

2013年
无纸化考试专用

新思路
新思路教育
NEW IDEAS FOR EDUCATION

全国计算机等级考试一本通关

考点详解 真题研究 拓展训练 考试题库 学习软件 模拟考场

六合一

二级公共基础知识与 Access数据库程序设计



新思路教育科技研究中心◎编

6大模块：
全能学习，一本通关

以考点为线索，囊括考点详解、真题研究、拓展训练、考试题库、学习软件、模拟考场6大模块

61个考点：
高效掌握常考、必考点

让考生知道考什么、怎么考。相当于老师“画重点”，直击重点难点，快速提高学习、备考的效率

1200道试题：
巩固考点，以练促学

真题+专项训练题+模拟题，通过试题的训练和研究，将考点的理论学习快速转换成实际的“考试能力”

50套题库：
日常练习，模拟考试

源自上机考试模拟软件，逼真模拟真实考试环境，具备自动评分、试题评析、学习笔记、错题复习等多项功能

机械工业出版社
CHINA MACHINE PRESS



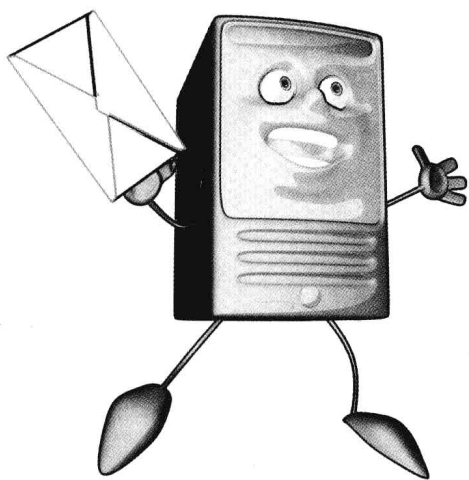
日常练习 / 模拟考场
附赠 无纸化考试系统

2013年
无纸化考试专用

全国计算机等级考试一本通关

考点详解 真题研究 拓展训练 考试题库 学习软件 模拟考场

六合一



二级公共基础知识与 Access数据库程序设计

新思路教育科技研究中心◎编

藏书章

为了帮助考生顺利通过计算机等级考试,我们在多年的考试辅导经验基础上,精心策划编写了本书。

本书根据最新的考试大纲要求,依据5年来历年试题的研究成果和对命题规律的把握,结合历年的真题讲解各个考点。真切、实用;学习不跑题、切中要点!另外,本书使用“画书法”,画出重点,点明必考内容,讲解重基础,易看懂。

本书既可以作为计算机等级考试考生的自学用书,也可以作为计算机等级考试培训班的教学参考书和辅导用书。

图书在版编目(CIP)数据

二级公共基础知识与 Access 数据库程序设计/新思路教育科技研究中心编.
—北京:机械工业出版社,2012.11
(全国计算机等级考试一本通关六合一)
ISBN 978-7-111-40251-0

I. ①二… II. ①新… III. ①电子计算机—水平考试—自学参考资料 ②关系数据库
系统—数据库管理系统—程序设计—水平考试—自学参考资料 IV. ①TP3

中国版本图书馆 CIP 数据核字(2012)第 259701 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:张晓娟 责任编辑:张晓娟
版式设计:墨格文慧 责任印制:李妍

北京诚信伟业印刷有限公司印刷

2013 年 1 月第 1 版第 1 次印刷

210mm×285mm·15.75 印张·593 千字

0 001—5000 册

标准书号:ISBN 978-7-111-40251-0

ISBN 978-7-89433-702-3(光盘)

定价:32.00 元(含 1CD)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

编委会

主 编：兰天静

编 委：（排名不分先后）

付江辉	迟 剑	房向阳	李爱菊
李 婧	董晓晓	赵江静	韩雪松
安玉彦	孙 科	刘永阳	马天宝
刘艳飞	柳寒冰	葛立军	唐保存
黄 和	兰天静	任大鹏	黄琴华
赵 诚	王艳梅	金 利	刘本发

编 校：（排名不分先后）

赵 波	李 润	周 雪	丁 建
丁力刚	杨宏舟	肖维萍	李 欣
梁 继	章金博	孟翠苗	李 瑞
胡昭昀	孟祥萍	张 宇	王 丹

为了帮助更多的考生顺利地通过考试，我们在深入研究、全面解读考试大纲的基础上，组织具有丰富的计算机等级考试辅导教学经验的一线教师编写了本书。本书分为考点详解、真题研究、拓展训练、考试题库、学习软件和模拟考场，共六大模块。

● 考点详解：知识全面，讲解详细

本书把原本繁杂的知识分成一个考点一个考点地去讲解，条理清晰、目标明确。除此之外，考点分等级，各知识点讲解详细，相当于一本精简版教程，对于有可能考核的内容做到一网打尽。还记得每次考试前，都希望老师能给我们“画重点”吗？画出一些重要的内容，大多都是必考的。我们推出的“画书法”就是这样的道理，使用各类标记符号为你提前把必考点和常考点一一标明，这些也就是学习的重点。

● 真题研究：深度剖析，分块训练

每一章后面是该章近年来考过的真题，题目越多表示该考点其实就是考试的“热点”。通过按考点学习真题，不仅可以了解具体考核形式，还可以温习试题（很多试题每年的考核形式都是相似的，甚至有些试题每年都会重复考核）。解题方案均由一线老师根据实践教学所得经验编写，注重教授考生解题思路、方法和技巧，让你通过做历年真题和模拟试题快速掌握各知识点及答题技巧。

● 拓展训练：巩固练习，实战演练

每个知识点都有经典试题的解题示范，并且在每章后留有大量练习题。通过试题训练，对前面的考点讲解做到真正地实战演练、消化理解，从而达到巩固学习成果的目的。

● 考试题库：无纸化考试新题型

本书最后安排了6套无纸化考试试题。该试题完全按照最新无纸化题型及最新考核要求设计，不论在形式和难度上，都与真实考试类似。每套试题都有详细的解题步骤，教你掌握每一题的解题思路和操作过程。

● 超级模拟考试软件

本书配有强大的考试模拟软件练习光盘，具有如下特点：

- 学习软件：含多套新版考试模拟试题，可日常练习，强化学习效果；同时配有多套历年真题及答案。
- 模拟考场：模拟考试，自测学习效果，登录、抽题、答题、提交与真实考试一样，并提供详细的答题步骤和标准答案。同时，还可以自动生成试卷、自动计时、自动评分。

本书由北京新思路教育科技有限公司组织编写，参加编写的有谭彪、钱勇、王海丽、张冬梅、潘菲、任晓军、钱磊、王丽丽、潘梅梅、张明轩、徐保才、白冬雪、聂钰桢、于建新。由于时间仓促，书中难免有不当之处，敬请指正。

新思路教育科技有限公司

最新版考试大纲

二级公共基础知识

1. 基本要求

- ① 掌握算法的基本概念。
- ② 掌握基本数据结构及其操作。
- ③ 掌握基本排序和查找算法。
- ④ 掌握逐步求精的结构化程序设计方法。
- ⑤ 掌握软件工程的基本方法,具有初步应用相关技术进行软件开发的能力。
- ⑥ 掌握数据库的基本知识,了解关系数据库的设计。

2. 考试内容

(1) 基本数据结构与算法

- ① 算法的基本概念:算法复杂度的概念和意义(时间复杂度与空间复杂度)。
- ② 数据结构的定义:数据的逻辑结构与存储结构;数据结构的图形表示;线性结构与非线性结构的概念。
- ③ 线性表的定义:线性表的顺序存储结构及其插入与删除运算。
- ④ 栈和队列的定义:栈和队列的顺序存储结构及其基本运算。
- ⑤ 线性单链表、双向链表与循环链表的结构及其基本运算。
- ⑥ 树的基本概念:二叉树的定义及其存储结构;二叉树的前序、中序和后序遍历。
- ⑦ 顺序查找与二分法查找算法:基本排序算法(交换类排序,选择类排序,插入类排序)。

(2) 程序设计基础

① 程序设计方法与风格。

② 结构化程序设计。

③ 面向对象的程序设计方法:对象、方法、属性及继承与多态性。

(3) 软件工程基础

① 软件工程基本概念:软件生命周期概念、软件工具与软件开发环境。

② 结构化分析方法:数据流图、数据字典、软件需求规格说明书。

③ 结构化设计方法:总体设计与详细设计。

④ 软件测试的方法:白盒测试与黑盒测试、测试用例设计、软件测试的实施、单元测试、集成测试和系统测试。

⑤ 程序的调试:静态调试与动态调试。

(4) 数据库设计基础

① 数据库的基本概念:数据库、数据库管理系统、数据库系统。

② 数据模型:实体联系模型及 E-R 图、从 E-R 图导出关系数据模型。

③ 关系代数运算:集合运算及选择、投影、连接运算、数据库规范化理论。

④ 数据库设计方法和步骤:需求分析、概念设计、逻辑设计和物理设计的相关策略。

3. 考试方式

无纸化考试中,公共基础知识的考试方式为上机考试,题型全部为选择题,共 10 分。

二级 ACCESS

1. 基本要求

- ① 具有数据库系统的基础知识。
- ② 基本了解面向对象的概念。
- ③ 掌握关系数据库的基本原理。
- ④ 掌握数据库程序设计方法。
- ⑤ 能使用 Access 建立一个小型数据库应用系统。

2. 考试内容

(1) 数据库基础知识

① 基本概念:数据库、数据模型、数据库管理系统、类和对象、事件。

② 关系模型(实体的完整性、参照的完整性、用户定义的完整性):关系模式、关系、元组、属性、字段、域、值、主关键字等。

③ 关系运算基本概念:选择运算、投影运算、连接运算。

④ SQL 基本命令:查询命令、操作命令。

⑤ Access 系统简介:

- Access 系统的基本特点。
- 基本对象:表、查询、窗体、报表、页、宏、模块。

(2) 数据库和表的基本操作

① 创建数据库:

- 创建空数据库。
- 使用向导创建数据库。

② 表的建立:

- 建立表结构:使用向导、使用表设计器、使用数据表。
- 设置字段属性。
- 输入数据:直接输入数据、获取外部数据。

③ 表间关系的建立与修改:

- 表间关系的概念: 一对一、一对多。
- 设置参照完整性。
- 建立表间关系。

④ 表的维护:

● 修改表结构: 添加字段、修改字段、删除字段、重新设置主关键字。

● 编辑表内容: 添加记录、个性记录、删除记录、复制记录。

● 调整表外观。

⑤ 表的其他操作:

- 查找数据。
- 替换数据。
- 排序记录。
- 筛选记录。

(3) 查询的基本操作

① 查询分类:

- 选择查询。
- 参数查询。
- 交叉表查询。
- 操作查询。
- SQL 查询。

② 查询准则:

- 运算符。
- 函数。
- 表达式。

③ 创建查询:

- 操作已创建的查询。
- 编辑查询中的字段。
- 编辑查询中的数据源。
- 排序查询的结果。

(4) 窗体的基本操作

① 窗体分类:

- 纵栏式窗体。
- 表格式窗体。
- 主/子窗体。
- 数据表窗体。
- 图表窗体。
- 数据透视表窗体。

② 创建窗体:

- 使用向导创建窗体。
- 使用设计器创建窗体: 控件的含义及种类、在窗体中添加和修改控件、设置控件的常见属性。

(5) 报表的基本操作

① 报表分类:

- 纵栏式报表。
- 表格式报表。
- 图表报表。
- 标签报表。

② 使用向导创建报表。

③ 使用设计器编辑报表。

④ 在报表中进行计算和汇总。

(6) 页的基本操作

① 数据访问页的概念。

② 创建数据访问页:

- 自动创建数据访问页。
- 使用向导创建数据访问页。

(7) 宏

① 宏的基本概念。

② 宏的基本操作:

- 创建宏: 创建一个宏、创建宏组。
- 运行宏。
- 在宏中使用条件。
- 设置宏操作参数。
- 常用的宏操作。

(8) 模块

① 模块的基本概念:

- 类模块。
- 标准模块。
- 将宏转换为模块。

② 创建模块:

● 创建 VBA 模块: 在模块中加入过程、在模块中执行宏。

- 编写事件过程: 键盘事件、鼠标事件、窗口事件、操作事件和其他事件。

③ 调用和参数传递。

④ VBA 程序设计基础:

- 面向对象程序设计的基本概念。
- VBA 编程环境: 进入 VBE 界面。
- VBA 编程基础: 常量、变量、表达式。
- VBA 程序流程控制: 顺序控制、选择控制、循环控制。
- VBA 程序的调试: 设置断点、单步跟踪、设置监视点。

3. 考试方式

全部采用上机操作方式, 考试时间为 120 分钟。

选择题: 共 40 小题, 共 40 分。

编程题: 共 3 大题, 共 60 分。

目 录

前言

最新版考试大纲

第 1 章 公共基础知识

知识清单	1
考点 1 算法的复杂度	1
考点 2 数据结构	1
考点 3 栈	2
考点 4 队列	2
考点 5 链表	2
考点 6 二叉树	2
考点 7 查找	3
考点 8 排序	4
考点 9 程序设计的方法与风格	4
考点 10 结构化程序设计	4
考点 11 面向对象方法	5
考点 12 软件工程基本概念	5
考点 13 软件生命周期	5
考点 14 软件设计	6
考点 15 结构化分析方法	7
考点 16 软件测试	7
考点 17 数据库	8
考点 18 关系	9
考点 19 数据库设计与原理	10
历年真题必学	10
巩固拓展训练	13

第 2 章 数据库基础知识

知识清单	19
考点 1 数据库基础知识	19
考点 2 关系数据库	20
考点 3 数据库设计基础	20
考点 4 Access 简介	21
历年真题必学	22
巩固拓展训练	24

第 3 章 数据库和表

知识清单	31
考点 1 创建数据库	31
考点 2 创建表	32
考点 3 维护表	38
考点 4 操作表	44
历年真题必学	48
巩固拓展训练	52

第 4 章 查询

知识清单	61
考点 1 查询概述	61
考点 2 创建选择查询	63
考点 3 创建交叉表查询	65
考点 4 创建参数查询	66
考点 5 创建操作查询	66
考点 6 创建 SQL 查询	68
考点 7 编辑和使用查询	70
历年真题必学	70
巩固拓展训练	75

第 5 章 窗体

知识清单	87
考点 1 认识窗体	87
考点 2 创建窗体	88
考点 3 设计窗体	89
考点 4 美化窗体	92
历年真题必学	92
巩固拓展训练	93

第 6 章 报表

知识清单	101
考点 1 报表的基本概念与组成	101
考点 2 创建报表	102
考点 3 编辑报表	102
考点 4 报表排序和分组	103
考点 5 使用计算控件	104
考点 6 创建子报表	104
考点 7 创建多列报表	105
考点 8 设计复杂的报表	105
历年真题必学	105
巩固拓展训练	107

第 7 章 数据访问页

知识清单	115
考点 1 数据访问页的基本概念	115
考点 2 创建数据访问页	115
考点 3 编辑数据访问页	116
历年真题必学	117
巩固拓展训练	117

第 8 章 宏

知识清单	121
考点 1 报表的基本概念与组成	121
考点 2 建立宏	121
考点 3 通过事件触发宏	123
历年真题必学	124
巩固拓展训练	126

第 9 章 模块与 VBA 编程基础

知识清单	133
考点 1 模块的基本概念	133
考点 2 创建模块	133
考点 3 VBA 程序设计基础	134
考点 4 VBA 流程控制语句	139
考点 5 过程调用和参数传递	142
考点 6 VBA 程序运行错误处理	143
考点 7 VBA 程序的调试	143
历年真题必学	144
巩固拓展训练	154

第 10 章 VBA 数据库编程

知识清单	175
考点 1 VBA 常见操作	175
考点 2 VBA 的数据库编程	177
历年真题必学	179
巩固拓展训练	180

第 11 章 无纸化考试试题、答案及详解

第 1 套 无纸化考试试题	183
一、选择题	183
二、基本操作题	187
三、简单应用题	187
四、综合应用题	187
第 2 套 无纸化考试试题	187
一、选择题	187
二、基本操作题	192
三、简单应用题	192
四、综合应用题	192
第 3 套 无纸化考试试题	192
一、选择题	192
二、基本操作题	196
三、简单应用题	196
四、综合应用题	197
第 4 套 无纸化考试试题	197
一、选择题	197

二、基本操作题	200
三、简单应用题	201
四、综合应用题	201
第 5 套 无纸化考试试题	201
一、选择题	201
二、基本操作题	205
三、简单应用题	205
四、综合应用题	205
第 6 套 无纸化考试试题	205
一、选择题	205
二、基本操作题	210
三、简单应用题	209
四、综合应用题	210
第 1 套 无纸化考试试题答案及详解	210
一、选择题	210
二、基本操作题	213
三、简单应用题	214
四、综合应用题	215
第 2 套 无纸化考试试题答案及详解	216
一、选择题	216
二、基本操作题	219
三、简单应用题	219
四、综合应用题	221
第 3 套 无纸化考试试题答案及详解	221
一、选择题	221
二、基本操作题	224
三、简单应用题	225
四、综合应用题	226
第 4 套 无纸化考试试题答案及详解	227
一、选择题	227
二、基本操作题	229
三、简单应用题	231
四、综合应用题	232
第 5 套 无纸化考试试题答案及详解	233
一、选择题	233
二、基本操作题	235
三、简单应用题	236
四、综合应用题	237
第 6 套 无纸化考试试题答案及详解	238
一、选择题	238
二、基本操作题	240
三、简单应用题	241
四、综合应用题	243

第1章 公共基础知识

本章导读

本章作为二级公共基础知识部分,主要讲述了数据结构与算法、程序设计基础、软件工程基础和数据库设计基础等考核内容。

另外,在二级各科的笔试考试中,选择题前10题和填空题前5题皆为本章知识,共占30分,可见本章知识的重要性,考生应好好掌握。

本章考试大纲

- (1) 数据结构与算法
- (2) 程序设计基础
- (3) 软件工程基础
- (4) 数据库设计基础

知识清单

考点1 算法的复杂度

1. 算法的基本概念

利用计算机算法为计算机解题的过程实际上是在实施某种算法。

① 算法的基本特征: 算法一般具有4个基本特征: 可行性、确定性、有穷性、拥有足够的情报。

② 算法的基本运算和操作: 算术运算、逻辑运算、关系运算、数据传输。

③ 算法的3种基本控制结构: 顺序结构、选择结构、循环结构。

④ 算法基本设计方法: 列举法、归纳法、递推、递归、

减半递推技术、回溯法。

⑤ 指令系统: 计算机系统能执行的所有指令的集合。

2. 算法复杂度

算法复杂度包括时间复杂度和空间复杂度。两者的区别见表1-1。

表 1-1 算法复杂度

名称	描述
时间复杂度	执行算法所需要的计算工作量
空间复杂度	执行这个算法所需要的内存空间

考点2 数据结构

1. 逻辑结构和存储结构

(1) 数据结构的基本概念

① 数据结构: 指相互有关联的数据元素的集合。

② 数据结构研究的3个方面: 数据集合中各数据元素之间所固有的逻辑关系, 即数据的逻辑结构; 在对数据进行处理时, 各数据元素在计算机中的存储关系, 即数据的存储结构; 对各种数据结构进行的运算。

(2) 逻辑结构

数据的逻辑结构是对数据元素之间的逻辑关系的描述, 它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。数据的逻辑结构有两个要素: 一是数据元素的集合, 通常记为 D ; 二是 D 上的关系, 它反映了数据元素之间的前后件关系, 通常记为 R 。一个数据结构可以表示成: $B=(D,R)$ 。

其中, B 表示数据结构。为了反映 D 中各数据元素之间的前后件关系, 一般用二元组来表示。

(3) 存储结构

数据的逻辑结构在计算机存储空间中的存放形式称为数据的存储结构 (也称数据的物理结构)。

由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同, 因此, 为了表示存放在计算机存储空间中的各

数据元素之间的逻辑关系 (即前后件关系), 在数据的存储结构中不仅要存放各数据元素的信息, 还需要存放各数据元素之间的前后件关系信息。

一种数据的逻辑结构根据需要可以表示成多种存储结构, 常用的存储结构有顺序、链接等存储结构。

顺序存储方式主要用于线性的数据结构, 它把逻辑上相邻的数据元素存储在物理上相邻的存储单元里, 结点之间的关系由存储单元的邻接关系来体现。

链式存储结构就是在每个结点中至少包含一个指针域, 用指针来体现数据元素之间逻辑上的联系。

2. 线性结构和非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度, 一般将数据结构分为两大类型: 线性结构与非线性结构。

(1) 如果一个非空的数据结构满足下列两个条件:

① 有且只有一个根结点。

② 每一个结点最多有一个前件, 也最多有一个后件。

则称该数据结构为线性结构。线性结构又称线性表。在一个线性结构中插入或删除任何一个结点后还应是线性结构。栈、队列、串等都为线性结构。

如果一个数据结构不是线性结构, 则称之为非线性结构。

数组、广义表、树和图等数据结构都是非线性结构。

(2) 线性表的顺序存储结构具有以下两个基本特点。

① 线性表中所有元素所占的存储空间是连续的。

② 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

(3) 顺序表的运算有查找、插入、删除 3 种。

考点 3 栈

1. 栈的基本概念

栈(stack)是一种特殊的线性表,是限定只在一端进行插入与删除的线性表。

在栈中,一端是封闭的,既不允许插入元素,也不允许删除元素;另一端是开口的,允许插入和删除元素。通常,称插入、删除的这一端为栈顶,另一端为栈底。当表中没有元素时称为空栈。栈顶元素总是最后被插入的元素,也是最先被删除的元素;栈底元素总是最先被插入的元素,也是最

后才能被删除的元素。栈是按照先进后出或后进先出的原则组织数据的。

2. 栈的顺序存储及其运算

栈的基本运算有 3 种:入栈、退栈与读栈顶元素。

① 入栈运算:在栈顶位置插入一个新元素。

② 退栈运算:取出栈顶元素并赋给一个指定的变量。

③ 读栈顶元素:将栈顶元素赋给一个指定的变量。

考点 4 队列

1. 队列的基本概念

队列是只允许在一端进行删除,在另一端进行插入的顺序表。通常,将允许删除的这一端称为队头,将允许插入的这一端称为队尾。当表中没有元素时称为空队列。

队列的修改是依照先进先出的原则进行的,因此队列也称为先进先出的线性表或者后进后出的线性表。

2. 队列运算

入队运算是往队列队尾插入一个数据元素;退队运算是从队列的队头删除一个数据元素。

队列的顺序存储结构一般采用队列循环的形式。循环队列 $s=0$,表示队列空; $s=1$ 且 $front=rear$,表示队列满。计算循环队列元素个数的公式为:尾指针-头指针(若为负数,再加其容量即可)。

考点 5 链表

在链式存储方式中,要求每个结点由两部分组成:一部分用于存放数据元素值,称为数据域;另一部分用于存放指针,称为指针域。其中,指针用于指向该结点的前一个或后一个结点(即前件或后件)。链式存储方式既可用于表示线性结构,也可用于表示非线性结构。

(1) 线性链表

线性表的链式存储结构称为线性链表。在某些应用中,对线性链表中的每个结点设置两个指针,一个称为左指针,用以指向其前件结点;另一个称为右指针,用以指向其后件结点。这样的表称为双向链表。在线性链表中,各数据元素

结点的存储空间可以是不连续的,且各数据元素的存储顺序与逻辑顺序可以不一致。在线性链表中进行插入与删除不需要移动链表中的元素。线性单链表中,HEAD 称为头指针,HEAD=NULL(或 0)称为空表。如果是双项链表的两指针,左指针(Llink)指向前件结点,右指针(Rlink)指向后件结点。线性链表的基本运算有 3 种:查找、插入、删除。

(2) 带链的栈

栈也是线性表,也可以采用链式存储结构。带链的栈可以用来收集计算机存储空间中所有空闲的存储结点,这种带链的栈称为可利用栈。

考点 6 二叉树

1. 二叉树概念及其基本性质

(1) 二叉树及其基本概念

二叉树是一种很有用的非线性结构,具有以下两个特点。

① 非空二叉树只有一个根结点。

② 每一个结点最多有两棵子树,且分别称为该结点的左子树和右子树。

在二叉树中,每一个结点的度最大为 2,即所有子树(左子树或右子树)也均为二叉树。另外,二叉树中的每个结点的子树被明显地分为左子树和右子树。

在二叉树中,一个结点可以只有左子树而没有右子树,也可以只有右子树而没有左子树。当一个结点既没有左子树

也没有右子树时,该结点即为叶子结点。

下面详细讲解二叉树的基本概念,见表 1-2。

表 1-2 二叉树的基本概念

名称	说明
父结点(根)	在树结构中,每一个结点只有一个前件,称为父结点。没有前件的结点只有一个,称为树的根结点,简称树的根
子结点和叶子结点	在树结构中,每一个结点可以有多个后件,称为该结点的子结点。没有后件的结点称为叶子结点
度	在树结构中,一个结点所拥有的后件的个数称为该结点的度,所有结点中最大的度称为树的度

(续)

名称	说明
深度	定义一棵树的根结点所在的层次为 1, 其他结点所在的层次等于它的父结点所在的层次加 1。树的最大层次称为树的深度
子树	在树中, 以某结点的一个子结点为根构成的树称为该结点的一棵子树

(2) 二叉树的基本性质

二叉树具有以下几个性质。

① 性质 1: 在二叉树的第 k 层上, 最多有 2^{k-1} ($k \geq 1$) 个结点。

② 性质 2: 深度为 m 的二叉树最多有 $2^m - 1$ 个结点。

③ 性质 3: 在任意一棵二叉树中, 度为 0 的结点 (即叶子结点) 总是比度为 2 的结点多一个。

④ 性质 4: 具有 n 个结点的二叉树, 其深度至少为 $\lceil \log_2 n \rceil + 1$ 。其中, $\lceil \log_2 n \rceil$ 表示取 $\log_2 n$ 的整数部分。

(3) 满二叉树与完全二叉树

满二叉树是指这样的一种二叉树: 除最后一层外, 每一层上的所有结点都有两个子结点。在满二叉树中, 每一层上的结点数都达到最大值, 即在满二叉树的第 k 层上有 2^{k-1} 个结点, 且深度为 m 的满二叉树有 $2^m - 1$ 个结点。

完全二叉树是指这样的二叉树: 除最后一层外, 每一层上的结点数均达到最大值; 在最后一层上只缺少右边的若干结点。

对于完全二叉树来说, 叶子结点只可能在层次最大的两层上出现: 对于任何一个结点, 若其右分支下的子孙结点的最大层次为 p , 则其左分支下的子孙结点的最大层次为 p 或 $p+1$ 。

完全二叉树具有以下两个性质。

① 性质 1: 具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。

② 性质 2: 设完全二叉树共有 n 个结点。如果从根结点开始, 按层次 (每一层从左到右) 用自然数 $1, 2, \dots, n$ 给结点进行编号, 则对于编号为 k ($k=1, 2, \dots, n$) 的结点有以下结论:

- 若 $k=1$, 则该结点为根结点, 它没有父结点; 若 $k>1$, 则该结点的父结点编号为 $\text{INT}(k/2)$ 。
- 若 $2k \leq n$, 则编号为 k 的结点的左子结点编号为 $2k$; 否则该结点无左子结点 (显然也没有右子结点)。
- 若 $2k+1 \leq n$, 则编号为 k 的结点的右子结点编号为 $2k+1$; 否则该结点无右子结点。

2. 二叉树的遍历

在遍历二叉树的过程中, 一般先遍历左子树, 再遍历右子树。在先左后右的原则下, 根据访问根结点的次序, 二叉树的遍历分为: 前序遍历、中序遍历和后序遍历3类。

(1) 前序遍历

先访问根结点, 然后遍历左子树, 最后遍历右子树; 并且在遍历左、右子树时, 仍需先访问根结点, 然后遍历左子树, 最后遍历右子树。

(2) 中序遍历

先遍历左子树, 然后访问根结点, 最后遍历右子树; 并且, 在遍历左、右子树时, 仍然先遍历左子树, 然后访问根结点, 最后遍历右子树。

(3) 后序遍历

先遍历左子树, 然后遍历右子树, 最后访问根结点; 并且, 在遍历左、右子树时, 仍然先遍历左子树, 然后遍历右子树, 最后访问根结点。

考点 7 查找

1. 顺序查找

查找是指在一个给定的数据结构中查找某个指定的元素。从线性表的第一个元素开始, 依次将线性表中的元素与被查找的元素相比较, 若相等则表示查找成功; 若线性表中所有的元素都与被查找元素进行了比较但都不相等, 则表示查找失败。

下列两种情况下只能采用顺序查找方法。

① 如果线性表为无序表, 则不管是顺序存储结构还是链式存储结构, 只能用顺序查找。

② 即使是有序线性表, 如果采用链式存储结构, 也只能用顺序查找。

2. 二分法查找

二分法查找, 也称折半查找, 是一种高效的查找方法。能使用二分法查找的线性表必须满足用顺序存储结构和线性表是有序表两个条件。

“有序”是特指元素按非递减排列, 即从小到大排列,

但允许相邻元素相等。下一节排序中, 有序的含义也是如此。

对于长度为 n 的有序线性表, 利用二分法查找元素 X 的过程如下。

步骤 1: 将 X 与线性表的中间项比较。

步骤 2: 如果 X 的值与中间项的值相等, 则查找成功, 结束查找。

步骤 3: 如果 X 小于中间项的值, 则在线性表的前半部分以二分法继续查找。

步骤 4: 如果 X 大于中间项的值, 则在线性表的后半部分以二分法继续查找。

顺序查找法每比较一次, 只将查找范围减少 1, 而二分法查找每比较一次, 可将查找范围减少为原来的一半, 效率大大提高。

对于长度为 n 的有序线性表, 在最坏情况下, 二分法查找只需比较 $\log_2 n$ 次, 而顺序查找需要比较 n 次。

考点 8 排序

1. 交换类排序法

(1) 冒泡排序法

首先,从表头开始往后扫描线性表,逐次比较相邻两个元素的大小,若前面的元素大于后面的元素,则将它们互换。不断地将两个相邻元素中的大者往后移动,最后,最大者到了线性表的最后。

然后,从后到前扫描剩下的线性表,逐次比较相邻两个元素的大小,若后面的元素小于前面的元素,则将它们互换。不断地将两个相邻元素中的小者往前移动,最后,最小者到了线性表的最前面。

对剩下的线性表重复上述过程,直到剩下的线性表变空为止。此时,已经排好序。

在最坏的情况下,冒泡排序需要比较次数为 $n(n-1)/2$ 。

(2) 快速排序法

任取待排序序列中的某个元素作为基准(一般取第一个元素),通过一次排序,将待排元素分为左右两个子序列,左

子序列元素的排序码均小于或等于基准元素的排序码,右子序列的排序码则大于基准元素的排序码,然后分别对两个子序列继续进行排序,直至整个序列有序。

2. 插入类排序法

(1) 简单插入排序法

最坏情况需要 $n(n-1)/2$ 次比较。

(2) 希尔排序法

最坏情况需要 $O(n^{1.5})$ 次比较。

3. 选择类排序法

(1) 简单选择排序法

最坏情况需要 $n(n-1)/2$ 次比较。

(2) 堆排序法

最坏情况需要 $O(n\log_2 n)$ 次比较。

相比以上几种(除希尔排序法外),堆排序法的时间复杂度最小。

考点 9 程序设计的方法与风格

要养成良好的程序设计习惯,应遵循如下原则。

(1) 源程序文档化

① 符号名的命名。符号名的命名应具有一定的实际含义,以便于人们对程序功能的理解。

② 程序注释。在源程序中添加正确的注释可帮助人们理解程序。程序注释可分为序言性注释和功能性注释。语句结构清晰第一,效率第二。

③ 视觉组织。通过在程序中添加一些空格、空行和缩进等,使人们在视觉上对程序的结构一目了然。

(2) 数据说明的方法

为使程序中数据说明易理解和维护,可采用表 1-3 中的数据说明风格。

表 1-3 数据说明风格

数据说明风格	详细说明
次序应规范化	使数据说明次序固定,使数据的属性容易查找,也有利于测试、排错和维护
变量安排有序化	当多个变量出现在同一个说明语句中时,变量名应按字母顺序排序,以便于查找
使用注释	在定义一个复杂的数据结构时,应通过注解来说明该数据结构的特点

(3) 语句的结构程序

语句的结构程序应该简单易懂,语句构造应该简单直接。

(4) 输入和输出

输入输出比较简单,这里就不作介绍。

考点 10 结构化程序设计

1. 结构化程序设计的原则

结构化程序设计方法引入了工程思想和结构化思想,使大型软件的开发和编程得到了极大的改善。结构化程序设计方法的主要原则为自顶向下、逐步求精、模块化和限制使用 goto 语句。

① 自顶向上:先考虑整体,再考虑细节;先考虑全局,再考虑局部。

② 逐步求精:对复杂问题应设计一些子目标作为过渡,逐步细化。

③ 模块化:把程序要解决的总目标分解为分目标,再进一步分解为具体的小目标,把每个小目标称为一个模块。

④ 限制使用 goto 语句:在程序开发过程中要限制使用 goto 语句。

2. 结构化程序的基本结构

结构化程序的基本结构有 3 种类型:顺序结构、选择结构和循环结构。

① 顺序结构:是最基本、最普通的结构形式,按照程序中的语句行的先后顺序逐条执行。

- ② 选择结构：又称为分支结构，它包括简单选择和分支选择结构。
- ③ 循环结构：根据给定的条件，判断是否要重复执行某

一相同的或类似的程序段。循环结构对应两类循环语句：先判断后执行的循环体称为当型循环结构；先执行循环体后判断的称为直到型循环结构。

考点 11 面向对象方法

面向对象方法涵盖对象及对象属性与方法、类、继承、多态性等几个基本要素。

1. 对象

通常把对象的操作也称为方法或服务。

属性即对象所包含的信息，通常在设计对象时确定，一般只能通过执行对象的操作来改变。属性值通常指的是纯粹的数据值，而不是指对象。

操作描述了对对象执行的功能，若通过信息的传递，还可以为其他对象使用。对象具有如下特征：标识唯一性、分类性、多态性、封装性和模块独立性。

2. 类和实例

类是具有共同属性、共同方法的对象的集合。它描述了属于该对象类型的所有对象的性质，而一个对象则是其对应类的一个实例。

类是关于对象性质的描述，它同对象一样，包括一组数

据属性和在数据上的一组合法操作。

3. 消息

消息是实例之间传递的信息，它请求对象执行某一处理或回答某一要求的信息，它统一了数据流和控制流。一个消息由三部分组成：接收消息的对象的名称、消息标识符（消息名）以及零或多个参数。

4. 继承

广义地说，继承是指能够直接获得已有的性质和特征，而不必重复定义它们。

继承分为单继承与多重继承。单继承是指一个类只允许有一个父类，即类等级为树形结构。多重继承是指一个类允许有多个父类。

5. 多态性

对象根据所接受的消息做出动作，同样的消息被不同的对象接受时可导致完全不同的行动，该现象称为多态性。

考点 12 软件工程基本概念

1. 软件定义与软件特点

软件指的是计算机系统中与硬件相互依存的另一部分，包括程序、数据和相关文档的完整集合。程序是软件开发人员根据用户需求开发的、用程序设计语言描述的、适合计算机执行的指令序列。数据是使程序能正常操纵信息的数据结构。文档是与程序的开发、维护和使用有关的图文资料。

可见，软件由两部分组成。

- ① 机器可执行的程序和数据。
- ② 机器不可执行的，与软件的开发、运行、维护、使用等有关的文档。

根据应用目标的不同，软件可分应用软件、系统软件和支撑软件（或工具软件），如表 1-4 所示。

表 1-4 软件的分类

名称	描述
应用软件	为解决特定领域的应用而开发的软件
系统软件	计算机管理自身资源，提高计算机使用效率并为计算机用户提供各种服务的软件

(续)

名称	描述
支撑软件（或工具软件）	支撑软件是介于两者之间，协助用户开发软件的工具性软件

2. 软件工程

为了摆脱软件危机，提出了软件工程的概念。软件工程学研究软件开发和维护的普遍原理与技术的一门工程学科。所谓软件工程，是指采用工程的概念、原理、技术和方法指导软件的开发与维护。软件工程学的主要研究对象包括软件开发与维护的技术、方法、工具和管理等方面。

软件工程包括 3 个要素：方法、工具和过程，其作用见表 1-5。

表 1-5 软件工程三要素

名称	描述
方法	方法是完成软件工程项目的手段
工具	工具支持软件的开发、管理和文档生成
过程	过程支持软件开发的各个环节的控制、管理

考点 13 软件生命周期

1. 软件生命周期概念

软件产品从提出、实现、使用维护到停止使用，退役的过程称为软件生命周期。软件生命周期分为 3 个时期，共 8

个阶段。

- ① 软件定义期：包括问题定义、可行性研究和需求分析 3 个阶段。

② 软件开发期：包括概要设计、详细设计、实现和测试 4 个阶段。

③ 运行维护期：即运行维护阶段。

软件生命周期各个阶段的活动可以有重复，执行时也可以有迭代。

2. 软件生命周期各阶段的主要任务

软件生命周期各阶段的主要任务，见表 1-6。

表 1-6 软件生命周期各阶段的主要任务

任务	描述
问题定义	确定要求解决的问题是什么
可行性研究与计划制定	决定该问题是否存在一个可行的解决办法，指定完成开发任务的实施计划
需求分析	对待开发软件提出的需求进行分析，并给出详细定义。编写软件规格说明书及初步的用户手册，提交评审

(续)

任务	描述
软件设计	通常又分为概要设计和详细设计两个阶段，给出软件的结构、模块的划分、功能的分配以及处理流程。这阶段提交评审的文档有概要设计说明书、详细设计说明书和测试计划初稿
软件实现	在软件设计的基础上编写程序。这阶段完成的文档有用户手册、操作手册等面向用户的文档，以及为下一步作准备而编写的单元测试计划
软件测试	在设计测试用例的基础上，检验软件的各个组成部分，编写测试分析报告
运行维护	将已交付的软件投入运行，同时不断维护，进行必要且可行的扩充和删改

考点 14 软件设计

1. 软件设计基本概念

(1) 按技术观点分

从技术观点上看，软件设计包括软件结构设计、数据设计、接口设计和过程设计。

① 结构设计：定义软件系统各主要部件之间的关系。

② 数据设计：将分析时创建的模型转化为数据结构的定义。

③ 接口设计：描述软件内部、软件和协作系统之间以及软件与人之间如何通信。

④ 过程设计：把系统结构部件转换为软件的过程性描述。

(2) 从工程管理角度分

从工程管理角度来看，软件设计分为概要设计和详细设计两个阶段。

① 概要设计：将软件的需求转化为软件体系结构、确定系统级接口、全局数据结构或数据库模式。

② 详细设计：确立每个模块的实现算法和局部数据结构，用适当方法表示算法和数据结构的细节。

2. 软件设计的基本原理

(1) 软件设计中应该遵循的基本原理和与软件设计有关的概念

① 抽象。软件设计中考虑模块化解决方案时，可以定出多个抽象级别。抽象的层次从概要设计到详细设计逐步降低。

② 模块化。模块是指把一个待开发的软件分解成若干小的简单的部分。模块化是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。

③ 信息隐蔽。信息隐蔽是指在一个模块内包含的信息（过程或数据），对于不需要这些信息的其他模块来说是不能访问的。

④ 模块独立性。模块独立性是指每个模块只完成系统要求的独立的子功能，并且与其他模块的联系最少且接口简单。模块的独立程度是评价设计好坏的重要度量标准。衡量软件的模块独立性使用耦合性和内聚性两个定性的度量标准。内聚性是信息隐蔽和局部化概念的自然扩展。一个模块的内聚性越强，则该模块的模块独立性越强。一个模块与其他模块的耦合性越强，则该模块的模块独立性越弱。

(2) 衡量软件模块独立性使用耦合性和内聚性两个定性的度量标准

① 内聚性是度量一个模块功能强度的相对指标。内聚是从功能角度来衡量模块的联系，它描述的是模块内的功能联系。内聚有如下种类，它们之间的内聚度由弱到强排列为：偶然内聚、逻辑内聚、时间内聚、过程内聚、通信内聚、顺序内聚、功能内聚。

② 耦合性是衡量模块之间互相连接的紧密程度的度量指标。耦合性取决于各个模块之间接口的复杂度、调用方式以及哪些信息通过接口。耦合可以分为多种形式，它们之间的耦合度由高到低排列为：内容耦合、公共耦合、外部耦合、控制耦合、标记耦合、数据耦合、非直接耦合。

在程序结构中，各模块的内聚性越强，则耦合性越弱。一般较优秀的软件设计，应尽量做到高内聚、低耦合，即减弱模块之间的耦合性和提高模块内的内聚性，有利于提高模块的独立性。

考点 15 结构化分析方法

1. 结构化分析方法的定义

结构化分析方法就是使用数据流图 (DFD)、数据字典 (DD)、结构化英语、判定表和判定树的工具,来建立一种新的、称为结构化规格说明的目标文档。

结构化分析方法的实质是着眼于数据流,自顶向下,对系统的功能进行逐层分解,以数据流图和数据字典为主要工具,建立系统的逻辑模型。

2. 结构化分析方法常用工具

(1) 数据流图 (DFD)

数据流图是系统逻辑模型的图形表示。即使不是专业的计算机技术人员也容易理解它,因此,它是分析员与用户之间极好的通信工具。

(2) 数据字典 (DD)

数据字典是对数据流图中所有元素的定义的集合,是结构化分析的核心。

数据流图和数据字典共同构成系统的逻辑模型。没有数

据字典,数据流图就不严格;若没有数据流图,数据字典也难以发挥作用。数据字典中有 4 种类型的条目:数据流、数据项、数据存储和加工。

(3) 判定表

有些加工的逻辑用语言形式不容易表达清楚,而用表的形式则一目了然。如果一个加工逻辑有多个条件、多个操作,并且在不同的条件组合下执行不同的操作,那么可以使用判定表来描述。

(4) 判定树

判定树和判定表没有本质的区别,可以用判定表表示的加工逻辑都能用判定树表示。

3. 软件需求规格说明书

软件需求规格说明书是需求分析阶段的最后成果,是软件开发的重要文档之一。它的特点是具有正确性、无歧义性、完整性、可验证性、一致性、可理解性、可修改性和可追踪性。

考点 16 软件测试

1. 软件测试的目的和准则

(1) 软件测试的目的

Grenford.J.Myers 给出了软件测试的目的:测试是为了发现程序中的错误而执行程序的过程。好的测试用例 (test case) 能发现迄今为止尚未发现的错误,一次成功的测试是能发现至今为止尚未发现的错误。

测试的目的是发现软件中的错误,但是,暴露错误并不是软件测试的最终目的,测试的根本目的是尽可能多地发现并排除软件中隐藏的错误。

(2) 软件测试的准则

根据上述软件测试的目的,为了能设计出有效的测试方案以及好的测试用例,软件测试人员必须深入理解并正确运用以下软件测试的基本准则。

- 所有测试都应追溯到用户需求。
- 在测试之前制订测试计划,并严格执行。
- 充分注意测试中的群集现象。
- 避免由程序的编写者测试自己的程序。
- 不可能进行穷举测试。
- 妥善保存测试计划、测试用例、出错统计和最终分析报告,为维护提供方便。

2. 软件测试的方法和实施

(1) 软件测试方法

软件测试具有多种方法,依据软件是否需要被执行,可以分为静态测试和动态测试方法。如果依照功能划分,可以分为白盒测试和黑盒测试方法。

① 静态测试和动态测试。

静态测试包括代码检查、静态结构分析、代码质量度量等。其中,代码检查分为代码审查、代码走查、桌面检查、静态分析等具体形式。

静态测试不实际运行软件,主要通过人工进行分析。动态测试就是通常所说的上机测试,是通过运行软件来检验软件中的动态行为和运行结果的正确性。

动态测试的关键是使用设计高效、合理的测试用例。测试用例就是为测试设计的数据,由测试输入数据和预期的输出结果两部分组成。测试用例的设计方法一般分为两类:黑盒测试方法和白盒测试方法。

② 黑盒测试和白盒测试。

白盒测试是把程序看成装在一只有透明的白盒子里,测试者完全了解程序的结构和处理过程。它根据程序的内部逻辑来设计测试用例,检查程序中的逻辑通路是否都按预定的要求正确地工作。

黑盒测试是把程序看成一只黑盒子,测试者完全不了解或不考虑程序的结构和处理过程。它根据规格说明书的功能来设计测试用例,检查程序的功能是否符合规格说明的要求。

(2) 软件测试的实施

软件测试过程分 4 个步骤,即单元测试、集成测试、验收测试和系统测试。

单元测试是对软件设计的最小单位——模块(程序单元)进行正确性检验测试。单元测试的技术可以采用静态分析和动态测试。

集成测试是测试和组装软件的过程,主要目的是发现与接口有关的错误,主要依据是概要设计说明书。集成测试所设计的内容包括:软件单元的接口测试、全局数据结构测试、

边界条件和非法输入的测试等。集成测试时将模块组装成程序,通常采用两种方式:非增量方式组装和增量方式组装。

确认测试的任务是验证软件的功能和性能以及其他特性是否满足了需求规格说明中确定的各种需求,包括软件配置是否完全、正确。确认测试的实施首先要运用黑盒测试方法,对软件进行有效性测试,即验证被测软件是否满足需求规格说明确认的标准。

系统测试是通过测试确认的软件,作为整个基于计算机系统的一个元素,与计算机硬件、外设、支撑软件、数据和人员等其他系统元素组合在一起,在实际运行(使用)环境下对计算机系统进行一系列的集成测试和确认测试。

系统测试的具体实施包括功能测试、性能测试、操作测试、配置测试、外部接口测试、安全性测试等。

3. 程序的调试

在对程序进行了成功的测试之后,将进入程序调试(通

常称 Debug,即排错)阶段。

程序的调试任务是诊断和改正程序中的错误。调试主要在开发阶段进行。

程序调试活动由两部分组成,一是根据错误的迹象确定程序中错误的确切性质、原因和位置;二是对程序进行修改,排除这个错误。

程序调试的基本步骤如下。

① 错误定位。从错误的外部表现形式入手,研究有关部分的程序,确定程序中出错位置,找出错误的内在原因。

② 修改设计和代码,以排除错误。

③ 进行回归测试,防止引进新的错误。

软件调试可分为静态调试和动态调试。静态调试主要是指通过人的思维来分析源程序代码和排错,是主要的设计手段,而动态调试是辅助静态调试的。主要的调试方法有:强行排错法、回溯法和原因排除法3种。

考点 17 数据库

1. 数据库的基本概念

数据是数据库中存储的基本对象,是描述事物的符号记录。

数据库是长期储存在计算机内、有组织、可共享的大量数据的集合,它具有统一的结构形式并存放于统一的存储介质内,是多种应用数据的集成,并可被各个应用程序所共享。所以,数据库技术的根本目标是解决数据共享问题。

数据库管理系统(Database Management System, DBMS)是数据库的机构,它是一种系统软件,负责数据库中的数据组织、数据操作、数据维护、控制及保护和数据服务等。数据库管理系统是数据系统的核心。

为完成数据管理功能,数据库管理系统提供了相应的数据语言:数据定义语言、数据操纵语言和数据控制语言。

2. 数据库系统的发展和基本特点

(1) 数据库系统的发展

数据管理技术的发展经历了3个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

关于数据管理3个阶段中的软硬件背景及处理特点,简单概括见表1-7。

表 1-7 数据管理 3 个阶段的比较

		人工管理阶段	文件管理阶段	数据库系统管理阶段
背景	应用目的	科学计算	科学计算、管理	大规模管理
	硬件背景	无直接存取设备	磁盘、磁鼓	大容量磁盘
	软件背景	无操作系统	有文件系统	有数据库管理系统
	处理方式	批处理	联机实时处理、批处理	分布处理、联机实时处理和批处理

(续)

		人工管理阶段	文件管理阶段	数据库系统管理阶段
特点	数据管理者	人	文件系统	数据库管理系统
	数据面向的对象	某个应用程序	某个应用程序	现实世界
	数据共享程度	无共享,冗余度大	共享性差,冗余度大	共享性大,冗余度小
	数据的独立性	不独立,完全依赖于程序	独立性差	具有高度的物理独立性和一定的逻辑独立性
	数据的结构化	无结构	记录内有结构,整体无结构	整体结构化,用数据模型描述
	数据控制能力	由应用程序控制	应用程序控制	由 DBMS 提供数据安全性、完整性、并发控制和恢复

(2) 数据库系统的特点

数据独立性指数据与程序间的互不依赖性,即数据库中的数据独立于应用程序而不依赖于应用程序。

数据的独立性一般分为物理独立性与逻辑独立性两种。

① 物理独立性:当数据的物理结构(包括存储结构、存取方式等)发生改变时,如存储设备更换、物理存储更换、存取方式改变等,应用程序可以不发生改变。

② 逻辑独立性:当数据的逻辑结构发生改变时,如修改数据模式、增加新的数据类型、改变数据间联系等,用户程序可以不发生改变。

3. 数据库系统的内部体系结构

(1) 数据系统的3级模式

① 概念模式。也称为逻辑模式,是对数据库系统中全局