

考点命中率99%

严格按照教育部最新颁布的无纸化考试大纲编写

全国计算机等级考试

教程全解 上机考题解答与训练



适用于Windows 7/XP系统

二级公共基础知识和 Visual FoxPro数据库程序设计

无纸化考试新版教程

- ▶ **内容全面** 包括二级VFP和二级公基全部课程考点详解
- ▶ **精讲精解** 针对考点超详细教学，零基础也能学会
- ▶ **一书多用** 课程详解+真题研密+分块训练+模拟演练+考试题库
- ▶ **真题研密** 真题按考点归类，专家权威讲解
- ▶ **配套软件** 新版无纸化考试模拟软件



赠 价值350元全套备考资料

- 二级VFP新教程考点详解
- 二级公共基础知识考点详解
- 无纸化考试新版模拟软件
- 100套无纸化题库试题（软件中）
- 《二级公基速学教程》（电子书）
- 《历年考试试题大全》（下载）

课程考点详解

知识点结构图

历年真题研密

同步训练及详解

无纸化综合模拟题

新版考试模拟软件



全国计算机等级考试 教程全解·上机 考题解答与训练

二级公共基础知识和 Visual FoxPro数据库程序设计

新思路教育科技研究中心
电子科技大学出版社

内 容 简 介

为了适应计算机等级考试的需求,帮助考生顺利通过考试,我们总结了多年考试经验和众多图书的优点,精心策划编写了本书。

本书根据最新考试大纲要求,依据近4年考试试题的研究成果和对命题规律的把握,结合历年真题讲解各考点,知识点讲解详细,重、难点突出,真切、实用;学习不跑题、切中要点!另外,本书所有试题的答案及解析均由一线老师根据实践教学中所经验编写,注重教授考生解题思路、方法和技巧,帮助考生系统全面的掌握各个笔试及上机考点。

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

上机考题解答与训练. 二级公共基础知识和 Visual FoxPro 语言程序设计 / 房向阳主编. -- 成都: 电子科技大学出版社, 2010.11

(全国计算机等级考试)

ISBN 978-7-5647-0689-0

I. ①上… II. ①房… III. ①电子计算机—水平考试—解题②关系数据库—数据库管理系统, Visual FoxPro—程序设计—水平考试—解题 IV. ①TP3-44

中国版本图书馆 CIP 数据核字(2010)第 212631 号

全国计算机等级考试教程全解·上机考题解答与训练 二级公共基础知识和 Visual FoxPro 数据库程序设计

全国计算机等级考试命题研究组 编
新思路教育科技研究中心

出 版: 电子科技大学出版社(成都市一环路东一段159号电子信息产业大厦 邮编: 610051)
策划编辑: 张 鹏
责任编辑: 张 鹏
主 页: www.uestcp.com.cn
电子邮箱: uestcp@uestcp.com.cn
发 行: 新华书店经销
印 刷: 北京万友印刷有限公司
成品尺寸: 210mm×285mm 印张 15.5 字数 396.8 千字
版 次: 2013年5月第一版
印 次: 2013年5月第一次印刷
书 号: ISBN 978-7-5647-0689-0
定 价: 30.00 元

版权所有 侵权必究

本书如有缺页、破损、装订错误,请寄回印刷厂调换。

邮购热线: 010-62165518 010-59796026

二级 Visual FoxPro 知识点考核点评

1. Visual FoxPro 基础知识

(1) 考核知识点

- 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，方法。
- ② 关系数据库：
 - 关系数据库：关系模型，关系模式，关系，元组，属性，域，主关键字和外部关键字。
 - 关系运算：选择，投影，连接。
 - 数据的一致性和完整性：实体完整性，域完整性，参照完整性。
- ③ Visual FoxPro 系统特点与工作方式：
 - Windows 版本数据库的特点。
 - 数据类型和主要文件类型。
 - 各种设计器和向导。
 - 工作方式：交互方式（命令方式，可视化操作）和程序运行方式。
- ④ Visual FoxPro 的基本数据元素：
 - 常量，变量，表达式。
 - 常用函数：字符处理函数，数值计算函数，日期时间函数，数据类型转换函数，测试函数。

(2) 知识点点评

此部分的重点是 Visual FoxPro 的基本数据元素，多属于理解的知识点，对于函数的运算格式需要大家牢记，经常和多个函数结合在一起考核。其他知识点都属于比较基础的，相对来说比较简单，对于关系数据库中的一些基本概念及运算需要大家了解和掌握。

(3) 考核情况

选择题：所占分值一般在 5 分以上。

操作题：此部分内容只会涉及一些向导及设计器的基本使用，非常简单，很少单独考核，经常是与其他内容融合到一起考核。

2. Visual FoxPro 数据库的基本操作

(1) 考核知识点

- ① 数据库和表的建立、修改与有效性检验：
 - 表结构的建立与修改。
 - 表记录的浏览、增加、删除与修改。
 - 创建数据库，向数据库添加或移出表。
 - 设定字段级规则和记录级规则。
 - 表的索引：主索引，候选索引，普通索引，惟一索引。
- ② 多表操作：
 - 选择工作区。
 - 建立表之间的关联，一对一的关联，一对多的关联。
 - 建立表间临时关联、设置参照完整性。
- ③ 建立视图与数据查询：
 - 查询或视图文件的建立、执行与修改。
 - 建立多表查询或视图。

(2) 知识点点评

此部分内容无论是选择题还是操作题考试都是重点，数据库的完整性及索引的概念是选择题的重点也是难点，利用查询或视图设计器建立查询或视图文件是操作题部分的难点，另外，对于数据库及表的基本概念也需要大家掌握。

(3) 考核情况：

选择题：所占的分值也比较高，基本上都在 8 分以上。

操作题：重点考核，基本上每套题都会涉及到此部分知识点的考核。

3. 关系数据库标准语言 SQL

(1) 考核知识点

- ① SQL 的数据定义功能：CREATE TABLE-SQL 和 ALTER TABLE-SQL 命令。

② SQL 的数据修改功能: DELETE-SQL、INSERT-SQL 和 UPDATE-SQL 命令。

③ SQL 的数据查询功能: 简单查询、嵌套查询、连接查询、分组与计算查询和集合的并运算。

(2) 知识点点评

此部分属于考试的难点,无论是选择题还是操作题中,都占有很重的比例。对于 SQL 的数据定义功能,重点掌握利用 ALTER 命令修改表结构的命令语句;对于 SQL 的数据修改功能,很少单独考核,一般与 SQL 查询语句结合考查;而对于 SQL 的数据查询功能,简单查询和简单连接查询比较容易,SQL 嵌套查询与分组计算查询则属于考试中的难点,也是常考的知识点。

(3) 考核情况

选择题:所占分值在 10 分以上,主要考查 SQL 各种功能的命令关键字,SQL 的查询结果等。

操作题:在第 1、2 大题中会涉及到相关 SQL 语句的考核,但这两大题中所考核的 SQL 语句都比较简单,主要是难在第 3 题的 SQL 语句设计,有可能需要编写很长一段查询代码。

4. 项目管理器、设计器和向导的使用

(1) 考核知识点

① 使用项目管理器:使用“数据”和“文档”选项卡。

② 使用表单设计器:

- 在表单中加入和修改控件对象。
- 设定数据环境。

③ 使用菜单设计器:建立主选项、设计子菜单以及设定菜单选项程序代码。

④ 使用报表设计器:生成快速报表、修改报表布局、设计分组报表和设计多栏报表。

⑤ 使用应用程序向导。

⑥ 应用程序生成器与连编应用程序。

(2) 知识点点评

对项目管理器中各选项卡所包含的文件类型都要有所了解;操作方面,重点需要掌握项目管理器的建立和使用;表单设计在考试中越来越受重视,且考核面也越来越广,难度不断提升。重点是表单及控件的各种属性、事件和方法的设置。难点是事件代码的设计,一般需要结合前面所学的知识点进行设置;菜单和报表的设计在选择题中考核的几率很小,且难度也不大,在操作题考试中,经常考核的是设计一个简单的菜单文件,或利用报表向导建立一个报表,难度也不是很大。

(3) 考核情况

选择题:所占分值在 10 分以上。

操作题:基本上每套题都有表单或控件的相关操作考核,一般包括控件或表单属性的设置、事件代码的编写,在第 3 大题中出现得比较多;关于项目管理器的相关操作,在操作题中考核频率也比较高,重点掌握项目管理器的使用;对于菜单和报表的设计,在第 2 大题简单应用题中出现比较多,第 3 大题中也会出现菜单文件设计的试题。

5. Visual FoxPro 程序设计

(1) 考核知识点

① 命令文件的建立与运行:

- 程序文件的建立。
- 简单的交互式输入、输出命令。
- 应用程序的调试与执行。

② 结构化程序设计:顺序、选择和循环结构程序设计。

③ 过程与过程调用:

- 子程序设计及调用。
- 过程与过程文件。
- 局部变量和全局变量,过程调用中的参数传递。

④ 用户定义对话框(MESSAGEBOX)的使用。

(2) 知识点点评

此部分知识点重点考核的是结构化程序设计的内容,其中,循环结构程序设计是重中之重,选择题中经常是给出一段程序代码,然后要求考生给出程序运行结果。另外,对于局部变量和全局变量的知识,也是经常会考到的。

(3) 考核情况

选择题:所占分值在 1~3 分。

操作题:很少单独考核,考核几率也比较低,一般与 SQL 查询语句结合考查。

编委会

主 编：房向阳

编委组：（排名不分先后）

白 杰	高强华	孟令全	付江辉
王 毅	房向阳	李 婧	刘 明
赵江静	韩雪松	安玉彦	张 龙
刘永阳	马天宝	刘艳飞	柳寒冰
唐保存	黄 和	兰天静	董 昶
黄琴华	王艳梅	金 利	刘本发
齐文斌	迟 剑	李爱菊	张铁军

责任编校：张 鹏

编校组：（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	范 伟	李 瑞
胡昭昀	孟祥萍	张 宇	李金娜

前言

教材是学习一门课程最重要、最基础的学习用书。如果有充沛的时间，教材是每一个考生都应该研习的图书。然而，很多考生复习时间有限，无法将教材从头到尾、深入地学习一遍。此外，普通教材也存在结构复杂庞大，语言严谨生硬，讲述性、辅助性内容过多，而真正考核重点无法凸显等众多缺点。大多数考生如果按照传统的方式一点一点地“啃”教材，势必会花很多的时间和精力，甚至很容易引起疲劳感和挫折感，最终导致不能坚持下去。这就需要一本真正的“高效教材”来满足广大考生的实际需求。

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，组织有着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——一本真正的“高效教材”。

● 知识点全面，考点讲解详细

这本“高效教材”目的就是为了解读复杂的传统教材。教材的核心是考核知识点（即“考点”），抓住这些点就是突破考试的保证，也是考试成功的捷径。而另外一些对考点影响不大甚至没有任何关联的内容，我们尽量地予以精简甚至取缔。有详有略，才能突出重点。这本“高效教材”还会根据考试的实际情况，对课程内容进行提示，分列出重难点，并明确指出考生对于各考点应该掌握的层次要求。“把钢用到刀刃上”，把时间花在有价值的内容上！这就是这本书的精髓。

● 分析历年真卷，科学归纳考点

考点的判断是依据五年来的对历年试题的研究成果和对命题规律的把握而得，所有判断均有具体数据支持。如果你初次接触等级考试，对“考什么”、“如何学”都一无所知。本书就是直接地告诉你这些重要的信息，且分清哪些是重要考点、哪些是必考考点……同时拿出研究数据来，让你真正信服我们的这些判断。

● 真题研密，分块训练

本书除了考点的系统学习之外，还为考生安排了大量不同类型的试题，以练促学、练中复习，有着极高的实用价值。本书所有考点都配有同步历年真题及模拟试题的讲解和训练，且试题的答案及解析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，让考生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。

● 无纸化考试新题型

本书最后安排了数套无纸化考试试题。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，手把手教你掌握每一题的解题思路和过程。

● 配套模拟软件

本书配有强大的考试模拟软件练习光盘，具有如下特点：

- 含多套新版考试模拟试题，让你既可日常练习，强化学习效果；同时也可以模拟考试，自测学习效果，提前感受考场氛围。
- 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。

由于时间仓促，书中难免有不当之处，敬请指出！

目 录

第 1 章 公共基础知识

1.1 考试必备知识	2
1.1.1 数据库结构与算法	2
1.1.2 程序设计基础	8
1.1.3 软件工程基础	9
1.1.4 数据库设计基础	14
1.2 历年试题汇编	19
1.2.1 历年试题	19
1.2.2 参考答案	22
1.3 全真模拟试题汇编	25
1.3.1 模拟试题	25
1.3.2 参考答案	27

第 2 章 Visual FoxPro 数据库基础

2.1 数据库基础知识	30
2.1.1 考试必备知识	30
2.1.2 历年试题汇编	30
2.1.3 全真模拟试题汇编	30
2.2 关系数据库	31
2.2.1 考试必备知识	31
2.2.2 历年试题汇编	32
2.2.3 全真模拟试题汇编	33
2.3 数据库设计基础	33
2.3.1 考试必备知识	33
2.3.2 历年试题汇编	34
2.3.3 全真模拟试题汇编	34
2.4 Visual FoxPro 系统概述	35
2.4.1 考试必备知识	35
2.4.2 历年试题汇编	38
2.4.3 全真模拟试题汇编	38
2.5 项目管理器	38
2.5.1 考试必备知识	39
2.5.2 历年试题汇编	40
2.5.3 全真模拟试题汇编	40
2.6 向导、设计器、生成器简介	41
2.6.1 考试必备知识	41
2.6.2 历年试题汇编	42
2.6.3 全真模拟试题汇编	42

第 3 章 Visual FoxPro 程序设计基础

3.1 常量和变量	44
3.1.1 考试必备知识	44
3.1.2 历年试题汇编	48
3.1.3 全真模拟试题汇编	49
3.2 表达式	50
3.2.1 考试必备知识	51
3.2.2 历年试题汇编	52
3.2.3 全真模拟试题汇编	53
3.3 函数	54
3.3.1 考试必备知识	54
3.3.2 历年试题汇编	61
3.3.3 全真模拟试题汇编	63
3.4 程序与程序文件	64
3.4.1 考试必备知识	64
3.4.2 历年试题汇编	66

3.4.3 全真模拟试题汇编	67
3.5 程序基本结构	68
3.5.1 考试必备知识	68
3.5.2 历年试题汇编	70
3.5.3 全真模拟试题汇编	70
3.6 多模块程序设计	72
3.6.1 考试必备知识	72
3.6.2 历年试题汇编	75
3.6.3 全真模拟试题汇编	76

第 4 章 Visual FoxPro 数据库及其操作

4.1 Visual FoxPro 数据库及其建立	78
4.1.1 考试必备知识	78
4.1.2 历年试题汇编	79
4.1.3 全真模拟试题汇编	79
4.2 建立数据库表	80
4.2.1 考试必备知识	80
4.2.2 历年试题汇编	82
4.2.3 全真模拟试题汇编	83
4.3 表的基本操作	83
4.3.1 考试必备知识	83
4.3.2 历年试题汇编	87
4.3.3 全真模拟试题汇编	88
4.4 索引	90
4.4.1 考试必备知识	90
4.4.2 历年试题汇编	92
4.4.3 全真模拟试题汇编	92
4.5 数据完整性	94
4.5.1 考试必备知识	94
4.5.2 历年试题汇编	95
4.5.3 全真模拟试题汇编	96
4.6 自由表	97
4.6.1 考试必备知识	97
4.6.2 历年试题汇编	98
4.6.3 全真模拟试题汇编	98
4.7 多个表的同时使用	99
4.7.1 考试必备知识	99
4.7.2 历年试题汇编	100
4.7.3 全真模拟试题汇编	101
4.8 排序	102
4.8.1 考试必备知识	102
4.8.2 历年试题汇编	102
4.8.3 全真模拟试题汇编	102

第 5 章 关系数据库标准语言 SQL

5.1 SQL 概述	104
5.1.1 考试必备知识	104
5.1.2 历年试题汇编	104
5.1.3 全真模拟试题汇编	105
5.2 SQL 的查询功能建立	105
5.2.1 考试必备知识	105
5.2.2 历年试题汇编	108
5.2.3 全真模拟试题汇编	113
5.3 SQL 的操作功能	118
5.3.1 考试必备知识	118
5.3.2 历年试题汇编	119

5.3.3 全真模拟试题汇编	120
5.4 SQL 的定义功能	122
5.4.1 考试必备知识	122
5.4.2 历年试题汇编	123
5.4.3 全真模拟试题汇编	124

第 6 章 查询与视图

6.1 查 询	128
6.1.1 考试必备知识	128
6.1.2 历年试题汇编	130
6.1.3 全真模拟试题汇编	131
6.2 视 图	132
6.2.1 考试必备知识	132
6.2.2 历年试题汇编	134
6.2.3 全真模拟试题汇编	135

第 7 章 表单设计与应用

7.1 面向对象的概念	138
7.1.1 考试必备知识	138
7.1.2 历年试题汇编	138
7.1.3 全真模拟试题汇编	139
7.2 Visual FoxPro 中的类	139
7.2.1 考试必备知识	139
7.2.2 历年试题汇编	141
7.2.3 全真模拟试题汇编	142
7.3 表单的创建和运行	143
7.3.1 考试必备知识	143
7.3.2 历年试题汇编	144
7.3.3 全真模拟试题汇编	144
7.4 表单设计器	144
7.4.1 考试必备知识	144
7.4.2 历年试题汇编	146
7.4.3 全真模拟试题汇编	147
7.5 表单的事件、属性和方法	148
7.5.1 考试必备知识	148
7.5.2 历年试题汇编	150
7.5.3 全真模拟试题汇编	151
7.6 基本型控件	152
7.6.1 考试必备知识	152
7.6.2 历年试题汇编	154
7.6.3 全真模拟试题汇编	155
7.7 容器类控件	156
7.7.1 考试必备知识	156
7.7.2 历年试题汇编	158
7.7.3 全真模拟试题汇编	158
7.8 自定义类	159
7.8.1 考试必备知识	159
7.8.2 历年试题汇编	162
7.8.3 全真模拟试题汇编	162

第 8 章 菜单的设计与应用

8.1 Visual FoxPro 系统菜单	164
8.1.1 考试必备知识	164
8.1.2 历年试题汇编	165
8.1.3 全真模拟试题汇编	165
8.2 菜单的基本设计	166
8.2.1 考试必备知识	166
8.2.2 历年试题汇编	168
8.2.3 全真模拟试题汇编	168

8.3 快捷菜单设计	169
8.3.1 考试必备知识	169
8.3.2 历年试题汇编	169
8.3.3 全真模拟试题汇编	169

第 9 章 报表的设计与应用

9.1 创建报表	172
9.1.1 考试必备知识	172
9.1.2 历年试题汇编	174
9.1.3 全真模拟试题汇编	174
9.2 设计报表	175
9.2.1 考试必备知识	175
9.2.2 历年试题汇编	178
9.2.3 全真模拟试题汇编	178
9.3 数据分组和多栏报表	179
9.3.1 考试必备知识	179
9.3.2 历年试题汇编	180
9.3.3 全真模拟试题汇编	180

第 10 章 应用程序的开发和生成

10.1 应用程序项目综合实践	182
10.1.1 考试必备知识	182
10.1.2 历年试题汇编	184
10.1.3 全真模拟试题汇编	184
10.2 使用应用程序生成器	185
10.2.1 考试必备知识	185
10.2.2 历年试题汇编	187
10.2.3 全真模拟试题汇编	187

第 11 章 无纸化考试试题

第 1 套 无纸化考试试题	227
第 2 套 无纸化考试试题	194
第 3 套 无纸化考试试题	200
第 4 套 无纸化考试试题	204
第 1 套 无纸化考试试题答案及详解	211
第 2 套 无纸化考试试题答案及详解	218
第 3 套 无纸化考试试题答案及详解	224
第 4 套 无纸化考试试题答案及详解	230

第1章 公共基础知识

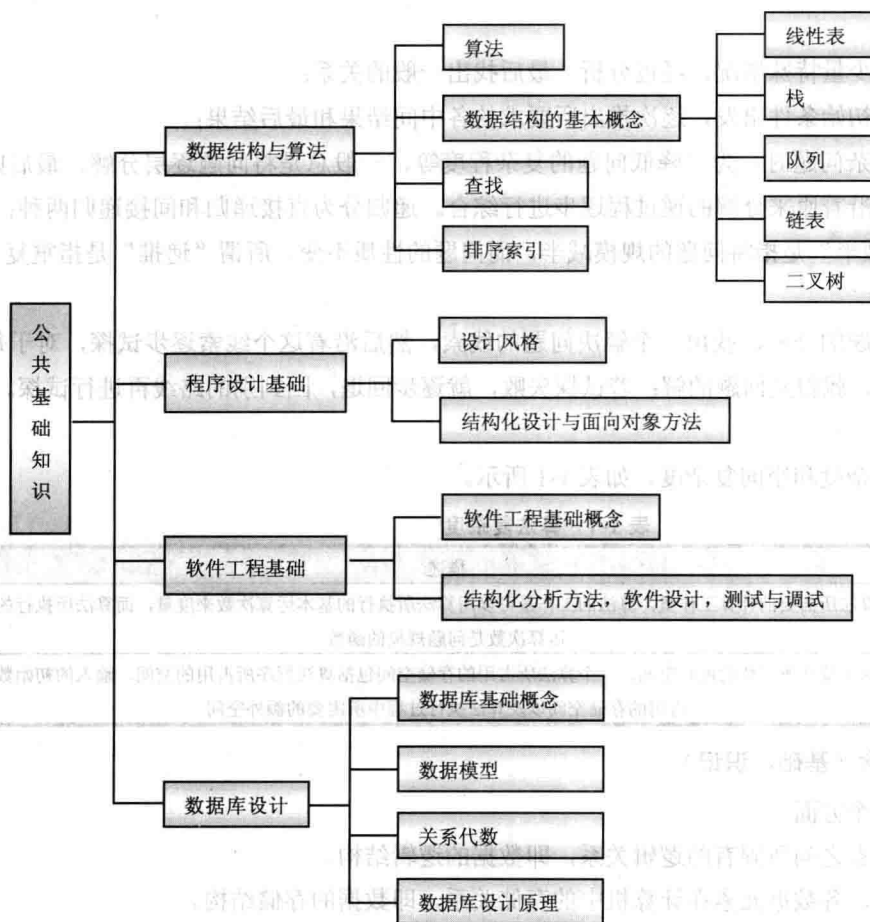
本章导读

本章介绍了公共基础知识的相关知识。本章内容在选择题中所占比例基本是固定的。作为后面章节的学习基础，建议考生理解学习，以便更好地掌握本章的相关概念和知识点。

本章考试大纲

- (1) 数据结构与算法（基础，识记）
- (2) 程序设计基础（基础，理解）
- (3) 软件工程基础（基础，理解）
- (4) 数据库基础（基础，了解）

本章知识的逻辑结构图



1.1 考试必备知识

1.1.1 数据库结构与算法

1. 算法（基础，识记）

(1) 算法的基本概念

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。所谓算法是对解题方案准确而完善的描述。

①算法的基本特征：可行性、确定性、有穷性、拥有足够的情报。

②基本运算和操作包括：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。

- 列举法：根据提出的问题，列举所有可能的情况，并用问题中给定的条件检验哪些是需要的，哪些是不需要的；
- 归纳法：通过列举少量特殊情况，经过分析，最后找出一般的关系；
- 递推：指从已知的初始条件出发，逐次推出所要求的各中间结果和最后结果；
- 递归：解决一些复杂问题时，为了降低问题的复杂程度等，一般总是将问题逐层分解，最后归结为一些简单的问题，再沿着原来分解的逆过程逐步进行综合。递归分为直接递归和间接递归两种；
- 减半递推技术：“减半”是指将问题的规模减半，而问题的性质不变。所谓“递推”是指重复“减半”过程；
- 回溯法：通过对问题的分析，找出一个解决问题的线索，然后沿着这个线索逐步试探，对于每一步的试探，若试探成功，就得到问题的解；若试探失败，就逐步回退，回到别的路线再进行试探。

(2) 算法复杂度

算法复杂度包括时间复杂度和空间复杂度。如表 1-1 所示。

表 1-1 算法复杂度

名称	描述
时间复杂度	指执行算法所需要的计算工作量。算法的工作量可以用算法所执行的基本运算次数来度量，而算法所执行的基本运算次数是问题规模的函数
空间复杂度	指执行这个算法所需要的内存空间。一个算法所占用的存储空间包括算法程序所占用的空间、输入的初始数据所占用的存储空间以及算法执行过程中所需要的额外空间

2. 数据结构的基本概念（基础，识记）

(1) 数据结构研究的3个方面

①数据集中各数据元素之间所固有的逻辑关系，即数据的逻辑结构。

②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。

③对各种数据结构进行的运算。

(2) 数据结构

数据结构是指相互有关联的数据元素的集合。

①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。

②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的信息。

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

- 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中，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。

(3) 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：

- 有且只有一个根结点；
- 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、串等都是线性结构。

②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。

3. 线性表（基础，识记）

(1) 线性表的基本概念

线性表是由 $n(n \geq 0)$ 个数据元素 $a_1, a_2, \dots, a_n$ 组成的一个有限序列，表中的每一个数据元素，除了第一个外，有且只有一个前件，除了最后一个外，有且只有一个后件。

(2) 线性表的顺序存储结构的基本特点

- 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。

元素 a_i 的存储地址为： $ADR(a_i) = ADR(a_1) + (i-1)k$ ， $ADR(a_1)$ 为第一个元素的地址， k 代表每个元素占的字节数。

(3) 线性表的运算

线性表的运算有查找、插入、删除 3 种。

①线性表的插入。

- 插入的实现方法：要在第 $i(1 \leq i \leq n)$ 个元素之前插入一个新元素时，首先要从最后一个元素开始，直到第 i 个元素之间的 $n-i+1$ 个元素依次向后移动一个位置，移动结束后，第 i 个元素位置被空出，然后将新元素插入到第 i 项。插入结束后，线性表的长度就增加 1。
- 线性表插入操作算法的复杂度分析：由线性表插入过程可知，如果插入位置在第 $i(1 \leq i \leq n)$ 个元素之前，则原来第 i 个元素之后的所有元素都必须向后移动。在平均情况下，要在线性表中插入一个新元素，需要移动表中一半的元素，最坏情况下要移动表中的所有元素。

②线性表的删除。

- 删除的实现方法：要删除第 $i(1 \leq i \leq n)$ 个元素时，首先要从第 $i+1$ 个元素开始，直到最后一个元素之间的 $n-i$ 个元素依次向前移动一个位置。删除结束后，线性表的长度就减少 1。
- 线性表删除操作算法的复杂度分析：如果删除的是第 $i(1 \leq i \leq n)$ 个元素，则原来第 i 个元素之后的所有

元素都必须向前移动一个位置。在平均情况下,要在线性表中删除一个已有的元素,需要移动表中一半的元素,最坏情况下要移动表中的所有元素。

(4) 线性表顺序存储结构的适用场合

线性表的顺序存储结构对于小线性表或其中元素不常变动的线性表来说是合适的,因为顺序存储的结构比较简单。但是这种顺序存储结构对于元素需要变动的大线性表就不合适了,因为插入和删除的效率比较低。

4. 栈(基础,识记)

(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,既不允许进行插入元素,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,从而也是最先被删除的元素;栈底元素总是最先被插入的元素,从而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。例如,枪械的子弹匣就可以用来形象地表示栈结构。子弹匣的一端是完全封闭的,最后被压入弹匣的子弹总是最先被弹出,而最先被压入的子弹最后被弹出。

(2) 栈的顺序存储及其运算

栈的基本运算有三种: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。首先将栈顶指针加1,然后将新元素插入到栈顶指针指向的位置。

②退栈运算。取出栈顶元素并赋给一个指定变量。首先将栈顶元素赋给一个指定的变量,然后将栈顶指针减1。当栈顶指针为0时,说明栈空,不可能进行退栈操作。此时,发生栈“下溢”错误。

③读栈顶元素。将栈顶元素赋给一个指定的变量,栈指针不变。当栈顶指针为0时,读栈顶元素操作失败。

5. 队列(基础,识记)

(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,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。若有队列:

$$Q = (q_1, q_2, \dots, q_n)$$

那么, q_1 为队头元素(排头元素), q_n 为队尾元素。队列中的元素是按照 $q_1, q_2, \dots, q_n$ 顺序进入的,退出队列也只能按照这个次序依次退出,即只有在 $q_1, q_2, \dots, q_{n-1}$ 都退队之后, q_n 才能退出队列。因最先进入队列的元素将最先出队,所以队列具有先进先出特性,体现“先来先服务”的原则。

队头元素 q_1 是最先被插入的元素,也是最先被删除的元素。队尾元素 q_n 是最后被插入的元素,也是最后被删除的元素。因此,与栈相反,队列又称为“先进先出”(First In First Out, FIFO)或“后进后出”(Last In Last Out, LILO)的线性表。

(2) 队列运算

①入队运算是往队列队尾插入一个数据元素。首先将队尾指针加1(rear=rear+1),并当rear=m+1(m表示队列的长度)时置rear=1;然后将新元素插入到指针指向的位置。

②退队运算是从队列的队头删除一个数据元素。首先将排头指针加1(front=front+1),并当front=m+1(m表示队列的长度)时置front=1;然后将排头指针指向的元素赋给指定变量。

(3) 循环队列

在实际应用中,队列的顺序存储结构一般采用循环队列的形式。所谓循环队列,就是将队列存储空间的一个位置绕到第一个位置,形成逻辑上的环状空间。循环队列s=0表示队列空;s=1且front=rear表示队列满。计算循环队列的元素个数:“尾指针减头指针”,若为负数,再加其容量即可。

6. 链表（基础，识记）

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

①线性表的链式存储结构称为线性链表。

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。

②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。

线性单链表中，HEAD 称为头指针，HEAD=NULL(或 0)称为空表。如果是双向链表的两指针：左指针(Link)指向前件结点，右指针(Rlink)指向后件结点。

（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。

（3）带链的队列

队列的链式结构和线性链表的存储结构基本相同。只不过队列的链式结构保持有两个指针，一个是指向队列头的头指针，另一个是指向队列尾的尾指针。

（4）线性链表的基本运算

线性链表的基本运算：查找、插入、删除。

①线性链表的查找。从头指针指向的结点开始往后沿指针进行扫描，直到后面已没有结点或下一个结点的数据域为查找值为止。

②线性链表的插入。先从栈中为新元素分配一个新的结点，并赋值。然后利用线性链表的查找算法找到待插入位置的前一个结点指针，将新结点指向该结点的后件，然后将新结点挂在该结点的后面。

③线性链表的删除。利用线性链表的查找算法找到要删除元素的前一个结点，将该结点的指针指向删除元素的结点的指针指向的结点，将包含删除元素的结点送回可利用栈。

7. 二叉树（基础，识记）

（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：

- 非空二叉树只有一个根结点；
- 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。

例如，一个家族中的族谱关系如图 1-1 所示：

A 有后代 B，C；

B 有后代 D，E；

C 有后代 F；

典型的二叉树如图 1-1 所示：

如表 1-2 所示，详细讲解二叉树的一些基本概念。

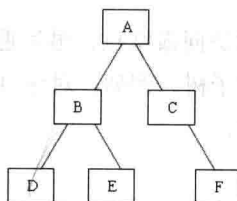


图 1-1 族谱二叉树

表 1-2 二叉树概念

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。例如, 在图 1-1 中, 结点 A 是树的根结点
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。例如, 在图 1-1 中, 结点 D, E, F 均为叶子结点
度	在树结构中, 一个结点所拥有的后件个数称为该结点的度, 所有结点中最大的度称为树的度。例如, 在图 1-1 中, 根结点 A 和结点 B 的度为 2, 结点 C 的度为 1, 叶子结点 D, E, F 的度为 0。所以, 该树的度为 2
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。例如, 在图 1-1 中, 根结点 A 在第 1 层, 结点 B, C 在第 2 层, 结点 D, E, F 在第 3 层。该树的深度为 3
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

②二叉树基本性质。

二叉树具有以下几个性质:

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

性质 2: 深度为 m 的二叉树最多有 $2^m - 1$ 个结点;

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。

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。

(2) 满二叉树与完全二叉树

满二叉树是指除最后一层外, 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, 若其右分支下子孙结点的最大层次为 p , 则其左分支下子孙结点的最大层次为 p , 或为 $p+1$ 。

完全二叉树具有以下两个性质:

性质 5: 具有 n 个结点的完全二叉树的深度为 $\lceil \log_2 n \rceil + 1$ 。

性质 6: 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:

①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 $\text{INT}(k/2)$ 。

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)。

③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。

(3) 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 3 类: 前序遍历、中序遍历和后序遍历。

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, 仍然先访问根结点, 然后遍历左子树, 最后遍历右子树。例如, 对图 1-1 中二叉树进行前序遍历的结果 (或称为该二叉树的前序序列) 为: A, B, D, E, C, F。

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。例如, 对图 1-1 中二叉树进行中序遍历的结果 (或称为该二叉树的中序序列) 为: D, B, E, A, C, F。

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。例如, 对图 1-1 中二叉树进行后序遍历的结果 (或称为该二叉树的后序序列)

为: D, E, B, F, C, A。

8. 查找 (基础, 识记)

(1) 顺序查找

顺序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。

在下列两种情况下只能采用顺序查找:

- 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。

在本书中, 为了简化问题, 更方便讨论, “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。后面讲解的排序中, 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。

顺序查找法每一次比较, 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。

9. 排序 (基础, 识记)

(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), 通过一趟排序, 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。

1.1.2 程序设计基础

1. 程序设计的方法与风格（基础，了解）

养成良好的程序设计风格，主要考虑下述因素：

（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。
- 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

为使程序中的数据说明易于理解和维护，可采用的数据说明风格，如表 1-3 所示。

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，应通过注释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。对语句的结构安排要遵循以下原则：

- 在一行内只写一条语句，程序编写应优先考虑清晰性；
- 除非对效率有特殊要求，语句结构清晰第一，效率第二；
- 首先要保证程序正确，然后才要求提高速度；
- 避免使用临时变量而使程序的可读性下降；
- 避免不必要的转移，尽可能使用库函数；
- 尽量减少使用“否定”条件语句，避免采用复杂的条件语句；
- 要模块化，使模块的功能尽可能单一化。

（4）输入和输出

输入输出比较简单，对输入输出界面要注意以下几点：

- 对所有输入数据进行合法检验；
- 输入格式要尽量简单，应允许使用自由格式与默认值；
- 输入批量数据时，做好结束标志；
- 交互方式输入时，在屏幕上要显示提示信息；
- 当程序设计对输入格式有严格要求时，应保持输入格式与输入语句的一致性。

2. 结构化程序设计（基础，了解）

（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；先考虑全局目标，再考虑局部目标。
- 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，它包括简单选择和多分支选择结构。