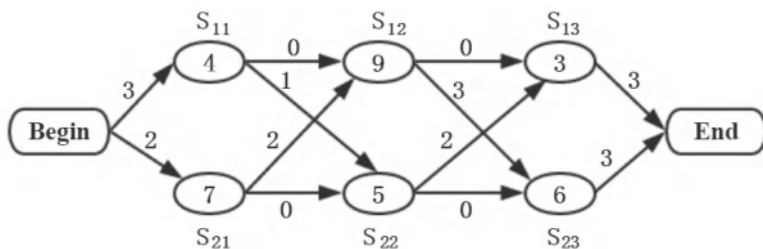
$$f_{\min} = \min(f_{1,n} + x_1, f_{2,n} + x_2) \quad (2)$$

由于在求解经过 $L1$ 和 $L2$ 的第 j 个工位的最短时间均包含了经过 $L1$ 的第 $j-1$ 个工位的最短时间或经过 $L2$ 的第 $j-1$ 个工位的最短时间, 该问题具有重复子问题的性质, 故采用迭

代方法求解。

该问题采用的算法设计策略是 (62), 算法的时间复杂度为 (63)。

以下是一个装配调度实例, 其最短的装配时间为 (64), 装配路线为 (65)。



- (62) A. 分治 B. 动态规划 C. 贪心 D. 回溯
 (63) A. $O(\lg n)$ B. $O(n)$ C. $O(n^2)$ D. $O(n \lg n)$
 (64) A. 21 B. 23 C. 20 D. 26
 (65) A. $S_{11} \rightarrow S_{12} \rightarrow S_{13}$ B. $S_{11} \rightarrow S_{22} \rightarrow S_{13}$
 C. $S_{21} \rightarrow S_{12} \rightarrow S_{23}$ D. $S_{21} \rightarrow S_{22} \rightarrow S_{23}$

- 在浏览器地址栏中输入一个正确的网址后, 本地主机将首先在 (66) 查询该网址对应的 IP 地址。

- (66) A. 本地 DNS 缓存 B. 本机 hosts 文件
 C. 本地 DNS 服务器 D. 根域名服务器

- 下面关于 Linux 目录的描述中, 正确的是 (67)。

- (67) A. Linux 只有一个根目录, 用 “/root” 表示
 B. Linux 有多个根目录, 用 “/” 加相应目录名称表示
 C. Linux 只有一个根目录, 用 “/” 表示
 D. Linux 有多个根目录, 用相应目录名称表示

- 以下关于 TCP/IP 协议栈中协议和层次的对应关系, 正确的是 (68)。

- (68) A.

TFTP	Telnet
UDP	TCP
ARP	

 B.

RIP	Telnet
UDP	TCP
ARP	

 C.

HTTP	SNMP
TCP	UDP
IP	

 D.

SMTP	FTP
UDP	TCP
IP	

- 在异步通信中, 每个字符包含 1 位起始位、7 位数据位和 2 位终止位, 若每秒钟传送 500 个字符, 则有效数据速率为 (69)。

- (69) A. 500b/s B. 700b/s C. 3500b/s D. 5000b/s

- 以下路由策略中, 依据网络信息经常更新路由的是 (70)。

(70) A. 静态路由 B. 洪泛式 C. 随机路由 D. 自适应路由

- The beauty of software is in its function, in its internal structure, and in the way in which it is created by a team. To a user, a program with just the right features presented through an intuitive and (71) interface is beautiful. To a software designer, an internal structure that is partitioned in a simple and intuitive manner, and that minimizes internal coupling is beautiful. To developers and managers, a motivated team of developers making significant progress every week, and producing defect-free code, is beautiful. There is beauty on all these levels.

Our world needs software—lots of software. Fifty years ago software was something that ran in a few big and expensive machines. Thirty years ago it was something that ran in most companies and industrial settings. Now there is software running in our cell phones, watches, appliances, automobiles, toys, and tools. And need for new and better software never (72). As our civilization grows and expands, as developing nations build their infrastructures, as developed nations strive to achieve ever greater efficiencies, the need for more and more Software (73) to increase. It would be a great shame if, in all that software, there was no beauty.

We know that software can be ugly. We know that it can be hard to use, unreliable, and carelessly structured. We know that there are software systems whose tangled and careless internal structures make them expensive and difficult to change. We know that there are software systems that present their features through an awkward and cumbersome interface. We know that there are software systems that crash and misbehave. These are (74) systems. Unfortunately, as a profession, software developers tend to create more ugly systems than beautiful ones.

There is a secret that the best software developers know. Beauty is cheaper than ugliness. Beauty is faster than ugliness. A beautiful software system can be built and maintained in less time, and for less money, than an ugly one. Novice software developers don't understand this. They think that they have to do everything fast and quick. They think that beauty is (75). No! By doing things fast and quick, they make messes that make the software stiff, and hard to understand. Beautiful systems are flexible and easy to understand. Building them and maintaining them is a joy. It is ugliness that is impractical. Ugliness will slow you down and make your software expensive and brittle. Beautiful systems cost the least build and maintain, and are delivered soonest.

- | | | | |
|---------------------|--------------|-----------------|---------------|
| (71) A. simple | B. hard | C. complex | D. duplicated |
| (72) A. happens | B. exists | C. stops | D. starts |
| (73) A. starts | B. continues | C. appears | D. stops |
| (74) A. practical | B. useful | C. beautiful | D. ugly |
| (75) A. impractical | B. perfect | C. time-wasting | D. practical |

第22小时

2017 年 5 月考试真题（下午案例分析）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

试题一（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某医疗器械公司作为复杂医疗产品的集成商，必须保持高质量部件的及时供应。为了实现这一目标，该公司欲开发一个采购系统。系统的主要功能如下：

- （1）检查库存水平。采购部门每天检查部件库存量，当特定部件的库存量降至其订货点时，返回低存量部件及库存量。
- （2）下达采购订单。采购部门针对低存量部件及库存量提交采购请求，向其供应商（通过供应商文件访问供应商数据）下达采购订单，并存储于采购订单文件中。
- （3）交运部件。当供应商提交提单并交运部件时，运输和接收（S/R）部门通过执行以下三个过程接收货物：
 - 1）验证装运部件。通过访问采购订单并将其与提单进行比较来验证装运的部件，并将提单信息发给 S/R 职员。如果收货部件项目出现在采购订单和提单上，则已验证的提单和收货部件项目将被送去检验。否则，将 S/R 职员提交的装运错误信息生成装运错误通知发送给供应商。
 - 2）检验部件质量。通过访问质量标准来检查装运部件的质量，并将已验证的提单发给检验员。

如果部件满足所有质量标准，则将其添加到接受的部件列表用于更新部件库存。如果部件未通过检查，则将检验员创建的缺陷装运信息生成缺陷装运通知发送给供应商。

3) 更新部件库存。库管员根据收到的接受的部件列表添加本次采购数量，与原有库存量累加来更新库存部件中的库存量，标记订单采购完成。

现在采用结构化方法对该采购系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和如图 1-2 所示的 0 层数据流图。

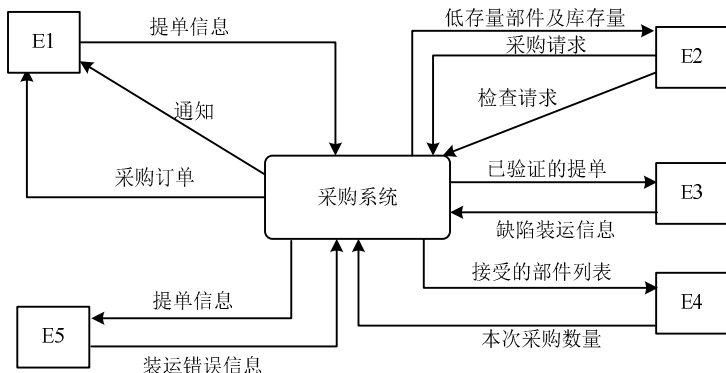


图 1-1 上下文数据流图

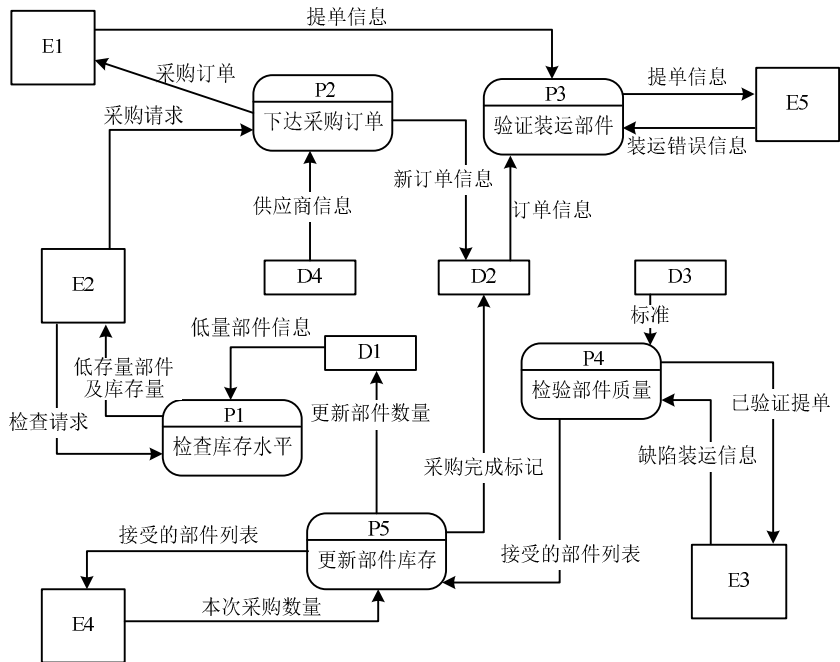


图 1-2 0 层数据流图

【问题 1】（5 分）

使用说明中的词语，给出图 1-1 中的实体 E1~E5。

【问题 2】（4 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1~D4 的名称。

【问题 3】（4 分）

根据说明和图中术语，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】（2 分）

用 200 字以内的文字，说明建模图 1-1 和图 1-2 时如何保持数据流图平衡。

试题二（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某房屋租赁公司拟开发一个管理系统用于管理其持有的房屋、租客及员工信息。请根据下述需求描述完成系统的数据库设计。

【需求描述】

（1）公司拥有多幢公寓楼，每幢公寓楼有唯一的楼编号和地址。每幢公寓楼中有多套公寓，每套公寓在楼内有唯一的编号（不同公寓楼内的公寓号可以相同）。系统需要记录每套公寓的卧室数和卫生间数。

（2）员工和租客在系统中有唯一的编号（员工编号和租客编号）。

（3）对于每个租客，系统需要记录姓名、多个联系电话、一个银行账号（方便自动扣房租）、一个紧急联系人的姓名及联系电话。

（4）系统需要记录每个员工的姓名、一个联系电话和月工资。员工类别可以是经理或维修工，也可兼任。每个经理可以管理多幢公寓楼，每幢公寓楼必须由一个经理管理。系统需要记录每个维修工的业务技能，如水暖维修、电工、木工等。

（5）租客租赁公寓必须和公司签订租赁合同。一份租赁合同通常由一个或多个租客（合租）与该公寓楼的经理签订，一个租客也可以租赁多套公寓。合同内容应包含签订日期、开始时间、租期、押金和月租金。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

联系电话（电话号码，租客编号）

租客（租客编号，姓名，银行账号，联系人姓名，联系人电话）

员工（员工编号，姓名，联系电话，类别，月工资，(a)）

公寓楼（(b)，地址，经理编号）

公寓（楼编号，公寓号，卧室数，卫生间数）

合同（合同编号，租客编号，楼编号，公寓号，经理编号，签订日期，起始日期，租期，（c），押金）

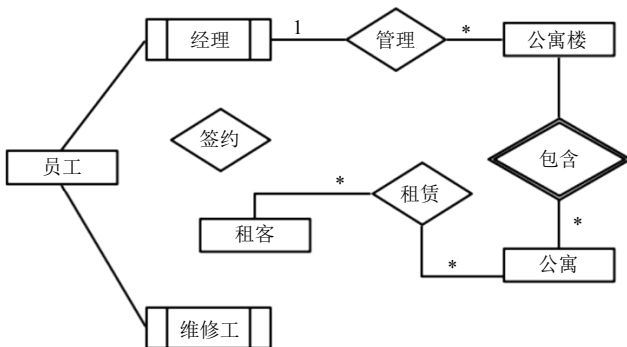


图 2-1 实体联系图

【问题 1】（4.5 分）

补充图 2-1 中的“签约”联系所关联的实体及联系类型。

【问题 2】（4.5 分）

补充逻辑结构设计中的（a）、（b）、（c）三处空缺。

【问题 3】（6 分）

在租期内，公寓内的设施如出现问题，租客可以在系统中进行故障登记，填写故障描述，每项故障由系统自动生成唯一的故障编号，由公司派维修工进行故障维修，系统需要记录每次维修的维修日期和维修内容。请根据此需求，对图 2-1 进行补充，并将所补充的 ER 图内容转换为一个关系模式，请写出该关系模式。

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某玩具公司正在开发一套电动玩具在线销售系统，用于向注册会员提供端对端的玩具定制和销售服务。在系统设计阶段，“创建新订单”（New Order）的设计用例详细描述如表 3-1 所示，候选设计类分类如表 3-2 所示，并根据该用例设计出部分类图如图 3-1 所示。

表 3-1 创建新订单（New Order）设计用例

用例名称	创建新订单（New Order）
用例编号	ETM-R002
参与者	会员
前提条件	会员已经注册并成功登录系统

续表

典型事件流	(1) 会员（C1）单击“新的订单”按钮。 (2) 系统列出所有正在销售的电动玩具清单及价格（C2）。 (3) 会员单击复选框选择所需的电动玩具并输入对应的数量，单击“结算”按钮。 (4) 系统自动计算总价（C3），显示销售清单和会员预先设置个人资料的收货地址和支付方式（C4）。 (5) 会员单击“确认支付”按钮。 (6) 系统自动调用支付系统（C5）接口支付该账单。 (7) 若支付系统返回成功标识，系统生成完整订单信息持久存储到数据库订单表（C6）中。 (8) 系统将以表格形式显示完整订单信息（C7），同时自动发送完整订单信息（C8）至会员预先配置的邮箱地址（C9）。	
候选事件流	3a	(1) 会员单击“定制”按钮。 (2) 系统以列表的形式显示所有可以定制的电动玩具清单和定制属性（如尺寸、颜色等）（C10）。 (3) 会员单击单选按钮选择所需要定制的电动玩具并填写所需要定制的属性要求，单击“结算”按钮。 (4) 回到步骤 4。
	7a	(1) 若支付系统返回失败标识，系统显示会员当前默认的支付方式（C11）让会员确认。 (2) 若会员单击“修改付款”按钮，调用“修改付款”用例，可以新增并存储为默认的支付方式（C12），回到步骤 4。 (3) 若会员单击“取消订单”按钮，则该用例终止执行。

表 3-2 候选设计类分类

接口类（Interface，负责系统与用户之间的交互）	(a)
控制类（Control，负责业务逻辑的处理）	(b)
实体类（Entity，负责持久化数据的存储）	(c)

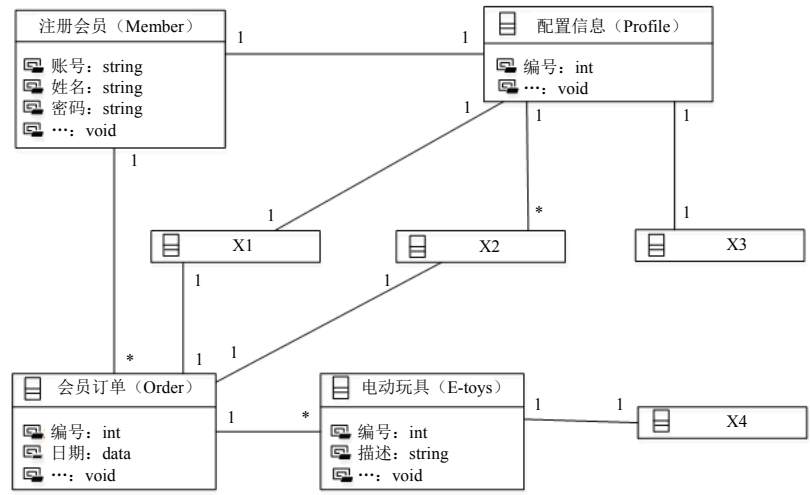


图 3-1 该用例设计的部分类图

在订单处理的过程中，会员可以单击“取消订单”按钮取消该订单。如果支付失败，该订单将被标记为挂起状态，可以后续重新支付；如果挂起超时 30 分钟未支付，系统将自动取消该订单。订单支付成功后，系统判断订单类型：①对于常规订单，标记为备货状态，订单信息发送到货运部，完成打包后交付快递发货；②对于定制订单，会自动进入定制状态，定制完成后交付快递发货。会员在系统中单击“收货”按钮变为收货状态，结束整个订单的处理流程。根据订单处理过程，设计的状态图如图 3-2 所示。

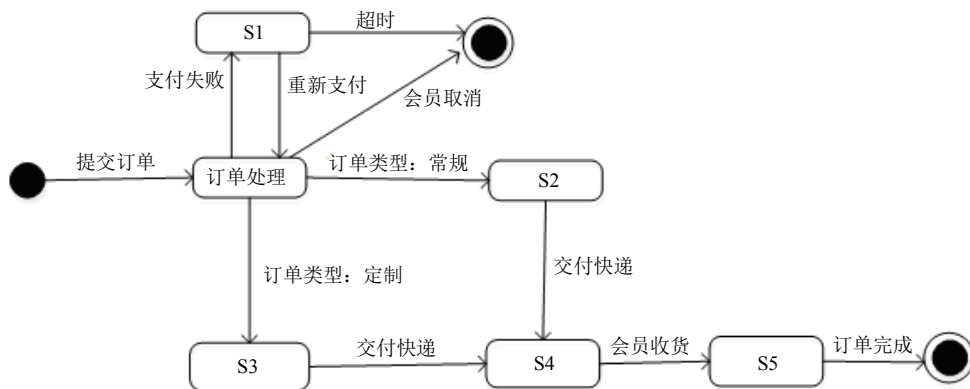


图 3-2 订单状态图

【问题 1】(6 分)

根据表 3-1 中所标记的候选设计类，请按照其类别将编号 C1~C12 分别填入表 3-2 中的 (a)、(b) 和 (c) 处。

【问题 2】(4 分)

根据创建新订单的用例描述，给出图 3-1 中的 X1~X4 处对应类的名称。

【问题 3】(5 分)

根据订单处理过程的描述，在图 3-2 中的 S1~S5 处分别填入对应的状态名称。

试题四 (15 分)

阅读下列说明和 C 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

假币问题：有 n 枚硬币，其中有一枚是假币，已知假币的重量较轻。现只有一个天平，要求用尽量少的比较次数找出这枚假币。

【分析问题】

将 n 枚硬币分成相等的两部分：

(1) 当 n 为偶数时，将前后两部分（即 $1, \dots, n/2$ 和 $n/2+1, \dots, n$ ）放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币。

(2) 当 n 为奇数时, 将前后两部分 (即 $1, \dots, (n-1)/2$ 和 $(n+1)/2+1, \dots, 0$) 放在天平的两端, 较轻的一端里有假币, 继续在较轻的这部分硬币中用同样的方法找出假币; 若两端重量相等, 则中间的硬币即第 $(n+1)/2$ 枚硬币是假币。

【C 代码】

下面是算法的 C 语言实现, 其中:

coins[]: 硬币数组

first, last: 当前考虑的硬币数组中的第一个和最后一个下标

```
#include <stdio.h>

int getCounterfeitCoin(int coins[], int first, int last)
{
    int firstSum=0, lastSum=0;
    int i;
    If(first==last-1){ /*只剩两枚硬币*/
        if(coins[first] < coins[last])
            return first;
        return last;
    }

    if((last-first+1)%2 ==0){ /*偶数枚硬币*/
        for(i=first; i < (1); i++){
            firstSum+=coins[i];
        }
        for(i=first + (last-first)/2 + 1; i < last + 1; i++){
            lastSum+=coins[i];
        }
        if(2){
            Return getCounterfeitCoin(coins, first, first +(last-first)/2;)
        }else{
            Return getCounterfeitCoin(coins, first+(last-first)/2+1, last;)
        }
    }
    else{ /*奇数枚硬币*/
        For(i=first; i<first+(last-first)/2; i++){
            firstSum+=coins[i];
        }
        For(i=first+(last-first)/2+1; i<last+1; i++){
            lastSum+=coins[i];
        }
    }
}
```



```

If(firstSum<lastSum){
return getCounterfeitCoin(coins, first, first+(last-first)/2-1);
}else if(firstSum>lastSum){
return getCounterfeitCoin(coins, first+(last-first)/2-1, last);
}else{
Return(3)
}
}
}
}

```

【问题 1】(6 分)

根据题干说明, 填充 C 代码中的 (1) ~ (3)。

【问题 2】(4 分)

根据题干说明和 C 代码, 算法采用了 (4) 设计策略。

函数 getCounterfeitCoin 的时间复杂度为 (5) (用 O 表示)。

【问题 3】(4 分)

若输入的硬币数为 30, 则最少的比较次数为 (6), 最多的比较次数为 (7)。

试题五 (15 分)

阅读下列说明和 C++ 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某快餐厅主要制作并出售儿童套餐, 一般包括主餐 (各类比萨)、饮料和玩具, 其餐品种类可能不同, 但其制作过程相同。前台服务员 (Waiter) 调度厨师制作套餐。现采用生成器 (Builder) 模式实现制作过程, 得到如图 5-1 所示的类图。

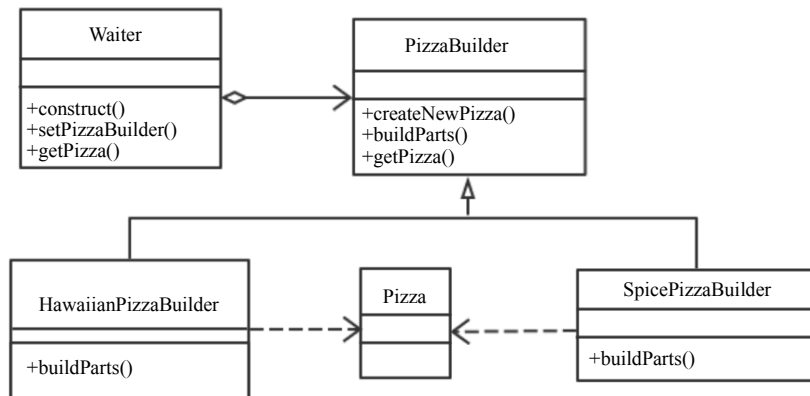


图 5-1 类图

【C++代码】

```

#include<iostream>
#include<string>
using namespace std;

class Pizza {
private: string parts;
public:
    void setParts(string parts) { this->parts = parts; }
    string getParts() { return parts; }
};

class PizzaBuilder {
protected: Pizza*pizza;
public:
    Pizza*getPizza() { return pizza; }
    void createNewPizza() { pizza = new Pizza(); }
    (1);
};

class HawaiianPizzaBuilder: public PizzaBuilder {
public:
    void buildParts() { pizza->setParts("cross + mild + ham&pineapple"); }
};

class SpicyPizzaBuider: public PizzaBuilder {
public:
    void buildParts() { pizza->setParts("pan baked + hot + ham&pineapple"); }
};

class Waiter{
private:
    PizzaBuilder*pizzaBuilder;
public:
    void setPizzaBuilder(PizzaBuilder*pizzaBuilder) { /*设置构建器*/
        (2);
    }
    Pizza*getPizza() { return pizzaBuilder->getPizza(); }
    void construct() { /*构建*/
        pizzaBuilder->createNewPizza();
        (3);
    }
};

int main()
{
    Waiter*waiter = new Waiter();
    PizzaBuilder*hawaiian_pizzabuilder = new HawaiianPizzaBuilder();
    (4);
    (5);
    Cout<<"pizza: "<<waiter->getPizza()->getParts()<<endl;
}

```

程序的输出结果为:

Pizza:cross + mild + ham&pineapple

【问题】(15 分)

阅读上述说明和 C++ 代码，将应填入的字句写在答题纸的对应栏内。

试题六 (15 分)

阅读下列说明和 Java 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某快餐厅主要制作并出售儿童套餐，一般包括主餐（各类比萨）、饮料和玩具，其餐品种类可能不同，但其制作过程相同。前台服务员（Waiter）调度厨师制作套餐。现采用生成器（Builder）模式实现制作过程，得到如图 6-1 所示的类图。

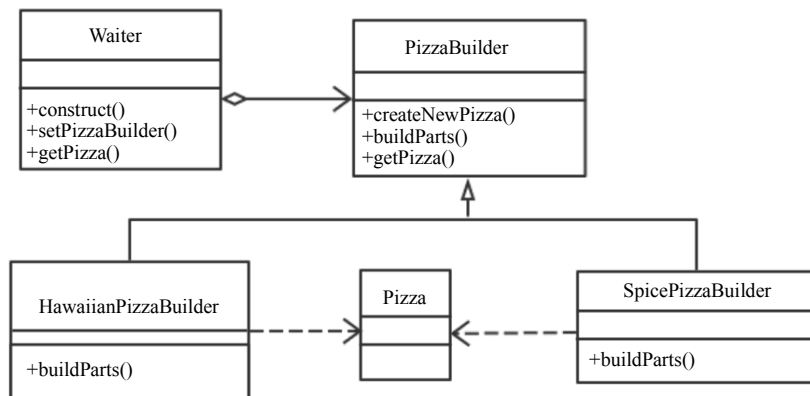


图 6-1 类图

【Java 代码】

```

class Pizza {
    private String parts;
    public void setParts(String parts) { this.parts=parts; }
    public String toString() { return this.parts; }
}

abstract class PizzaBuilder {
    protected Pizza pizza;
    public Pizza getPizza() { return pizza; }
    public void createNewPizza() { pizza=new Pizza(); }
    public ____ (1) ____;
}

class HawaiianPizzaBuilder extends PizzaBuilder {
    public void buildParts() { pizza.setParts("cross + mild + ham&pineapple"); }
}

class SpicyPizzaBuilder extends PizzaBuilder {

```

```
public void buildParts() { pizza.setParts("pan baked + hot + pepperoni&salami"); }  
}  
  
class Waiter {  
    private PizzaBuilder pizzaBuilder;  
  
    public void setPizzaBuilder(PizzaBuilder pizzaBuilder) { /*设置构建器*/  
        (2);  
    }  
    public Pizza getPizza(){ return pizzaBuilder.getPizza(); }  
  
    public void construct() { /*构建*/  
        pizzaBuilder.createNewPizza();  
        (3);  
    }  
}  
  
Class FastFoodOrdering {  
    public static void main(String[] args) {  
        Waiter waiter=new Waiter();  
        PizzaBuilder hawaiian_pizzabuilder=new HawaiianPizzaBuilder();  
  
        (4);  
        (5);  
        System.out.println("pizza: " + waiter.getPizza());  
    }  
}
```

程序的输出结果为：

Pizza:cross + mild + ham&pineapple

【问题】（15 分）

阅读上述说明和 Java 代码，将应填入的字句写在答题纸的对应栏内。

第23小时

2017 年 5 月真题解析

上午基础知识

(1) 试题分析:

A 选项程序计数器 (PC) 是存放执行指令的地方, 计算之前就要用到。

B 选项累加寄存器用来暂时存放算术逻辑运算部件 ALU 运算的结果信息。

C 选项指令寄存器 (IR) 保存当前正在执行的一条指令。

D 选项地址寄存器 (AR) 用来保存当前 CPU 所要访问的内存单元的地址。

答案: (1) B

(2) 试题分析: 在逻辑运算中, 设 A 和 B 为两个逻辑变量, 当且仅当 A 和 B 的取值都为“真”时, A 与 B 的值为“真”; 否则, A 与 B 的值为“假”。当且仅当 A 和 B 的取值都为“假”时, A 或 B 的值为“假”; 否则, A 或 B 的值为“真”。当且仅当 A 和 B 的值不同时, A 异或 B 为“真”, 否则 A 异或 B 为“假”。对于 16 位二进制整数 a, 其与 0000000000001111 (即十六进制数 000F) 进行逻辑与运算后, 结果的高 12 位都为 0, 低 4 位则保留 a 的低 4 位, 因此, 当 a 的低 4 位全为 0 时, 上述逻辑与运算的结果等于 0。

答案: (2) A

(3) 试题分析: 由于 DMA 方式是在 DMA 控制器硬件的控制下实现数据的传送, 不需要 CPU 执行程序, 故这种方式传送的速度最快。另外, 三种都是通过 CPU 执行某一段程序, 实现计算机内存与外设间的数据交换。

答案: (3) D

(4) 试题分析: 串联系统可靠性公式为 $R=R_1 \times R_2 \times \cdots \times R_n$; 并联系统可靠性公式为 $R=1-(1-R_1) \times (1-R_2) \times \cdots \times (1-R_n)$ 。

答案: (4) B

(5) 试题分析: 在海明码中, 用 k 代表其中有效信息位数, r 表示添加的校验码位, 它们之间的关系应满足 $2^r \geq k+r+1$ 。

本题中 $k=16$, 则要求 $2^r \geq 16+r+1$, 根据计算可以得知 r 的最小值为 5。

答案: (5) C

(6) 试题分析: Cache 存储器用来存放主存的部分拷贝(副本)。它是按照程序的局部性原理选取出来的最常使用或将来仍将使用的内容。

答案: (6) A

(7) 试题分析: HTTPS (Hyper Text Transfer Protocol over Secure Socket Layer) 是以安全为目标的 HTTP 通道, 简单讲即是 HTTP 的安全版。HTTPS 在 HTTP 的基础上加入了 SSL 协议, SSL 依靠证书来验证服务器的身份, 并为浏览器和服务器之间的通信加密。

答案: (7) B

(8) 试题分析: A 选项的 RSA 是非对称加密算法; B 选项的 SHA-1 与 C 选项的 MD5 属于信息摘要算法; D 选项的 RC-5 属于非对称加密算法。这些算法中 SHA-1 与 MD5 是不能用来加密数据的, 而 RSA 由于效率问题, 一般不直接用于大量的明文加密。

答案: (8) D

(9) 试题分析: 两个证书发放机构 I1 和 I2 互换公钥是 A、B 互信的必要条件。A 选项和 C 选项的私钥是不能互换的。B 选项中要 AB 互信, 其信任基础是建立在 CA 之上的, 如果仅交换 AB 的公钥并不能解决信任的问题。

答案: (9) D

(10) 试题分析: 根据《著作权法》第 17 条的规定, 著作权归属由委托人和受托人通过合同约定。合同中未作明确约定的, 著作权属于受托人。

答案: (10) A

(11) 试题分析: 根据我国法律法规的规定必须使用注册商标的是烟草类商品。

答案: (11) D

(12) 试题分析: 根据“同一的发明创造只能被授予一项专利”的规定, 两个不同的人在同一天就同样的发明创造申请专利时, 专利局将分别向各位申请人通报有关情况, 请他们自己去协商解决这一问题。解决的方法一般有两种, 一种是两位申请人作为专利申请的共同申请人; 另一种是其中一方放弃权利并从另一方得到适当的补偿。

答案: (12) D

(13) 试题分析: 根据尼奎斯特取样定理, 如果取样速率大于模拟信号最高频率的两倍, 则可以用得到的样本恢复原来的模拟信号。

答案: (13) A

(14) 试题分析: DPI 为像素/英寸, 可以得到 $(300 \times 3) \times (300 \times 4) = 900 \times 1200$ 。

答案: (14) D

(15)、(16) 试题分析: 接口设计的主要依据是数据流图, 接口设计的任务主要是描述软件与外部环境之间的交互关系, 软件内模块之间的调用关系。定义软件的主要结构元素及其之间的关系是架构阶段的任务; 确定软件涉及的文件系统的结构及数据库的表结构是数据存储设计阶段的任务; 确定软件各个模块内部的算法和数据结构是详细设计阶段的任务。

答案: (15) A (16) C

(17)、(18) 试题分析: 项目的工期是从开始到结束持续时间最长的工作。题目中持续时间最长的是 ABDIJL, 需要时间 20 天。

BD 活动在 AB 活动结束后便可以开始, 所以最早的开始时间为 4。HK 活动需要在 AEGH 与 ACFH 两条路径上的活动均完成后才能开始, 所以最早的开始时间为 11。

答案: (17) D (18) B

(19) 试题分析: 沟通渠道 $= N(N-1)/2 = 28$, 其中 N 是指参加沟通的人数。由于成员之间没有沟通, 所以沟通路径为 7。

答案: (19) D

(20) 试题分析: 常见的命名对象有变量、函数、数据类型。

答案: (20) B

(21) 试题分析: 首先所有的选项都是以 b 结尾的, 但只有 D 选项中的 $(a|b)^*$ 可以表示 $\{\epsilon, a, b, aa, ab, \dots\}$ 由字符 a 、 b 构成的所有字符串。

答案: (21) D

(22) 试题分析:

词法分析阶段: 输入源程序, 对构成源程序的字符串进行扫描和分解, 识别出一个个的单词, 删掉无用的信息, 报告分析时的错误。

语法分析阶段: 语法分析器以单词符号作为输入, 分析单词符号是否形成符合语法规则的语法单位, 如表达式、赋值、循环等, 按语法规则分析检查每条语句是否有正确的逻辑结构。

语义分析阶段: 主要检查源程序是否存在语义错误, 并收集类型信息供后面的代码生成阶段使用, 如赋值语句的右端和左端的类型不匹配、表达式的除数是否为零等。

答案: (22) B

(23) 试题分析: $300 \times 1024 / 4 / 32 = 2400$ 。

答案: (23) B

(24) 试题分析: 首先给每个进程分配所需资源数减 1 个资源, 然后系统还有 1 个资源, 则不可能发生死锁, 即 $3 \times 4 + 1 = 13$ 个。

答案: (24) B

(25) 试题分析: 逻辑地址 = 页号 + 页内地址, 为 32 位。物理地址 = 物理块号 + 物理地址的页内地址, 其中, 页内地址 = 物理地址的页内地址。页面大小为 4K, 占 0~11 位 ($4K = 4096$)。

$= 2^{12}$), 即页内地址有 12 位, 所以十六进制数中的 D16H 是页内地址, 因为 D16H 转化成二进制是 110100010110 正好是 12 位, 那么逻辑页号就为 2。查表可知, 对应的物理块号为 4, 所以物理地址为 4D16H。

答案: (25) C

(26)、(27)、(28) 试题分析: 图中有五条箭线, 因此需要设置五个信号量, 按照从小到大的顺序分别是: $P1 \rightarrow P2$ 是 $S1$, $P2 \rightarrow P3$ 是 $S2$, $P2 \rightarrow P4$ 是 $S3$, $P3 \rightarrow P4$ 是 $S4$, $P4 \rightarrow P5$ 是 $S5$ 。每个进程开始的时候执行 P 操作 ($P1$ 没有 P 操作, 只有 V 操作), 结束的时候执行 V 操作, 如 $P2$ 开始时执行 $P(S1)$ 的操作, 结束时执行 $V(S2)$ 的操作。其他同理。

答案: (26) B (27) C (28) A

(29) 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和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。

答案: (29) D

(30) 试题分析: XP 提倡结对编程, 代码所有权归于整个开发队伍。其中, 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。

答案: (30) D

(31) 试题分析: D 选项是 B/S 结构的特点。

答案: (31) D

(32) 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, 或者其实现代码行数在 50~200 行, 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, 一般来说, 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。

答案：(32) D

(33)、(34) 试题分析：内聚按强度从低到高有以下几种类型：

1) 偶然内聚。如果一个模块的各成分之间毫无关系，则称为偶然内聚，也就是说模块完成一组任务，这些任务之间的关系松散，实际上没有什么联系。

2) 逻辑内聚。几个逻辑上相关的功能被放在同一模块中，则称为逻辑内聚，如一个模块读取各种不同类型外设的输入。尽管逻辑内聚比偶然内聚合理一些，但逻辑内聚的模块各成分在功能上并无关系，即使局部功能的修改有时也会影响全局，因此这类模块的修改也比较困难。

3) 时间内聚。如果一个模块完成的功能必须在同一时间内执行（如系统初始化），但这些功能只是因为时间因素关联在一起，则称为时间内聚。

4) 通信内聚。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。

5) 顺序内聚。如果一个模块的各个成分和同一个功能密切相关，而且一个成分的输出作为另一个成分的输入，则称为顺序内聚。

6) 过程内聚。构件或操作的组合方式是，允许在调用前面的构件或操作之后，马上调用后面的构件或操作，即使两者之间没有数据进行传递。

模块完成多个需要按一定的步骤一次完成的功能（过程相关—控制耦合）。例如，在用程序流程图设计模块时，若将程序流程图中的一部分划出各自组成模块，便形成过程内聚。

7) 信息内聚。模块完成多个功能，各个功能都在同一数据结构上操作，每一项功能有一个唯一的入口点。这个模块将根据不同的要求，确定该模块执行哪一个功能。由于这个模块的所有功能都是基于同一个数据结构（符号表），因此，它是一个信息内聚的模块。

8) 功能内聚。模块的所有成分对于完成单一的功能都是必须的，则称为功能内聚。

答案：(33) A (34) D

(35)、(36) 试题分析：覆盖两条路径就能达到语句覆盖的要求，用两个测试用例即可。路径覆盖需要把程序中的所有路径均覆盖一遍，需要四个用例。

整个程序流程图转化为节点图之后，一共 11 个节点、13 条边，根据环路复杂度公式有 $13-11+2=4$ 。

答案：(35) B (36) D

(37)、(38) 试题分析：

多重继承是编程语言中的概念，多重继承指的是一个类可以继承另外一个类，而另外一个类又可以继承别的类，比如 A 类继承 B 类，而 A 类又可以继承 C 类，这就是多重继承。多重继承可能造成混淆的情况，出现二义性的成员。

答案：(37) A (38) D

(39) 试题分析：采用面向对象方法进行软件开发，在分析阶段中架构师主要关注系统的行为，即系统应该做什么。

答案：(39) D

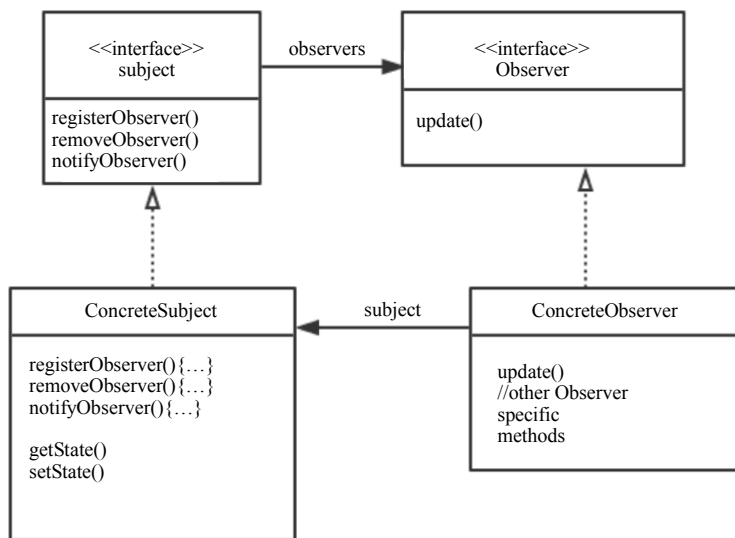
(40) 试题分析：多态按字面的意思就是“多种状态”。在面向对象语言中，接口多种不同的实现方式即为多态。例如，不同的足球运动员在进行射门这个动作时，会产生不同的结果。在运行时，可以通过指向基类的指针，来调用实现派生类中的方法。也就是说，客户类其实在调用方法时，并不需要知道特定子类的实现，都会用统一的方式来调用。

答案：(40) A

(41)、(42)、(43) 试题分析：通信图也叫作协作图，描述的是对象和对象之间的关系，即一个类操作的实现。简而言之就是对象和对象之间的调用关系，体现的是一种组织关系。“协作”作为一个结构事物用于表达静态结构和动态行为的概念组合，表达不同事物相互协作完成一个复杂功能。文本框中的“:”号表示一个对象，“:”号前的部分是对象名，“:”号后的部分是类名，对象之间连线上面的箭头所标识的是对象之间通信的消息。

答案：(41) C (42) B (43) D

(44)、(45) 试题分析：观察者将自己注册到事件，那么具体的事件就知道了自己的观察者。观察者和事件都有自己的抽象，当实现具体的观察者和事件的时候都要实现相应的接口，所以对扩展是开放的，如下图所示。



答案：(44) C (45) A

(46)、(47) 试题分析：装饰模式是一种对象结构型模式，可以动态地给一个对象增加一些额外的职责，就增加对象功能来说，装饰模式比生成子类实现更为灵活。通过装饰模式，可以在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责；当需要动态地给一个对象增加功能，这些功能可以再动态地被撤销时可以使用装饰模式；当不能采用生成子类的方法进行扩充时也可以使用装饰模式。

外观模式是对象的结构模式，要求外部与一个子系统的通信必须通过一个统一的外观对象进行，为子系统中的一组接口提供一个一致的界面，外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

答案：(46) B (47) D

(48) 试题分析：解析路径为 ACEEBDD。

答案：(48) C

(49) 试题分析：传值与传引用的区别是：值传递时，实参被拷贝了一份，只能在函数体内使用。因此，a 代表的其实就是 x 本身，只有 a 的变化才能导致 main 函数里面的 x 值的变化。

答案：(49) B

(50) 试题分析：表达式语法树后缀形式，就是对树进行后序遍历(左右根)，结果为 $x5y+*a/b-$ 。

答案：(50) A

(51)、(52) 试题分析：若事务 T1 对数据 D1 加上共享锁，则其他事务只能再对 D1 加共享锁，而不能加排他锁；若事务 T2、T3 对数据 D2、D3 加上排他锁，则其他事务不能再对 D2、D3 加任何锁。

答案：(51) D (52) C

(53) 试题分析：候选关键字是 A1A3、A1A2，必有的属性是 A1。

答案：(53) A

(54)、(55)、(56) 试题分析：从“每个供应商可以为多个项目供应多种零件，每个项目可由多个供应商供应多种零件”中判断 SP_P 的联系类型是多对多对多的关系。而选择供应商号和供应零件的数量只能从新的关系模式 SP_P 中选择。

查询条件 Where 与 Having 的区别：Where 是针对单条记录的判断条件，而 Having 是针对分组之后的判断条件。同时，由于考虑到项目号可能重复，所以要加 Distinct 去掉重复的项目。

答案：(54) A (55) D (56) C

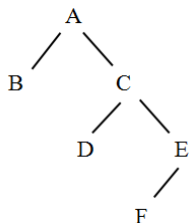
(57) 试题分析：在程序设计中，字符串(string)为符号或数值的一个连续序列，如符号串(一串字符)或二进制数字串(一串二进制数字)。A 选项中的空格也属于一种字符，都是空格的字符串叫作空格串，而不是空串；B 选项的字符串是属于线性结构的；C 选项字符串的长度是指字符串所有字符个数的总和(包括空格)。

答案：(57) C

(58) 试题分析：操作步骤：a1a2 入，a2 出，a3a4 入，a4 出，a5 入，a5 出，a3 出，a1 出。

答案：(58) A

(59) 试题分析：先序遍历是根左右的方式，中序遍历是左根右的方式，最终的结果如下图所示。



答案: (59) B

(60) 试题分析: 按照条件 “ $k_i \leq k_{2i}$ 且 $k_i \leq k_{2i+1}$ ” 的要求, 带入四个选项。以 A 选项为例, 当 $i=1$ 时, $k_1(16) < k_2(25)$, $k_1(16) < k_3(40)$, 依此类推, 可得 D 选项不满足要求。

答案: (60) D

(61) 试题分析: 二分查找法的原则是: 首先, 假设表中元素是按升序排列, 将表中间位置记录的关键字与查找关键字比较, 如果两者相等, 则查找成功; 否则利用中间位置记录将表分成前、后两个子表, 如果中间位置记录的关键字大于查找关键字, 则进一步查找前一子表, 否则进一步查找后一子表。重复以上过程, 直到找到满足条件的记录, 使查找成功, 或直到子表不存在为止, 此时查找不成功。

答案: (61) B

(62) ~ (65) 试题分析: 动态规划算法与分治法不同的是, 适合于用动态规划求解的问题, 经分解得到子问题往往不是互相独立的。若用分治法来解这类问题, 则分解得到的子问题数目太多, 有些子问题被重复计算了很多次。如果能够保存已解决的子问题的答案, 而在需要时再找出已求得的答案, 这样就可以避免大量的重复计算, 节省时间。可以用一个表来记录所有已解的子问题的答案。不管该子问题以后是否被用到, 只要它被计算过, 就将其结果填入表中。这就是动态规划法的基本思路。本题中的时间复杂度为 $O(n)$ 。

贪心选择是指所求问题的整体最优解可以通过一系列局部最优的选择, 即贪心选择来达到。这是贪心算法可行的第一个基本要素, 也是贪心算法与动态规划算法的主要区别。

回溯算法实际上是一个类似枚举的搜索尝试过程, 主要是在搜索尝试的过程中寻找问题的解, 当发现已不满足求解条件时, 就“回溯”返回, 尝试别的路径。回溯法是一种选优搜索法, 按选优条件向前搜索, 以达到目标。但当探索到某一步时, 发现原先选择并不优或达不到目标, 就退回一步重新选择, 这种走不通就退回再走的技术即为回溯法, 而满足回溯条件的某个状态的点称为“回溯点”。

求最短的装配时间与装配路线只需要将选项按照公式带入计算(将图上每条路径上的所有数字相加)可得最短路线为 $S11 \rightarrow S22 \rightarrow S13$, 时间为 21。

答案: (62) B (63) B (64) A (65) B

(66) 试题分析: 本地主机将首先在本机 hosts 文件中查询该网址对应的 IP 地址。

答案: (66) B

(67) 试题分析: Linux 中只有一个根目录, 用 “/” 表示。

答案：(67) C

(68) 试题分析：TCP 与 UDP 是基于 IP 协议的，SMTP 是基于 TCP 协议的。

答案：(68) C

(69) 试题分析：总的数据速率为 $(1+7+2) \times 500 = 5000$ ，其中有效数据是 7 位，那么有效数据速率为 $5000 \times 7/10 = 3500\text{b/s}$ 。

答案：(69) C

(70) 试题分析：静态路由是指由用户或网络管理员手工配置的路由信息。当网络的拓扑结构或链路的状态发生变化时，网络管理员需要手工去修改路由表中相关的静态路由信息。

随机路由使用前向代理收集网络中的有限全局信息，即当前节点到其源节点的旅行时间，并以此来更新节点的旅行时间表。

洪泛路由是一种简单的路由算法，将收到的封包往所有可能连结的路径上递送，直到封包到达为止。

动态路由就是自适应路由选择算法，是指路由器能够自动地建立自己的路由表，并且能够根据实际情况的变化适时地进行调整。

答案：(70) D

(71) ~ (75) 试题分析：

翻译：

软件的优点在于其功能、内部结构和由团队创建的方式。对于用户来说，通过直观和 (71) 界面呈现的正确功能的程序是出色的软件。对于软件设计师来说，分割的内部结构是一种简单而直观的方式，最小化内部耦合是美观的。对于开发人员和经理来说，一个积极的开发团队每周都取得重大进展，并且生产无缺陷的代码是件美好的事情。

我们的世界需要大量的软件。五十年前，软件是在大多数公司和工业环境中运行的。现在软件存在在我们的手机、手表、电器、汽车、玩具和工具中，并且对新的和更好的软件的需求永远不会 (72)。随着我们文明的发展和壮大，随着发展中国家建设基础设施，发达国家努力实现更高的效率，越来越多的软件需求 (73) 增长。如果在所有的软件中没有美存在的话，这将是一个很大的耻辱。

我们知道软件可能很难使用，有不可靠、容易出错的结构。我们知道这些结构使得它们变得昂贵和难以改变。我们知道有一些软件系统通过尴尬和繁琐的界面来呈现其功能。我们知道有软件系统崩溃和捣乱行为。这些都是 (74) 系统。不幸的是，作为专业人士，软件开发人员开发出难用的系统多过好用的系统。

这是优秀的软件开发者知道的秘密。好用的软件比难用的更便宜、更快。一个好用的软件系统相当于一个难用的系统来说，建立和维护要花的时间与金钱会少得多。很多新手软件开发人员不明白这一点。他们认为做每一个事情必须快速，更快速。他们认为软件之美是 (75)。不！快速开发使软件变得僵硬，难以理解而好用的系统灵活易懂，使得开发和维护工作成为一种快乐。难用的软件不切实际，会减慢速度，会使软件昂贵而脆弱。美观的系统成本最低，建立和维护成本最低，交

货时间也最短。

- | | | | |
|--------------|--------|----------|--------|
| (71) A. 简单 | B. 困难 | C. 复杂 | D. 复制品 |
| (72) A. 发生 | B. 存在 | C. 停止 | D. 开始 |
| (73) A. 开始 | B. 持续 | C. 出现 | D. 停止 |
| (74) A. 实用的 | B. 有用的 | C. 美丽的 | D. 丑陋的 |
| (75) A. 不实用的 | B. 完美的 | C. 浪费时间的 | D. 实用的 |

答案: (71) A (72) C (73) B (74) D (75) A

下午案例分析

试题一 (15 分)

【问题 1】(5 分)

答案: E1: 供应商; E2: 采购部门; E3: 检验员; E4: 库管员; E5: S/R 职员。

【问题 2】(4 分)

答案: D1: 库存表; D2: 采购订单表; D3: 质量标准表; D4: 供应商表。

【问题 3】(4 分)

答案:

检查库存信息: P1 (检查库存水平) —— D1 (部件库存表)

产品送检: P3 (验证装运部件) —— P4 (校验部件质量)

装运错误通知: P3 (验证装运部件) —— E1 (供应商)

缺陷装运通知: P4 (校验部件质量) —— E1 (供应商)

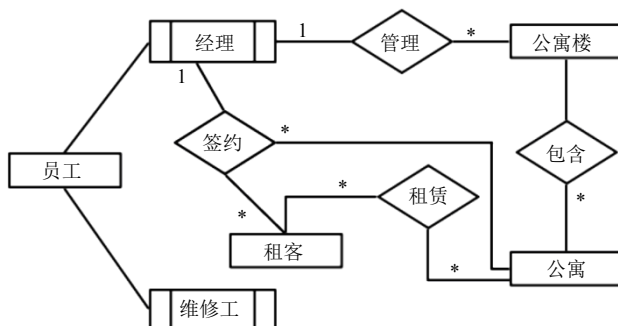
【问题 4】(2 分)

答案: 父图中某个加工的输入输出数据流必须与其子图的输入输出数据流在数量上和内容上保持一致。父图的一个输入 (或输出) 数据流对应子图中几个输入 (或输出) 数据流, 而子图中组成的这些数据流的数据项全体正好是父图中的这一个数据流。

试题二 (15 分)

【问题 1】(4.5 分)

答案: 如下图所示。

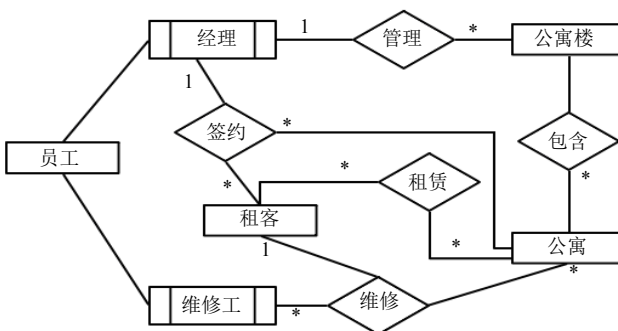


【问题 2】(4.5 分)

答案：(a) 业务技能；(b) 楼编号；(c) 月租金。

【问题 3】(6 分)

答案：如下图所示。



新增维修关系：维修工维修公寓，关系模式为维修情况。

维修情况（故障编号，员工编号，楼编号，公寓号，维修日期，维修内容）。

试题三 (15 分)

【问题 1】(6 分)

答案：

(a) C4、C5、C7、C10、C11

(b) C3、C8

(c) C1、C2、C6、C9、C12

【问题 2】(4 分)

答案：

X1：收货地址

X2：支付方式

X3：邮箱地址

X4: 定制属性

【问题 3】(5 分)

答案:

S1: 订单挂起

S2: 订单备货

S3: 订单定制

S4: 订单发货

S5: 订单收货

试题四 (14 分)

【问题 1】(6 分)

答案:

(1) $\text{first} + (\text{last} - \text{first}) / 2$ 或 $(\text{first} + \text{last}) / 2$

(2) $\text{firstSum} < \text{lastSum}$

(3) $\text{first} + (\text{last} - \text{first}) / 2$ 或 $(\text{first} + \text{last}) / 2$

【问题 2】(4 分)

答案: 分治法、 $O(n \log n)$ 。

【问题 3】(4 分)

答案: 2、4。

试题五 (16 分)

答案:

(1) `virtual void buildParts()`

(2) `this->pizzaBuilder=pizzaBuilder`

(3) `pizzaBuilder->buildParts()`

(4) `waiter->setPizzaBuilder(hawaiian_pizzabuilder)`

(5) `waiter->construct()`

试题六 (16 分)

答案:

(1) `abstract void buildParts()`

(2) `this.pizzaBuilder=pizzaBuilder`

(3) `pizzaBuilder.buildParts()`

(4) `waiter.setPizzaBuilder(hawaiian_pizzabuilder)`

(5) `waiter.construct()`

第24小时

2017年11月考试真题（上午基础知识）



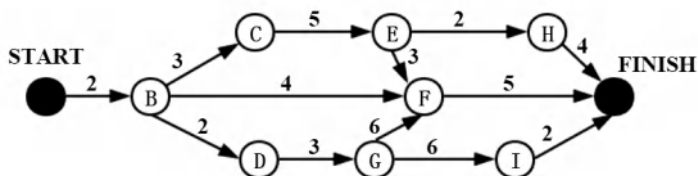
【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

- 在程序执行过程中，Cache 与主存的地址映射是由（1）完成的。
(1) A. 操作系统 B. 程序员调度 C. 硬件自动 D. 用户软件
- 某四级指令流水线分别完成取指、取数、运算、保存结果四步操作。若完成上述操作的时间依次为 8ns、9ns、4ns、8ns，则该流水线的操作周期应至少为（2）ns。
(2) A. 4 B. 8 C. 9 D. 33
- 内存按字节编址，若用存储容量为 $32\text{K} \times 8\text{bit}$ 的存储器芯片构成地址从 A0000H 到 DFFFFH 的内存，则至少需要（3）片芯片。
(3) A. 4 B. 8 C. 16 D. 32
- 计算机系统的主存主要是由（4）构成的。
(4) A. DRAM B. SRAM C. Cache D. EEPROM
- 以下关于海明码的叙述中，正确的是（5）。
(5) A. 海明码利用奇偶性进行检错和纠错
 B. 海明码的码距为 1

- C. 海明码可以检错但不能纠错
D. 海明码中数据位的长度与校验位的长度必须相同
- 计算机运行过程中, CPU 需要与外设进行数据交换。采用__(6)__控制技术时, CPU 与外设可以并行工作。
(6) A. 程序查询方式和中断方式
B. 中断方式和 DMA 方式
C. 程序查询方式和 DMA 方式
D. 程序查询方式、中断方式和 DMA 方式
 - 与 HTTP 相比, HTTPS 协议对传输的内容进行加密, 更加安全。HTTPS 基于__(7)__安全协议, 其默认端口是__(8)。
(7) A. RSA B. DES C. SSL D. SSH
(8) A. 1023 B. 443 C. 80 D. 8080
 - 下列攻击行为中, 属于典型被动攻击的是__(9)。
(9) A. 拒绝服务攻击 B. 会话拦截 C. 系统干涉 D. 修改数据命令
 - __(10)__不属于入侵检测技术。
(10) A. 专家系统 B. 模型检测 C. 简单匹配 D. 漏洞扫描
 - 以下关于防火墙功能特性的叙述中, 不正确的是__(11)。
(11) A. 控制进出网络的数据包和数据流向
B. 提供流量信息的日志和审计
C. 隐藏内部 IP 和网络结构细节
D. 提供漏洞扫描功能
 - 某软件公司项目组的程序员在程序编写完成后均按公司规定撰写文档, 并上交公司存档。此情形下, 该软件文档著作权应由__(12)__享有。
(12) A. 程序员 B. 公司与项目组共同 C. 公司 D. 项目组全体人员
 - 我国商标法规定了申请注册的商标不得使用的文字和图形, 其中包括县级以上行政区的地名(文字)。以下商标若要注册申请, 经审查, 能获准注册的商标是__(13)。
(13) A. 青岛(市) B. 黄山(市) C. 海口(市) D. 长沙(市)
 - 李某购买了一张有注册商标的应用软件光盘, 则李某享有__(14)。
(14) A. 注册商标专用权 B. 该光盘的所有权
C. 该软件的著作权 D. 该软件的所有权
 - 某医院预约系统的部分需求为: 患者可以查看医院发布的专家特长介绍及其就诊时间; 系统记录患者信息, 患者预约特定时间就诊。用 DFD 对其进行功能建模时, 患者是__(15); 用 ERD 对其进行数据建模时, 患者是__(16)。
(15) A. 外部实体 B. 加工 C. 数据流 D. 数据存储
(16) A. 实体 B. 属性 C. 联系 D. 弱实体

- 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，链接顶点的边表示包含的活动，变色数字表示活动的持续时间（天）。完成该项目的最少时间为（17）天。由于某种原因，现在需要同一个开发人员完成 BC 和 BD，则完成该项目的最少时间为（18）天。



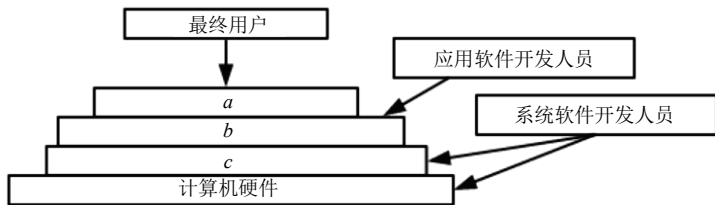
- (17) A. 11 B. 18 C. 20 D. 21
- (18) A. 11 B. 18 C. 20 D. 21
- 某企业财务系统的需求中，属于功能需求的是（19）。

(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
 - 更适合用来开发操作系统的编程语言是（20）。

(20) A. C/C++ B. Java C. Python D. JavaScript
 - 以下关于程序设计语言的叙述中，不正确的是（21）。

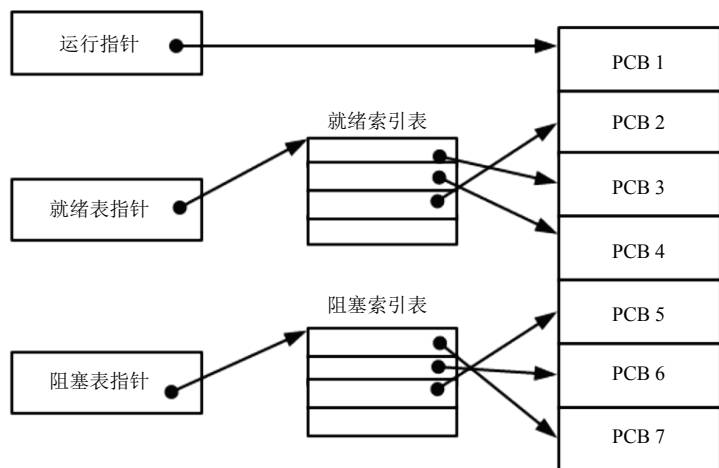
(21) A. 脚本语言中不使用变量和函数 B. 标记语言常用于描述格式化和链接
 C. 脚本语言采用解释方式实现 D. 编译型语言的执行效率更高
 - 将高级语言源程序通过编译或解释方式进行翻译时，可以先生成与源程序等价的某种中间代码。以下关于中间代码的叙述中，正确的是（22）。

(22) A. 中间代码常采用符号表来表示
 B. 后缀式和三地址码是常用的中间代码
 C. 对中间代码进行优化要依据运行程序的机器特性
 D. 中间代码不能跨平台
 - 计算机系统的层次结构如下图所示，基于硬件上的软件可以分为 a、b、c 三个层次。图中 a、b 和 c 分别表示（23）。



- (23) A. 操作系统、系统软件和应用软件 B. 操作系统、应用软件和系统软件
C. 应用软件、系统软件和操作系统 D. 应用软件、操作系统和系统软件

- 下图所示的 PCB (进程控制块) 的组织方式是 (24), 图中 (25)。



- (24) A. 链接方式 B. 索引方式
C. 顺序方式 D. Hash

- (25) A. 有 1 个运行进程、2 个就绪进程、4 个阻塞进程
B. 有 2 个运行进程、3 个就绪进程、2 个阻塞进程
C. 有 1 个运行进程、3 个就绪进程、3 个阻塞进程
D. 有 1 个运行进程、4 个就绪进程、2 个阻塞进程

- 某文件系统采用多级索引结构。若磁盘块的大小为 1K 字节, 每个块号占 3 字节, 那么采用二级索引时的文件最大长度为 (26) K 字节。

- (26) A. 1024 B. 2048
C. 116281 D. 232562

- 某操作系统采用分页存储管理方式, 下图给出了进程 A 和进程 B 的页表结构。如果物理页的大小为 1K 字节, 那么进程 A 中逻辑地址为 1024 (十进制) 用变量存放在 (27) 号物理内存页中。假设进程 A 的逻辑页 4 与进程 B 的逻辑页 5 要共享物理页 4, 那么应该在进程 A 页表的逻辑页 4 和进程 B 页表的逻辑页 5 的物理页处分别填 (28)。

进程 A 页表

逻辑页	物理页
0	8
1	3
2	5
3	2
4	
5	

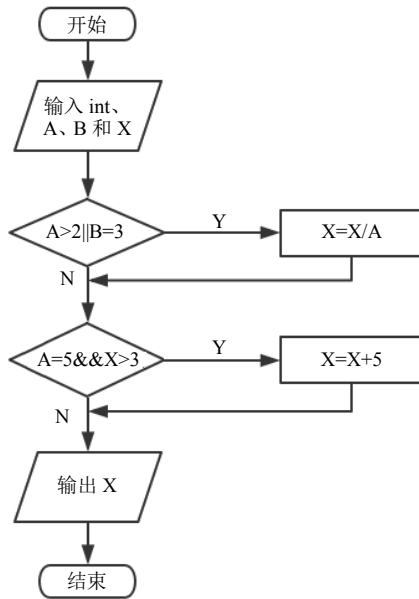
进程 B 页表

逻辑页	物理页
0	1
1	6
2	9
3	7
4	0
5	

0
1
2
3
4
5
6
7
8
9

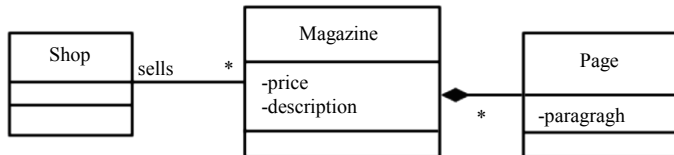
- (27) A. 8 B. 3 C. 5 D. 2
 (28) A. 4、4 B. 4、5 C. 5、4 D. 5、5

- 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,则至少要 (29) 个测试用例,正确的测试用例对是 (30) (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. (2,2,16;16) 和 (5,2,21;9)

- 配置管理贯穿软件开发的整个过程。以下内容中,不属于配置管理的是__(31)___。
(31) A. 版本控制 B. 风险管理 C. 变更管理 D. 配置状态报告
- 极限编程(XP)的12个最佳实践不包括__(32)___。
(32) A. 小的发布 B. 结对编程 C. 持续集成 D. 精心设计
- 以下关于管道过滤器体系结构的优点的叙述中,不正确的是__(33)___。
(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
- 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__(34)___耦合。
(34) A. 数据 B. 标记 C. 控制 D. 内容
- 某模块涉及多个功能,这些功能必须以特定的次序执行,则该模块的内聚类型为__(35)___内聚。
(35) A. 实践 B. 过程 C. 信息 D. 功能
- 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__(36)___维护。
(36) A. 改正性 B. 适应性 C. 改善性 D. 预防性
- 在面向对象方法中,将逻辑上相关的数据和行为绑定在一起,使信息对使用者隐蔽称为__(37)___。当类中的属性或方法被设计为 private 时,__(38)___可以对其进行访问。
(37) A. 抽象 B. 继承 C. 封装 D. 多态
(38) A. 应用程序中的所有的方法 B. 只有此类中定义的方法
C. 只有此类中定义的 public 方法 D. 同一个包的类中定义的方法
- 采用继承机制创建子类时,子类中__(39)___。
(39) A. 只能有父类中的属性 B. 只能有父类中的行为
C. 只能新增行为 D. 可以有新的属性和行为
- 面向对象分析过程中,从给定需求描述中选择__(40)___来识别对象。
(40) A. 动词短语 B. 名词短语 C. 形容词 D. 副词
- 如图所示的UML类图中,Shop和Magazine之间为__(41)___关系,Magazine和Page之间为__(42)___关系。UML类图通常不用于对__(43)___进行建模。



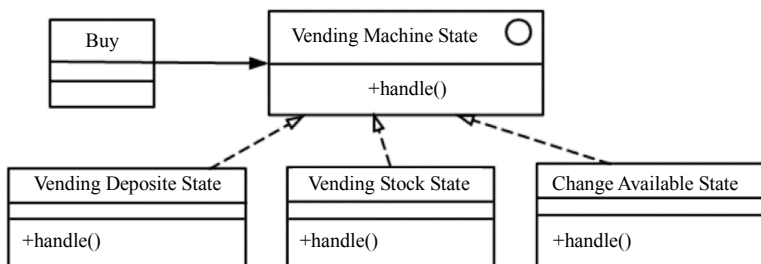
- (41) A. 关联 B. 依赖 C. 组合 D. 继承

(42) A. 关联 B. 依赖 C. 组合 D. 继承

(43) A. 系统的词汇 B. 简单的协作

C. 逻辑数据库模式 D. 对象快照

- 自动售货机根据库存、存放货币量、找零能力、所选项目等不同，在货币存入并进行选择时具有如下行为：交付产品不找零；交付产品找零；存入货币不足而不提供任何产品；库存不足而不提供任何产品。这一业务需求适合采用 (44) 模式设计实现，其类图如下图所示，其中 (45) 是客户程序使用的主要接口，可用状态来对其进行配置。此模式为 (46)，体现的最主要的意图是 (47)。



(44) A. 观察者 (Observer)

B. 状态 (State)

C. 策略 (Strategy)

D. 访问者 (Visitor)

(45) A. Vending Machine State

B. Buy

C. Vending Deposit State

D. Vending Stock State

(46) A. 创建型对象模式

B. 结构型对象模式

C. 行为型类模式

D. 行为型对象模式

(47) A. 当一个对象状态改变时，所有依赖它的对象得到通知并自动更新

B. 在不破坏封装性的前提下，捕获对象的内部状态并在对象之外保存

C. 一个对象在其内部状态改变时改变其行为

D. 将请求封装为对象，从而可以使用不同的请求对客户进行参数化

- 编译过程中进行的语法分析主要是分析 (48)。

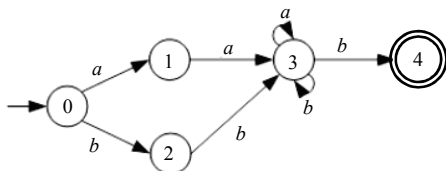
(48) A. 源程序中的标识符是否合法

B. 程序语句的含义是否合法

C. 程序语句的结构是否合法

D. 表达式的类型是否合法

- 某确定的有限自动机 (DFA) 的状态转换图如下图所示 (0 是初态，4 是终态)，则该 DFA 能识别 (49)。



(49) A. *aaab* B. *abab* C. *bbba* D. *abba*

- 函数 `main()`、`f()` 的定义如下所示。调用函数 `f()` 时, 第一个参数采用传值 (call by value) 方式, 第二个参数采用传引用 (call by reference) 方式, 则函数 `main()` 执行后输出的值为 (50)。

`main()`

```
int x=10;
f(x,x);
print(x);
```

`f(int x, int & a)`

```
x=2*x-1;
a=a+x;
return;
```

(50) A. 10 B. 19 C. 20 D. 29

- 采用三级结构或两级映像的数据库体系结构, 如果对数据库的一张表创建聚簇索引, 改变的是数据库的 (51)。

(51) A. 用户模式 B. 外模式 C. 模式 D. 内模式

- 某企业的培训关系模式 R (培训科目, 培训师, 学生, 成绩, 时间, 教室), R 的函数依赖集 $F=\{\text{培训科目} \rightarrow \text{培训师}, (\text{学生}, \text{培训科目}) \rightarrow \text{成绩}, (\text{时间}, \text{教室}) \rightarrow \text{培训科目}, (\text{时间}, \text{培训师}) \rightarrow \text{教室}, (\text{时间}, \text{学生}) \rightarrow \text{教室}\}$ 。关系模式 R 的主键为 (52), 其规范化程度最高达到 (53)。

(52) A. (学生, 培训科目) B. (时间, 教室)
C. (时间, 培训师) D. (时间, 学生)

(53) A. 1NF B. 2NF C. 3NF D. BCNF

- 设关系模式 $R(U, F)$, 其中, $U=\{A, B, C, D, E\}$, $F=\{A \rightarrow B, DE \rightarrow B, CB \rightarrow E, E \rightarrow A, B \rightarrow D\}$ 。(54) 为关系模式 R 的候选关键字, 分解 (55) 是无损连接, 并保持函数依赖性。

(54) A. AB B. DE C. DB D. CE

(55) A. $\rho=\{R_1(AC), R_2(ED), R_3(B)\}$ B. $\rho=\{R_1(AC), R_2(E), R_3(DB)\}$
C. $\rho=\{R_1(AC), R_2(ED), R_3(AB)\}$ D. $\rho=\{R_1(ABC), R_2(ED), R_3(ACE)\}$

- 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)。

(56) A. JDBC B. XML C. CGI D. COM

- 设 S 是一个长度为 n 的非空字符串, 其中的字符各不相同, 则其互异的非平凡子串 (非空且不同于 S 本身) 个数为 (57)。

(57) A. $2n-1$ B. n^2 C. $n(n+1)/2$ D. $(n+2)(n-1)/2$

- 假设某消息中只包含七个字符 $\{a, b, c, d, e, f, g\}$, 这七个字符在消息中出现的次数为 $\{5, 24, 8, 17, 34, 4, 13\}$, 利用哈夫曼树 (最优二叉树) 为该消息中的字符构造符合前缀编码要求的不等长编码。各字符的编码长度分别为 (58)。

(58) A. $a:4, b:2, c:3, d:3, e:2, f:4, g:3$ B. $a:6, b:2, c:5, d:3, e:1, f:6, g:4$
C. $a:3, b:3, c:3, d:3, e:3, f:2, g:3$ D. $a:2, b:6, c:3, d:5, e:6, f:1, g:4$

- 设某二叉树采用二叉链表表示（即节点的两个指针分别指示左、右孩子）。当该二叉树包含 k 个节点时，其二叉链表节点中必有 (59) 个空的孩子指针。

(59) A. $k-1$ B. k C. $k+1$ D. $2k$

- 以下关于无向连通图 G 的叙述中，不正确的是 (60)。

(60) A. G 中任意两个顶点之间均有边存在
 B. G 中任意两个顶点之间存在路径
 C. 从 G 中任意顶点出发可以遍历图中所有顶点
 D. G 的邻接矩阵是对称矩阵

- 两个递增序列 A 和 B 的长度分别为 m 和 n ($m < n$ 且 m 与 n 接近)，将二者归并为一个长度为 $m+n$ 的递增序列。当元素关系为 (61)，归并过程中元素的比较次数最少。

(61) A. $a_1 < a_2 < \dots < a_{m-1} < a_m < b_1 < b_2 < \dots < b_{n-1} < b_n$
 B. $b_1 < b_2 < \dots < b_{n-1} < b_n < a_1 < a_2 < \dots < a_{m-1} < a_m$
 C. $a_1 < b_1 < a_2 < b_2 < \dots < a_{m-1} < b_{m-1} < a_m < b_m < b_{m+1} < \dots < b_{n-1} < b_n$
 D. $b_1 < b_2 < \dots < b_{m-1} < b_m < a_1 < a_2 < \dots < a_{m-1} < a_m < b_{m+1} < \dots < b_{n-1} < b_n$

- 求解两个长度为 n 的序列 X 和 Y 的一个最长公共子序列（如序列 ABCBDAB 和 BDCABA 的一个最长公共子序列为 BCBA），可以采用多种计算方法。例如，可以采用蛮力法，对 X 的每一个子序列，判断其是否也是 Y 的子序列，最后求出最长的即可，该方法的时间复杂度为 (62)。经分析发现该问题具有最优子序列，可以定义序列长度分别为 i 和 j 的两个序列 X 和 Y 的最长公共子序列的长度为 $C[i, j]$ ，如下式所示。该方法的时间复杂度为 (63)。

$$C[i, j] = \begin{cases} 0 & i = 0 \text{ 或 } j = 0 \\ C[i-1, j-1] + 1 & i, j > 0 \text{ 且 } X_i = Y_j \\ \max\{C[i-1, j], C[i, j-1]\} & \text{其他} \end{cases}$$

(62) A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n^{2^n})$
 (63) A. $O(n^2)$ B. $O(n^2 \lg n)$ C. $O(n^3)$ D. $O(n^{2^n})$

- 现需要对一个基本有序的数组进行排序。此时最适宜采用的算法为 (64) 排算法，时间复杂度为 (65)。

(64) A. 插入 B. 快速 C. 归并 D. 堆
 (65) A. $O(n)$ B. $O(n \lg n)$ C. $O(n^2)$ D. $O(n^2 \lg n)$

- 相比于 TCP，UDP 的优势为 (66)。

(66) A. 可靠传输 B. 快消较小 C. 拥塞控制 D. 流量控制

- 若一台服务器只开放了 25 和 110 两个端口，那么这台服务器可以提供 (67) 服务。

(67) A. E-Mail B. WEB C. DNS D. FTP

- SNMP 是一种异步请求或响应协议，采用 (68) 协议进行封装。

(68) A. IP B. ICMP C. TCP D. UDP

- 在一台安装好 TCP/IP 协议的计算机上，当网络连接不可用时，为了测试编写好的网络程序，

通常使用的目的主机 IP 地址为 (69)。

- (69) A. 0.0.0.0 B. 127.0.0.0 C. 10.0.0.1 D. 210.225.21.255/24

- 测试网络连通性通常采用的命令是 (70)。

- (70) A. Nestat B. Ping C. msconfig D. cmd

- The development of the Semantic Web proceeds in steps, each step building a layer on top of another. The pragmatic justification for this approach is that it is easier to achieve (71) on small steps, whereas it is much harder to get everyone on board if too much is attempted. Usually there are several research groups moving in different directions; this (72) of ideas is a major driving force for scientific progress. However, from an engineering perspective there is a need to standardize. So, if most researchers agree on certain issues and disagree on others, it makes sense to fix the point of agreement. This way, even if the more ambitious research efforts should fail, there will be at least (73) positive outcomes. Once a (74) has been established, many more groups and companies will adopt it, instead of waiting to see which of the alternative research lines will be successful in the end. The nature of the Semantic Web is such that companies and single users must build tools, add content, and use that content. We cannot wait until the full Semantic Web vision materializes-it may take another ten years for it to be realized to its full (75) (as envisioned today, of course).

- (71) A. conflicts B. consensus C. success D. disagreement

- (72) A. competition B. agreement C. cooperation D. collaboration

- (73) A. total B. complete C. partial D. entire

- (74) A. technology B. standard C. pattern D. model

- (75) A. area B. goal C. object D. extent

第25小时

2017 年 11 月考试真题（下午案例分析）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

试题一（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某公司拟开发一个共享单车系统，采用北斗定位系统进行单车定位，提供针对用户的 APP 及微信小程序、基于 Web 的管理与监控系统。该共享单车系统的主要功能如下：

（1）用户注册登录。用户在 APP 端输入手机号并获取验证码后进行注册，将用户信息进行存储。用户登录后显示用户所在位置周围的单车。

（2）使用单车。

1) 扫码或手动开锁。通过扫描二维码或手动输入编码获取开锁密码，系统发送开锁指令进行开锁，系统修改单车状态，新建单车行程。

2) 骑行单车。单车定时上传位置，更新行程。

3) 锁车结账。用户停止使用或手动锁车结束行程后，系统根据已设置好的计费规则及使用时间自动结算，更新本次骑行的费用并显示给用户，用户确认支付后，记录行程的支付状态。系统还将重置单车的开锁密码和单车状态。

(3) 辅助管理。

1) 查询。用户可以查看行程列表和行程详细信息。

2) 报修。用户上传所在位置或单车位置及单车故障信息并进行记录。

(4) 管理与监控。

1) 单车管理及计费规则设置。商家对单车的基础信息、状态等进行管理，对计费规则进行设置并存储。

2) 单车监控。对单车、故障、行程等进行查询统计。

3) 用户管理。管理用户信用与状态信息，对用户进行查询统计。现采用结构化方法对共享单车系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和如图 1-2 所示的 0 层数据流图。

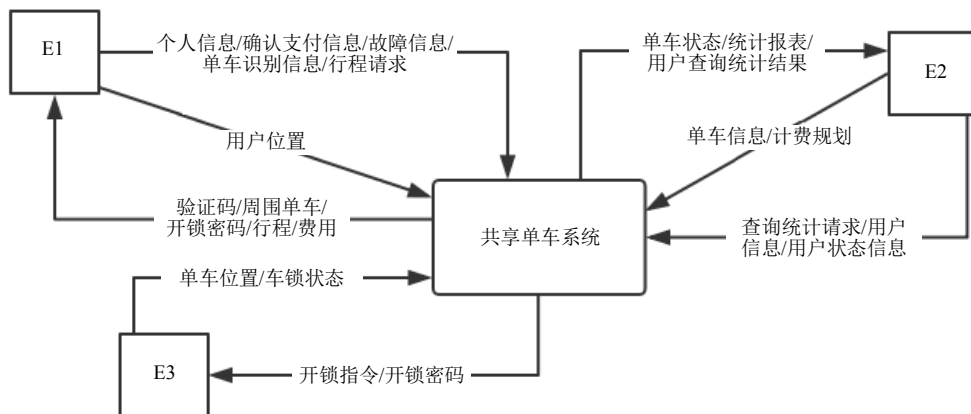


图 1-1 上下文数据流图

【问题 1】（3 分）

使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

【问题 2】（5 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1~D5 的名称。

【问题 3】（5 分）

根据说明和图中术语及符号，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】（2 分）

根据说明中的术语，说明“使用单车”可以分解为哪些子加工？

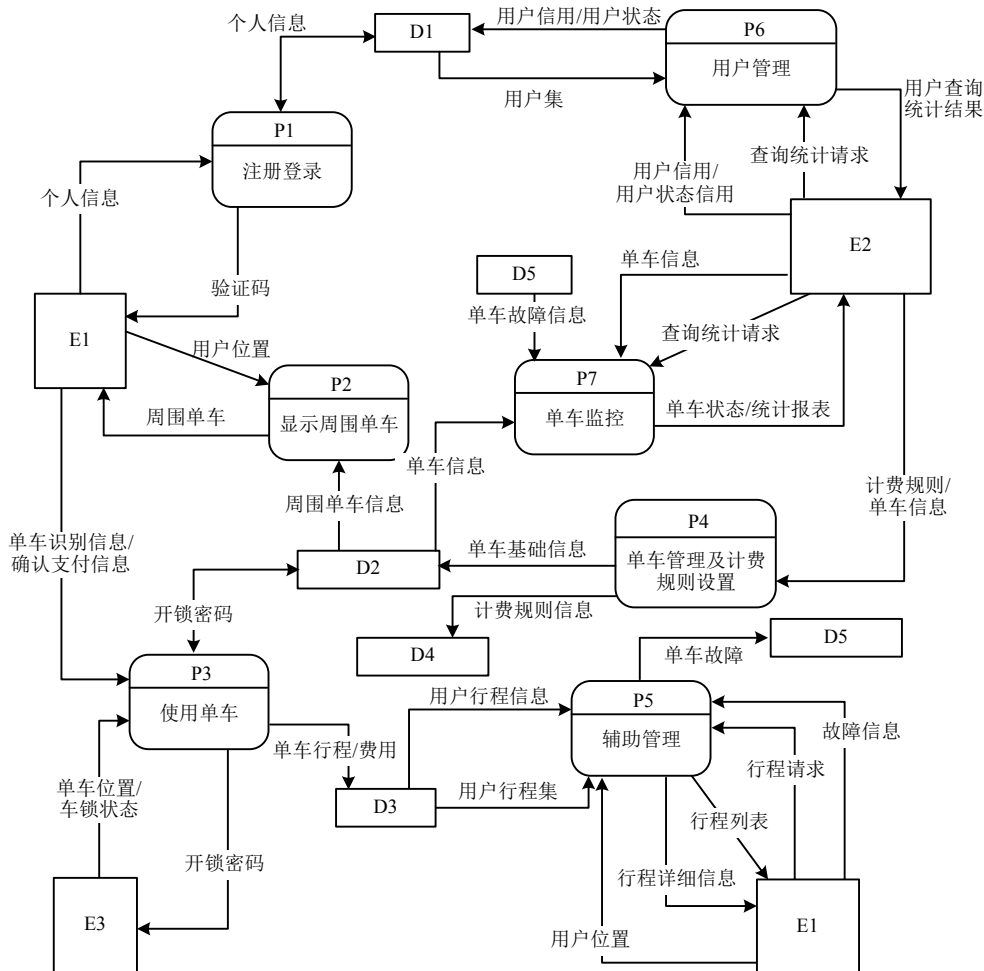


图 1-2 0 层数据流图

试题二（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

M 公司为了便于开展和管理各项业务活动，提高公司的知名度和影响力，拟构建一个基于网络的会议策划系统。

【需求分析结果】

该系统的部分功能及初步需求分析的结果如下：

（1）M 公司旗下有业务部、策划部和其他部门。部门信息包括部门号、部门名、主管、联系电话和邮箱号；每个部门只有一名主管，只负责管理本部门的工作，而且主管参照员工关系的员工

号：一个部门有多名员工，每名员工属于且仅属于一个部门。

（2）员工信息包括员工号、姓名、职位、联系方式和薪资。职位包括主管、业务员、策划员等。业务员负责受理用户申请，设置受理标志。一名业务员可以受理多个用户申请，但一个用户申请只能由一名业务员受理。

（3）用户信息包括用户号、用户名、银行账号、电话、联系地址。用户号唯一标识用户信息中的每一个元组。

（4）用户申请信息包括申请号、用户号、会议日期、天数、参会人数、地点、预算和受理标志。申请号唯一标识用户申请信息中的每一个元组，而且一个用户可以提交多个申请，但一个用户申请只对应一个用户号。

（5）策划部主管为已受理的用户申请制定会议策划任务。策划任务包括申请号、任务明细和要求完成时间。申请号唯一标识策划任务的每一个元组。一个策划任务只对应一个已受理的用户申请，但一个策划任务可由多名策划员参与执行，而且一名策划员可以参与执行且在项策划任务。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。

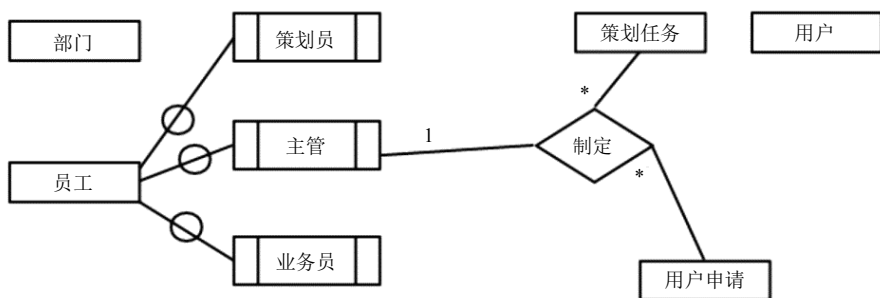


图 2-1 实体联系图

【关系模型设计】

部门（部门号，部门名，部门主管，联系电话，邮箱号）

员工（员工号，姓名，（a），联系方式，薪资）

用户（用户名，（b），电话，联系地址）

用户申请（申请号，用户号，会议日期，天数，参会人数，地点，受理标志，（c））

执行（申请号，任务明细，（d））

【问题 1】（5 分）

根据问题描述，补充五个联系，完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3、联系 4 和联系 5，联系类型为 1:1、1:n 和 m:n（或 1:1、1:* 和 *: *）。

【问题 2】（4 分）

根据题意，将关系模型中的（a）～（d）补充完整，并填入答题纸对应的位置上。

【问题 3】(4 分)

给出“用户申请”和“策划任务”关系模式的主键和外键。

【问题 4】(2 分)

请问“执行”关系模式的主键为全码的说法正确吗？为什么？

试题三 (15 分)

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某大学拟开发一个用于管理学术出版物(Publication)的数字图书馆系统，用户可以从该系统查询或下载已发表的学术出版物。系统的主要功能如下：

(1) 登录系统。系统的用户(User)仅限于该大学的学生(Student)、教师(Faculty)和其他工作人员(Staff)。在访问系统之前，用户必须使用其校园账户和密码登录系统。

(2) 查询某位作者(Author)的所有出版物。系统中保存了会议文章(ConfPaper)、期刊文章(JurnalArticle)和校内技术报告(TechReport)等学术出版物的信息，如题目、作者及出版年份等。除此之外，系统还存储了不同类型出版物的一些特有信息。

1) 对于会议文章，系统还记录了会议名称、召开时间和召开地点。

2) 对于期刊文章，系统还记录了期刊名称、出版月份、期号和主办单位。

3) 对于校内技术报告，系统记录了由学校分配的唯一 ID。

(3) 查询指定会议集(Proceedings)或某个期刊特定期(Edition)的所有文章。会议集包含了发表在该会议(在某个特定时间段、特定地点召开)上的所有文章。期刊的每一期在特定时间发行，其中包含若干篇文章。

(4) 下载出版物。系统记录了每个出版物被下载的次数。

(5) 查询引用了某篇出版物的所有出版物。在学术出版物中引用他人或早期的文献作为相关工作或背景资料是很常见的现象。用户也可以在系统中为某篇出版物注册引用通知，若有新的出版物引用了该出版物，系统将发送电子邮件通知该用户。

现在采用面向对象方法对该系统进行开发，得到系统的初始设计类图如图 3-1 所示。

【问题 1】(9 分)

根据说明中的描述，给出图 3-1 中 C1~C9 所对应的类名。

【问题 2】(4 分)

根据说明中的描述，给出图 3-1 中类 C6~C9 的属性。

【问题 3】(2 分)

图 3-1 中包含了哪种设计模式？实现的是该系统的哪个功能？

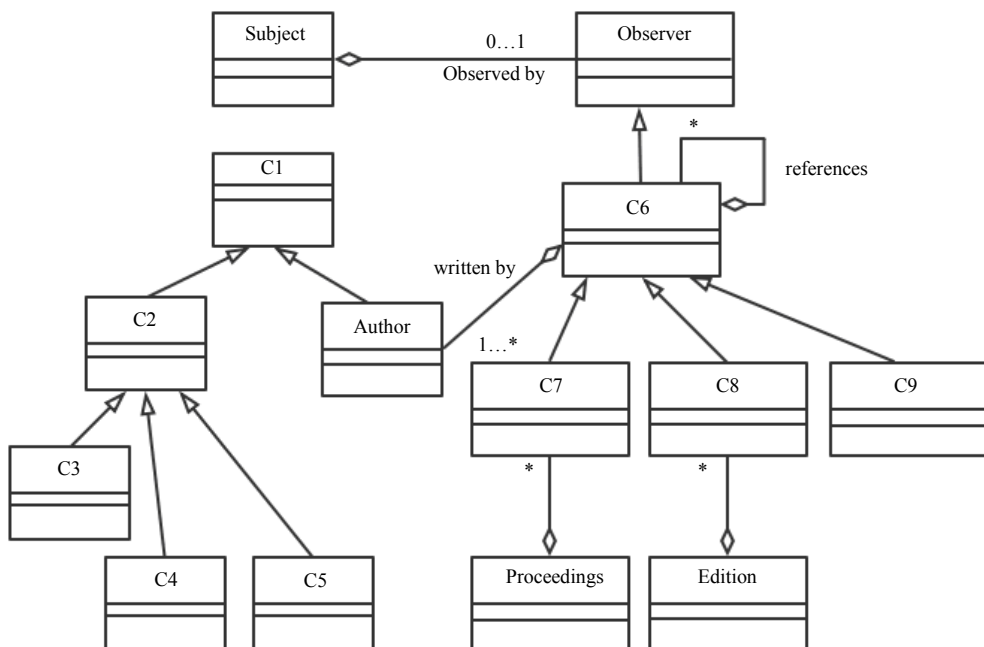


图 3-1 初始设计类图

试题四（15 分）

阅读下列说明和 C 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

一个无向连通图 G 点上的哈密顿（Hamilton）回路是指从图 G 上的某个顶点出发，经过图上所有其他顶点一次且仅一次，最后回到该顶点的路径。哈密顿回路算法的基础如下：假设图 G 存在一个从顶点 V_0 出发的哈密顿回路 $V_1-V_2-V_3-\dots-V_{n-1}-V_0$ 。算法从顶点 V_0 出发，访问该顶点的一个未被访问的邻接顶点 V_1 ，接着从顶点 V_1 出发，访问 V_1 一个未被访问的邻接顶点 V_2 ……对顶点 V_i 重复进行以下操作，访问 V_i 的一个未被访问的邻接顶点 V_{i+1} ；若 V_i 的所有邻接顶点均已被访问，则返回到顶点 V_{i-1} ，考虑 V_{i-1} 的下一个未被访问的邻接顶点，仍记为 V_i ；直到找到一条哈密顿回路或找不到哈密顿回路，算法结束。

【C 代码】

下面是算法的 C 语言实现。

（1）常量和变量说明。

n ：图 G 中的顶点数。

$c[i][j]$ ：图 G 的邻接矩阵。

k ：统计变量，当前已经访问的顶点数为 $k+1$ 。

$x[k]$: 第 k 个访问的顶点编号, 从 0 开始。

$visited[x[k]]$: 第 k 个顶点的访问标志, 0 表示未访问, 1 表示已访问。

(2) C 程序。

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
void Hamilton(int n, int x[MAX], int c[MAX][MAX]){
    int;
    int visited[MAX];
    int k;
    /*初始化 x 数组和 visited 数组*/
    for (i=0;i<n; i++){
        x[i]=0;
        visited[i]=0;
    }
    /*访问起始顶点*/
    k=0
    ____ (1) ____;
    x[0]=0
    k=k+1
    /*访问其他顶点*/
    while(k>=0){
        x[k]=x[k]+1;
        while(x[k]<n){
            if ____ (2) ____ &&c[x[k-1]][x[k]==1]){/*邻接顶点 x[k] 未被访问过*/
                break;
            }else{
                x[k]=x[k]+1
            }
        }
        if(x[k] < n && ____ (3) ____ { /*找到一条哈密顿回路*/
            for (k=0;k<n;k++){
                printf("%d--",x[k]; /*输出哈密顿回路*/
            }
            printf("%d--",x[0];
            return;
        }else if x[k]<n&&k<n-1){ /*设置当前顶点的访问标志, 继续下一个顶点*/
            ____ (4) ____;
            k=k+1;
        }else{ /*没有未被访问过的邻接顶点,回退到上一个顶点*/
            x[k]=0;
            visited x[k]=0;
            ____ (5) ____;
        }
    }
}
```

【问题 1】(10 分)

根据题干说明, 填充 C 代码中的 (1) ~ (5)。

【问题 2】（5 分）

根据题干说明和 C 代码，算法采用的设计策略为（6），该方法在遍历图的顶点时，采用的是（7）方法（深度优先或广度优先）。

试题五（15 分）

阅读下列说明和 C++ 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某图像预览程序要求能够查看 BMP、JPEG 和 GIF 三种格式的文件，而且能够 Windows 和 Linux 两种操作系统上运行。程序需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，现采用桥接（Bridge）模式进行设计，得到如图 5-1 所示的类图。

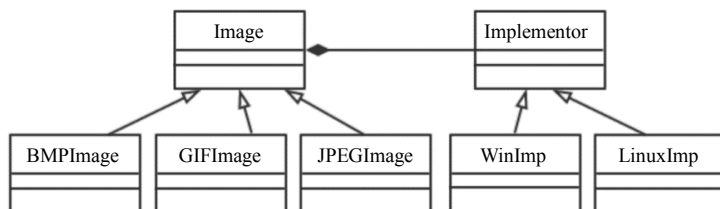


图 5-1 类图

【C++ 代码】

```

#include <iostream>
#include <string>;
Using namespace std;
class Matrix{//各种格式的文件最终都被转化为像素矩阵
//此处代码省略
};
class Implementor{
public:
    ____ (1) ____; //显示像素矩阵 m
};
class WinImp: public Implementor{
public:
    Viod doPaint(Matrix m){/*调用 Windows 系统的绘制函数绘制像素矩阵*/
};
class LinuxImp: public Implementor{/*调用 Linux 系统的绘制函数绘制像素矩阵*/
};
class Image{
public:
    viod setImp(Implement*imp) {this->imp=imp;}
    Virtual viod parseFile(string filename)=0
protected:
    Implementor*imp;
};
  
```

```

class BMPImage: public Image{
//此处省略代码
};
class GIFImage: public Image{
Public:
void parseFile(string fileName) {
//此处解析 GIF 文件并获得一个像素矩阵对象 m
__ (2) __; //显示像素矩阵 m
}
};
class JPEGImage: public Image{
//此处代码省略
};
int main(){
//在 linux 操作系统上查看 demo.gif 图像文件
Image*image=__ (3) __;
Implementor*imageImp=__ (4) __;
__ (5) __;
Image->parseFile("demo.gif");
return 0;
}

```

【问题】(15 分)

阅读上述说明和 C++代码，将应填入的字句写在答题纸的对应栏内。

试题六 (15 分)

阅读下列说明和 Java 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某图像预览程序要求能够查看 BMP、JPEG 和 GIF 三种格式的文件，而且能够在 Windows 和 Linux 两种操作系统上运行。程序需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，现采用桥接模式进行设计，得到如图 6-1 所示的类图。

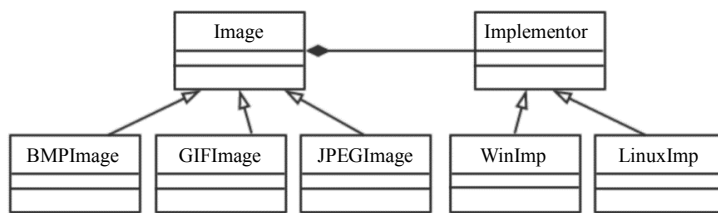


图 6-1 类图

【Java 代码】

```

import java.util.*;
class Matrix{ //各种格式的文件最终都被转化为像素矩阵
//此处代码省略
};

```

```
abstract class Implementor{
Public ____ (1) ____;    //显示像素矩阵 m
};
class WinImp extends Implementor{
public void doPaint(Matrix m){ //调用 Windows 系统的绘制函数绘制像素矩阵
}
};
class LinuxImp extends Implementor{
public void doPaint(Matrix m){ //调用 Linux 系统的绘制函数绘制像素矩阵
}
};
abstract class Image{
public void setImp(Implementor imp){ this.imp= imp; }
public abstract void parseFile(String fileName);
protected Implementor imp;
};
class BMPImage extends Image{
//此处代码省略
};
class GIFImage extends Image{
public void parseFile(String fileName) {
//此处解析 BMP 文件并获得一个像素矩阵对象 m
____ (2) ____;    //显示像素矩阵 m
}
};
Class Main{
Public static void main(String[] args){
//在 Linux 操作系统上查看 demo.gif 图像文件
Image image= ____ (3) ____;
Implementor imageImp= ____ (4) ____;
____ (5) ____;
Image.parseFile("demo.gif");
}
}
```

【问题】（15 分）

阅读上述说明和 Java 代码，将应填入的字句写在答题纸的对应栏内。

第26小时

2017 年 11 月真题解析

上午基础知识

(1) 试题分析: Cache 与主存之间的映射由硬件实现, 主存与辅存之间的交互是硬件与软件结合起来实现的。

答案: (1) C

(2) 试题分析: 流水线的周期为指令执行时间最长的一段。

答案: (2) C

(3) 试题分析: 存储器的字节数= $\text{DFFFFH}-\text{A0000H}+1=40000\text{H}=262144$, 用 $32\text{K}\times 8\text{bit}$ 的存储器芯片需要 $262144/32=8192/1024=8$ 片。

答案: (3) B

(4) 试题分析: 主存主要采用动态随机存储器 DRAM, Cache 采用静态随机存储器 SRAM, EEPROM 是带电可擦除可编程的只读存储器。

答案: (4) A

(5) 试题分析: 海明码是利用奇偶性来检错和纠错的校验方法, 码距最小为 $2n+1$ 。

答案: (5) A

(6) 试题分析: 程序查询方式的原理是: 当主机进行 I/O 操作时, 首先发出询问信号, 读取设备的状态并根据设备状态决定下一步操作究竟是进行数据传输还是等待。这种控制下, CPU 一旦启动 I/O, 必须停止现程序的运行, 并在现行程序中插入一段程序。程序查询方式的主要特点是 CPU 有踏步等待现象, CPU 与 I/O 串行工作。

程序中断是指计算机执行现程序的过程中,出现某些急需处理的异常情况和特殊请求,CPU 暂时终止现程序,而转去对随机发生的更紧迫的事件进行处理,在处理完毕后,CPU 将自动返回原来的程序继续执行。在中断方式中 CPU 与外设可并行工作。

直接内存存取 DMA 是指在内存与 I/O 设备间传送数据块的过程中,不需要 CPU 的任何干涉,只需要 CPU 在过程考试启动与过程结束时的处理,实际操作由 DMA 硬件直接执行完成,CPU 在此传送过程中可做别的事情。在 DMA 方式中 CPU 与外设可并行工作。

答案:(6) B

(7)、(8) 试题分析:HTTPS (Secure Hypertext Transfer Protocol) 安全超文本传输协议。它是一个安全通信通道,基于 HTTP 开发,用于在客户计算机和服务器之间交换信息。HTTPS 使用安全套接字层 (SSL) 进行信息交换,简单来说它是 HTTP 的安全版。

HTTPS 和 HTTP 的区别:

- 1) HTTPS 协议需要到 CA 申请证书,一般免费证书很少,需要交费。
- 2) HTTP 是超文本传输协议,信息是明文传输,HTTPS 则是具有安全性的 SSL 加密传输协议。
- 3) HTTP 和 HTTPS 使用的是完全不同的连接方式,用的端口也不一样,前者是 80,后者是 443。
- 4) HTTP 的连接很简单,是无状态的。

HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议,要比 HTTP 协议安全。

答案:(7) C (8) B

(9) 试题分析:显然 A、B、D 选项都属于主动攻击。

答案:(9) C

(10) 试题分析:漏洞扫描是指基于漏洞数据库,通过扫描等手段对指定的远程或者本地计算机系统的安全脆弱性进行检测,发现可利用漏洞的一种安全检测(渗透攻击)行为。漏洞扫描技术是一类重要的网络安全技术。它和防火墙、入侵检测系统互相配合,能够有效提高网络的安全性。

答案:(10) D

(11) 试题分析:防火墙最基本的功能就是控制在计算机网络中,不同信任程度区域间传送的数据流。防火墙对流经它的网络通信进行扫描,这样能够过滤掉一些攻击,以免其在目标计算机上被执行。

防火墙还可以关闭不使用的端口,隐蔽内部细节。所有的访问都经过防火墙,防火墙就能记录下这些访问并作出日志记录,同时也能提供网络使用情况的统计数据。

答案:(11) D

(12) 试题分析:按照公司规定完成并交由公司保管,属于职务作品,著作权归公司所有。

答案:(12) C

(13) 试题分析:根据《中华人民共和国商标法》第十条,下列标志不得作为商标使用:县级以上行政区划的地名或公众知晓的外国地名,不得作为商标。但是,地名具有其他含义或作为集体商标、证明商标组成部分的除外;已经注册的使用地名的商标继续有效。

答案：(13) B

(14) 试题分析：购买的软件光盘只有该光盘的使用权和所有权。

答案：(14) B

(15)、(16) 试题分析：按对于数据流图来说，患者属于外部实体；对于实体关系图来说，患者属于实体。D 选项的弱实体是指某些实体对于另一些实体有很强的依赖关系，即一个实体的存在必须以另一实体的存在为前提。如果题干改成“患者家属”，答案则是弱实体。

答案：(15) A (16) A

(17)、(18) 试题分析：从开始节点到结束节点，持续时间最长的路径是项目的工期，也就是完成项目最少需要的时间，图中 Start—B—C—E—F—Finish，持续时间最长为 18 天。

BC 与 BD 需要同一个开发人员，或者先完成 BC，或者先完成 BD，先完成 BC 需要 21 天，先完成 BD 需要 20 天。

答案：(17) B (18) C

(19) 试题分析：功能性需求即软件必须完成哪些事，必须实现哪些功能。B、C、D 选项属于性能需求。

答案：(19) A

(20) 试题分析：C 语言是处于汇编语言和高级语言之间的一种中间型程序设计语言，常被称为中级语言。它既有高级语言的基本特点，又具有汇编语言面向硬件和系统，可以直接访问硬件的功能。C 语言适用于开发操作系统。C++ 是 C 语言的继承，它既可以进行 C 语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。

答案：(20) A

(21) 试题分析：脚本语言又被称为扩建的语言，或者动态语言，是为了缩短传统的编写—编译—链接—运行 (edit-compile-link-run) 过程而创建的计算机编程语言，用来控制软件应用程序。脚本通常以文本 (如 ASCII) 保存，只在被调用时进行解释或编译。脚本语言与编程语言也有很多相似地方，其函数与编程语言比较相像一些，其也涉及变量。与编程语言之间最大的区别是编程语言的语法规则更为严格和复杂一些。脚本语言常用于描述格式化和链接，编程效率不如编译型语言。

答案：(21) A

(22) 试题分析：中间代码的作用是可使程序的结构在逻辑上更为简单明确，特别是可使目标代码的优化比较容易实现。中间代码有多种形式，常见的有逆波兰记号 (后缀式)、四元式和三元式，它们的共同特点是与具体的机器无关，不依赖于具体的计算机。

答案：(22) B

(23) 试题分析：计算机系统的层次结构为硬件层—操作系统层—语言处理程序层—应用程序层。

答案：(23) C

(24)、(25) 试题分析：进程控制块 PCB 的组织方式有以下三种。

1) 线性表方式：不论进程的状态如何，将所有的 PCB 连续地存放在内存的系统区。这种方式

适用于系统中进程数目不多的情况。

2) 索引表方式: 该方式是线性表方式的改进, 系统按照进程的状态分别建立就绪索引表、阻塞索引表等。

3) 链接表方式: 系统按照进程的状态将进程的 PCB 组成队列, 从而形成就绪队列、阻塞队列、运行队列等。

答案: (24) B (25) C

(26) 试题分析: 磁盘块的大小为 1K 字节, 每个块号占 3 字节, 一共有 341 个磁盘块存一级索引文件, 每个磁盘块还可以指向 341 个磁盘块, 因此一共有 $341 \times 341 = 116281$ 个磁盘块存二级索引文件, 每个磁盘块 1k 字节, 则该二级索引文件最大长度为 11628k 字节。

答案: (26) C

(27)、(28) 试题分析: 物理页大小=逻辑页大小等于 1k, 逻辑地址为 1024 的进程存放在逻辑页的第 1 页, 对应物理页号是 3。进程 A 的逻辑页 4 与进程 B 的逻辑页 5 要共享物理页 4, 则它们对应的物理页号都是 4。

答案: (27) B (28) A

(29)、(30) 试题分析: 先用代入法求出第 30 题, 然后判断第 29 题需要两个测试用例即可。

答案: (29) B (30) D

(31) 试题分析: 配置管理是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程, 确保软件开发者在软件生命周期中各个阶段都能得到精确的产品配置。

配置管理的活动包括: 制定配置管理计划、配置库管理、配置控制、配置状态报告、配置审计、发布管理和交付。

风险管理是与配置管理并列的项目管理过程。

答案: (31) B

(32) 试题分析: 极限编程是一种轻量级(敏捷)、高效、低风险、柔性、可预测的、科学的软件开发方式。

1) 四大价值观: 沟通、简单性、反馈和勇气。

2) 五个原则: 快速反馈、简单性假设、逐步修改、提倡更改和优质工作。

3) 十二个最佳实践: 计划游戏(快速制定计划、随着细节的不断变化而完善)、小型发布(系统的设计要能够尽可能早地交付)、隐喻(找到合适的比喻传达信息)、简单设计(只处理当前的需求, 使设计保持简单)、测试先行(先写测试代码, 然后再编写程序)、重构(重新审视需求和设计, 重新明确地描述它们以符合新的和现有的需求)、结对编程、集体代码所有制、持续集成(可以按日甚至按小时为客户提供可运行的版本)、每周工作 40 个小时、现场客户和编码标准。

答案: (32) D

(33) 试题分析: 管道或过滤器体系结构的优点:

1) 使得软件组件具有良好的隐蔽性和高内聚、低耦合的特点。

2) 允许设计者将整个系统的输入或输出行为看成是多个过滤器的行为的简单合成。
3) 支持软件重用。提供适合在两个过滤器之间传送的数据, 任何两个过滤器都可被连接起来。
4) 系统维护和增强系统性能简单。新的过滤器可以添加到现有系统中来; 旧的可以被改进的过滤器替换掉。

5) 允许对一些如吞吐量、死锁等属性的分析。

6) 支持并行执行。每个过滤器是作为一个单独的任务完成, 因此可与其他任务并行执行。

答案: (33) D

(34) 试题分析:

1) 数据耦合: 如果一个模块访问另一个模块时, 彼此之间是通过数据参数 (不是控制参数、公共数据结构或外部变量) 来交换输入、输出信息的, 则称这种耦合为数据耦合。

2) 印记耦合: 如果一组模块通过参数表传递记录信息, 就是标记耦合。

3) 控制耦合: 如果一个模块通过传送开关、标志、名字等控制信息, 明显地控制选择另一模块的功能, 就是控制耦合。

4) 内容耦合: 如果发生下列情形, 两个模块之间就发生了内容耦合。

一个模块直接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内部; 两个模块有一部分程序代码重叠 (只可能出现在汇编语言中); 一个模块有多个入口。

答案: (34) A

(35) 试题分析: 内聚是一个模块内部各成分之间相关联程度的度量。内聚按强度从低到高有以下几种类型:

1) 偶然内聚。如果一个模块的各成分之间毫无关系, 则称为偶然内聚, 也就是说模块完成一组任务, 这些任务之间的关系松散, 实际上没有什么联系。

2) 逻辑内聚。几个逻辑上相关的功能被放在同一模块中, 则称为逻辑内聚, 如一个模块读取各种不同类型外设的输入。尽管逻辑内聚比偶然内聚合理一些, 但逻辑内聚的模块各成分在功能上并无关系, 即使局部功能的修改有时也会影响全局, 因此这类模块的修改也比较困难。

3) 时间内聚。如果一个模块完成的功能必须在同一时间内执行 (如系统初始化), 但这些功能只是因为时间因素关联在一起, 则称为时间内聚。

4) 通信内聚。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。

5) 顺序内聚。如果一个模块的各个成分和同一个功能密切相关, 而且一个成分的输出作为另一个成分的输入, 则称为顺序内聚。

6) 过程内聚。构件或操作的组合方式是, 允许在调用前面的构件或操作之后, 马上调用后面的构件或操作, 即使两者之间没有数据进行传递。

模块完成多个需要按一定的步骤一次完成的功能 (过程相关—控制耦合)。例如, 在用程序流程图设计模块时, 若将程序流程图中的一部分划出各自组成模块, 便形成过程内聚。

7) 信息内聚。模块完成多个功能, 各个功能都在同一数据结构上操作, 每一项功能有一个唯

一的入口点。这个模块将根据不同的要求,确定该模块执行哪一个功能。由于这个模块的所有功能都是基于同一个数据结构(符号表),因此,它是一个信息内聚的模块。

8) 功能内聚。模块的所有成分对于完成单一的功能都是必须的,则称为功能内聚。

答案:(35) B

(36) 试题分析:软件的维护内容主要有以下几个方面:

1) 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) 适应性维护是指使应用软件适应信息技术变化和管理需求变化而进行的修改。

3) 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) 预防性维护是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。

答案:(36) C

(37)、(38) 试题分析:逻辑上相关的数据和行为绑定在一起,使信息对使用者隐蔽称为封装。对于私有成员来说,只能是该类中定义的方法才能对其进行访问。

答案:(37) C (38) B

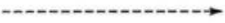
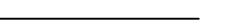
(39) 试题分析:在令子类继承父类的同时,可以重新定义某些属性,并重写某些方法,即覆盖父类的原有属性和方法,使其获得与父类不同的功能。另外,为子类追加新的属性和方法也是常见的做法。

答案:(39) D

(40) 试题分析:在面向对象分析过程中,选择名词短语来识别对象,动词短语识别对象的操作。

答案:(40) B

(41)、(42)、(43) 试题分析:在 UML 类图中,常见的有以下几种关系。

- 1) 依赖 (Dependency) 
- 2) 泛化 (Generalization) 
- 3) 关联 (Association) 
 - 组合 (Composition) 
 - 聚合 (Aggregation) 
- 4) 实现 (Realization) 

类图用于对项目的静态设计视图建模。这种视图主要支持系统的功能需求,即系统要提供给最终用户的服务。当对系统的静态设计视图建模时,通常以下述三种方式之一使用类图:系统的词汇、简单的协作、逻辑数据库模式。

D 选项的对象快照用于对象图中。

答案:(41) A (42) C (43) D

(44)~(47) 试题分析:状态型模式的特点是对对象内部的状态,允许其在不同的状态下,

拥有不同的行为，对状态单独封装成类。题干中根据投入货币的情况，售货机分别有交付产品不找零和交付产品找零以及不提供货物等行为判断该模式是属于状态模式。Vending Machine State 是客户程序使用的主要接口。

设计模式的类型如下所示：

创建型模式	用于创建对象	工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式，共 5 种
结构型模式	处理类或对象的组合	适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式，共 7 种
行为型模式	描述类与对象怎样交互、怎样分配职责	策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式，共 11 种

观察者模式（Observer Pattern）：观察者模式定义了对对象间的一种一对多依赖关系，使得每当一个对象改变状态，则所有依赖于它的对象都会得到通知并被自动更新。

访问者模式（Visitor Pattern）：表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用与这些元素的新操作。即对于某个对象或一组对象，不同的访问者，产生的结果不同，执行操作也不同。

备忘录模式（Memento Pattern）：在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样就可以将该对象恢复到原先保存的状态。

策略模式（Strategy Pattern）：策略模式定义了一系列的算法，并将每一个算法封装起来，而且使它们还可以相互替换。策略模式让算法独立于使用它的客户而独立变化。

命令模式（Command Pattern）：将一个请求封装成一个对象，从而使得用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。

答案：（44）B （45）A （46）D （47）C

（48）试题分析：编译程序的工作过程一般划分为五个阶段：词法分析、语法分析、语义分析、优化、目标代码生成。

- 1）词法分析：也就是从左到右一个一个的读入源程序，识别一个单词或符号，并进行归类。
- 2）语法分析：在词法分析的基础上，将单词序列分解成各类语法短语，如“程序”“语句”“表达式”等。
- 3）语义分析：审查源程序是否有语义的错误，当不符合语言规范的时候，程序就会报错。
- 4）代码优化：这个阶段是对前阶段的中间代码进行变换或改造，目的是使生成的目标代码更为高效，即节省时间和空间。
- 5）目标代码生成：也就是把优化后的中间代码变换成指令代码或汇编代码。

词法分析和语法分析本质上都是对源程序的结构进行分析。

答案：（48）C

第 26 小时

(49) 试题分析：将选项分别带入判断。

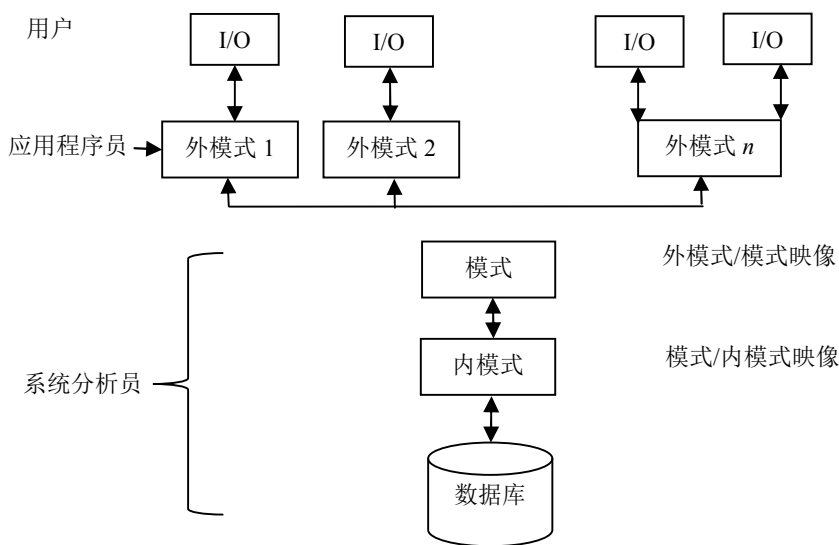
答案：(49) A

(50) 试题分析： $x=19$, $a=29$ 。因为 a 属于传引用，最后结果是 29。

传值与传引用的区别是：值传递时，实参被拷贝了一份，只能在函数体内使用。因此题干中的 $x=19$ ，只能作用于 f 函数中。

答案：(50) D

(51) 试题分析：数据库采用“三级模式、两级映像”的结构，如下图所示。



1) 概念模式也称模式，是数据库中全部数据的整体逻辑结构的描述。

2) 外模式也称用户模式或子模式，是用户与数据库系统的接口，是用户用到的那部分数据的描述。

3) 内模式也称存储模式，是数据库在物理存储方面的描述，定义所有内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。

4) 两级映像：①模式/内模式映像存在于概念级和内部级之间，用于定义概念模式和内模式之间的对应性；②外模式/模式映像存在于外部级和概念级之间，用于定义外模式和概念模式之间的对应性。

聚簇索引也叫簇类索引，是一种对磁盘上实际数据重新组织以按指定的一个或多个列的值排序，它改变的是数据库的内模式。

答案：(51) D

(52)、(53) 试题分析：关系型数据库中的一条记录中有若干个属性，若其中某一个属性组（注意是组）能唯一标识一条记录，该属性组就可以成为一个主键。选项 D（时间，学生）的组合中，

(时间, 学生) → 教室, (时间, 教室) → 培训科目, 培训科目 → 培训师, (学生, 培训科目) → 成绩, 可见, 包括了关系模式 R 的全部属性, 因此 (时间, 学生) 是该关系模式的主键。

数据库的关系范式有:

1) 第一范式 (1NF): 属性不可拆分或无重复的列。

2) 第二范式 (2NF): 完全函数依赖。在一个关系中, 若某个非主属性数据项依赖于全部关键字称为完全函数依赖。例如, 成绩表 (学号, 课程号, 成绩) 关系中, (学号, 课程号) → 成绩, 只有学号或只有课程号都无法推导出成绩, 因此该关系式属于完全函数依赖。

3) 第三范式 (3NF): 消除传递依赖, 不依赖于其他非主属性 (消除传递依赖)。满足第三范式的数据库必须先满足第二范式。也就是说, 数据库中的属性依赖仅能依赖于主属性, 不存在于其他非主属性的关联。例如, 图书和图书室的关系。图书包括编号、出版商、页码等信息, 图书室包括图书室编号、所存图书 (外键)。其中, 图书室的表中不应该存储任何图书的具体信息 (如出版商), 而只能通过主键图书编号来获得对应图书的信息。

4) BC 范式 (BCNF): 所有非主属性对每一个码都是完全函数依赖; 所有的主属性对于每一个不包含它的码, 也是完全函数依赖; 没有任何属性完全函数依赖于非码的任意一个组合。如果说关系模式 R 属于 3NF, 不一定属于 BCNF, 如果 R 属于 BCNF, 则一定属于 3NF。

5) 第四范式 (4NF): 对于每一个 $X \rightarrow Y$, X 都能找到一个候选码 (若关系中的某一属性组的值能唯一地表示一个元组, 而其真子集不行, 则称该属性组为候选码)。

题干中“(时间, 教室) → 培训科目”且“培训科目 → 培训师”, 这之间存在传递依赖, 不满足 3NF, 属于 2NF。

答案: (52) D (53) B

(54)、(55) 试题分析:

第 54 题 D 选项中, $E \rightarrow A$ 、 $A \rightarrow B$ 、 $B \rightarrow D$ 、 $CB \rightarrow E$ 包括了全部的属性。

第 55 题判断无损连接有多种方法, 下面介绍一种构造二维表进行判断的方法 (以正确的 D 选项为例)。

1) 构造一个初始的二维表, 若“属性”属于“模式”中的属性, 则填 a_j , 否则填 b_{ij} 。

属性 \ 模式	A	B	C	D	E
ABC	a_1	a_2	a_3	b_{14}	b_{15}
ED	b_{21}	b_{22}	b_{23}	a_4	a_5
ACE	a_1	b_{32}	a_3	b_{34}	a_5

2) 逐一考察关系式中的函数依赖, 将表中的 b_{ij} 修改成 a_j 。如 $CB \rightarrow E$, 则上表中第一行第六列的 b_{15} , 修改成 a_5 。修改结果如下表所示。

模式 \ 属性	A	B	C	D	E
ABC	a_1	a_2	a_3	a_4	a_5
ED	a_1	a_2	b_{23}	a_4	a_5
ACE	a_1	a_2	a_3	b_{34}	a_5

3) 修改后的表格若任一行存在 a_1 、 a_2 、 a_3 、 a_4 、 a_5 的情况, 该分解就属于无损连接, 如上表格中第一行存在 a_1 、 a_2 、 a_3 、 a_4 、 a_5 , 因此属于无损连接。

答案: (54) D (55) D

(56) 试题分析:

Java 数据库连接 (Java DataBase Connectivity, JDBC) 是一种用于执行 SQL 语句的 Java API, 可以为多种关系数据库提供统一访问, 它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准, 据此可以构建更高级的工具和接口, 使数据库开发人员能够编写数据库应用程序, 简单地说, JDBC 可做三件事: 与数据库建立连接、发送操作数据库的语句并处理结果。

XML 可扩展标记语言是一种元标记语言, 它可以对文档和数据进行结构化处理, 从而能够在部门、客户和供应商之间进行交换, 实现动态内容生成, 企业集成和应用开发。XML 简化了网络中数据交换和表示, 使得代码、数和表示分离, 并作为数据交换的标准格式, 因此它常被称为智能数据文档。

公共网关接口 (Common Gateway Interface, CGI) 是 WWW 技术中最重要的技术之一, 有着不可替代的重要地位。CGI 是外部应用程序 (CGI 程序) 与 WEB 服务器之间的接口标准, 是在 CGI 程序和 Web 服务器之间传递信息的过程。

COM 是组件对象模型 (Component Object Model) 的英文缩写。COM 是微软公司为了计算机工业的软件生产更加符合人类的行为方式开发的一种新的软件开发技术。在 COM 构架下, 人们可以开发出各种各样的功能专一的组件, 然后将它们按照需要组合起来, 构成复杂的应用系统。

答案: (56) A

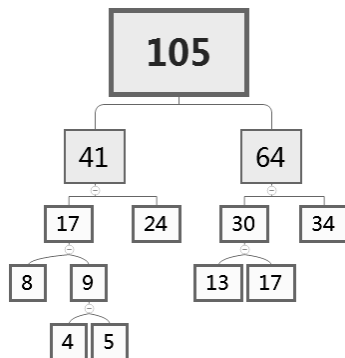
(57) 试题分析: 任取一个字符串 $abcd$, 它的非空子集有 a 、 b 、 c 、 d 、 ab 、 bc 、 cd 、 abc 、 bcd , 共 9 个。注意, abd 、 acd 、 bd 、 ad 等不属于它的子串。

答案: (57) D

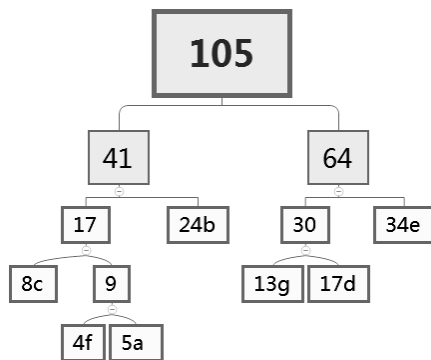
(58) 试题分析: 哈夫曼的构造过程如下:

1) 将给定的序列从小到大进行排列得到 $\{4, 5, 8, 13, 17, 24, 34\}$, 选取序列中最小的两个数 4 和 5 画出一棵树, 得到数字 9。

2) 将 4 和 5 的和 9 放入序列中, 继续重复上一步, 直至画出整个哈夫曼树, 如下图所示。

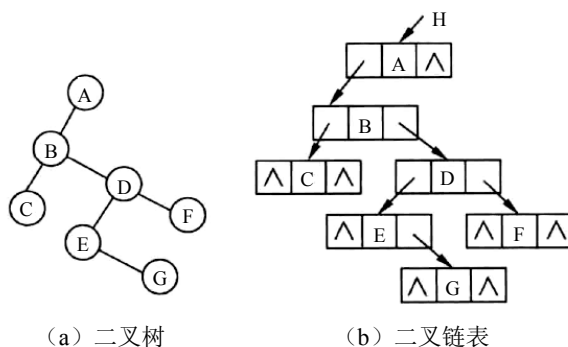


各个字符对应的节点为：



答案：(58) A

(59) 试题分析：所谓二叉树的链式存储结构是指用链表来表示一棵二叉树，即用链来指示着元素的逻辑关系，如下图所示。

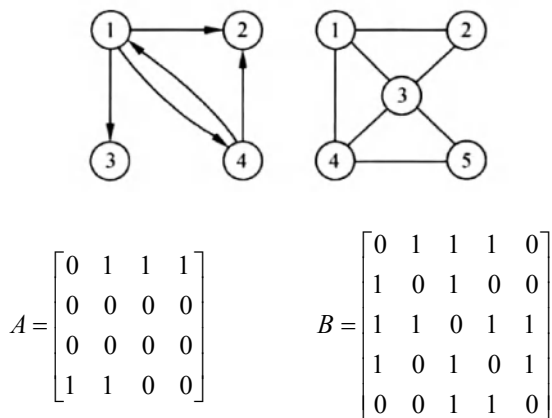


二叉链表中每个节点有 2 个指针，一共有 $2k$ 个指针。在二叉树中除了根节点之外，其他的节

点都有一条边进入该节点,即一个指针指向该节点,所以二叉树中边的总个数为 $k-1$,也就说明非空指针的个数为 $k-1$ 个。那么,空指针的个数为:总的节点数-非空指针的个数 $=2k-(k-1)=k+1$ 。

答案:(59) C

(60) 试题分析:有向连通图与无向连通图如下图所示。



答案:(60) A

(61) 试题分析:两个递增序列 A、B 进行归并时,从序列的第一个元素开始,分别从这两个序列中取一个元素并进行比较,将较小者输出,然后从较小者所在序列取下一个元素再进行比较,循环往复,直到某个序列的全部元素已经输出,再将另一个序列的剩余元素依次输出即可。

若 $a_m < b_1$,则需要依次比较 a_1 与 b_1 、 a_2 与 b_1 、 a_3 与 b_1 、 a_{m-1} 与 b_1 、 a_m 与 b_1 ,共需要比较 m 次,这是归并时比较次数最少的情况。

答案:(61) A

(62)、(63) 试题分析:第 62 题是一个给定的序列的子序列,就是将给定序列中零个或多个元素去掉之后得到的结果。序列 ABCBDAB 去掉划线的字符 ABCBDAB 得到的 BCBA 就是它的一个子序列。一个序列的子序列有 2^n 个,将 X、Y 的所有子序列都检查过后,即可求出 X、Y 的最长公共子序列。这种方式的时间复杂度为 $O(n2^n)$ 。

第 63 题引进一个二维数组 $c[i][j]$ 表示 X 的 i 位和 Y 的 j 位之前的最长公共子序列的长度,按照公式将数值填入下表。

j		0	1	2	3	4	5	6
		y_j	B	D	C	A	B	A
0	x_i	0	0	0	0	0	0	0
1	A	0	0	0	0	1	1	1
2	B	0	1	1	1	1	2	2
3	C	0	1	1	2	2	2	2

续表

$\begin{matrix} & j \\ i & \end{matrix}$		0	1	2	3	4	5	6
		y_j	B	D	C	A	B	A
4	B	0	1	1	2	2	3	3
5	D	0	1	2	2	2	3	3
6	A	0	1	2	2	3	3	4
7	B	0	1	2	2	3	4	4

此时， $c[i][j]$ 中最大的数就是 X 和 Y 的最长公共子序列的长度，等于 4。该算法的时间复杂度为 $O(n^2)$ 。

答案：(62) D (63) A

(64)、(65) 试题分析：对于基本有序数组采用插入排序效率是最高的，时间复杂度为 $O(n)$ ，快速排序适用于无序数组，对于有序数组来说时间复杂度是 $O(n^2)$ ，属于最坏的情况。

答案：(64) A (65) A

(66) 试题分析：UDP 与 TCP 的主要区别在于 UDP 不一定提供可靠的数据传输。事实上，该协议不能保证数据准确无误地到达目的地。UDP 在许多方面非常有效，在数据传输过程中延迟小、数据传输效率高。当某个程序的目标是尽快地传输尽可能多的信息时，可以使用 UDP。

答案：(66) B

(67) 试题分析：25 端口为 SMTP (Simple Mail Transfer Protocol，简单邮件传输协议) 服务器所开放，主要用于发送邮件。

110 端口是为 POP3 (邮件协议 3) 服务开放的，POP2、POP3 都是主要用于接收邮件的，目前 POP3 使用的比较多，许多服务器都同时支持 POP2 和 POP3。客户端可以使用 POP3 协议来访问服务端的邮件服务。

答案：(67) A

(68) 试题分析：简单网络管理协议 (SNMP) 由一组网络管理的标准组成，包含一个应用层协议 (Application Layer Protocol)、数据库模型 (Database Schema) 和一组资源对象。该协议能够支持网络管理系统，用以监测连接到网络上的设备是否有任何引起管理上关注的情况。SNMP 使用 UDP (用户数据报协议) 作为第四层协议 (传输协议)，进行无连接操作。

答案：(68) D

(69) 试题分析：127.0.0.1 是回送地址，指本地机，一般用来测试使用。回送地址 (127.x.x.x) 是本地回送地址 (Loopback Address)，即主机 IP 堆栈内部的 IP 地址，主要用于网络软件测试及本地机进程间通信，无论什么程序，一旦使用回送地址发送数据，协议软件立即返回，不进行任何网络传输。

答案：(69) B

(70) 试题分析：Netstat 是控制台命令，是一个监控 TCP/IP 网络的非常有用的工具，它可以

第 26 小时

显示路由表、实际的网络连接及每一个网络接口设备的状态信息。Netstat 用于显示与 IP、TCP、UDP 和 ICMP 协议相关的统计数据，一般用于检验本机各端口的网络连接情况。

Ping 是一个通信协议，是 TCP/IP 协议的一部分。利用 ping 命令可以检查网络是否连通，可以很好地帮助分析和判定网络故障。

msconfig 即系统配置实用程序，是 Microsoft System Configuration 的缩写。在开始菜单里运行中输入然后确认就可以找到程序开启或禁用，可以帮助电脑禁止不需要运行的程序，加快电脑运行。

cmd 命令提示符是在操作系统中，提示进行命令输入的一种工作提示符。在不同的操作系统环境下，命令提示符各不相同。

答案：(70) B

(71) ~ (75) 试题分析：

翻译：

语义网目前正在稳步发展中，每次在已经完成的基础之上又增加一些内容。客观来说这种方式会成功是因为实现一个小的 (71) 比较容易，而迈出一大步实现所有功能则比较困难。研究开发者们通常会按照不同方向进行开发工作，这种 (72) 的思想是促进科学进步的重要驱动力。然而就软件工程的观点来说，应该把开发过程标准化，若是大家就某些观点能达成共识，而其他的问题有不同的意见，那么就很有必要落实这些已经达成共识的部分。这么做的话，就算是那些野心勃勃的开发者在开发的道路上遭遇了滑铁卢，整个行业还是会产生 (73) 积极成果。这个 (74) 一旦被建立，就会有更多的开发者采用它而不是观望哪种研究路线会取得最终的胜利。语义网的本质是企业 and 单个用户必须构建工具，添加内容并使用。我们已经迫不及待地看到语义网真正实现那一天——也许 (75) 实现它的内容需要再过十年时间（当然这是按照今天所设想的那样）。

- | | | | |
|------------|---------|-------|-------|
| (71) A. 冲突 | B. 一致同意 | C. 成功 | D. 异议 |
| (72) A. 竞争 | B. 同意 | C. 合作 | D. 协作 |
| (73) A. 全部 | B. 完整 | C. 部分 | D. 整体 |
| (74) A. 技术 | B. 标准 | C. 方法 | D. 模型 |
| (75) A. 局部 | B. 目的 | C. 目标 | D. 程度 |

答案：(71) C (72) A (73) C (74) A (75) D

下午案例分析

试题一（共 15 分）

【问题 1】（3 分）

答案：E1：用户；E2：商家；E3：单车。

【问题 2】（5 分）

答案：

- D1: 存储用户信息 (用户信息表)。
 D2: 记录单车位置及状态信息 (单车信息表)。
 D3: 记录单车行程信息 (行程信息表)。
 D4: 存储计费规则 (计费规则信息表)。
 D5: 记录单车故障信息 (单车故障信息表)。

【问题 3】(5 分)

答案:

- 起点: P3; 终点: E1; 数据流名称: 开锁密码。
 起点: P3; 终点: E1; 数据流名称: 锁车结账。
 起点: P3; 终点: D2; 数据流名称: 单车状态。
 起点: D4; 终点: P3; 数据流名称: 计费规则。
 起点: D3; 终点: P7; 数据流名称: 行程查询统计。

【问题 4】(2 分)

答案: 扫码或手动开锁, 骑行单车, 锁车结账。

试题二 (共 15 分)

【问题 1】(5 分)

答案:

- 联系 1: 部门和员工, $1:n$ 。
 联系 2: 业务员和用户申请, $1:n$ 。
 联系 3: 用户和用户申请, $1:n$ 。
 联系 4: 策划员和策划任务, $m:n$ 。
 联系 5: 策划任务和用户申请, $1:1$ 。

【问题 2】(4 分)

答案:

- (1) 职位, 部门号
- (2) 用户名, 银行账号
- (3) 预算费用, 业务员或员工号
- (4) 要求完成时间

【问题 3】(4 分)

答案:

- 用户申请: 主键: 申请号; 外键: 用户号, 业务员。
 策划任务: 主键: 申请号; 外键: 申请号。

【问题 4】(2 分)

答案: 不正确。All-key 关系模型的所有属性组组成该关系模式的候选码, 称为全码。即所有

属性当作一个码。若关系中只有一个候选码,而且这个候选码中包含全部属性,则该候选码为全码。

策划任务中的任务明细和要求完成时间不包括全部属性,因此不是全码。

试题三 (15 分)

【问题 1】(9 分)

答案:

C1: System

C2: User

C3: Student

C4: Faculty

C5: Staff

C6: Publication

C7: ConfPaper

C8: JournalArticle

C9: TechReport

【问题 2】(4 分)

答案:

C6: 题目, 作者, 出版年份

C7: 会议名称, 召开时间, 召开地点

C8: 期刊名称, 出版月份, 期号, 主办单位

C9: ID

【问题 3】(2 分)

答案:

观察者模式。实现: 引用他人学术出版物发送电子邮件通知该用户。

策略模式。实现: 查询作者出版的不同类型的出版物——会议文章、期刊文章、校内技术报告。

试题四 (15 分)

【问题 1】(10 分)

答案:

(1) visited[0]=1

(2) visited[x[k]]==0

(3) k==n-1&&v[x[k]][x[0]]==1

(4) visited[x[k]]=1

(5) k=k-1

【问题 2】(5 分)

答案：回溯法、深度优先。

试题五（共 15 分）

答案：

- (1) virtual void doPaint(Matrix m)=0
- (2) imp->doPaint(m)
- (3) new GIFImage()
- (4) new LinuxImp()
- (5) image->setImp(imageImp)

试题六（共 15 分）

答案：

- (1) abstract void doPaint(Matrix m)
- (2) imp.doPaint(m)
- (3) new GIFImage()
- (4) new LinuxImp()
- (5) image.setImp(imageImp)

第27小时

2018年5月考试真题（上午基础知识）



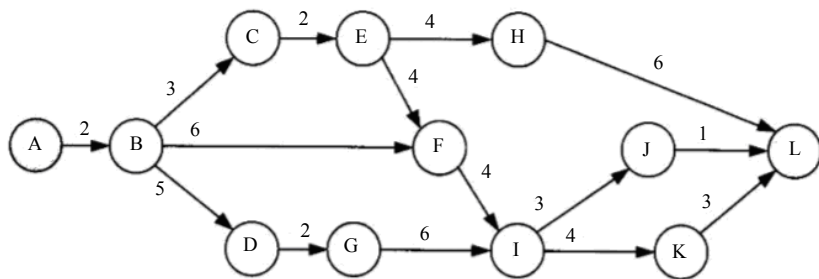
【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

- 对有 n 个结点、 e 条边且采用数组表示法（即邻接矩阵存储）的无向图进行深度优先遍历，时间复杂度为 (1)。
(1) A. $O(n^2)$ B. $O(e^2)$ C. $O(n+e)$ D. $O(n \times e)$
- 浮点数的表示分为阶和尾数两部分。两个浮点数相加时，需要先对阶，即 (2)（ n 为阶差的绝对值）。
(2) A. 将大阶向小阶对齐，同时将尾数左移 n 位
 B. 将大阶向小阶对齐，同时将尾数右移 n 位
 C. 将小阶向大阶对齐，同时将尾数左移 n 位
 D. 将小阶向大阶对齐，同时将尾数右移 n 位
- 计算机运行过程中，遇到突发事件，要求 CPU 暂时停止正在运行的程序，转去为突发事件服务，服务完毕，再自动返回原程序继续执行，这个过程称为 (3)，其处理过程中保存现场的目的是 (4)。
(3) A. 阻塞 B. 中断 C. 动态绑定 D. 静态绑定
(4) A. 防止丢失数据 B. 防止对其他部件造成影响

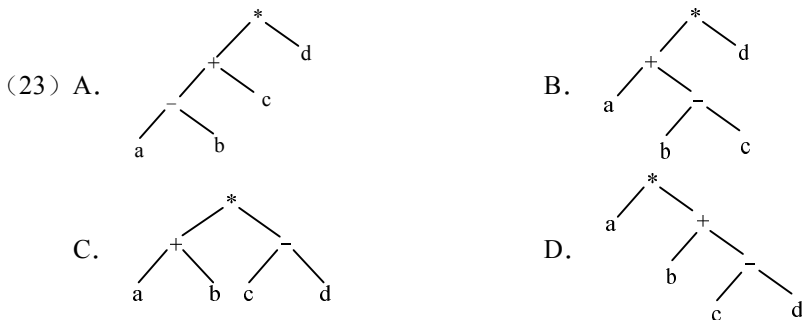
- C. 返回去继续执行原程序 D. 为中断处理程序提供数据
- 海明码是一种纠错码，其方法是为需要校验的数据位增加若干校验位，使得校验位的值决定于某些被校位的数据，当被校数据出错时，可根据校验位的值的变化找到出错位，从而纠正错误。对于 32 位的数据，至少需要加 (5) 个校验位才能构成海明码。
以 10 位数据为例，其海明码表示为 D9D8D7D6D5D4P4D3D2D1P3D0P2P1 中，其中 D_i ($0 \leq i \leq 9$) 表示数据位， P_j ($1 \leq j \leq 4$) 表示校验位，数据位 D9 由 P4、P3 和 P2 进行校验（从右至左 D9 的位序为 14，即等于 $8+4+2$ ，因此用第 8 位的 P4、第 4 位的 P3 和第 2 位的 P2 校验），数据位 D5 由 (6) 进行校验
(5) A. 3 B. 4 C. 5 D. 6
(6) A. P4P1 B. P4P2 C. P4P3P1 D. P3P2P1
 - 流水线的吞吐率是指单位时间流水线处理的任务数，如果各段流水的操作时间不同，则流水线的吞吐率是 (7) 的倒数。
(7) A. 最短流水段操作时间 B. 各段流水的操作时间总和
C. 最长流水段操作时间 D. 流水段数乘以最长流水段操作时间
 - 网络管理员通过命令行方式对路由器进行管理，要确保 ID，口令和会话话内存的保密性，应采取的访问方式是 (8)。
(8) A. 控制台 B. AUX C. TELNET D. SSH
 - 在安全通信中，S 将所发送的信息使用 (9) 进行数字签名，T 收到该消息后可利用 (10) 验证该消息的真实性。
(9) A. S 的公钥 B. S 的私钥 C. T 的公钥 D. T 的私钥
(10) A. S 的公钥 B. S 的私钥 C. T 的公钥 D. T 的私钥
 - 在网络安全管理中,加强内防内控可采取的策略有 (11)。
①控制终端接入数量。
②终端访问授权，防止合法终端越权访问。
③加强终端的安全检查与策略管理。
④加强员工上网行为管理与违规审计。
(11) A. ②③ B. ②④ C. ①②③④ D. ②③④
 - 攻击者通过发送一个目的主机已经接收过的报文来达到攻击目的，这种攻击方式属于 (12) 攻击。
(12) A. 重放 B. 拒绝服务 C. 数据截获 D. 数据流分析
 - 以下有关计算机软件著作权的叙述中，正确的是 (13)。
(13) A. 非法进行拷贝、发布或更改软件的人被称为软件盗版者
B. 《计算机软件保护条例》是国家知识产权局颁布的，用来保护软件著作权人的权益
C. 软件著作权属于软件开发者，软件著作权自软件开发完成之日起产生
D. 用户购买了具有版权的软件，则具有对该软件的使用权和复制权

- 王某是某公司的软件设计师, 完成某项软件开发后按公司规定进行软件归档。以下有关该软件的著作权的叙述中, 正确的是 (14)。
(14) A. 著作权应由公司和王某共同享有
B. 著作权应由公司享有
C. 著作权应由王某享有
D. 除署名权以外, 著作权的其它权利由王某享有
- 著作权中, (15) 的保护期不受限制。
(15) A. 发表权 B. 发行权 C. 署名权 D. 展览权
- 数据字典是结构化分析的一个重要输出。数据字典的条目不包括 (16)。
(16) A. 外部实体 B. 数据流 C. 数据项 D. 基本加工
- 某商店业务处理系统中, 基本加工“检查订货单”的描述为: 若订货单金额大于 5000 元, 而且欠款时间超过 60 天, 则不予批准; 若订货单金额大于 5000 元, 而且欠款时间不超过 60 天, 则发出批准书和发货单; 若订货单金额小于或等于 5000 元, 则发出批准书和发货单, 若欠款时间超过 60 天, 则还要发催款通知书。现采用决策表表示该基本加工, 则条件取值的组合数最少是 (17)。
(17) A. 2 B. 3 C. 4 D. 5
- 某软件项目的活动图如下图所示, 其中顶点表示项目里程碑, 连接顶点的边表示包含的活动, 边上的数字表示活动的持续天数, 则完成该项目的最少时间为 (18) 天。活动 EH 和 II 的松弛时间分别为 (19) 天。

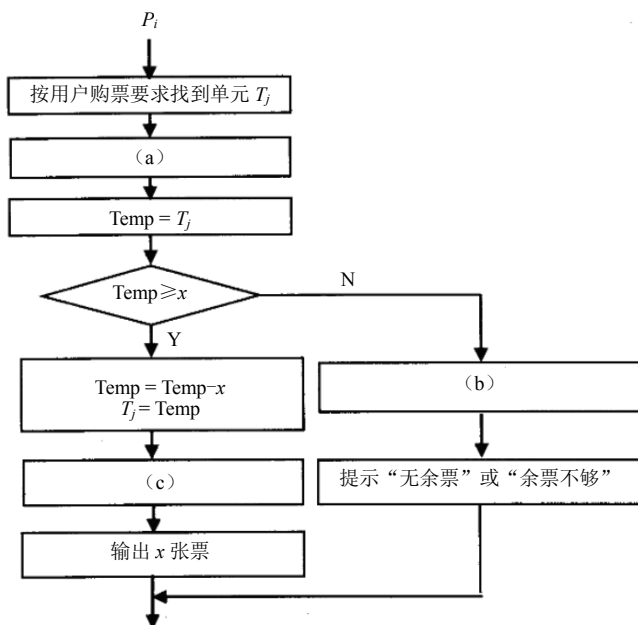


- (18) A. 17 B. 19 C. 20 D. 22
- (19) A. 3 和 3 B. 3 和 6 C. 5 和 3 D. 5 和 6
- 工作量估算模型 COCOMO II 的层次结构中, 估算选择不包括 (20)。
(20) A. 对象点 B. 功能点 C. 用例数 D. 源代码行
- (21) 是一种函数式编程语言。
(21) A. Lisp B. Prolog C. Python D. Java/C++
- 将高级语言源程序翻译为可在计算机上执行的形式有多种不同的方式, 其中 (22)。
(22) A. 编译方式和解释方式都生成逻辑上与源程序等价的目标程序

- B. 编译方式和解释方式都不生成逻辑上与源程序等价的目标程序
 C. 编译方式生成逻辑上与源程序等价的目标程序, 解释方式不生成
 D. 解释方式生成逻辑上与源程序等价的目标程序, 编译方式不生成
- 对于后缀表达式 $a b c - + d *$ (其中, $-$ 、 $+$ 、 $*$ 表示二元算术运算减、加、乘), 与该后缀式等价的语法树为 (23)。



- 假设铁路自动售票系统有 n 个售票终端, 该系统为每个售票终端创建一个进程 P_i ($i=1,2,\dots,n$) 管理车票销售过程。假设 T_j ($j=1,2,\dots,m$) 单元存放某日某趟车的车票剩余票数, Temp 为 P_i 进程的临时工作单元, x 为某用户的购票张数。 P_i 进程的工作流程如下图所示, 用 P 操作和 V 操作实现进程间的同步与互斥。初始化时系统应将信号量 S 赋值为 (24)。图中 (a)、(b) 和 (c) 处应分别填入 (25)。



(24) A. $n-1$ B. 0 C. 1 D. 2

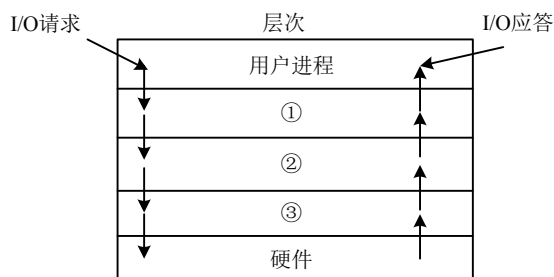
(25) A. V(S)、P(S)和 P(S) B. P(S)、P(S)和 V(S)

C. V(S)、V(S)和 P(S) D. P(S)、V(S)和 V(S)

- 若系统在将 (26) 文件修改的结果写回磁盘时发生崩溃, 则对系统的影响相对较大。

(26) A. 目录 B. 空闲块 C. 用户程序 D. 用户数据

- I/O 设备管理软件一般分为四个层次, 如下图所示。图中①②③分别对应 (27)。



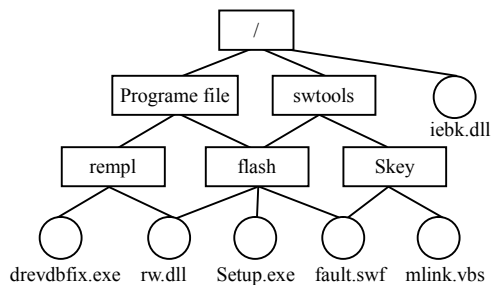
(27) A. 设备驱动程序、虚设备管理、与设备无关的系统软件

B. 设备驱动程序、与设备无关的系统软件、虚设备管理

C. 与设备无关的系统软件、中断处理程序、设备驱动程序

D. 与设备无关的系统软件、设备驱动程序、中断处理程序

- 若某文件系统的目录结构如下图所示, 假设用户要访问文件 `rw.dll`, 而且当前工作目录为 `swtools`, 则该文件的全文名为 (28), 相对路径和绝对路径分别为 (29)。



(28) A. `rw.dll`

B. `flash/rw.dll`

C. `/swtools/flash/rwd11`

D. `/Programe file/Skey/rw.dll`

(29) A. `/swtools/flash/和/flash/`

B. `flash/和/swtools/flash/`

C. `/swtools/flash/和 flash/`

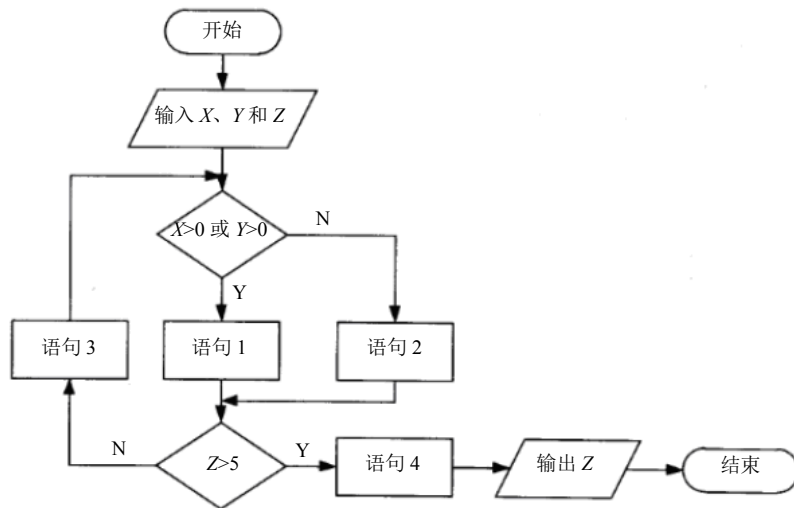
D. `/flash/和 swtools/flash/`

- 以下关于增量模型的叙述中, 不正确的是 (30)。

(30) A. 容易理解, 管理成本低

B. 核心的产品往往首先开发, 因此经历最充分的“测试”

- 软件调试的任务就是根据测试时所发现的错误，找出原因和具体的位置进行改正。其常用的



- 软件调试的任务就是根据测试时所发现的错误，找出原因和具体的位置进行改正。其常用的

方法中, (37) 是指从测试所暴露的问题出发, 收集所有正确或不正确的数据, 分析它们之间的关系, 提出假想的错误原因, 用这些数据来证明或反驳, 从而查出错误所在。

(37) A. 试探法 B. 回溯法 C. 归纳法 D. 演绎法

- 对象的 (38) 标识了该对象的所有属性 (通常是静态的) 及每个属性的当前值 (通常是动态的)。

(38) A. 状态 B. 唯一 ID C. 行为 D. 语义

- 在下列机制中, (39) 是指过程调用和响应调用所需执行的代码在运行时加以结合; 而 (40) 是过程调用和响应调用所需执行的代码在编译时加以结合。

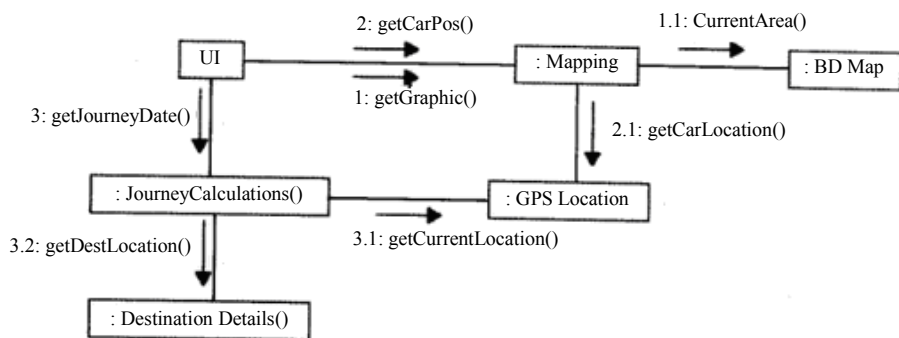
(39) A. 消息传递 B. 类型检查 C. 静态绑定 D. 动态绑定

(40) A. 消息传递 B. 类型检查 C. 静态绑定 D. 动态绑定

- 同一消息可以调用多种不同类的对象的方法, 这些类有某个相同的超类, 这种现象是 (41)。

(41) A. 类型转换 B. 映射 C. 单态 D. 多态

- 如下所示的图为 UML 的 (42), 用于展示某汽车导航系统中 (43)。Mapping 对象获取汽车当前位置 (GPS Location) 的消息为 (44)。

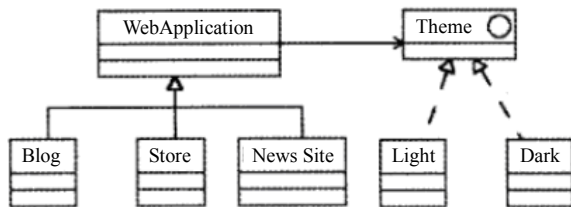


(42) A. 类图 B. 组件图 C. 通信图 D. 部署图

(43) A. 对象之间的消息流及其顺序 B. 完成任务所进行的活动流
C. 对象的状态转换及其事件顺序 D. 对象之间消息的时间顺序

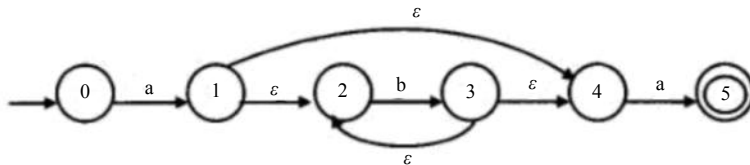
(44) A. 1: getGraphic() B. 2: getCarPos()
C. 1.1: CurrentArea() D. 2.1: getCarLocation()

- 假设现在要创建一个 Web 应用框架, 基于此框架能够创建不同的具体 Web 应用, 如博客、新闻网站和网上商店等; 并可以为每个 Web 应用创建不同的主题样式, 如浅色或深色等。这一业务需求的类图设计适合采用 (45) 模式 (如下图所示)。其中 (46) 是客户程序使用的主要接口, 维护对主题类型的引用。此模式为 (47), 体现的最主要的意图是 (48)。



- (45) A. 观察者 (Observer) B. 访问者 (Visitor)
 C. 策略 (Strategy) D. 桥接 (Bridge)
- (46) A. Webapplication B. Blog
 C. Theme D. Light
- (47) A. 创建型对象模式 B. 结构型对象模式
 C. 行为型类模式 D. 行为型对象模式
- (48) A. 将抽象部分与其实现部分分离,使它们都可以独立地变化
 B. 动态地给一个对象添加一些额外的职责
 C. 为其他对象提供一种代理以控制对这个对象的访问
 D. 将一个类的接口转换成客户希望的另外一个接口

- 下图所示为一个不确定有限自动机 (NFA) 的状态转换图。该 NFA 识别的字符串集合可用正规式 (49) 描述。



- (49) A. ab^*a B. $(ab)^*a$ C. a^*ba D. $a(ba)^*$

- 简单算术表达式的结构可以用下面的上下文无关文法进行描述 (E 为开始符号), (50) 是符合该文法的句子。

$E \rightarrow T | E + T$

$T \rightarrow F | T * F$

$F \rightarrow -F | N$

$N \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

- (50) A. $2-3*4$ B. $2+-3*4$ C. $(2+3)*4$ D. $2*4-3$

- 语法制导翻译是一种 (51) 方法。

- (51) A. 动态语义分析 B. 中间代码优化
 C. 静态语义分析 D. 目标代码优化

- 给定关系模式 $R<U,F>$, 其中 U 为属性集, F 是 U 上的一组函数依赖, 那么 Armstrong 公理系统的伪传递律是指 (52)。

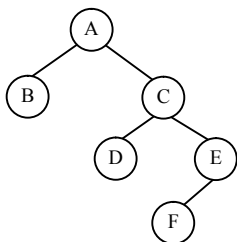
- 第27小时

D. 入队列操作不需要遍历链表而出队列操作需要

- 设有 n 阶三对角矩阵 A , 即非零元素都位于主对角线以及与主对角线平行且紧邻的两条对角线上, 现对该矩阵进行按行压缩存储, 若其压缩空间用数组 B 表示, A 的元素下标从 0 开始, B 的元素下标从 1 开始。已知 $A[0,0]$ 存储在 $B[1]$, $A[n-1,n-1]$ 存储在 $B[3n-2]$, 那么非零元素 $A[i,j]$ ($0 \leq i < n$, $0 \leq j < n$, $|i-j| \leq 1$) 存储在 $B[\underline{(59)}]$ 。

(59) A. $2i+j-1$ B. $2i+j$ C. $2i+j+1$ D. $3i-j+1$

- 对下面的二叉树进行顺序存储 (用数组 MEM 表示), 已知结点 A、B、C 在 MEM 中对应元素的下标分别为 1、2、3, 那么结点 D、E、F 对应的数组元素下标为 (60)。



(60) A. 4、5、6 B. 4、7、10 C. 6、7、8 D. 6、7、14

- 用哈希表存储元素时, 需要进行冲突 (碰撞) 处理, 冲突是指 (61)。

(61) A. 关键字被依次映射到地址编号连续的存储位置

B. 关键字不同的元素被映射到相同的存储位置

C. 关键字相同的元素被映射到不同的存储位置

D. 关键字被映射到哈希表之外的位置

- 现需要申请一些场地举办一批活动, 每个活动有开始时间和结束时间。在同一个场地, 如果一个活动结束之前, 另一个活动开始, 即两个活动冲突。若活动 A 从 1 时间开始, 5 时间结束, 活动 B 从 5 时间开始, 8 时间结束, 则活动 A 和 B 不冲突。现要计算 n 个活动需要的最少场地数。

求解该问题的基本思路如下 (假设需要场地数为 m , 活动数为 n , 场地集合为 $P_1, P_2, \dots, P_m$), 初始条件 P_i 均无活动安排:

(1) 采用快速排序算法对 n 个活动的开始时间从小到大排序, 得到活动 $a_1, a_2, \dots, a_n$ 。对每个活动 a_i , i 从 1 到 n , 重复步骤 (2)、(3) 和 (4);

(2) 从 P_1 开始, 判断 a_i 与 P_1 的最后一个活动是否冲突, 若冲突, 考虑下一个场地 $P_2, \dots$

(3) 一旦发现 a_i 与某个 P_j 的最后一个活动不冲突, 则将 a_i 安排到 P_j , 考虑下一个活动;

(4) 若 a_i 与所有已安排活动的 P_j 的最后一个活动均冲突, 则将 a_i 安排到一个新的场地, 考虑下一个活动;

(5) 将 n 减去没有安排活动的场地数, 即可得到所用的最少场地数。首先采用了快速排序算法进行排序, 其算法设计策略是 (62); 后面步骤采用的算法设计策略是 (63)。整个

算法的时间复杂度是 (64)。下表给出了 $n=11$ 的活动集合, 根据上述算法, 得到最少的场地数为 (65)。

i	1	2	3	4	5	6	7	8	9	10	11
开始时间 s_i	0	1	2	3	3	5	5	6	8	8	12
结束时间 f_i	6	4	13	5	8	7	9	10	11	12	14

- (62) A. 分治 B. 动态规划 C. 贪心 D. 回溯
 (63) A. 分治 B. 动态规划 C. 贪心 D. 回溯
 (64) A. $\Theta(\lg n)$ B. $\Theta(n)$ C. $\Theta(n \lg n)$ D. $\Theta(n^2)$
 (65) A. 4 B. 5 C. 6 D. 7

- 下列网络互连设备中, 属于物理层的是 (66)。

- (66) A. 交换机 B. 中继器 C. 路由器 D. 网桥

- 在地址 <http://www.dailynews.com.cn/channel/welcome.htm> 中, www.dailynews.com.cn 表示 (67), [welcome.htm](#) 表示 (68)。

- (67) A. 协议类型 B. 主机
 C. 网页文件名 D. 路径
 (68) A. 协议类型 B. 主机域名
 C. 网页文件名 D. 路径

- 在 Linux 中, 要更改一个文件的权限设置可以使用 (69) 命令。

- (69) A. attrib B. modify
 C. chmod D. change

- 主域名服务器在接收到域名请求后, 首先查询的是 (70)。

- (70) A. 本地 hosts 文件 B. 转发域名服务器
 C. 本地缓存 D. 授权域名服务器

- Creating a clear map of where the project is going is an important first step. It lets you identify risks, clarify objectives, and determine if the project even makes sense. The only thing more important than the release plan is not to take it too seriously. Please planning is creating a game plan for your Web project (71) what you think you want your Web site to be. The plan is a guide for the content, design elements, and functionality of a Web site to be released to the public, to partners, or internally. It also (72) how long the project will take and how much it will cost. What the plan is not is a functional (73) that defines the project in detail or that produces a budget you can take to the bank. Basically you use a release Plan to do an initial sanity check of the project's (74) and worthiness. Release Plans are useful road maps, but don't think of them as guides to the interstate road system. Instead, think of them as the (75) used by early explorers—half humor and guess and half hope and expectation. It's always a good idea to have a map of where a project is headed.

- | | | | |
|-----------------------|------------------|-----------------|-------------------|
| (71) A. constructing | B. designing | C. implementing | D. outlining |
| (72) A. defines | B. calculates | C. estimates | D. knows |
| (73) A. specification | B. structure | C. requirement | D. implementation |
| (74) A. correctness | B. modifiability | C. feasibility | D. traceability |
| (75) A. navigators | B. maps | C. guidance | D. goals |

第28小时

2018 年 5 月考试真题（下午案例分析）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

试题一（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某医疗护理机构为老年人或有护理需求的人提供专业护理，现欲开发一个基于 Web 的医疗管理系统，以改善医疗护理效率，该系统的主要功能如下：

- （1）通用信息查询。客户提交通用信息查询请求，查询通用信息表，返回查询结果。
- （2）医生聘用。医生提出应聘或辞职申请，交由主管进行聘用或解聘审批，更新医生表，并给医生反馈聘用或解聘结果；删除解聘医生的出诊安排。
- （3）预约处理。医生安排出诊时间，存入医生出诊时间表，根据客户提交的预约查询请求，查询在职医生及其出诊时间等预约所需数据并返回；创建预约，提交预约请求，在预约表中新增预约记录，更新所约医生的出诊时间，并给医生发送预约通知；给客户反馈预约结果。
- （4）药品管理。医生提交处方，根据药品名称从药品数据中查询相关药品库存信息，开出药品，更新对应药品的库存以及预约表中的治疗信息；给医生发送“药品已开出”反馈。

(5) 创建报表。根据主管提交的报表查询请求(报表类型和时间段),从预约数据、通用信息、药品库存数据、医生和医生出诊时间中进行查询,生成报表返回给主管。

现在采用结构化方法对医疗管理系统进行分析与设计,获得如图 1-1 所示的上下文数据流图和如图 1-2 所示的 0 层数据流图。

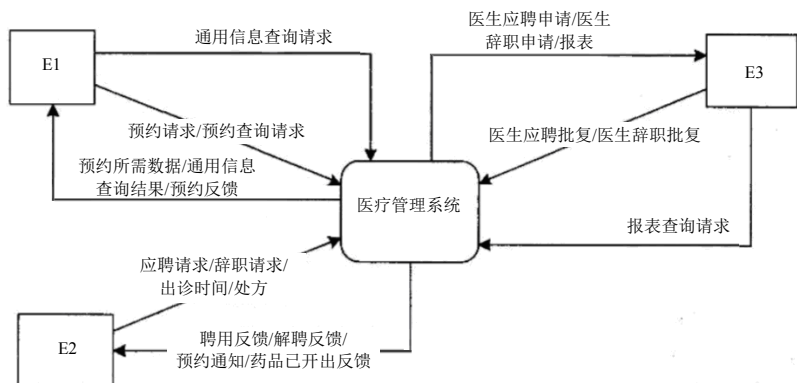


图 1-1 上下文数据流图

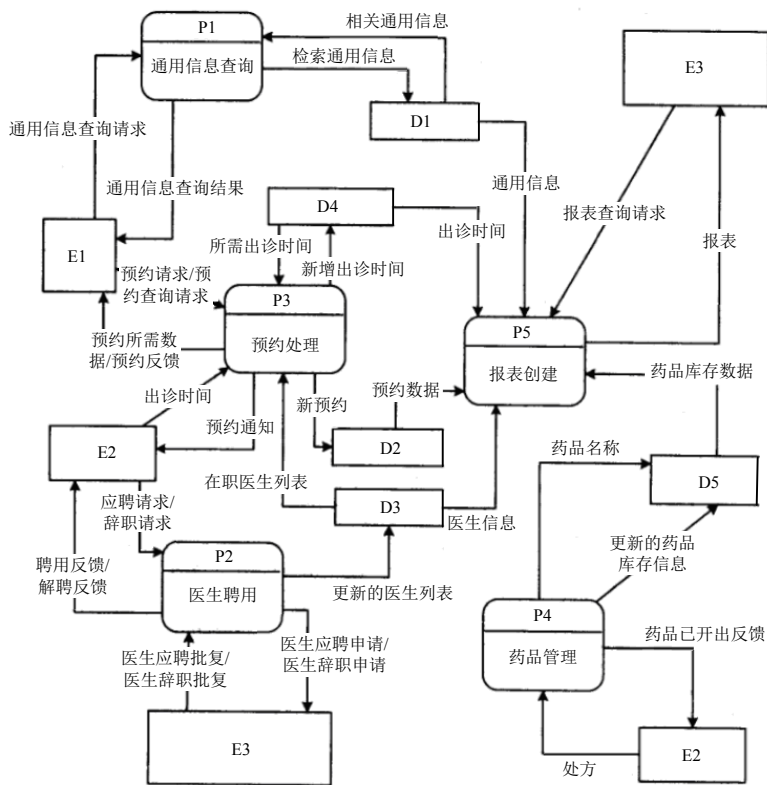


图 1-2 0 层数据流图

【问题 1】（3 分）

使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

【问题 2】（5 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1~D5 的名称。

【问题 3】（4 分）

使用说明和图中术语，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】（3 分）

使用说明中的词语，说明“预约处理”可以分解为哪些子加工，并说明建模图 1-1 和图 1-2 是如何保持数据流图平衡的。

试题二（15 分）

某海外代购公司为扩展公司业务，需要开发一个信息化管理系统。请根据公司现有业务及需求完成该系统的数据库设计。

【需求描述】

（1）记录公司员工信息。员工信息包括工号、身份证号、姓名、性别和一个手机号，工号唯一标识每位员工，员工分为代购员和配送员。

（2）记录采购的商品信息。商品信息包括商品名称、所在超市名称、采购价格、销售价格和商品介绍，系统内部用商品条码唯一标识每种商品。一种商品只在一家超市代购。

（3）记录顾客信息。顾客信息包括顾客真实姓名、身份证号（清关缴税用）、一个手机号和一个收货地址，系统自动生成唯一的顾客编号。

（4）记录托运公司信息。托运公司信息包括托运公司名称、电话和地址，系统自动生成唯一的托运公司编号。

（5）顾客登录系统之后，可以下订单购买商品。订单支付成功后，系统记录唯一的支付凭证编号，顾客需要在订单里指定运送方式（空运或海运）。

（6）代购员根据顾客的订单在超市采购对应商品，一份订单所含的多个商品可能由多名代购员从不同超市采购。

（7）采购完的商品交由配送员根据顾客订单组合装箱，然后交给托运公司运送。托运公司按顾客订单核对商品名称和数量，然后按顾客的地址进行运送。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）。

员工（工号，身份证号，姓名，性别，手机号）

商品（条码，商品名称，所在超市名称，采购价格，销售价格，商品介绍）

顾客（编号，姓名，身份证号，手机号，收货地址）

托运公司（托运公司编号，托运公司名称，电话，地址）

订单（订单 ID，(a)，商品数量，运送方式，支付凭证编号）

代购（代购 ID，代购员工号，(b)）

运送（运送 ID，配送员工号，托运公司编号，订单 ID，发运时间）

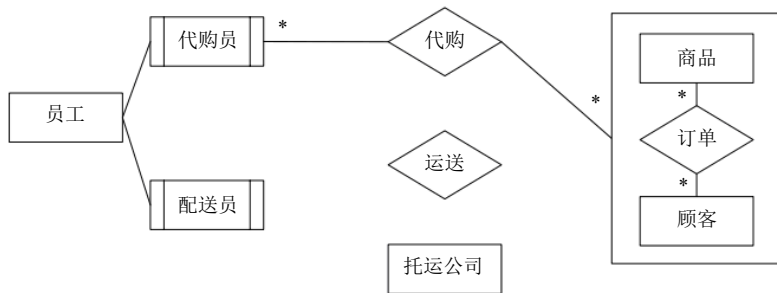


图 2-1 实体联系图

【问题 1】（3 分）

根据问题描述，补充图 2-1 的实体联系图。

【问题 2】（6 分）

补充逻辑结构设计结果中 (a)、(b) 两处空缺。

【问题 3】（6 分）

为了方便顾客，允许顾客在系统中保存多组收货地址。请根据此需求，增加“顾客地址”弱实体，对图 2-1 进行补充，并修改“运送”关系模式。

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某 ETC（Electronic Toll Collection，不停车收费）系统在高速公路沿线的特定位置上设置一个横跨道路上空的龙门架（Toll gantry），龙门架下包括六条车道（Trafficlanes），每条车道上安装有雷达传感器（Radar Sensor）、无线传输器（Radio Transceiver）和数码相机（Digital Camera）等用于不停车收费的设备，以完成正常行驶速度下的收费工作。该系统的基本工作过程如下：

（1）每辆汽车上安装有车载器，驾驶员（Driver）将一张具有唯一识别码的磁卡插入车载器中。磁卡中还包含有驾驶员账户的当前信用记录。

（2）当汽车通过某条车道时，不停车收费设备识别车载器内的特有编码，判断车型，将收集到的相关信息发送到该路段所属的区域系统（Regional Center）中，计算通行费用创建收费交易（Transaction），从驾驶员的专用账户中扣除通行费用。如果驾驶员账户透支，则记录透支账户交易信息。区域系统再将交易后的账户信息发送到维护驾驶员账户信息的中心系统（Central System）。

(3) 车载器中的磁卡可以使用邮局的付款机进行充值。充值信息会传送至中心系统，以更新驾驶员账户的余额。

(4) 当没有安装车载器或者车载器发生故障的车辆通过车道时，车道上的数码相机将对车辆进行拍照，并将车辆照片及拍摄时间发送到区域系统，记录失败的交易信息；并将该交易信息发送到中心系统。

(5) 区域系统会获取不停车收费设备所记录的交通事件(Traffic Events); 交通广播电台(Traffic Advice Center) 根据这些交通事件进行路况分析并播报路况。

现采用面向对象方法对上述系统进行分析与设计，得到如表 3-1 所示的用例列表和如图 3-1 所示的用例图及如图 3-2 所示的分析类图。

表 3-1 用例列表

用例名称	说明
Create Transaction	记录收费交易
Charge Card	磁卡充值
Underpaid Transaction	记录透支账户交易信息
Record Illegal Use	记录非法交易信息
Record Traffic Event	记录交通事件

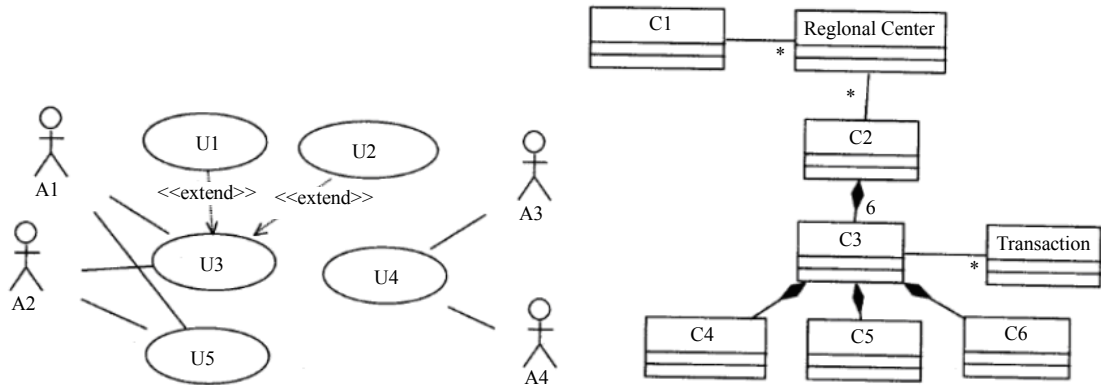


图 3-1 用例图

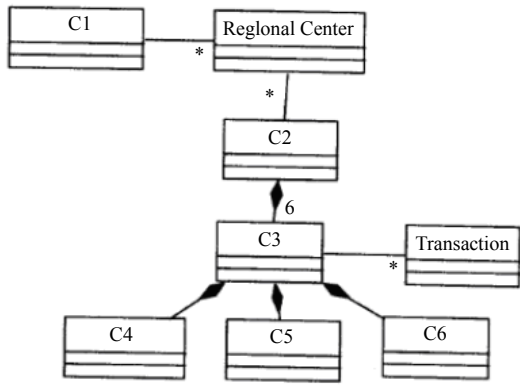


图 3-2 分析类图

【问题 1】(4 分)

根据说明中的描述，给出图 3-1 中 A1~A4 所对应的参与者名称。

【问题 2】(5 分)

根据说明中的描述及表 3-1，给出图 3-1 中 U1~U5 所对应的用例名称。

【问题 3】(6 分)

根据说明中的描述，给出图 3-2 中 C1~C6 所对应的类名。

试题四 (15 分)

阅读下列说明和 C 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某公司购买长钢条, 将其切割后进行出售。切割钢条的成本可以忽略不计, 钢条的长度为整英寸。已知价格表示为 P , 其中 $P_i (i=1,2,\dots,m)$ 表示长度为 i 英寸的钢条的价格。现要求解使销售收益最大的切割方案。

求解此切割方案算法的基本思想如下: 假设长钢条的长度为 n 英寸, 最佳切割方案的最左边切割段长度为 i 英寸, 则继续求解剩余长度为 $n-i$ 英寸钢条的最佳切割方案。考虑所有可能的 i , 得到的最大收益 rn 对应的切割方案即为最佳切割方案。 rn 的递归定义如下:

$$rn = \max_{1 \leq i \leq n} (P_i + rn - i)$$

对此递归式, 给出自顶向下和自底向上两种实现方式。

【C 代码】

```
/*常量和变量说明
n: 长钢条的长度
P[]: 价格数组
*/
#define LEN 100
int Top_Down_Cut_Rod(int P[], int n){/*自顶向下*/
    int r=0;
    int i;
    if(n==0){
        return 0;
    }
    for(i=1; i<=n; i++){
        int tmp=p[i]+Top_Down_Cut_Rod(p,n-i);
        r=(r>tmp)?r: tmp;
    }
    return r;
}
int Bottom_Up_Cut_Rod(int p[],int n){/*自底向上*/
    int r[LEN]={0};
    int temp=0;
    int i,j;
    for(j=1;j<=n;j++){
        temp=0;
        for(i=1; i<=j; i++){
            temp= (3) ;
        }
        (4)
    }
    return r[n];
}
```

【问题 1】(8 分)

根据说明，填充 C 代码中的（1）～（4）。

【问题 2】（7 分）

根据说明和 C 代码，算法采用的设计策略为__（5）__。

求解 r_n 时，自顶向下方法的时间复杂度为__（6）__；自底向上方法的时间复杂度为__（7）__。

（用 O 表示）。

试题五（15 分）

阅读下列说明和 C++ 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

生成器（Builder）模式的意图是将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示，如图 5-1 所示为其类图。

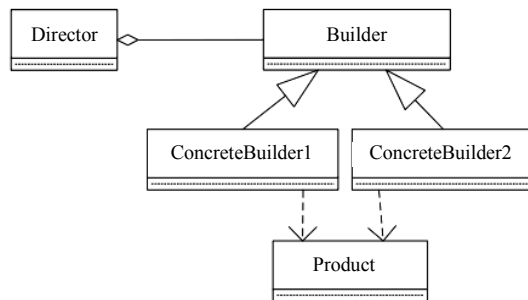


图 5-1 生成器模式类图

【C++ 代码】

```

#include <iostream>
#include <string>
using namespace std;

class Product {
private:
    string partA, partB;
public:
    Product() { }
    void setPartA(const string& s) { PartA = s;}
    void setPartB(const string& s) { PartB = s;}
    //其余代码省略
};

class Builder {
public:
    (1) ;
    virtual void buildPartB()=0;
    (2) ;
};
  
```



```

class ConcreteBuilder1: public Builder {
private:
    Product*product;
public:
    ConcreteBuilder1() {product = new Product(); }
    void buildPartA() { (3) ("Component A"); }
    void buildPartB() { (4) ("Component B"); }
    Product* getResult() {return product;}
    //其余代码省略
};
class ConcreteBuilder2: public Builder {

    /*代码省略*/

};
class Director {
private:
    Builder* builder;
public:
    Director(Builder* pBuilder) { builder= pBuilder;}
    void construct() {
        (5);
        //其余代码省略
    }
    //其余代码省略
};
int main() {
    Director* director1=new Director(new ConcreteBuilder1());
    director1->construct();
    delete director1;
    return 0;
}

```

【问题】(15 分)

阅读上述说明和 C++ 代码，将应填入的字句写在答题纸的对应栏内。

试题六 (15 分)

阅读下列说明和 Java 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

生成器 (Builder) 模式的意图是将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示，如图 6-1 所示为其类图。

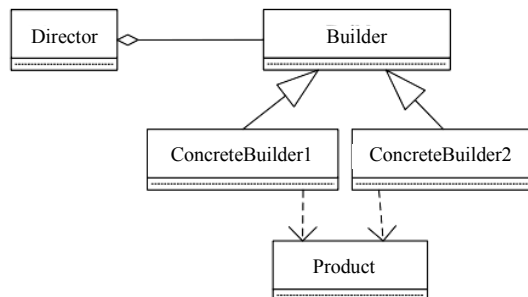


图 6-1 生成器模式类图

【Java 代码】

```

import java.util.*;

class Product {
    private String partA;
    private String partB;
    public Product() {}
    public void setPartA(String s) {partA = s;}
    public void setPartB(String s) {partB = s;}
}

interface Builder {
    public ____ (1) ____;
    public void buildPartB();
    public ____ (2) ____;
}

class ConcreteBuilder1 implements Builder {
    private Product product;
    public ConcreteBuilder1() { product = new Product(); }
    public void buildPartA() { ____ (3) ____ ("Component A"); }
    public void buildPartB() { ____ (4) ____ ("Component B"); }
    public Product getResult() { return product;}
}

class ConcreteBuilder2 implements Builder {
    //代码省略
}

class Director {
    private Builder builder;
    public Director(Builder builder) {this.builder = builder; }
    public void construct() {
        ____ (5) ____;
        //代码省略
    }
}

```

```
class Test {  
    public static void main(String[] args) {  
        Director director1 = new Director(new ConcreteBuilder1());  
        director1.construct();  
    }  
}
```

【问题】(15 分)

阅读上述说明和 Java 代码，将应填入的字句写在答题纸的对应栏内。

第29小时

2018年5月真题解析

上午基础知识

(1) 试题分析：图的邻接矩阵是指用一个矩阵来表示图中顶点之间的关系。对有 n 个结点的图，其邻接矩阵是一个 n 阶方阵。对于无向图来说，其邻接矩阵如下图所示。当采用深度优先进行遍历的时候，查找所有邻接点所需要的时间是 $O(n^2)$ 。

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

答案：(1) A

(2) 试题分析：两个浮点数对阶的时候要把阶码小的数的尾数右移 n 位，与阶码大的对齐。

答案：(2) D

(3)、(4) 试题分析：为了提高 I/O 能力和 CPU 效率，计算机系统引进了中断方式。程序中断是指计算机执行现行程序的过程中，出现某些急需处理的异常情况和特殊请求，CPU 暂时中止现行程序（保护现场），而转去对随机发生的更紧迫的事件进行处理，在处理完毕后，CPU 将自动返回原来的程序继续执行（恢复现场）。

答案：(3) B (4) C

(5)、(6) 试题分析：海明码数据位与校验位之间的关系可以采用公式表示为 $2^k \geq n+k+1$ ，其中数据位是 n 位，校验位是 k 位。因此第 5 题套入公式可得需要 6 位校验位。

第 6 题中 D5 在第 10 位, $10=8+2=2^3+2^1$, 由于校验码处于 2^0 、 2^1 、 2^2 、 2^3 ……, 分别对应 P1、P2、P3、P4……, 因此 D5 是由 P4P2 进行校验的。

答案: (5) D (6) B

(7) 试题分析: 流水线吞吐率是指令执行周期的倒数, 即 $1/\Delta t$, 而指令执行周期是最长流水段的操作时间。

答案: (7) C

(8) 试题分析: SSH 为 Secure Shell 的缩写, 由 IETF 的网络小组 (Network Working Group) 所制定; SSH 为建立在应用层基础上的安全协议。SSH 是目前较可靠, 专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。

答案: (8) D

(9)、(10) 试题分析: 数字签名采用私钥进行签名, 公钥进行验证。

答案: (9) B (10) A

(11) 试题分析: 上述措施均可。

答案: (11) C

(12) 试题分析: 重放攻击的基本原理就是把以前窃听到的数据原封不动地重新发送给接收方。很多时候, 网络上传输的数据是加密过的, 此时窃听者无法得到数据的准确意义。但如果他知道这些数据的作用, 就可以在不知道数据内容的情况下通过再次发送这些数据达到愚弄接收端的目的。例如, 有的系统会将鉴别信息进行简单加密后进行传输, 这时攻击者虽然无法窃听密码, 但他们却可以首先截取加密后的口令然后将其重放, 从而利用这种方式进行有效的攻击。再比如, 假设网上存款系统中, 一条消息表示用户支取了一笔存款, 攻击者完全可以多次发送这条消息而偷窃存款。

答案: (12) A

(13) 试题分析: B 选项《计算机软件保护条例》是由国务院颁布的。

C 选项根据《计算机软件保护条例》第十一条: 接受他人委托开发的软件, 其著作权的归属由委托人与受托人签订书面合同约定; 无书面合同或合同未作明确约定的, 其著作权由受托人享有。

D 选项只有使用权。

答案: (13) A

(14) 试题分析: 根据《著作权法》第十六条: 公民为完成法人或者其他组织工作任务所创作的作品是职务作品, 除本条第二款的规定以外, 著作权由作者享有, 但法人或者其他组织有权在其业务范围内优先使用。作品完成两年内, 未经单位同意, 作者不得许可第三人以与单位使用的相同方式使用该作品。

有下列情形之一的职务作品, 作者享有署名权, 著作权的其他权利由法人或者其他组织享有, 法人或者其他组织可以给予作者奖励: ①主要是利用法人或者其他组织的物质技术条件创作, 并由法人或者其他组织承担责任的工程设计图、产品设计图、地图、计算机软件等职务作品; ②法律、行政法规规定或合同约定著作权由法人或者其他组织享有的职务作品。

答案: (14) B

(15) 试题分析: 根据《著作权法》第二十条, 作者的署名权、修改权、保护作品完整权的保护期不受限制。

答案: (15) C

(16) 试题分析: 数据字典就是为数据流图中的每个数据流、文件、加工, 以及组成数据流或文件的数据项作出说明。数据字典有四类条目: 数据流、数据项、数据存储和基本加工。

答案: (16) A

(17) 试题分析: 根据题干做出下列判定表。

		1	2	3	4
条件	订单金额	>5000 元	>5000 元	≤5000 元	≤5000 元
	欠款时间	>60 天	≤60 天	>60 天	≤60 天
操作	不予批准	√			
	发出批准书		√	√	√
	发出发货单		√	√	√
	发催款通知书			√	

表中第 2 列与第 4 列操作相同, 可以进行合并, 最终的条件组合有 3 种。

答案: (17) B

(18)、(19) 试题分析: 关键路径为 ABCFEIKL 和 ABDGIKL, 工期 22 天。活动 EH 与 IJ 不是关键工作, 总时差分别为 5 天和 3 天。

答案: (18) D (19) C

(20) 试题分析: COCOMO II 是一种成本估算模型, 估算选择有对象点、功能点和代码行。

答案: (20) C

(21) 试题分析: LISP 是函数式程序设计语言, 长期以来垄断人工智能领域的应用。LISP 语言中大量使用了递归。

答案: (21) A

(22) 试题分析: 解释程序也称为解释器, 它或者直接解释执行源程序, 或者将源程序翻译成某种中间表示形式后再加以执行; 编译程序 (编译器) 则是将源程序翻译成目标语言程序, 然后在计算机上运行目标程序。

两种语言处理程序的根本区别是: 在编译方式下, 机器上运行的是与源程序等价的目标程序, 源程序和编译程序都不再参与目标程序的执行过程, 而在解释方式下, 解释程序和源程序 (或某种等价表示) 要参与到程序的运行过程中, 运行程序的控制权在解释程序。

解释器翻译源程序时不生成独立的目标程序, 而编译器则将源程序翻译成独立的目标程序。

答案: (22) C

(23) 试题分析: 后续遍历是采用左—右—根的方式进行遍历, 只有 B 选项符合题干要求。

答案：(23) B

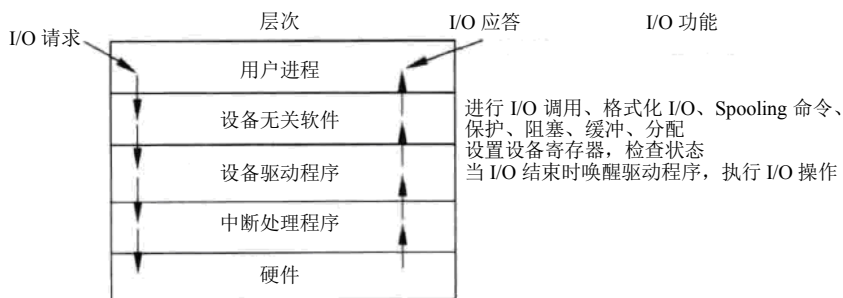
(24)、(25) 试题分析：按照用户购票要求找到了单元，那么 (a) 应该执行 P 操作来申请一个资源，互斥操作的信号量初值一般为 1 或资源的数量，剩下的 (b)、(c) 都应该是 V 操作，释放资源。

答案：(24) C (25) D

(26) 试题分析：很多文件系统是先读取磁盘块到主存，在主存进行修改，修改完毕再写回磁盘。但如果读取某磁盘块，修改后再将信息协会磁盘前系统崩溃，则文件系统可能会出现不一致状态。如果这些未被写回的磁盘块是索引结点、目录块或空闲块，那么后果是不堪设想的。通常，解决方案是采用文件系统的一致性检查，一致性检查包括块的一致性和文件的一致性检查。

答案：(26) B

(27) 试题分析：I/O 软件的所有层次及每一层的主要功能如下图所示。



答案：(27) D

(28)、(29) 试题分析：文件的全文件名应包括盘符及从根目录开始的路径名；文件的相对路径是从当前工作目录下的路径名；文件的绝对路径名是指目录下的绝对位置，直接到达目标位置。

答案：(28) C (29) B

(30) 试题分析：随着开发过程的进展，如果用户的需求发生了变更，一些增量可能需要重新开发，由此产生的管理成本也会增加甚至超出组织的能力。

答案：(30) A

(31) 试题分析：CL1（已执行的）的共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品，以实现支持过程域的特定目标。

答案：(31) A

(32) 试题分析：辅助软件维护工具辅助维护人员对软件代码及其文档进行各种维护活动。软件维护工具主要有：版本控制工具、文档分析工具、开发信息库工具、逆向工程工具、再工程工具。

答案：(32) B

(33) 试题分析：一般来讲，概要设计的内容可以包含系统架构、模块划分、系统接口、数据设计四个方面的主要内容，不包括模块内算法设计。

答案：(33) C

(34) 试题分析: 耦合是模块之间的相对独立性(互相连接的紧密程度)的度量。耦合取决于各个模块之间接口的复杂程度、调用模块的方式及通过接口的信息类型。

答案: (34) D

(35)、(36) 试题分析: 判定覆盖是设计足够多的测试用例, 使得程序中的每一个判断至少获得一次“真”和一次“假”, 即使得程序流程图中的每一个真假分支至少被执行一次。

计算环路复杂度可以采用公式法 $V(G) = m - n + 2$, 其中 m 为弧度(边的数量), n 为结点数量。或者可以数封闭区间的数量, 除了图中明显的两处外, 与外界连通的部分也是一个封闭区间。

答案: (35) A (36) B

(37) 试题分析: 归纳法是指从测试所暴露的问题出发, 收集所有正确或不正确的数据, 分析它们之间的关系, 提出假想的错误原因, 用这些数据来证明或反驳, 从而查出错误所在。

答案: (37) C

(38) 试题分析: 对象的状态标识了该对象的所有属性(通常是静态的)及每个属性的当前值(通常是动态的)。行为是对象根据它的状态和消息传递所采取的行动和所作出的反映; 操作则代表了一个类提供给它的对象的一种服务。

答案: (38) A

(39)、(40) 试题分析: 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。程序运行过程中, 把函数(或过程)调用与响应调用所需要的代码相结合的过程称为动态绑定。

静态绑定是指在程序编译过程中, 把函数(方法或者过程)调用与响应调用所需的代码结合的过程称之为静态绑定。

答案: (39) D (40) C

(41) 试题分析: 多态是指同一操作作用于不同的对象, 可以有不同的解释, 产生不同的执行结果。在运行时, 可以通过指向基类的指针, 来调用实现派生类中的方法。

答案: D

(42)~(44) 试题分析: 协作图(通信图)强调的是发送和接收消息的对象之间的组织结构。协作图的组成元素有:

1) 对象: 题干图形中的矩形元素即为对象, 其中冒号前面部分为对象名, 后面为类名, 表示类的一个实例。

2) 链接: 用两个对象之间的单一线条表示, 用来在通信图中关联对象, 目的是让消息在不同系统对象之间传递。可以理解链接是公路, 消息是车。

3) 消息: 通信图中对象之间通信的方式。

协作图与顺序图相比顺序图(序列图)强调的消息时间顺序的交互图, 描述类系统中类与类之间的交互, 它将这些交互建模成消息互换, 换句话说, 顺序图描述了类与类之间相互交换以完成期望行为的消息。

答案: (42) C (43) A (44) D

(45)~(48) 试题分析: 桥接模式将抽象部分与它的实现部分分离, 使它们都可以独立地变化。

答案: (45) D (46) A (47) B (48) A

(49) 试题分析: 将四个选项分别带入可以得出答案。

答案: (49) A

(50) 试题分析: 从开始出发, 不断推导与替换非终结符。 $E \rightarrow E+T \rightarrow T+T \rightarrow F+T \rightarrow N+T \rightarrow 2+T \rightarrow 2+(T * F) \rightarrow 2+(-F * N) \rightarrow 2+(-N) * N \rightarrow 2+(-3) * 4$

答案: (50) B

(51) 试题分析: 程序设计语言的语义分为静态语义和动态语义, 其中静态语义分析方法是语法制导翻译, 其基本思想是将语言结构的语义以属性的形式赋予代表此结构的文法符号, 而属性的计算以语义规则的形式赋予文法的产生式。

答案: (51) C

(52) 试题分析: 从已知的一些函数依赖, 可以推导出另外一些函数依赖, 这就需要一系列推理规则。函数依赖的推理规则最早出现在 1974 年 W.W.Armstrong 的论文里, 这些规则常被称作 Armstrong 公理。

合并规则: 若 $X \rightarrow Y$, $X \rightarrow Z$ 同时在 R 上成立, 则 $X \rightarrow YZ$ 在 R 上也成立。

分解规则: 若 $X \rightarrow W$ 在 R 上成立且属性集 Z 包含于 W , 则 $X \rightarrow Z$ 在 R 上也成立。

伪传递规则: 若 $X \rightarrow Y$ 在 R 上成立且 $WY \rightarrow Z$, 则 $XW \rightarrow Z$ 。

答案: (52) B

(53)、(54) 试题分析: 第 53 题中自然连接去掉重复的属性列, 结果集是(A, S.B, S.C, D, E, F, G), 选择第 2、4、6、7 列, 答案是(R.B, D, F, G)。

第 54 题相当于选择出满足属性列 2 的值<属性列 7 的那些行。

答案: (53) A (54) C

(55)、(56) 试题分析: 第 55 题是返回值的类型; 第 56 题是声明。

答案: (55) A (56) D

(57) 试题分析: 数据仓库是决策支持系统和联机分析应用数据源的结构化数据环境。数据仓库研究和解决从数据库中获取信息的问题。数据仓库的特征在于面向主题、集成性、稳定性和时变性。

OLAP 工具是针对特定问题的联机数据访问与分析。它通过多维的方式对数据进行分析、查询和报表。例如, 从时间、地区和商品种类三个维度来分析某家电商品的销售数据。

答案: (57) B

(58) 试题分析: 单链表只能向后遍历, 无法逆序遍历。

答案: (58) C

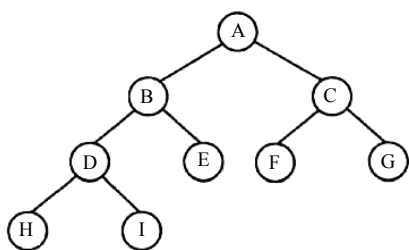
(59) 试题分析: 三对角矩阵如下所示。

$$A_{n \times n} = \begin{bmatrix} a_{1,1} & a_{1,2} & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & 0 \\ & a_{3,2} & a_{3,3} & a_{3,4} & \\ & & \dots & \dots & \dots \\ & 0 & & \dots & \dots \\ & & & & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

将 $i=0$ 、 $j=0$ 与 $1=n-1$ 、 $j=n-1$ 分别带入选项中, 可得 C 选项。

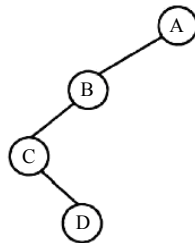
答案: (59) C

(60) 试题分析: 下列二叉树的顺序存储如下图所示。因此, 要为题干中 B 的子树构造一些虚拟的结点, 所以 D、E、F 对应的数组元素下标为 6、7、14。



1	2	3	4	5	6	7	8	9	10
A	B	C	D	E	F	G	H	I	

(a) 完全二叉树



1	2	3	4	5	6	7	8	9	10
A	B	φ	C	φ	φ	φ	φ	D	

(b) 一般二叉树

答案: (60) D

(61) 试题分析: 哈希表 (hashtable) 通过一个已记录的关键字为自变量的函数 (哈希函数) 得到该记录的存储地址, 所以在哈希表中进行查找操作时, 需用同一哈希函数计算得到待查记录的存储地址, 然后到相应的存储单元去获得有关信息再判定查找是否成功。用哈希表存储元素时, 需要进行冲突 (碰撞) 处理, 这个冲突指的是关键字不同的元素被映射到相同的存储位置。

答案: (61) B

(62) ~ (65) 试题分析: 快速排序由 C. A. R. Hoare 在 1962 年提出。它的基本思想是: 通过一趟排序将要排序的数据分割成独立的两部分, 其中一部分的所有数据都比另外一部分的所有数据都要小, 然后再按此方法对这两部分数据分别进行快速排序, 整个排序过程可以递归进行, 以此达到整个数据变成有序序列。快速排序采用的思想是分治思想。

贪心算法 (又称贪婪算法) 是指在对问题求解时, 总是做出在当前看来是最好的选择。也就是说, 不从整体最优上加以考虑, 所做出的是在某种意义上的局部最优解。

整个算法的时间复杂度是 $O(n \log n)$ 。

场地上可以安排活动 1、8、11 为一个场地；活动 2、6、9 一个场地；活动 3 为一个场地；活动 4、7 为一个场地；活动 5、10 为一个场地，共 5 个场地。

答案：(62) A (63) C (64) C (65) B

(66) 试题分析：网络设备工作的层次如下表所示。

互联设备	工作层次	主要功能
中继器	物理层	对接收信号进行再生和发送，只起扩展传输距离的作用，对高层协议是透明的，但使用个数有限（如在以太网中只能使用 4 个）
网桥	数据链路层	根据帧物理地址进行网络之间的信息转发，可以缓解网络通信繁忙度，提高效率，只能连接相同 MAC 层的网络
路由器	网络层	通过逻辑地址进行网络之间的信息转发，可以完成异构网络之间的互联互通，只能连接使用相同网络层协议的子网
网关	高层（第 4~7 层）	最复杂的网络互联设备，用于连接网络层以上的执行不同协议的子网
集线器	物理层	多端口中继器
二层交换机	数据链路层	传统意义上的交换机，多端口网桥
三层交换机	网络层	带路由功能的二层交换机
多层交换机	高层（第 4~7 层）	带协议转换功能的交换机

答案：(66) B

(67)、(68) 试题分析：www.dailynews.com.cn 表示主机，welcome.htm 表示网页文件名。

答案：(67) B (68) C

(69) 试题分析：在 Linux 中，要更改一个文件的权限设置可以使用 chmod 命令。

答案：(69) C

(70) 试题分析：当应用过程需要将一个主机域名映射为 IP 地址时，就调用域名解析函数，解析函数将待转换的域名放在 DNS 请求中，以 UDP 报文方式发给本地域名服务器。本地的域名服务器查到域名后，将对应的 IP 地址放在应答报文中返回。同时域名服务器还必须具有连向其他服务器的信息以支持不能解析时的转发。若域名服务器不能回答该请求，则此域名服务器就暂成为 DNS 中的另一个客户，向根域名服务器发出请求解析，根域名服务器一定能找到下面的所有二级域名的域名服务器，这样以此类推，一直向下解析，直到查询到所请求的域名。

答案：(70) C

(71) ~ (75) 试题分析：

翻译：

创建一个清晰的项目地图是重要的第一步。它让你识别风险，明确目标，并确定项目是否有意义。比发布计划更重要的是不要把这个计划太当真。发布计划可以为你的 Web 项目创建一个游戏计划，(71) 你希望网站能够做什么。该计划是有关网站内容，设计元素和功能的指南，供公众，

合作伙伴,或内部使用。它还 (72) 了项目的时间和成本费用。这个计划不是一个 (73) 的实现,它没有项目的细节,你也无法用它来向银行贷款。基本上,你使用发布计划来对项目的 (74) 和价值进行初步的完整性检查。发布计划是有用的路线图,但不要认为它们是州际公路系统的指南。相反,把它们看作是早期探险家使用的 (75) ——一半是谣传与猜测,一半是希望和期盼。有一个项目走向的地图总是一件好事。

- | | | | |
|-------------|---------|--------|---------|
| (71) A. 构建 | B. 设计 | C. 实施 | D. 概述 |
| (72) A. 定义 | B. 计算 | C. 预测 | D. 知晓 |
| (73) A. 说明 | B. 结构 | C. 需求 | D. 实现 |
| (74) A. 正确性 | B. 可修改性 | C. 可行性 | D. 可追踪性 |
| (75) A. 导航 | B. 地图 | C. 指南 | D. 目标 |

答案: (71) D (72) A (73) D (74) C (75) C

下午案例分析

试题一 (15 分)

【问题 1】(3 分)

答案: E1: 客户; E2: 医生; E3: 主管。

【问题 2】(5 分)

答案: D1: 通用信息表; D2: 预约表; D3: 在职医生表; D4: 出诊时间表; D5: 药品库存信息表。

【问题 3】(4 分)

答案: 数据流名称: 删除解聘医生的预约信息; 起点: P2; 终点: D2。

数据流名称: 删除解聘医生的出诊安排; 起点: P2; 终点: D4。

数据流名称: 查询相关药品库存信息; 起点: D5; 终点: P4。

数据流名称: 查询相关药品库存信息; 起点: P4; 终点: D2。

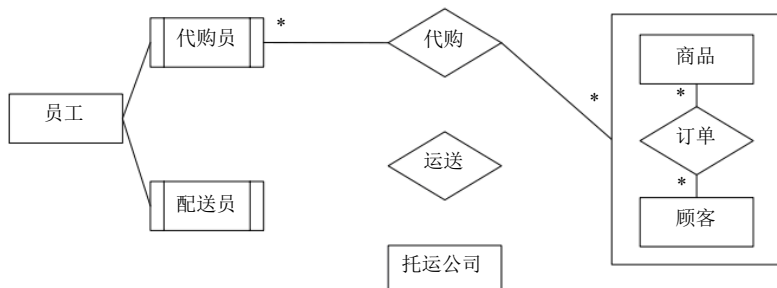
【问题 4】(3 分)

答案: “预约处理”可以分为预约就诊、安排出诊时间两个子加工。父图与子图要在数据流的数量与流向上保持一致。

试题二 (15 分)

【问题 1】(3 分)

答案: 如下图所示。

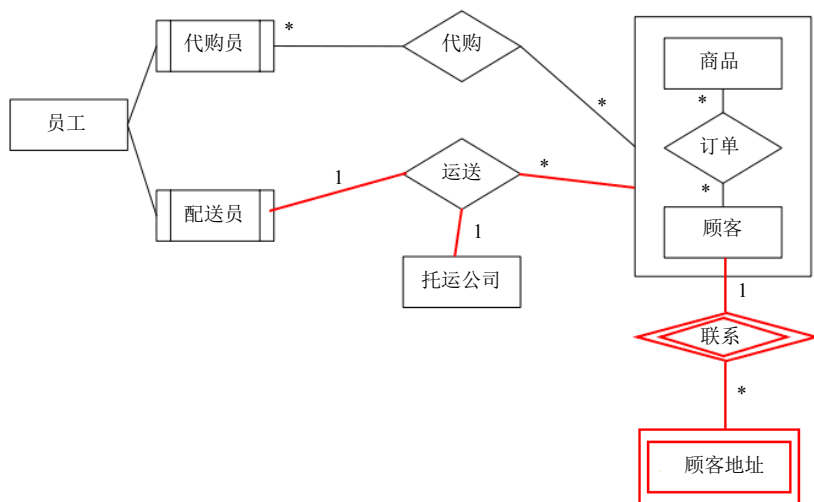


【问题 2】(6 分)

答案：(a) 商品条码、顾客编号；(b) 订单 ID。

【问题 3】(6 分)

答案：如下图所示。



运送关系模式新增联系与顾客地址。

试题三 (15 分)

【问题 1】(4 分)

答案：

A1: Central system/Driver

A2: Driver/Central system

A3: Regional center/Traffic advice center

A4: Traffic advice center/Regional center

【问题 2】(5 分)

答案:

U1: Underpaid transaction/Record Illegal use

U2: Record Illegal use/Underpaid transaction

U3: Create transaction

U4: Record traffic event

U5: Charge card

【问题 3】(6 分)

答案:

C1: Center system

C2: Toll gantry

C3: Traffic lanes

C4: Radar sensor/Radio transceiver/Digital Camera

C5: Radio transceiver/Radar sensor/Digital Camera

C6: Digital Camera/Radar sensor/Radio transceiver

试题四 (15 分)

【问题 1】(8 分)

答案:

(1) $i \leq n$

(2) $i \leq j$

(3) $\text{temp} = (\text{temp} \geq r[i] + r[j-1]) ? \text{temp} : (r[i] + r[j-1])$

(4) $r[j] = (\text{temp} > p[j]) ? \text{temp} : p[j]$

【问题 2】(7 分)

答案:

(5) 动态规划

(6) $O(2^n)$

(7) $O(n^2)$

试题五 (15 分)

答案:

(1) `virtual void buildPart A()=0`

(2) `virtual Product*getResult()=0`

(3) `product->setPartA`

- (4) product->setPartB
- (5) builder->buildPartA();
builder->buildPartB();
Product*p=builder->getResult();

试题六 (15 分)

答案:

- (1) void buildPart A()
- (2) Product getResult()
- (3) product.setPartA
- (4) product.setPartB
- (5) builder.buildPartA();
builder.buildPartB();
Product p=builder.getResult();

第30小时

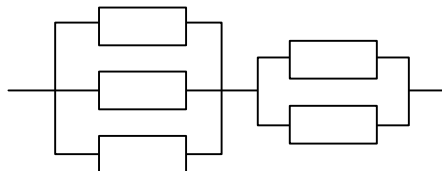
2018 年 11 月考试真题（上午基础知识）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

- CPU 在执行指令的过程中，会自动修改(1)的内容，以使其保存的总是将要执行的下一条指令的地址。
(1) A. 指令寄存器 B. 程序计数器 C. 地址寄存 D. 指令译码器
- 在微机系统中，BIOS（基本输入输出系统）保存在(2)中。
(2) A. 主板上的 ROM B. CPU 的寄存器
C. 主板上的 RAM D. 虚拟存储器
- 采用 n 位补码（包含一个符号位）表示数据，可以直接表示数值(3)。
(3) A. $2n$ B. $-2n$ C. $2n-1$ D. $-2n-1$
- 某系统由下图所示的部件构成，每个部件的千小时可靠度都为 R ，该系统的千小时可靠度为(4)。



- (4) A. $(3R+2R)/2$ B. $R/3+R/2$
C. $(1-(1-R)^3)(1-(1-R)^2)$ D. $(1-(1-R)^3)-(1-R)^2$
- 以下关于采用一位奇校验方法的叙述中, 正确的是 (5)。
(5) A. 若所有奇数位出错, 则可以检测出该错误但无法纠正错误
B. 若所有偶数位出错, 则可以检测出该错误并加以纠正
C. 若有奇数个数据位出错, 则可以检测出该错误但无法纠正错误
D. 若有偶数个数据位出错, 则可以检测出该错误并加以纠正
 - 下列关于流水线方式执行指令的叙述中, 不正确的是 (6)。
(6) A. 流水线方式可以提高单条指令的执行速度
B. 流水线方式下可以同时执行多条指令
C. 流水线方式提高了各部件的利用率
D. 流水线方式提高了系统的吞吐率
 - DES 是 (7) 算法。
(7) A. 公开密钥加密 B. 共享密钥加密
C. 数字签名 D. 认证
 - 计算机病毒的特征不包括 (8)。
(8) A. 传染性 B. 触发性 C. 隐蔽性 D. 自毁性
 - MD5 是 (9) 算法, 对任意长度的输入计算得到的结果长度为 (10) 位。
(9) A. 路由选择 B. 摘要 C. 共享密钥 D. 公开密钥
(10) A. 56 B. 128 C. 140 D. 160
 - 使用 Web 方式收发电子邮件时, 以下描述错误的是 (11)。
(11) A. 无需设置简单邮件传输协议 B. 可以不设置账号密码登录
C. 邮件可以插入多个附件 D. 未发送的邮件可以保存到草稿箱
 - 有可能无限期拥有的知识产权是 (12)。
(12) A. 著作权 B. 专利权 C. 商标权 D. 集成电路布图设计权
 - (13) 是构成我国保护计算机软件著作权的两个基本法律文件。
(13) A. 《软件法》和《计算机软件保护条例》
B. 《中华人民共和国著作权法》和《计算机软件保护条例》
C. 《软件法》和《中华人民共和国著作权法》
D. 《中华人民共和国版权法》和《计算机软件保护条例》
 - 某软件程序员接受一家公司 (软件著作权人) 委托开发完成一个软件, 三个月后又接受另一家公司委托开发功能类似的软件, 此程序员仅将受第一家公司委托开发的软件略作修改即提交给第二家公司, 此种行为 (14)。
(14) A. 属于开发者的特权 B. 属于正常使用著作权
C. 不构成侵权 D. 构成侵权

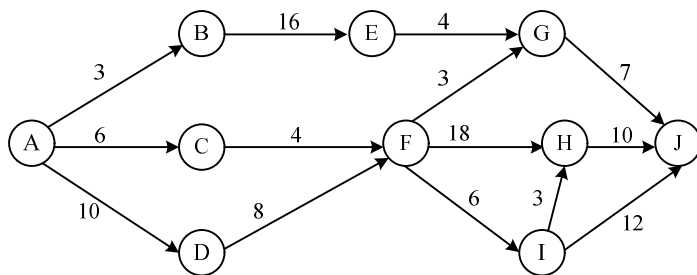
- 结构化分析的输出不包括 (15)。

(15) A. 数据流图 B. 数据字典 C. 加工逻辑 D. 结构图

- 某航空公司拟开发一个机票预订系统, 旅客预订机票时使用信用卡付款。付款通过信用卡公司的信用卡管理系统提供的接口实现。若采用数据流图建立需求模型, 则信用卡管理系统是 (16)。

(16) A. 外部实体 B. 加工 C. 数据流 D. 数据存储

- 某软件项目的活动图如下图所示, 其中顶点表示项目里程碑, 连接顶点的边表示包含的活动, 边上的数字表示活动的持续时间 (天), 则完成该项目的最少时间为 (17) 天, 活动 FG 的松弛时间为 (18) 天。



(17) A. 20 B. 37 C. 38 D. 46

(18) A. 9 B. 10 C. 18 D. 26

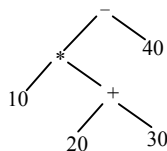
- 以下叙述中, (19) 不是一个风险。

(19) A. 由另一个小组开发的子系统可能推迟交付, 导致系统不能按时交付给客户
 B. 客户不清楚想要开发什么样的软件, 因此开发小组开发原型帮助其确定需求
 C. 开发团队可能没有正确地理解客户的需求
 D. 开发团队的核心成员可能在系统开发过程中离职

- 对布尔表达式进行短路求值是指无需对表达式中的所有操作数或运算符进行计算就可以确定表达式的值。对于表达式 “ $a \text{ or } ((c < d) \text{ and } b)$ ”, (20) 时可以进行短路计算。

(20) A. d 为 true B. a 为 true C. b 为 true D. c 为 true

- 下面的二叉树表示的简单算术表达式为 (21)。

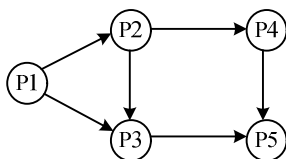


(21) A. $10 * 20 + 30 - 40$ B. $10 * (20 + 30 - 40)$

C. $10 * (20 + 30) - 40$ D. $10 * 20 + (30 - 40)$

- 在程序运行过程中, (22) 时涉及整型数据转换为浮点型数据的操作。

- (22) A. 将浮点型变量赋值给整型变量 B. 将整型常量赋值给整型变量
C. 将整型变量与浮点型变量相加 D. 将浮点型常量与浮点型变量相加
- 某计算机系统中互斥资源 R 的可用数为 8, 系统中有三个进程 P1、P2 和 P3 竞争 R , 而且每个进程都需要 i 个 R , 该系统可能会发生死锁的 i 的最小值为 (23)。
- (23) A. 1 B. 2 C. 3 D. 4
- 进程 P1、P2、P3、P4 和 P5 的前趋图如下图所示:



若用 PV 操作控制这五个进程同步与互斥的程序如下所示, 那么, 程序中的空①和空②处应分别为 (24); 空③和空④处应分别为 (25); 空⑤和空⑥处应分别为 (26)。

```

begin
    S1,S2,S3,S4,S5,S6: semaphore;           //定义信号量
    S1:=0; S2:=0; S3:=0; S4:=0; S5:=0; S6:=0;
    Cobegin
        process P1      process P2      process P3      process P4      process P5
        Begin           Begin           Begin           Begin           Begin
        P1 执行;        ②;             P(S2);        P(S4);        ⑥;
        V(S1);          P2 执行;        ③;             P4 执行;        P5 执行;
        ①;              V(S3);        P3 执行;        ⑤;             end;
        end;            V(S4);        ④;             end;
    Coend;              end;
end.                   end;

```

- (24) A. V(S1)和 P(S2) B. P(S1)和 V(S2)
C. V(S1)和 V(S2) D. V(S2)和 P(S1)
- (25) A. V(S3)和 V(S5) B. P(S3)和 V(S5)
C. V(S3)和 P(S5) D. P(S3)和 P(S5)
- (26) A. P(S6)和 P(S5)V(S6) B. V(S5)和 V(S5)V(S6)
C. V(S6)和 P(S5)P(S6) D. P(S6)和 P(S5)P(S6)
- 某文件管理系统在磁盘上建立了位示图 (bitmap), 用来记录磁盘的使用情况。若磁盘上物理块的编号依次为 0、1、2……系统中的字长为 32 位, 位示图中字的编号依次为 0、1、2……每个字中的一个二进制位对应文件存储器上的一个物理块, 取值 0 和 1 分别表示物理块是空闲或占用。假设操作系统将 2053 号物理块分配给某文件, 那么该物理块的使用情况在位示图中编号为 (27) 的字中描述。
- (27) A. 32 B. 33 C. 64 D. 65
- 某操作系统文件管理采用索引节点法。每个文件的索引节点有 8 个地址项, 每个地址项的大小为 4 字节, 其中, 5 个地址项为直接地址索引, 2 个地址项为一级间接地址索引, 1 个地址

项为二级间接地址索引, 磁盘索引块和磁盘数据块的大小均为 1KB。若要访问文件的逻辑块号分别为 1 和 518, 则系统应分别采用 (28)。

- (28) A. 直接地址索引和一级间接地址索引
B. 直接地址索引和二级间接地址索引
C. 一级间接地址索引和一级间接地址索引
D. 一级间接地址索引和二级间接地址索引

- 某企业拟开发一个企业信息管理系统, 系统功能与多个部门的业务相关。现希望该系统能够尽快投入使用, 系统功能可以在使用过程中不断改善, 则最适宜采用的软件过程模型为 (29)。

- (29) A. 瀑布模型 B. 原型模型
C. 演化 (迭代) 模型 D. 螺旋模型

- 能力成熟度模型集成 (CMMI) 是若干过程模型的综合和改进。连续式模型和阶段式模型是 CMMI 提供的两种表示方法, 而连续式模型包括六个过程域能力等级, 其中, (30) 使用量化 (统计学) 手段改变和优化过程域, 以应对客户要求的改变和持续改进计划中的过程域的功效。

- (30) A. CL2 (已管理的) B. CL3 (已定义级的)
C. CL4 (定量管理的) D. CL5 (优化的)

- 在 ISO/IEC 9126 软件质量模型中, 可靠性质量特性是指在规定的一段时间内和规定的条件下, 软件维持在其性能水平有关的能力, 其质量特性不包括 (31)。

- (31) A. 安全性 B. 成熟性 C. 容错性 D. 易恢复性

- 以下关于模块化设计的叙述中, 不正确的是 (32)。

- (32) A. 尽量考虑高内聚、低耦合, 保持模块的相对独立性
B. 模块的控制范围在其作用范围内
C. 模块的规模适中
D. 模块的宽度、深度、扇入和扇出适中

- 某企业管理信息系统中, 采购子系统根据材料价格、数量等信息计算采购的金额, 并给财务子系统传递采购金额、收款方和采购日期等信息, 则这两个子系统之间的耦合类型为 (33) 耦合。

- (33) A. 数据 B. 标记 C. 控制 D. 外部

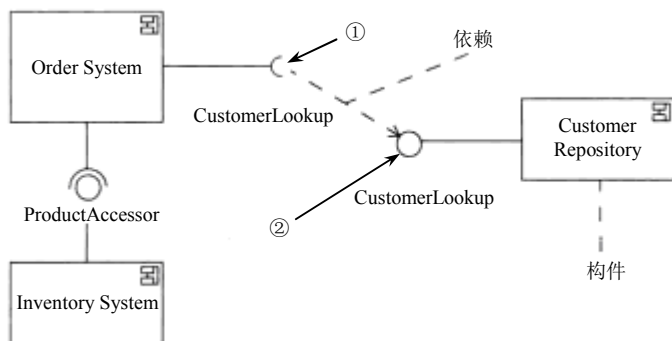
- 对以下的程序伪代码 (用缩进表示程序块) 进行路径覆盖测试, 至少需要 (34) 个测试用例。采用 McCabe 度量法计算其环路复杂度为 (35)。

```
输入 x,y,z  
语句 1  
If x>0  
    语句 2  
    If y>0  
        语句 3  
    Else
```

语句 4
Else
语句 5
If z>0
语句 6
Else
语句 7
输出语句

- (34) A. 2 B. 4 C. 6 D. 8
(35) A. 2 B. 3 C. 4 D. 5

- 某商场的销售系统所使用的信用卡公司信息系统的格式发生了更改，因此对该销售系统进行的修改属于 (36) 维护。
(36) A. 更正性 B. 适应性 C. 完善性 D. 预防性
- 在面向对象方法中，继承用于 (37)。
(37) A. 在已存在的类的基础上创建新类 B. 在已存在的类中添加新的方法
 C. 在已存在的类中添加新的属性 D. 在已存在的状态中添加新的状态
- (38) 多态是指操作（方法）具有相同的名称但在不同的上下文中代表不同的含义。
(38) A. 参数 B. 包含 C. 过载 D. 强制
- 在某销售系统中，客户采用扫描二维码的方式进行支付。若采用面向对象方法开发该销售系统，则客户类属于 (39) 类，二维码类属于 (40) 类。
(39) A. 接口 B. 实体 C. 控制 D. 状态
(40) A. 接口 B. 实体 C. 控制 D. 状态
- 下图所示的 UML 图为 (41)，用于展示 (42)，①和②分别表示 (43)。



- (41) A. 类图 B. 组件图 C. 通信图 D. 部署图
(42) A. 一组对象、接口、协作和它们之间的关系
 B. 收发消息的对象的结构组织
 C. 组件之间的组织和依赖
 D. 面向对象系统的物理模型

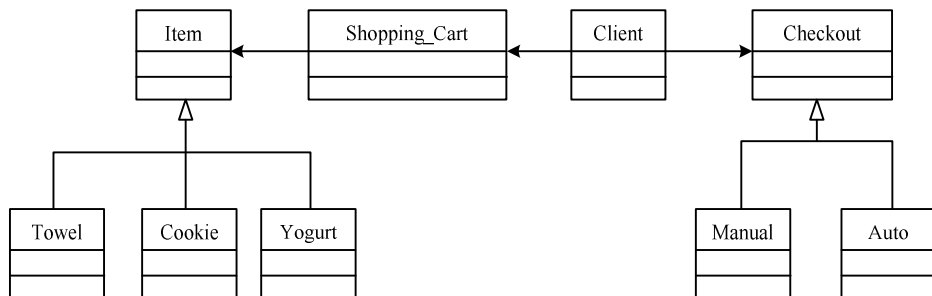
(43) A. 供接口和供接口

B. 需接口和需接口

C. 供接口和需接口

D. 需接口和供接口

- 假设现在要创建一个简单的超市销售系统, 顾客将毛巾、饼干、酸奶等物品 (Item) 加入购物车 (Shopping_Cart), 在收银台 (Checkout) 人工 (Manual) 或自动 (Auto) 地将购物车中每个物品的价格汇总成总价格后结账。这一业务需求的类图设计如下图所示 (方法略), 采用了 (44) 模式。其中, (45) 定义以一个 Checkout 对象为参数的 accept 操作, 由子类实现此 accept 操作。此模式为 (46), 适用于 (47)。



(44) A. 观察者 (Observer)

B. 访问者 (Visitor)

C. 策略 (Strategy)

D. 桥接器 (Bridge)

(45) A. Item

B. Shopping_Cart

C. Checkout

D. Manual 和 Auto

(46) A. 创建型对象模式

B. 结构型对象模式

C. 行为型类模式

D. 行为型对象模式

(47) A. 必须保存一个对象在某一个时刻的 (部分) 状态

B. 想在不明确指定接收者的情况下向多个对象中的一个提交一个请求

C. 需要对一个对象结构中的对象进行很多不同的且不相关的操作

D. 在不同的时刻指定、排列和执行请求

- 在以阶段划分的编译器中, (48) 阶段的主要作用是分析程序中的句子结构是否正确。

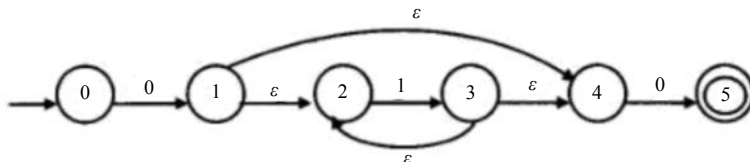
(48) A. 词法分析

B. 语法分析

C. 语义分析

D. 代码生成

- 下图所示为一个不确定有限自动机 (NFA) 的状态转换图, 该 NFA 可识别字符串 (49)。



(49) A. 0110

B. 0101

C. 1100

D. 1010

- 函数 f 和 g 的定义如下图所示。执行函数 f 时若采用引用 (call by reference) 方式调用函数 $g(a)$,

则函数 f 的返回值为 (50)。

$f()$

```
int a=5,c;
c=g(a);
return a+c;
```

g (形式参数 x)

```
int m=2;
m=x*m; x=m-1;
return x+m;
```

(50) A. 14 B. 18 C. 24 D. 28

- 数据库系统中的视图、存储文件和基本表分别对应数据库系统结构中的 (51)。

(51) A. 模式、内模式和外模式 B. 外模式、模式和内模式
C. 模式、外模式和内模式 D. 外模式、内模式和模式
- 在分布式数据库中, (52) 是指用户或应用程序不需要知道逻辑上访问的表具体如何分块存储。

(52) A. 逻辑透明 B. 位置透明 C. 分片透明 D. 复制透明
- 设有关系模式 $R(A_1, A_2, A_3, A_4, A_5, A_6)$, 函数依赖集为 $F=\{A_1 \rightarrow A_3, A_1 A_2 \rightarrow A_4, A_5 A_6 \rightarrow A_1, A_3 A_5 \rightarrow A_6, A_2 A_5 \rightarrow A_6\}$ 。关系模式 R 的一个主键是 (53), 从函数依赖集 F 可以推出关系模式 R (54)。

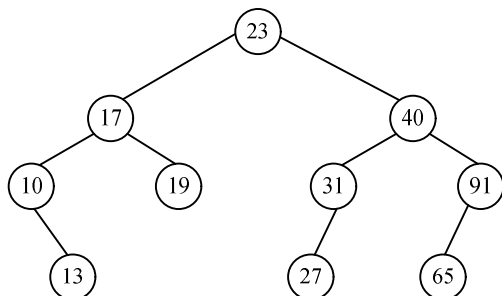
(53) A. $A_1 A_4$ B. $A_2 A_5$ C. $A_3 A_4$ D. $A_4 A_5$
(54) A. 不存在传递依赖, 故 R 为 1NF
B. 不存在传递依赖, 故 R 为 2NF
C. 存在传递依赖, 故 R 为 3NF
D. 每个非主属性完全函数依赖于主键, 故 R 为 2NF
- 给定关系 $R(A, B, C, D)$ 和 $S(C, D, E)$, 若关系 R 与 S 进行自然连接运算, 则运算后的元组属性列数为 (55); 关系代数表达式 $\pi_{1,4}(\sigma_{2=5}(R \bowtie S))$ 与 (56) 等价。

(55) A. 4 B. 5 C. 6 D. 7
(56) A. $\pi_{A,D}(\sigma_{C=D}(R \times S))$ B. $\pi_{R,A,R,D}(\sigma_{R,B=S,C}(R \times S))$
C. $\pi_{A,R,D}(\sigma_{R,C=S,D}(R \times S))$ D. $\pi_{R,A,R,D}(\sigma_{R,B=S,E}(R \times S))$
- 栈的特点是后进先出, 若用单链表作为栈的存储结构, 并用头指针作为栈顶指针, 则 (57)。

(57) A. 入栈和出栈操作都不需要遍历链表
B. 入栈和出栈操作都需要遍历链表
C. 入栈操作需要遍历链表而出栈操作不需要
D. 入栈操作不需要遍历链表而出栈操作需要
- 已知某二叉树的先序遍历序列为 A B C D E F, 中序遍历序列为 B A D C F E, 则可以确定该二叉树 (58)。

(58) A. 是单支树 (即非叶子结点都只有一个孩子)
B. 高度为 4 (即结点分布在四层上)
C. 根结点的左子树为空
D. 根结点的右子树为空

- 可以构造出下图所示二叉排序树(二叉检索树、二叉查找树)的关键码序列是 (59)。



- (59) A. 10 13 17 19 23 27 31 40 65 91 B. 23 40 91 17 19 10 31 65 27 13
C. 23 19 40 27 17 13 10 91 65 31 D. 27 31 40 65 91 13 10 17 23 19

- 图 G 的邻接矩阵如下图所示(顶点依次表示为 $v_0, v_1, v_2, v_3, v_4, v_5$), G 是 (60)。对 G 进行广度优先遍历(从 v_0 开始), 可能的遍历序列为 (61)。

$$\begin{bmatrix}
 \infty & 18 & 17 & \infty & \infty & \infty \\
 \infty & \infty & \infty & 20 & 16 & \infty \\
 \infty & 19 & \infty & 23 & \infty & \infty \\
 \infty & \infty & \infty & \infty & \infty & 15 \\
 \infty & \infty & \infty & \infty & \infty & 12 \\
 \infty & \infty & \infty & \infty & \infty & \infty
 \end{bmatrix}$$

- (60) A. 无向图 B. 有向图 C. 完全图 D. 强连通图
(61) A. $v_0, v_1, v_2, v_3, v_4, v_5$ B. $v_0, v_2, v_4, v_5, v_1, v_3$
C. $v_0, v_1, v_3, v_5, v_2, v_4$ D. $v_0, v_2, v_4, v_3, v_5, v_1$

- 在一条笔直公路的一边有许多房子, 现要安装消防栓, 每个消防栓的覆盖范围远大于房子的面积, 如下图所示。现求解能覆盖所有房子的最少消防栓数和安装方案(在问题求解过程中, 可以将房子和消防栓均视为直线上的点)。

该问题求解算法的基本思路为: 从左端的第一栋房子开始, 在其右侧 m 米处安装一个消防栓, 去掉被该消防栓覆盖的所有房子。在剩余的房子中重复上述操作, 直到所有房子被覆盖。算法采用的设计策略为 (62), 对应的时间复杂度为 (63)。



假设公路起点 A 的坐标为 0, 消防栓的覆盖范围(半径)为 20 米, 10 栋房子的坐标为(10, 20,

30, 35, 60, 80, 160, 210, 260, 300), 单位为米。根据上述算法, 共需要安装__(64)__个消防栓。以下关于该求解算法的叙述中, 正确的是__(65)___。

- (62) A. 分治 B. 动态规划 C. 贪心 D. 回溯
 (63) A. $\Theta(\lg n)$ B. $\Theta(n)$ C. $\Theta(n \lg n)$ D. $\Theta(n^2)$
 (64) A. 4 B. 5 C. 6 D. 7
 (65) A. 肯定可以求得问题的一个最优解 B. 可以求得问题所有的最优解
 C. 对有些实例可能得不到最优解 D. 只能得到近似最优解

- 使用 ADSL 接入 Internet, 用户端需要安装__(66)__协议。

(66) A. PPP B. SLIP C. PPTP D. PPPoE

- 下列命令中, 不能用于诊断 DNS 故障的是__(67)___。

(67) A. netstat B. nslookup C. ping D. tracer

- 以下关于 TCP/IP 协议和层次对应关系的表示中, 正确的是__(68)___。

- (68) A.

HTTP	SNMP
TCP	UDP
IP	

 B.

FTP	Telnet
UDP	TCP
ARP	
- C.

HTTP	SMTP
TCP	UDP
IP	

 D.

SMTP	FTP
UDP	TCP
ARP	

- 把 CSS 样式表与 HTML 网页关联, 不正确的方法是__(69)___。

(69) A. 在 HTML 文档的<head>标签内定义 CSS 样式
 B. 用@import 引入样式表文件
 C. 在 HTML 文档的<!-- -->标签内定义 CSS 样式
 D. 用<link>标签链接网上可访问的 CSS 样式表文件

- 使用__(70)__命令可以释放当前主机自动获取的 IP 地址。

(70) A. ipconfig/all B. ipconfig/reload
 C. ipconfig/release D. ipconfig/reset

- The project workbook is not so much a separate document as it is a structure imposed on the documents that the project will be producing anyway.

All the documents of the project need to be part of this__(71)_. This includes objectives, external specifications, interface specifications, technical standards, internal specifications and administrative memoranda (备忘录).

Technical prose is almost immortal. If one examines the genealogy (Ff) of a customer manual for a piece of hardware or software, one can trace not only the ideas, but also many of the very sentences and

paragraphs back to the first (72) proposing the product or explaining the first design. For the technical writer, the paste-pot is as mighty as the pen.

Since this is so, and since tomorrow's product-quality manuals will grow from today's memos, it is very important to get the structure of the documentation right. The early design of the project (73) ensures that the documentation structure itself is crafted, not haphazard. Moreover, the establishment of a structure molds later writing into segments that fit into that structure.

The second reason for the project workbook is control of the distribution of (74). The problem is not to restrict information, but to ensure that relevant information gets to all the people who need it.

The first step is to number all memoranda, so that ordered lists of titles are available and a worker can see if he has what he wants. The organization of the workbook goes well beyond this to establish a tree-structure of memoranda. The (75) allows distribution lists to be maintained by subtree, if that is desirable.

- | | | | |
|-------------------|------------------|-------------------|------------------|
| (71) A. structure | B. specification | C. standard | D. objective |
| (72) A. objective | B. memoranda | C. standard | D. specification |
| (73) A. title | B. list | C. workbook | D. quality |
| (74) A. product | B. manual | C. document | D. information |
| (75) A. list | B. document | C. tree-structure | D. number |

第31小时

2018 年 11 月考试真题（下午案例分析）



【导读小贴士】

经过前面章节的学习，相信广大考生已经掌握了软件设计师考试的重点内容，我们在接下来的几个章节中为大家准备了几套历年考试真题及解析，供大家巩固所学的知识。这部分内容不只是对之前所学知识的检验，也是对其他零散知识点的查缺补漏。俗话说“得真题者得天下”，只要反复做题、认真练习，相信大家一定能取得好的成绩。

试题一（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

某房产中介连锁企业欲开发一个基于 Web 的房屋中介信息系统，以有效地管理房源和客户，提高成交率。该系统的主要功能如下：

（1）房源采集与管理。系统自动采集外部网站的潜在房源信息，并保存为潜在房源。由经纪人联系确认的潜在房源变为房源，并添加出售或出租房源的客户。由经纪人或客户登记的出售或出租房源，系统将其保存为房源。房源信息包括基本情况、配套设施、交易类型、委托方式、业主等。经纪人可以对房源进行更新等管理操作。

（2）客户管理。求租或求购客户进行注册、更新，推送客户需求给经纪人，或者由经纪人对求租或求购客户进行登记、更新。客户信息包括身份证号、姓名、手机号、需求情况、委托方式等。

（3）房源推荐。根据客户的需求情况（求购或求租需求情况和出售或出租房源信息），向已登记的客户推荐房源。

（4）交易管理。经纪人对租售客户双方进行交易信息管理，包括订单提交和取消、设置收取中介费比例。财务人员收取中介费之后，表示该订单已完成，系统更新订单状态和房源状态，向客

户和经纪人发送交易反馈。

（5）信息查询。客户根据自身需求查询房屋供需信息。

现在采用结构化方法对房屋中介信息系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和如图 1-2 所示的 0 层数据流图。

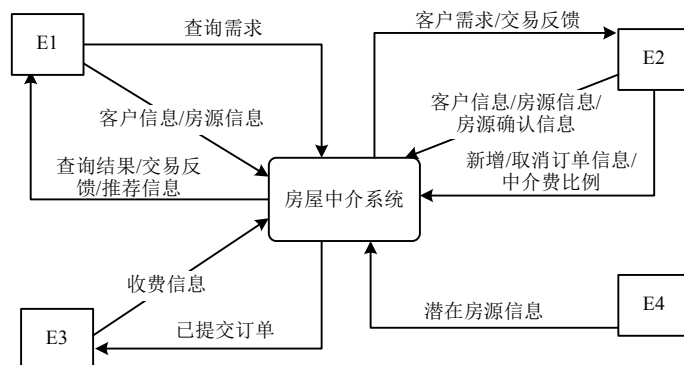


图 1-1 上下文数据流图

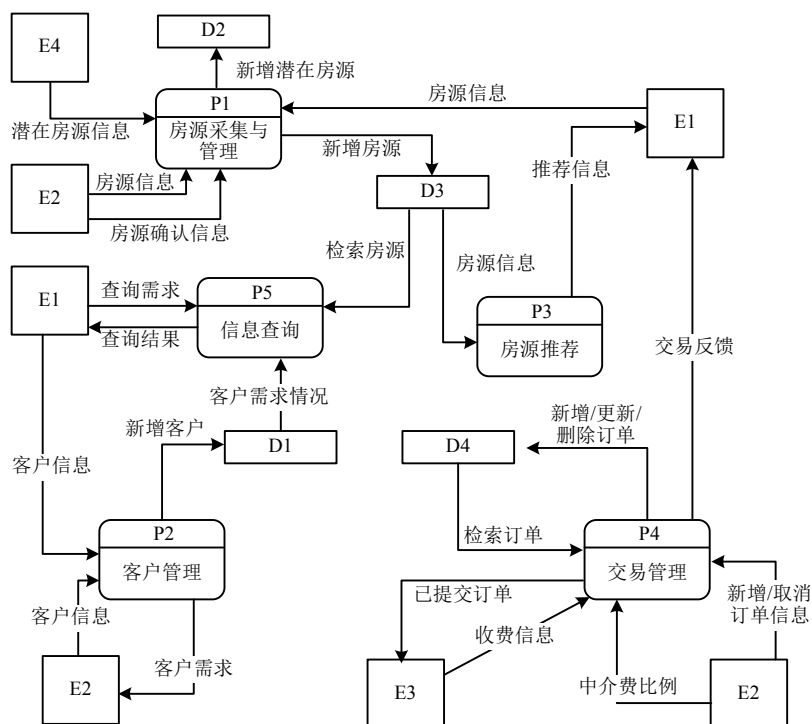


图 1-2 0 层数据流图

【问题 1】(4 分)

使用说明中的词语, 写出图 1-1 中的实体 E1~E4 的名称。

【问题 2】(4 分)

使用说明中的词语, 写出图 1-2 中的数据存储 D1~D4 的名称。

【问题 3】(3 分)

根据说明和图中术语, 补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】(4 分)

根据说明和图中术语, 写出图 1-1 中数据流的“客户信息”和“房源信息”的组成。

试题二 (15 分)

阅读下列说明, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某集团公司拥有多个分公司, 为了方便集团公司对分公司各项业务活动进行有效的管理, 集团公司决定构建一个信息系统以满足公司的业务管理需求。

【需求分析】

(1) 分公司关系需要记录的信息包括分公司编号、名称、经理、联系地址和电话。分公司编号唯一标识分公司信息中的每一个元组。每个分公司只有一名经理, 负责该分公司的管理工作。每个分公司设立仅为本分公司服务的多个业务部门, 如研发部、财务部、采购部、销售部等。

(2) 部门关系需要记录的信息包括部门号、部门名称、主管号、电话和分公司编号。部门号唯一标识部门信息中的每一个元组。每个部门只有一名主管, 负责部门的管理工作。每个部门有多名员工, 每名员工只能隶属于一个部门。

(3) 员工关系需要记录的信息包括员工号、姓名、隶属部门、岗位、电话和基本工资。其中, 员工号唯一标识员工信息中的每一个元组。岗位包括经理、主管、研发员、业务员等。

【概念模型设计】

根据需求阶段收集的信息, 设计的实体联系图和关系模式 (不完整) 如图 2-1 所示。

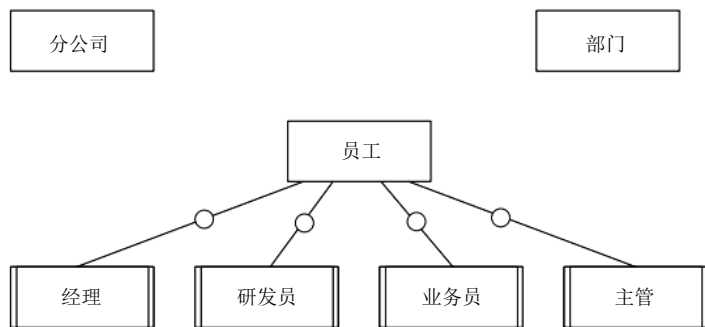


图 2-1 实体联系图

【关系模式设计】

分公司（分公司编号，名称，(a)，联系地址，电话）

部门（部门号，部门名称，(b)，电话，分公司编号）

员工（员工号，姓名，(c)，电话，基本工资）

【问题 1】（4 分）

根据问题描述，补充四个联系，完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型为 1:1、1:n 和 m:n（或 1:1、1:*和*:*）。

【问题 2】（5 分）

根据题意，将关系模式中的 (a) ~ (c) 空补充完整。

【问题 3】（4 分）

给出“部门”和“员工”关系模式的主键和外键。

【问题 4】（2 分）

假设集团公司要求系统能记录部门历任主管的任职时间和任职年限，那么，是否需要在设计数据库时增设一个实体？为什么？

试题三（15 分）

阅读下列说明和图，回答问题，将解答填入答题纸的对应栏内。

【说明】

社交网络平台（SNS）的主要功能之一是建立在线群组，群组中的成员之间可以互相分享或挖掘兴趣和活动。每个群组包括标题、管理员和成员列表等信息。

社交网络平台的用户可以自行选择加入某个群组。每个群组拥有一个主页，群组内的所有成员都可以查看主页上的内容。如果在群组的主页上发布或更新了信息，群组中的成员会自动接收到发布或更新后的信息。

用户可以加入一个群组，也可以退出这个群组。用户退出群组后，不会再接收到该群组发布或更新的任何信息。

现在采用面向对象方法对上述需求进行分析与设计，得到如表 3-1 所示的类列表和如图 3-1 所示的类图。

表 3-1 类列表

类名	描述
SNSSubject	群组主页的描述
SNSGroup	社交网络平台中的群组（在主页上发布信息）
SNSObserver	群组主页内容的关注者
SNSUser	社交网络平台的用户或群组成员
SNSAdmin	群组的管理员

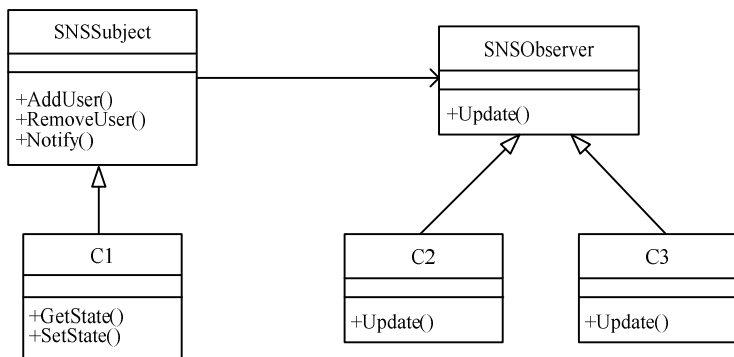


图 3-1 类图

【问题 1】(6 分)

根据说明中的描述，给出图 3-1 中 C1~C3 所对应的类名。

【问题 2】(6 分)

图 3-1 中采用了哪一种设计模式？说明该模式的意图及其适用场合。

【问题 3】(3 分)

现在对上述社交网络平台提出了新的需求：一个群体可以作为另外一个群体中的成员，如群体 A 加入群体 B，那么，群体 A 中的所有成员就自动成为群体 B 中的成员。

若要实现这个新需求，需要对图 3-1 进行哪些修改（以文字方式描述）？

试题四 (15 分)

阅读下列说明和 C 代码，回答问题，将解答写在答题纸的对应栏内。

【说明】

给定一个字符序列 $B=b_1b_2\dots b_n$ ，其中， $b_i \in \{A, C, G, U\}$ 。 B 上的二级结构是一组字符对集合 $S = \{(b_i, b_j)\}$ ，其中， $i, j \in \{1, 2, \dots, n\}$ ，并满足以下四个条件：

- (1) S 中的每对字符是 (A, U) 、 (U, A) 、 (C, G) 和 (G, C) 四种组合之一；
- (2) S 中的每对字符之间至少有四个字符将其隔开，即 $i < j - 4$ ；
- (3) S 中每一个字符（记为 b_k ）的配对存在两种情况： b_k 不参与任何配对； b_k 和字符 b_t 配对，其中， $t < k - 4$ ；
- (4)（不交叉原则）若 (b_i, b_j) 和 (b_k, b_l) 是 S 中的两个字符对且 $i < k$ ，则 $i < k < j < l$ 不成立。

B 的具有最大可能字符对数的二级结构 S 被称为最优配对方案，求解最优配对方案中的字符对数的方法如下：

假设用 $C(i, j)$ 表示字符序列 $b_i b_{i+1} \dots b_j$ 的最优配对方案（即二级结构 S ）中的字符对数，则 $C(i, j)$ 可以递归定义为：

$$C(i, j) = \begin{cases} \max(C(i, j-1), \max(C(i, t-1)+1+C(t+1, j-1))) & \text{若 } b_i \text{ 和 } b_j \text{ 匹配且 } i < j-4 \\ 0 & \text{否则} \end{cases}$$

下面的代码是算法的 C 语言实现, 其中:

n: 字符序列长度

B[]: 字符序列

C[][]: 最优配对数量数组

【C 代码】

```
#include<stdio.h>
#include<stdlib.h>
#define LEN 100

/*判断两个字符是否配对*/
int isMatch(char a,char b){
    if((a=='A'&& b=='U')||(a=='U' && b=='A'))
        return 1;
    if((a=='C'&& b=='G')||(a=='G' && b=='C'))
        return 1;
    return 0;
}

/*求最大配对数*/
int RNA_2(char B[LEN], int n){
    int i, j, k, t;
    int max;
    int C[LEN][LEN]={0};

    for(k =5; k<=n-1; k++){
        for(i =1; i<=n-k; i++){
            j = i+k;
            (1);
            for((2); t<=j-4; t++){
                if((3) && max < C[i][t-1]+1+C[t+1][j-1])
                    max = C[i][t-1]+1+C[t+1][j-1];
            }
            C[i][j] = max;
            printf("c[%d][%d] = %d--", i, j, C[i][j]);
        }
    }
    return (4);
}
```

【问题 1】(8 分)

根据题干说明, 填充 C 代码中的 (1) ~ (4)。

【问题 2】(4 分)

根据题干说明和 C 代码, 算法采用的设计策略为 (5)。

算法的时间复杂度为__(6)__ (用 O 表示)。

【问题 3】(3 分)

给定字符序列 ACCGGUAGU, 根据上述算法求得最大字符对数为__(7)__。

试题五 (15 分)

阅读下列说明和 C++ 代码, 回答问题, 将解答填入答题纸的对应栏内。

【说明】

某航空公司的会员积分系统将其会员划分为: 普卡 (Basic)、银卡 (Silver) 和金卡 (Gold) 三个等级。非会员 (NonMember) 可以申请成为普卡会员。会员的等级根据其一年内累积的里程数进行调整。描述会员等级调整的状态图如图 5-1 所示。现在采用状态 (State) 模式实现上述场景, 得到如图 5-2 所示的类图。

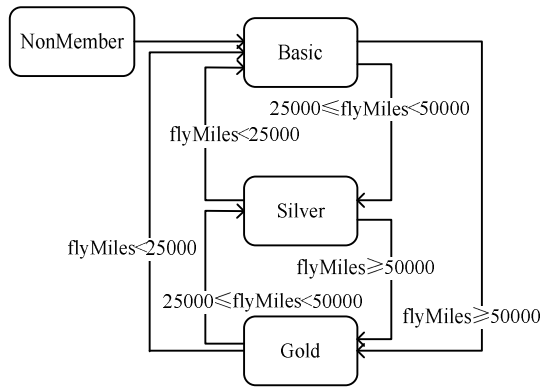


图 5-1 会员等级调整状态图

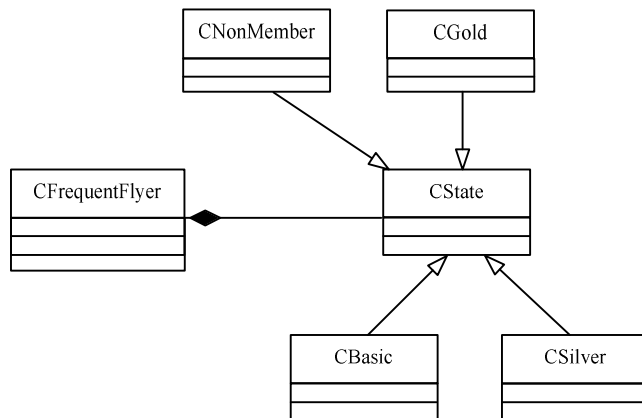


图 5-2 状态模式类图

【C++代码】

```

#include<iostream>
using namespace std;
class FrequentFlyer; class CBasic; class CSilver; class CGold; class CNoCustomer; //提前引用
class CState{
private: int flyMiles; //里程数
public:
    _____(1)_____; //根据累积里程数调整会员等级
};
class FrequentFlyer{
friend class CBasic; friend class CSilver; friend class CGold;
public:
    CState*state; CState*nocustomer; CState*basic; CState*silver; CState*gold;
    double flyMiles;
public:
    FrequentFlyer(){ flyMiles = 0; setState(nocustomer); }
    void setState(CState*state){ this->state = state;}
    void travel(int miles) {
        double bonusMiles = state-> travel(miles, this);
        flyMiles = flyMiles + bonusMiles;
    }
};
class CNoCustomer: public CState { //非会员
public:
    double travel(int miles, FrequentFlyer*context) {
        cout<<"Your travel will not account for points\n"; //不累积里程数
        return miles;
    }
};
class CBasic: public CState { //普卡会员
public:
    double travel(int miles, FrequentFlyer*context){
        if(context->flyMiles>=25000 && context->flyMiles<50000)
            _____(2)_____;
        if(context->flyMiles<25000)
            _____(3)_____;
        return miles + 0.5*miles; //累积里程数
    }
};
class CGold: public CState { //金卡会员
public:
    double travel(int miles, FrequentFlyer*context){
        if(context->flyMiles>=25000 && context->flyMiles<50000)
            _____(4)_____;
        if(context->flyMiles<25000)
            _____(5)_____;
        return miles + 0.5*miles; //累积里程数
    }
};
class CSilver: public CState { //银卡会员
public:
    double travel(int miles, FrequentFlyer*context){

```

```

if(context->flyMiles<25000)
    context->setState(context->basic);
if(context->flyMiles>=50000)
    context->setState(context->gold);
return (miles + 0.25*miles);    //累积里程数
}
};

```

【问题】（15 分）

阅读上述说明和 C++代码，将应填入的字句写在答题纸的对应栏内。

试题六（15 分）

阅读下列说明和 Java 代码，回答问题，将解答填入答题纸的对应栏内。

【说明】

某航空公司的会员积分系统将其会员划分为：普卡（Basic）、银卡（Silver）和金卡（Gold）三个等级。非会员（NonMember）可以申请成为普卡会员。会员的等级根据其一年内累积的里程数进行调整。描述会员等级调整的状态图如图 6-1 所示。现在采用状态（State）模式实现上述场景，得到如图 6-2 所示的类图。

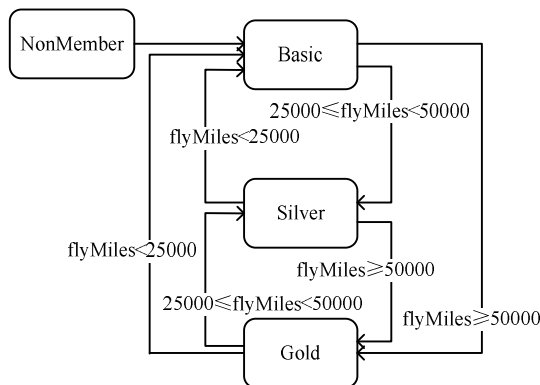


图 6-1 会员等级调整状态图

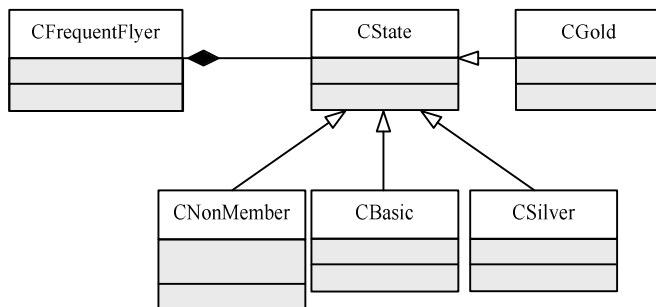


图 6-2 状态模式类图

【Java 代码】

```
import java.util.*;

abstract class CState {
    public int flyMiles;           //里程数
    public ____ (1) ____;         //根据累积里程数调整会员等级
}

class CNoCustomer extends CState { //非会员
    public double travel(int miles, FrequentFlyer context){
        System.out.println("Your travel will not account for points");
        return miles;             //不累积里程数
    }
}

class CBasic extends CState { //普卡会员
    public double travel(int miles, FrequentFlyer context){
        if(context.flyMiles>= 25000&&context.flyMiles<50000)
            ____ (2) ____;
        if(context.flyMiles >= 50000)
            ____ (3) ____;
        return miles;
    }
}

class CGold extends CState{ //金卡会员
    public double travel(int miles, FrequentFlyer context){
        if(context.flyMiles>=25000&&context.flyMiles<50000)
            ____ (4) ____;
        if(context.flyMiles<25000)
            ____ (5) ____;
        Return miles+0.5*miles;    //累积里程数
    }
}

class CSilver extends CState{ //银卡会员
    public double travel(int miles, FrequentFlyer context){
        if(context.flyMiles<=25000)
            context.setState(new CBasic());
        if(context.flyMiles>50000)
            context.setState(new CGold());
        return(miles+0.25*miles);  //累积里程数
    }
}

class FrequentFlyer{
    CState state;
    double flyMiles;
    public FrequentFlyer(){
        state = new CNoCustomer();
    }
}
```

```
        flyMiles = 0;
        setState(state);
    }
    public void setState(CState state){ this.state = state; }
    public void travel(int miles)
    {
        double bonusMiles = state.travel(miles, this);
        flyMiles = flyMiles + bonusMiles;
    }
}
```

【问题】（15 分）

阅读上述说明和 Java 代码，将应填入的字句写在答题纸的对应栏内。

第32小时

2018 年 11 月真题解析

上午基础知识

(1) 试题分析:

程序计数器 (PC) 是用于存放下一条指令所在单元地址的地方。

指令寄存器是临时放置从内存里面取得的程序指令的寄存器,用于存放当前从主存储器读出的正在执行的一条指令。

地址寄存器用来保存当前 CPU 所访问的内存单元的地址。

指令译码器是从内存中取出一条指令经数据总线送往指令寄存器中。

答案: (1) B

(2) 试题分析: BIOS 是一个程序, 固化在主板的 BIOS ROM 芯片里面, 它相当于硬件底层的一个操作系统, 控制和保存着硬件的输入及输出工作。

答案: (2) A

(3) 试题分析: 对于有 n 位的整数补码, 其取值范围为 $-(2^{n-1}) \sim (2^{n-1})-1$ 。以八位的整数补码举例, 其有效取值范围为 $-(2^7) \sim (2^7)-1$, 也就是 $-128 \sim 127$ 。C 选项的计算值为 128, 越界了; 只有 D 选项符合。其他选项都越界了。

答案: (3) D

(4) 试题分析:

串行系统的可靠性公式为: $R=R_1 \times R_2 \times \cdots \times R_n$

并联系统的可靠性公式为: $R=1-(1-R_1) \times (1-R_2) \times \cdots \times (1-R_n)$

答案: (4) C

(5) 试题分析: 如果以二进制数据中 1 的个数是奇数为依据, 则是奇校验; 如果以二进制数

据中 1 的个数是偶数为依据,则是偶校验。如果传输过程中包括校验位在内的奇数个数据位发生改变,那么,奇校验位将出错表示传输过程有错误发生,但是由于没有办法确定哪一位出错,所以它不能进行错误校正。

答案: (5) C

(6) 试题分析:流水线技术是指在程序执行时,多条指令重叠进行操作的一种任务分解技术。把一个任务分解为若干顺序执行的子任务,不同的子任务由不同的执行机构来负责执行,而这些执行机构可以同时并行工作。

答案: (6) A

(7) 试题分析:共享密钥加密又称对称加密算法,既用同一个密钥去加密和解密数据。常用的算法包括 DES、3DES、IDEA、AES 等。

答案: (7) B

(8) 试题分析:计算机病毒具有隐蔽性、传染性、潜伏性、触发性和破坏性等特性。因此,不包括 D 选项的自毁性。

答案: (8) D

(9)、(10) 试题分析:常见的摘要算法有 MD5、SHA 等。MD5 是第五版信息摘要算法,输入以 512 位为分组进行处理,产生一个 128 位的输出。

答案: (9) B (10) B

(11) 试题分析:谈到收发这两个关键字,使用 Web 方式收发邮件是必须设置账号密码进行登录的。如果只是简单地查看,可以不设置账号密码。

答案: (11) B

(12) 试题分析:根据我国《著作权法》第二十条的规定,作者的署名权、修改权、保护作品完整权的保护期不受限制。但是商标权可以每 10 年无限续期,所以也存在无限期拥有。但是题目中提到的是“有可能”,所以应该选择 C 选项,不然 A 选项就不用说有可能了,是必须无限期了。

答案: (12) C

(13) 试题分析:我国保护计算机软件著作权的两个基本法律文件是《中华人民共和国著作权法》和《计算机软件保护条例》。

答案: (13) B

(14) 试题分析:该题目指明了第一个软件公司为软件著作权人,所以再把略作修改的软件交给第二家公司,就构成了侵权行为。

答案: (14) D

(15) 试题分析:结构化方法的分析结果由以下几部分组成:一套分层的数据流图、一本数据词典、一组小说明(也称加工逻辑说明)、补充材料。因此,不包括 D 选项。

答案: (15) D

(16) 试题分析:数据流图的成分包括数据存储、数据流、加工、外部实体。其中,数据流、加工、数据存储用于系统内部数据处理的模型,而外部实体是指系统以外,又和系统有联系的人、

事物或软件系统。所以，应该选择 A 选项。

答案：(16) A

(17)、(18) 试题分析：从开始节点到结束节点，持续时间最长的路线（也叫关键路线）决定着项目的最短工期，也就是完成项目需要花费的最少时间。因此，ADFHJ 为项目活动图的关键路径，即等于 46 天。FG 的松弛时间为 18 天。46-7-3 和 46-10-18 相比，差值为 18，所以此选项为 18。

答案：(17) D (18) C

(19) 试题分析：一般而言，风险与不确定性有关，若某一事件的发生存在着两种或两种以上的可能性，即可认为该事件存在风险。但是 B 选项已经确定客户不清楚需求，所以是确定的事情，不存在风险。

答案：(19) B

(20) 试题分析：此题考查的是短路问题。or 代表“或”的意思，and 代表“且”的意思。所以，a 为 true 时后面的就可以不用执行下去直接判定为真，从而形成短路效应。

答案：(20) B

(21) 试题分析：此题考查的是二叉树排序问题。根据图示可知，最后计算的符号为“-”，其次为“*”，最后为“+”，所以答案为 C 选项。

答案：(21) C

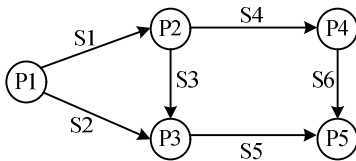
(22) 试题分析：B 选项和 D 选项为同数据类型；A 选项是将 float 数据类型转换成 int 数据类型；C 选项是将 int 数据类型和 float 数据类型进行相加，实现了将 int 数据类型转换成 float 数据类型。

答案：(22) C

(23) 试题分析：如果是每个进程都需要 3 个资源，按照给每个进行资源最大分配，会是 3、3、2 这种情况。那么，有两个进程资源就够了。最后的进程资源为 8 个，如果 i 是 4 的话，3、3、2 这种必是死锁。

答案：(23) D

(24)、(25)、(26) 试题分析：标明信号量的前趋图如下所示，可知答案依次为 D、B、C。



答案：(24) D (25) B (26) C

(27) 试题分析： $2053/32 = 64.156$ ，并且位示图是从 0 开始的，所以选择 C 选项。

答案：(27) C

(28) 试题分析： $1\text{KB}/4\text{b} = 256$ ，所以直接地址索引为 0~4 号，一级地址索引为 $256 \times 2 + 4 = 516$ 号，二级地址索引为 $256 \times 256 + 516 = 517 \sim 66052$ 号。

答案：(28) B

(29) 试题分析：增量模型在各个阶段并不交付一个可运行的完整产品，而是交付满足客户需求的一个子集的可运行产品。整个产品被分解成若干个构件，开发人员逐个构件地交付产品。这样做的好处是软件开发可以较好地适应变化，客户可以不断地看到所开发的软件，从而降低开发风险。

答案：(29) C

(30) 试题分析：

CL0（未完成的）：未执行或未达到 CL1 定义的所有目标。

CL1（已执行的）：共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品，以实现支持过程域的特定目标。

CL2（已管理的）：共性目标集中于已管理的过程的制度化。

CL3（已定义级的）：共性目标集中于以定义过程的制度化。

CL4（定量管理的）：共性目标集中于可定量管理的过程的制度化。

CL5（优化的）：使用量化（统计学）手段改变和优化过程域，以对付客户要求的可持续改进计划中的过程域的功效。

答案：(30) D

(31) 试题分析：软件质量特性如下表所示。

功能性	可靠性	易用性	效率	维护性	可移植性
适合性	容错性	易学性	时间特性	可测试性	适应性
准确性	易恢复性	易理解性	资源特性	可修改性	易安装性
互用性	成熟性	易操作性		稳定性	一致性
依从性				易分析性	可替换性
安全性					

答案：(31) A

(32) 试题分析：模块化设计的原则应该是尽量考虑高内聚、低耦合。

1) 模块独立。模块的独立性原则表现在模块完成独立的功能，与其他模块的联系应该尽可能简单，各个模块具有相对的独立性。

2) 模块的规模要适当。模块的规模不能太大，也不能太小。如果模块的功能太强，可读性就会较差；如果模块的功能太弱，就会有有很多的接口。读者需要通过较多的程序设计来进行经验的积累。

3) 分解模块时要注意层次。在进行多层次任务分解时，要注意对问题进行抽象化。在分解初期，可以只考虑大的模块，在中期再逐步进行细化，分解成较小的模块进行设计。

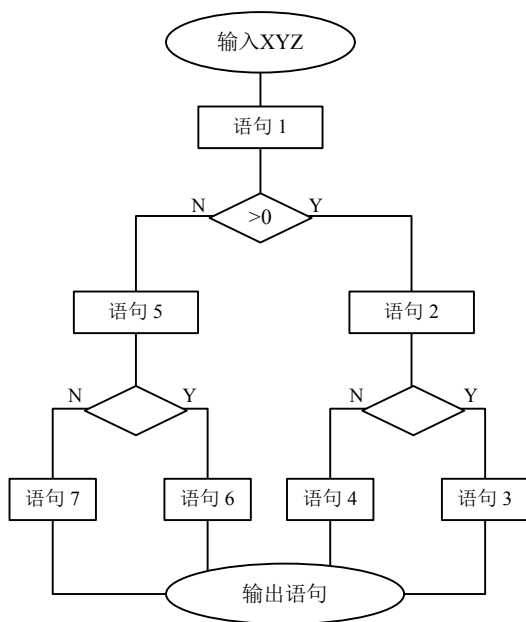
答案：(32) B

(33) 试题分析：数据耦合是指两个模块之间有调用关系，传递的是简单的数据值，相当于高级语言的值传递；标记耦合是指两个模块之间传递的是数据结构，如高级语言的数组名、记录名、文件名等这些名字即为标记，其实传递的是这个数据结构的地址；控制耦合是指一个模块调用另一个模块时，传递的是控制变量（如开关、标志等），被调用的模块通过该控制变量的值有选择地执

行块内的某一功能。这种耦合对系统的影响较大，它影响接收控制流模块的内部运行。

答案：(33) A

(34)、(35) 试题分析：根据题意可以画出流程图如下图所示。



由流程图可知，4 个测试用例即可覆盖全部路径。环路复杂度由公式可知， $V(G)=m-n+2$ ，得到 $14-12+2=4$ 。

答案：(34) B (35) C

(36) 试题分析：

1) 更正性。由于系统测试不可能揭露系统存在的所有错误，因此在系统投入运行后的频繁的实际应用过程中，就有可能暴露出系统内隐藏的错误。

2) 适应性维护。适应性维护是为了使系统适应环境的变化而进行的维护工作。

3) 完善性维护。在系统的使用过程中，用户往往要求扩充原有系统的功能，增加一些在软件需求规范书中没有规定的功能与性能特征，以及对处理效率和编写程序的改进。

4) 预防性维护。系统维护工作不应总是被动地等待用户提出要求后才进行，应进行主动的预防性维护，即选择那些还有较长使用寿命，目前尚能正常运行，但可能将要发生变化或调整的系统进行维护，目的是通过预防性维护为未来的修改与调整奠定更好的基础。

答案：(36) B

(37) 试题分析：本题考查的是继承的定义。继承是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上进行。因此，本题选择 A 选项。

答案：(37) A

(38) 试题分析:

- 1) 参数多态: 应用广泛、最纯的多态。
- 2) 包含多态: 同样的操作可以用于一个类型及其子类型。包含多态一般需要进行运行时的类型检查。
- 3) 过载多态: 同一个名(操作符、函数名)在不同的上下文中有不同的类型。
- 4) 强制多态: 编译程序通过语义操作, 把操作对象的类型强行加以变换, 以符合函数或操作符的要求。

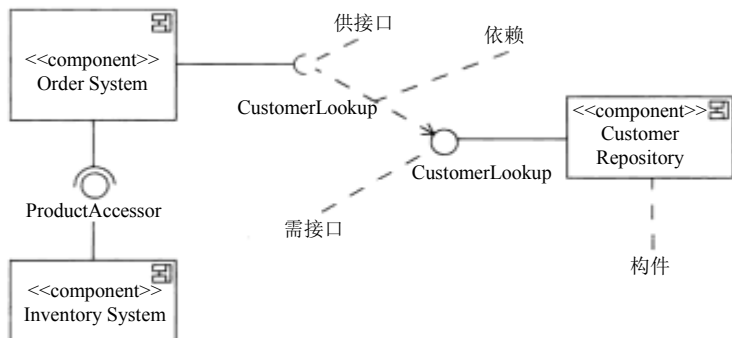
答案: (38) C

(39)、(40) 试题分析: 类可以分为三种: 实体类、接口类(边界类)和控制类。

- 1) 实体类的对象表示现实世界中真实的实体, 如人、物等。
- 2) 接口类(边界类)的对象为用户提供的一种与系统合作交互的方式, 分为人和系统两大类。其中, 人的接口可以是显示屏、窗口、Web 窗体、对话框、菜单、列表框、其他显示控制、条形码、二维码或用户与系统交互的其他方法; 系统接口涉及把数据发送到其他系统, 或者从其他系统接收数据。
- 3) 控制类的对象用来控制活动流, 充当协调者。

答案: (39) B (40) A

(41)、(42)、(43) 试题分析: 末端有一个完整的圆周的接口符号代表组件提供接口; 末端只有半个圆的接口符号代表组件需求接口, 如下图所示。



答案: (41) B (42) C (43) C

(44)、(45)、(46)、(47) 试题分析: 访问者模式包括以下角色: Visitor, 抽象访问者; Concrete Visitor, 具体访问者; Element, 抽象元素; Concrete Element, 具体元素; Object Structure, 对象结构。

答案: (44) B (45) A (46) D (47) C

(48) 试题分析:

- 1) 词法分析阶段。识别出一个个的单词, 删掉无用的信息, 报告分析时的错误。
- 2) 语法分析阶段。语法分析器以单词符号作为输入, 分析单词符号是否形成符合语法规则的语法单位, 如表达式、赋值、循环等, 按语法规则分析检查每条语句是否有正确的逻辑结构。语法

分析构造了一棵语法树。

3) 语义分析阶段。主要检查源程序是否存在语义错误,并收集类型信息供后面的代码生成阶段使用,如赋值语句的右端和左端的类型不匹配、表达式的除数是否为 0 等。

答案:(48) B

(49) 试题分析:由答案可知以 0 开始、以 0 结尾,所以只有 A 选项符合。

答案:(49) A

(50) 试题分析:代入可知, $m=2 \times 5=10$, $x=10-1=9$, 返回 $x+m=19$, $19+9=28$ 。

答案:(50) D

(51) 试题分析:本题考查的是数据库的体系结构:三层模式。对于题干给出的视图、存储文件、基本表分别对应外模式、内模式、模式。因此,本题选择 D 选项。

答案:(51) D

(52) 试题分析:本题考查的是分布式数据库的相关知识。

1) 逻辑透明性(局部映像透明性)。它是最低层次的透明性,该透明性提供数据到局部数据库的映像,即用户不必关心局部 DBMS 支持哪种数据模型、使用哪种数据操纵语言,数据模型和数据操纵语言的转换是由系统完成的。因此,局部映像透明性对异构型和同构异质的分布式数据库系统是非常重要的。

2) 位置透明性。用户不必知道所操作的数据放在何处,即数据分配到哪个或哪些站点存储对用户是透明的。因此,数据分片模式的改变,如把数据从一个站点转移到另一个站点将不会影响应用程序,因而应用程序不必改写。

3) 分片透明性。用户不必关心数据是如何分片的,他们对数据的操作是在全局关系上进行的,即关心如何分片对用户是透明的,因此,当分片改变时应用程序可以不变。

4) 复制透明性。用户不用关心数据库在网络中各个节点的复制情况,被复制的数据的更新都由系统自动完成。

答案:(52) C

(53)、(54) 试题分析:把每个选项代入求它的闭包, A_2A_5 能推出所有属性的主键。由此可知, R 的一个主键为 A_2A_5 , 由函数依赖集 F 可知, 存在 $A_2A_5 \rightarrow A_6$, $A_5A_6 \rightarrow A_1$, $A_1 \rightarrow A_3$, 这里存在传递函数依赖, 故 A、B 选项均不正确; C 选项本身不正确, 存在非主属性对候选键的传递函数依赖是不满足 3NF 的; 因此, 本题选择 D 选项。

答案:(53) B (54) D

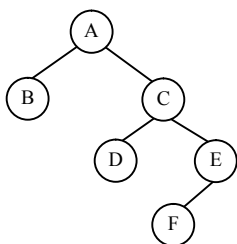
(55)、(56) 试题分析:对于自然连接,只要把重复的列去掉即可,所以列数为 5;与表达式等价的为 D 选项。

答案:(55) B (56) D

(57) 试题分析:本题用单链表作为栈的存储结构,因为栈的操作是先进后出,因此无论是入栈还是出栈,都只对栈顶元素进行操作,而在单链表中用头指针作为栈顶指针,此时无论是出栈还是入栈,都只需要对头指针指向的栈顶指针进行操作即可,不需要遍历链表。

答案：(57) A

(58) 试题分析：根据先序遍历和中序遍历序列，可以构造出如下图所示的二叉树，可知高度为 4，所以选择 B 选项。

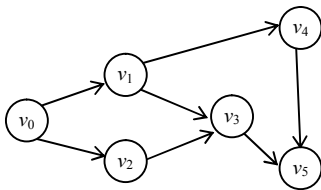


答案：(58) B

(59) 试题分析：根据排序二叉树的构造过程，可知 A 选项的根节点为 10，D 选项的根节点为 27，因此可以排除。对于 C 选项，构造根节点的子结点可知 19 为其左孩子结点，与图不符。本题只有 B 选项可以构造出如图所示的排序二叉树。

答案：(59) B

(60)、(61) 试题分析：由题可以作序列图如下图所示，由图可知，G 是有向图。对 G 进行广度优先遍历，可能的遍历序列为 v_0 、 v_1 、 v_2 、 v_3 、 v_4 、 v_5 。



答案：(60) B (61) A

(62)、(63)、(64)、(65) 试题分析：本题使用的是分治法。

1) 分治法特征：对于一个规模为 n 的问题，若该问题可以被容易地解决（如规模 n 较小），则直接解决；否则将其分解为 k 个规模较小的子问题。这些子问题互相独立且与原问题形式相同，递归地解决这些子问题，然后将各个子问题的解合并得到原问题的解。

2) 动态规划法：在求解问题的过程中，对于每一步决策，列出各种可能的局部解，再依据某种判定条件，舍弃那些肯定不能得到最优解的局部解，在每一步都经过筛选，以每一步都是最优解来保证全局是最优解。本题情景没有列出所有的可能解进行筛选，因此，本题不属于动态规划法。

3) 回溯法：回溯法是一种选优搜索法，按选优条件向前搜索以达到目标。但当搜索到某一步时，发现原先的选择并不优或达不到目标，就退回一步重新选择。这种走不通就退回再走的方法就是回溯法。本题情景没有探索和回退的过程，因此，本题不属于回溯法。

4) 贪心法：总是做出在当前来说最好的选择，而不从整体上加以考虑，故所做的每步的选择只是当前步骤的局部最优选择，但从整体来说不一定是最优的选择。由于它不必为了寻找最优解而穷尽所有的可能解，因此其耗费时间少，一般可以快速得到满意的解，但得不到最优解。本题情景

没有给出每步选择的局部最优判断条件, 因此, 本题不属于贪心法。

舍弃已被覆盖的房子, 可以将问题的规模逐步缩小, 形成规模较小的子问题, 而这些问题的求解与原问题的求解过程相同, 因此本题属于分治法的算法思想。

由于本题的算法过程是依次与各个房子进行判断, 当所有房子都被比较之后, 则问题结束。因此, 时间复杂度与房子的个数相关, 本题的时间复杂度应该趋于现象, 即为 $O(n)$ 。

关于对应序列(10, 20, 30, 35, 60, 80, 160, 210, 260, 300), 有如下放置方法:

1) 第一轮放置: 在第一栋房子 $x=10$ 的右侧 20 米处安装一个消防栓, 可以覆盖 10、20、30、35 这四栋房子。

2) 第二轮放置: 去掉前四栋房子, 在第五栋房子 $x=60$ 的右侧 20 米处安装一个消防栓, 可以覆盖 60 和 80 这两栋房子。

3) 第三轮放置: 去掉前面已覆盖的房子, 在第七栋房子 $x=160$ 的右侧 20 米处安装一个消防栓, 只能覆盖 160 这一栋房子。

4) 第四轮放置: 去掉前面已覆盖的房子, 在第八栋房子 $x=210$ 的右侧 20 米处安装一个消防栓, 可以覆盖 210 这一栋房子。

5) 第五轮放置: 去掉前面已覆盖的房子, 在第九栋房子 $x=260$ 的右侧 20 米处安装一个消防栓, 可以覆盖 260 和 300 这两栋房子。

房子全部覆盖完毕, 因此, 共需安装 5 个消防栓。

对于得到一个最优解是动态规划的特点, 可以得到问题所有的最优解是回溯法的特征, 因此可以排除 A、B 选项; 分治法对有些实例可能得不到最优解, 因此选择 C 选项。

答案: (62) A (63) B (64) B (65) C

(66) 试题分析: PPPoE (Point-to-Point Protocol over Ethernet) 是以太网 (Ethernet) 上的点对点协议 (PPP), 是将点对点协议封装在以太网框架中的一种网络隧道协议, 提供用户身份验证、用户管理和数据加密等功能。

答案: (66) D

(67) 试题分析:

1) netstat 是控制台命令, 是一个监控 TCP/IP 网络非常有用的工具, 它可以显示路由表、实际的网络连接和每一个网络接口设备的状态信息。

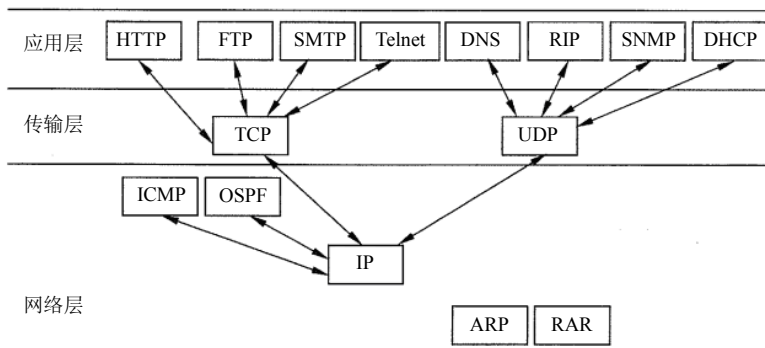
2) nslookup 可以指定查询的类型, 可以查到 DNS 记录的生存时间, 还可以指定使用哪个 DNS 服务器进行解释。在已安装 TCP/IP 协议的计算机上, 均可以使用这个命令。

3) ping 命令可以检查网络是否连通, 可以很好地帮助分析和判定网络故障。

4) tracert (跟踪路由) 是路由跟踪的实用程序, 用于确定 IP 数据包访问目标所采取的路径。tracert 命令使用 IP 生存时间 (TTL) 字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由。

答案: (67) A

(68) 试题分析: TCP/IP 协议和层次对应关系如下表所示。



答案: (68) A

(69) 试题分析: `<!-- -->` 是 HTML 注释的表示方式, 在这里定义 CSS 样式无效。

答案: (69) C

(70) 试题分析:

`/?`: 显示帮助信息。

`/all`: 显示完整配置信息。

`/release`: 释放指定适配器的 IPv4 地址。

`/release6`: 释放指定适配器的 IPv6 地址。

`/renew`: 更新指定适配器的 IPv4 地址。

`/renew6`: 更新指定适配器的 IPv6 地址。

`/flushdns`: 清除 DNS 解析程序缓存。

`/registerdns`: 刷新所有 DHCP 租用并重新注册 DNS 名称。

`/displaydns`: 显示 DNS 解析程序缓存的内容。

`/showclassid`: 显示适配器允许的所有 DHCP 类 ID。

`/setclassid`: 修改 DHCP 类 ID。

`/showclassid6`: 显示适配器允许的所有 IPv6 DHCP 类 ID。

`/setclassid6`: 修改 IPv6 DHCP 类 ID。

答案: (70) C

(71) ~ (75) 试题分析:

翻译:

项目工作手册与其说是一个独立的文档, 倒不如说它是一种加在文档上的 (71), 项目会按照项目工作手册来实施。

项目所有的文档都是项目工作手册的一部分, 其中包括目标、外部说明、接口说明、技术标准、内部说明和管理备忘录。

技术文档是其中至关重要的部分。如果客户通过用户手册去搜寻硬件或软件的信息, 他不止能找到简单的描述, 还能找到关于项目计划与初步设计方面的 (72)。对于技术文档工程师来说,

糊锅跟笔一样重要。

鉴于项目质量手册出自于这些备忘录，那么建立正确的文档结构尤为重要。早期的项目 (73) 可以确保文档的结构被精心制作而不是被粗制滥造。更进一步说，文档结构的建立也给后续的书写工作提供了一个模板。

再者，项目工作手册也能控制 (74) 的分发，它不是严格限制分发，而是把信息传递给需要它的人。

第一步要给所有备忘录进行编号，形成一个列表，工作手册再把这些备忘录建立成一个树形结构。如有必要，则在此 (75) 基础上形成分发列表的子结构。

- | | | | |
|------------|--------|---------|-------|
| (71) A. 结构 | B. 说明 | C. 标准 | D. 目标 |
| (72) A. 目标 | B. 备忘录 | C. 标准 | D. 说明 |
| (73) A. 标题 | B. 列表 | C. 工作手册 | D. 质量 |
| (74) A. 产品 | B. 手册 | C. 文档 | D. 信息 |
| (75) A. 列表 | B. 文档 | C. 树形结构 | D. 号码 |

答案：(71) A (72) B (73) C (74) D (75) C

下午案例分析

试题一 (15 分)

【问题 1】(4 分)

答案：E1：客户；E2：经纪人；E3：财务人员；E4：外部网站。

【问题 2】(4 分)

答案：D1：客户信息表或客户记录表；D2：潜在房源表；D3：房源表；D4：订单记录表。

【问题 3】(3 分)

答案：

数据流名称：交易反馈信息；起点：P4（交易管理）；终点：E2（经纪人）。

数据流名称：更新房源状态；起点：P4（交易管理）；终点：D3（房源表）。

数据流名称：客户信息；起点：D1（客户信息表）；终点：P3（房源推荐）。

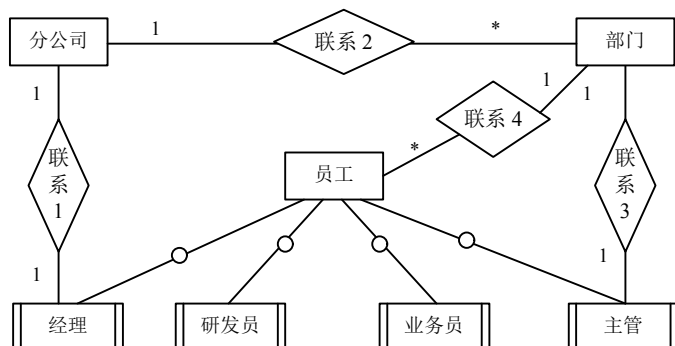
【问题 4】(4 分)

答案：房源信息包括基本情况、配套设施、交易类型、委托方式、主业等。客户信息包括身份证号、姓名、手机号、需求情况、委托方式等。

试题二 (15 分)

【问题 1】(4 分)

答案：如下图所示。



【问题 2】(5 分)

答案: (a) 经理; (b) 主管号, 分公司编号; (c) 隶属部门, 岗位。

【问题 3】(4 分)

答案:

部门主键: 部门号; 外键: 分公司编号, 主管号。

员工主键: 员工号; 外键: 隶属部门。

【问题 4】(2 分)

答案: 不需要增加实体。因为它可以直接归属于联系当中, 它的联系可以直接写成关系模式, 所以不需要增加实体。

试题三 (15 分)

【问题 1】(6 分)

答案:

C1: SNSGroup

C2: SNSUser

C3: SNSAdmin (C2 可以和 C3 互换位置)

【问题 2】(6 分)

答案: 采用了观察者模式。该模式的意图是当被观察者发生改变时, 会给观察者发送消息通知, 让它们随之发生改变。适用场合一般适用于一个被观察者改变时观察者也随之改变的场合。

【问题 3】(3 分)

答案: 可以在 SNSSubject 下面增加一个被观察者对象, 然后它可以在观察者对象这里增加一个加入另外群体的方法, 以实现接收被观察者发送的通知。

试题四 (15 分)

【问题 1】(8 分)

答案:

- (1) $\max = C[i][j-1]$
- (2) $t = 0$
- (3) $\text{isMatch}(B[t], B[j])$
- (4) $C[i][j-1]$

【问题 2】(4 分)

答案:

- (5) 动态规划
- (6) $O(n^3)$

【问题 3】(3 分)

答案: (7) 2

试题五 (15 分)

答案:

注意: 原版的题目应该是 CBasic、CSilver。

- (1) `virtual double travel(int miles, FrequentFlyer*context)=0`
- (2) `context->setState(context->CBasic)`
- (3) `context->setState(context->CSilver)`
- (4) `context->setState(context->CBasic)`
- (5) `context->setState(context->CSilver)`

试题六 (15 分)

答案:

- (1) `abstract double travel(int miles, FrequentFlyer*context)`
- (2) `context.setState(new CSilver())`
- (3) `context.setState(new CGold())`
- (4) `context.setState(new CSilver())`
- (5) `context.setState(new CBasic())`