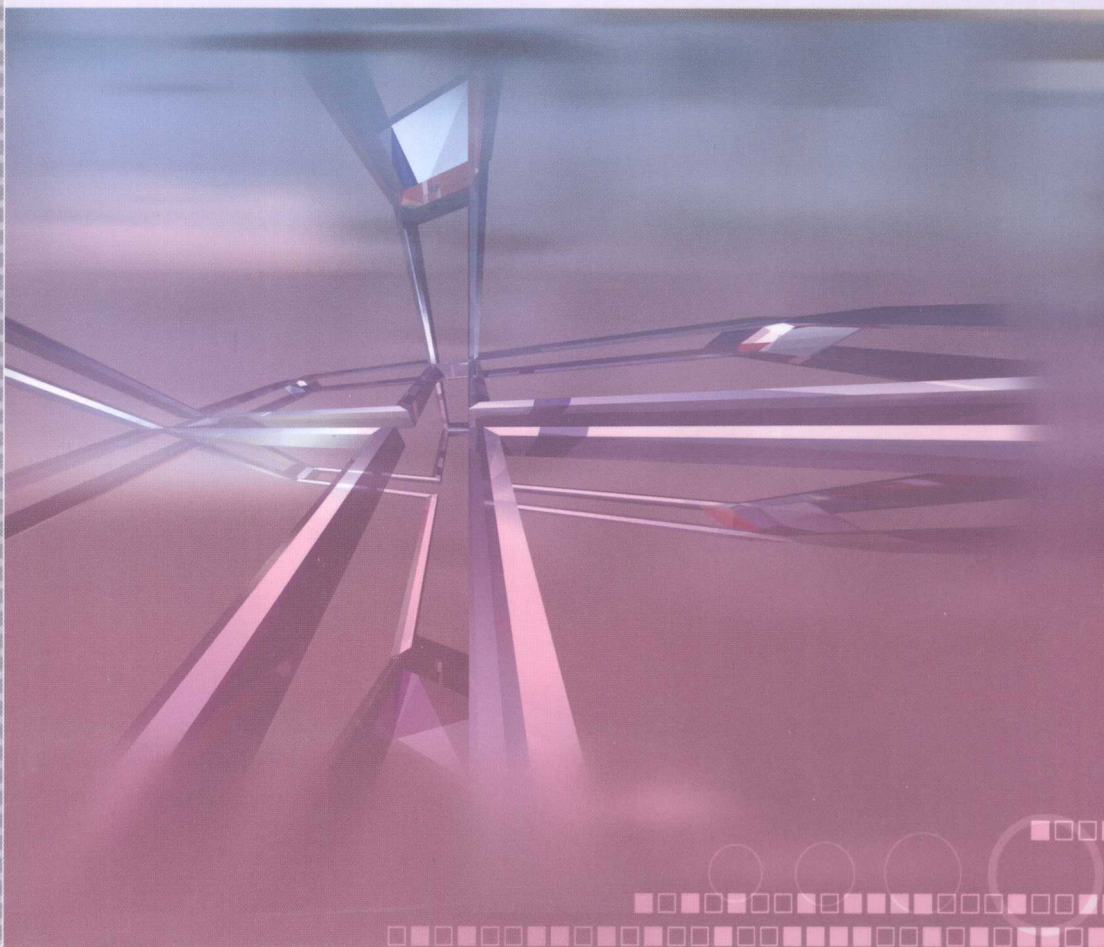




高职高专计算机实用教程系列规划教材

Visual FoxPro 程序设计

黄金凤 主编 周 琼 梁小利 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机实用教程系列规划教材

Visual FoxPro 程序设计

黄金凤 主 编

周 琼 梁小利 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本教材按照全国计算机等级考试二级 Visual FoxPro 考试大纲的要求,由从事精品课“数据库及程序设计”教学和教材建设的专业教师编写。

本教材将教学和实用技术相结合,理论联系实际,注重培养学生的实际应用能力。全书分为上下两篇共 10 章,分别介绍数据结构与算法、程序设计基础、软件工程基础、数据库基本原理、Visual FoxPro 6.0 基础、Visual FoxPro 基本数据元素、Visual FoxPro 数据库的基本操作、关系型数据库标准语言 SQL、项目管理器、设计器和向导使用、Visual FoxPro 程序设计基础,最后附有 Visual FoxPro 数据库程序设计试卷套题。

本书适合作为高等院校、高等职业技术学院的学生教材,也可以作为参加计算机等级考试的考生和计算机专业人员的参考书。

图书在版编目(CIP)数据

Visual FoxPro 程序设计/黄金凤主编. —北京:中国铁道出版社, 2008. 1

(高职高专计算机实用教程系列规划教材)

ISBN 978-7-113-08544-5

I. V… II. 黄… III. 关系数据库—数据库管理系统, Visual FoxPro—程序设计—高等学校:技术学校—教材
IV. TP311. 138

中国版本图书馆 CIP 数据核字 (2008) 第 014220 号

书 名: Visual FoxPro 程序设计

作 者: 黄金凤 周 琼 梁小利

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 秦绪好

责任编辑: 王春霞

特邀编辑: 薛秋沛

封面设计: 付 巍

封面制作: 白 雪

责任校对: 刘彦会

印 刷: 北京市兴顺印刷厂

开 本: 787×1092 1/16 印张: 15.75 字数: 366 千

版 本: 2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 978-7-113-08544-5/TP·2678

定 价: 24.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

为了提高全社会计算机应用水平,普及计算机知识,适应国民经济信息化的需求,顺应用人单位对计算机能力的要求,本书编者结合全国计算机等级考试,针对目前高职高专的培养要求,根据 2007 年计算机等级 Visual FoxPro 6.0 考试大纲,编写了《Visual FoxPro 程序设计》。

本书围绕学生掌握 Visual FoxPro 编程的基本方法、提高学生 Visual FoxPro 的使用与开发能力三方面来组织内容,主要以满足高职高专学生需求,对其他不同层次需要学习和使用 Visual FoxPro 人员(尤其是报考全国计算机二级考试人员)也同样适用。

本书的特点是针对全国计算机等级考试,设计公共基础部分和 Visual FoxPro 程序设计部分,剖析典型等级考试题并设计自测练习题。详细分析和覆盖等级考试大纲与试题,帮助考生熟悉等级考试知识点,给出大量的自测练习题,方便考生自我检查。本书的设计既有利于初学者尽快掌握必备的知识,又有利于以后进一步提高,为掌握中小型数据库的设计打下扎实的基础。本书给出的例题和程序全部在 Visual FoxPro 6.0 环境下调试通过。

本书由黄金凤主编,负责教材的大纲确定和主要章节的编写,周琼、梁小利任副主编。参编人员有曾凌静、郑瑾、于淑云、吴忠斌和张传娟,本书是大家共同研究与协作的结果。

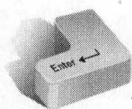
由于时间仓促,加之编者水平有限,书中疏漏之处在所难免,在此恳请广大读者批评指正。

编 者
2008 年 1 月

目 录

公共基础部分（上篇）

第 1 章 数据结构与算法	2
1.1 算法	2
1.1.1 算法的基本概念	2
1.1.2 算法的复杂度	4
1.2 数据结构	5
1.2.1 数据结构的定义	5
1.2.2 数据结构的图形表示	6
1.2.3 线性结构与非线性结构	6
1.3 线性表及顺序存储结构	7
1.3.1 线性表的定义	7
1.3.2 线性表的顺序存储结构	7
1.3.3 线性表的插入运算	8
1.3.4 线性表的删除运算	8
1.4 栈和队列	9
1.4.1 栈及其基本运算	9
1.4.2 队列及其基本运算	10
1.5 线性链表	11
1.5.1 线性单链表的结构及其基本运算	11
1.5.2 线性链表的基本运算	12
1.5.3 线性双向链表的结构及其基本运算	13
1.5.4 循环链表的结构及其基本运算	14
1.6 树与二叉树	14
1.6.1 树的基本概念	14
1.6.2 二叉树及其基本性质	15
1.6.3 二叉树的存储结构	18
1.6.4 二叉树的遍历	18
1.7 查找技术	19
1.7.1 顺序查找	19
1.7.2 二分法查找	19
1.8 排序技术	20



1.8.1	交换类排序法	20
1.8.2	插入类排序法	20
1.8.3	选择类排序法	21
1.9	本章小结	21
习题 1		22
第 2 章	程序设计基础	23
2.1	程序设计风格与方法	23
2.2	结构化程序设计	24
2.2.1	结构化程序设计的原则	24
2.2.2	结构化程序设计的基本结构与特点	25
2.2.3	结构化程序设计原则和方法的应用	26
2.3	面向对象的程序设计	26
2.3.1	关于面向对象方法	26
2.3.2	面向对象方法的基本概念	27
2.4	本章小结	29
习题 2		30
第 3 章	软件工程基础	31
3.1	软件工程概述	31
3.1.1	软件的组成与软件特点	31
3.1.2	软件危机与软件工程	32
3.1.3	软件工程过程	33
3.1.4	软件生命周期	33
3.1.5	软件工程的目标与原则	34
3.1.6	软件开发工具与软件开发环境	35
3.2	结构化分析方法概述	35
3.2.1	需求分析与需求分析方法	35
3.2.2	结构化分析方法	36
3.2.3	软件需求规格说明书	37
3.3	结构化设计方法	38
3.3.1	软件设计的基本概念	38
3.3.2	概要设计任务	40
3.3.3	面向数据流的设计方法	41
3.3.4	设计准则	42
3.3.5	详细设计	43
3.4	软件测试	46
3.4.1	软件测试的目的和准则	46
3.4.2	软件测试的技术与方法	46

3.4.3	软件测试的实施.....	48
3.5	程序的调试.....	49
3.5.1	程序调试的概念.....	49
3.5.2	程序调试的方法.....	50
3.6	软件工程管理.....	51
3.6.1	软件工程管理的职能.....	51
3.6.2	进度安排.....	51
3.6.3	标准化.....	51
3.6.4	软件配置.....	51
3.6.5	软件产权保护.....	52
3.7	本章小结.....	52
	习题 3.....	52
第 4 章	数据库基本原理.....	54
4.1	数据库技术.....	54
4.1.1	数据库的基本概念.....	54
4.1.2	数据管理技术的发展.....	56
4.1.3	数据库系统的组成.....	57
4.1.4	数据库系统的内部结构体系.....	60
4.2	数据模型.....	61
4.2.1	层次模型.....	61
4.2.2	网状模型.....	62
4.2.3	关系模型.....	62
4.2.4	实体-联系模型.....	63
4.2.5	面向对象模型.....	65
4.3	关系数据库.....	65
4.3.1	关系数据库的基本概念.....	65
4.3.2	关系模型.....	66
4.3.3	关系运算.....	67
4.4	数据库的设计与管理.....	70
4.4.1	数据库设计概述.....	70
4.4.2	数据库设计的需求分析.....	71
4.4.3	数据库概念设计.....	72
4.4.4	数据库的逻辑设计.....	72
4.4.5	数据库的物理设计.....	73
4.4.6	数据库管理.....	73
4.5	本章小结.....	75
	习题 4.....	76



Visual FoxPro 程序设计部分（下篇）

第 5 章 Visual FoxPro 6.0 基础	78
5.1 启动和退出 Visual FoxPro 6.0	78
5.1.1 启动 Visual FoxPro 6.0	78
5.1.2 Visual FoxPro 6.0 工作环境	79
5.1.3 退出 Visual FoxPro 6.0	79
5.2 设置工作环境	80
5.2.1 使用命令行开关	80
5.2.2 使用“选项”命令	80
5.2.3 使用 SET 命令	80
5.2.4 使用配置文件	81
5.3 Visual FoxPro 6.0 的工作方式	83
5.4 Visual FoxPro 6.0 的文件类型	83
5.4.1 数据库文件	83
5.4.2 文档文件	83
5.4.3 程序文件	84
5.5 项目管理器	85
5.5.1 创建项目	85
5.5.2 使用项目管理器	86
5.5.3 定制项目管理器	87
5.6 本章小结	87
习题 5	87
第 6 章 Visual FoxPro 基本数据元素	89
6.1 常量、变量和数据类型	89
6.1.1 常量	89
6.1.2 变量	89
6.1.3 内存变量的基本操作	89
6.1.4 数据类型	93
6.1.5 数组	94
6.2 表达式与运算符	94
6.2.1 基本概念	94
6.2.2 数值、字符与日期时间表达式	95
6.2.3 关系表达式	96
6.2.4 逻辑表达式	97
6.2.5 各种运算的优先级	97
6.3 常用函数	98

6.3.1	数值运算函数	98
6.3.2	字符处理函数	100
6.3.3	日期时间函数	103
6.3.4	数据类型转换函数	104
6.3.5	测试函数	105
6.3.6	其他函数	109
6.4	本章小结	109
习题 6		110
第 7 章	Visual FoxPro 数据库的基本操作	112
7.1	Visual FoxPro 项目	112
7.1.1	创建 Visual FoxPro 项目	112
7.1.2	打开 Visual FoxPro 项目	113
7.2	Visual FoxPro 数据库	113
7.2.1	创建数据库	113
7.2.2	打开数据库	115
7.2.3	指定数据库	116
7.2.4	修改数据库	116
7.2.5	关闭数据库	117
7.2.6	删除数据库	117
7.3	Visual FoxPro 表	118
7.3.1	表的介绍	118
7.3.2	表的创建	120
7.3.3	自由表和数据库中表的互换	121
7.3.4	表设计器	122
7.3.5	表结构的建立	126
7.3.6	表结构的修改	126
7.3.7	表记录的输入	128
7.3.8	与“学生成绩”管理相关表的建立	128
7.3.9	表的重命名	129
7.3.10	表的删除	130
7.4	数据记录的操作	131
7.4.1	数据表的打开和关闭	132
7.4.2	使用浏览窗口新增、修改与删除数据记录	137
7.4.3	表的命令操作方式	140
7.4.4	数据检索	147
7.5	多表的操作	152
7.5.1	关联方式的分类	152



80	7.5.2 创建关系	153
001	7.5.3 编辑关系	154
801	7.5.4 删除连接	154
401	7.5.5 数据表之间的参照完整性	154
201	7.6 查询与视图	155
901	7.6.1 查询与视图的介绍	155
901	7.6.2 使用查询设计器建立查询	156
101	7.6.3 使用视图设计器建立本地视图	160
511	7.6.4 使用视图更新数据	162
115	7.7 本章小结	163
511	习题 7	163
151	第 8 章 关系型数据库标准语言 SQL	166
811	8.1 SQL 概述	166
811	8.2 SQL 的数据定义功能	167
211	8.2.1 建立新表	167
811	8.2.2 为表添加新字段	168
811	8.2.3 编辑表中已有字段	169
911	8.2.4 删除表中指定字段	169
711	8.3 SQL 的数据操纵功能	170
811	8.3.1 插入记录	170
811	8.3.2 更新记录	170
051	8.3.3 删除记录	171
151	8.4 SQL 的数据查询功能	171
551	8.4.1 简单查询	173
551	8.4.2 嵌套查询	174
851	8.4.3 联接查询	175
851	8.4.4 分组计算查询	176
851	8.4.5 集合的并运算	177
951	8.5 视图	177
131	8.5.1 视图的概念及其定义	177
131	8.5.2 视图的删除	177
551	8.5.3 关于视图的说明	178
751	8.6 本章小结	178
041	习题 8	178
125	第 9 章 项目管理器、设计器和向导使用	180
125	9.1 项目管理器的功能和使用	180
125	9.1.1 项目管理器的功能	180

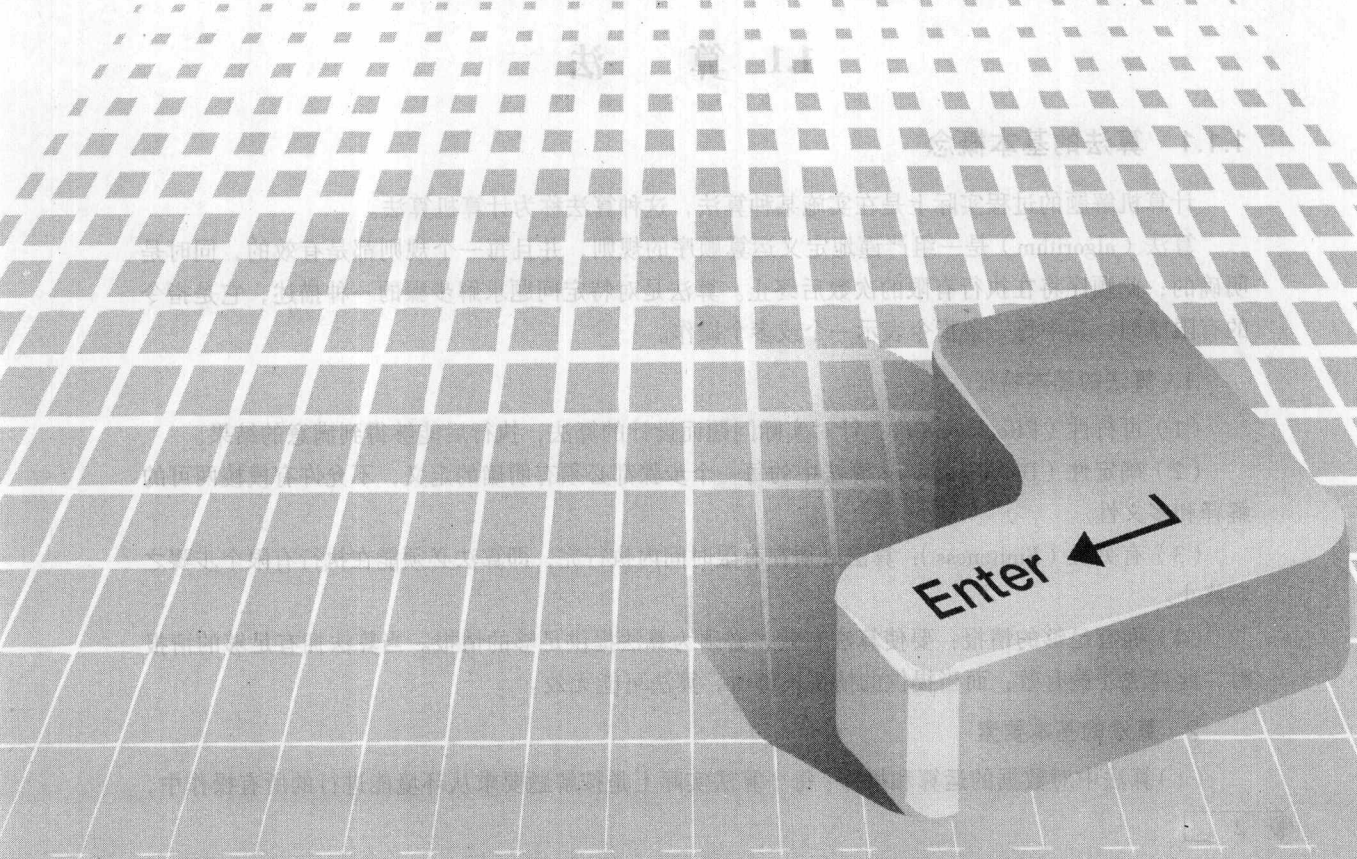
9.1.2 项目管理器的使用	180
9.2 设计器的使用	185
9.2.1 表单设计器的功能和使用	185
9.2.2 菜单设计器的功能和使用	190
9.2.3 报表设计器的功能和使用	194
9.3 向导的使用	198
9.3.1 表单向导的使用	200
9.3.2 报表向导的使用	201
9.4 本章小结	203
习题 9	204
第 10 章 Visual FoxPro 程序设计基础	205
10.1 程序文件的建立与运行	205
10.1.1 程序文件的建立与编辑	205
10.1.2 程序文件的运行	206
10.1.3 程序的终止运行	206
10.1.4 程序的书写规则	206
10.2 简单的输入/输出命令	207
10.3 程序设计的基本结构	209
10.3.1 顺序结构	209
10.3.2 选择结构	209
10.3.3 循环结构	211
10.4 模块化程序设计	216
10.4.1 过程与模块	216
10.4.2 过程的带参调用	218
10.4.3 自定义函数	218
10.4.4 内存变量的作用域	219
10.5 本章小结	221
习题 10	221
参考文献	224
附录 A Visual FoxPro 数据库程序设计试卷套题（一）	225
附录 B Visual FoxPro 数据库程序设计试卷套题（二）	230
附录 C 参考答案	235

公共基础部分（上篇）

第一章

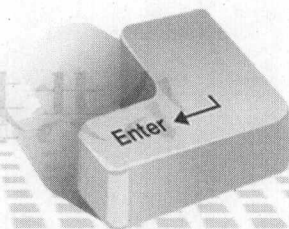
第一章 公共基础知识

公共基础知识是公务员录用考试的重要组成部分，也是公务员录用考试的重要内容。本章主要介绍公共基础知识的基本概念、基本理论、基本方法和基本技能。本章共分三节，第一节介绍公共基础知识的基本概念，第二节介绍公共基础知识的基本理论，第三节介绍公共基础知识的基本方法和基本技能。



第1章

数据结构与算法



从事各种工作和活动，都必须事先想好进行的步骤，然后按部就班地执行，这样才能避免产生错乱。因此为解决一个问题而采取的方法和步骤就称为“算法”。

对同一个问题，可以有不同的解题方法和步骤。方法有优劣之分。有的方法只需进行很少的步骤，而有些方法则需要较多的步骤。一般来说，希望采用简单且运算步骤少的方法。因此，为了有效地进行解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适的算法。

1.1 算 法

1.1.1 算法的基本概念

计算机解题的过程实际上是在实施某种算法，这种算法称为计算机算法。

算法 (algorithm) 是一组严谨地定义运算顺序的规则，并且每一个规则都是有效的，同时是明确的；此顺序将在执行有限的次数后终止。算法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。

1. 算法的基本特征

(1) 可行性 (Effectiveness)：针对实际问题而设计的算法，执行后能够得到满意的结果。

(2) 确定性 (Definiteness)：算法中的每一个步骤都必须有明确的定义，不允许有模棱两可的解释和多义性。

(3) 有穷性 (Finiteness)：算法必须在有限时间内执行完，即算法必须能在执行有限个步骤之后终止。

(4) 拥有足够的情报：要使算法有效，必须为算法提供足够的情报。当算法拥有足够的情报时，此算法才最有效；而当提供的情报不够时，算法可能无效。

2. 算法的基本要素

(1) 算法中对数据的运算和操作：每个算法实际上是按解题要求从环境能进行的所有操作中，

选择合适的操作所组成的一组指令序列。

计算机可以执行的基本操作是以指令的形式描述的。一个计算机系统能执行的所有指令的集合,称为该计算机系统的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令所组成的指令序列。在一般的计算机系统中,基本的运算和操作有以下4类。

① 算术运算:主要包括加、减、乘、除等运算。

② 逻辑运算:主要包括“与”、“或”、“非”等运算。

③ 关系运算:主要包括“大于”、“小于”、“等于”、“不等于”等运算。

④ 数据传输:主要包括赋值、输入、输出等操作。

(2) 算法的控制结构:一个算法的功能不仅仅取决于所选用的操作,而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

算法的控制结构给出了算法的基本框架,它不仅决定了算法中各操作的执行顺序,而且也直接反映了算法的设计是否符合结构化原则。描述算法的工具通常有传统流程图、N-S结构化流程图、算法描述语言等。一个算法一般可以由顺序、选择、循环3种基本控制结构组合而成。

3. 算法设计的基本方法

计算机算法不同于人工处理的方法,下面是工程上常用的几种算法设计,在实际应用时,各种方法之间往往存在着一定的联系。

(1) 列举法

列举法是计算机算法中的一个基础算法。列举法的基本思想是,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。

列举法的特点是算法比较简单。但当列举的可能情况较多时,执行列举算法的工作量将会很大。因此,在用列举法设计算法时,使方案优化,尽量减少运算工作量,是应该重点注意的。

(2) 归纳法

归纳法的基本思想是,通过列举少量的特殊情况,经过分析,最后找出一般的关系。从本质上讲,归纳法就是通过观察一些简单而特殊的情况,最后总结出一般性的结论。

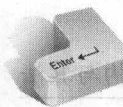
(3) 递推法

递推法是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。其中初始条件或是问题本身已经给定,或是通过对问题的分析与化简而确定。递推法本质上属于归纳法,工程上许多递推关系式实际上是通过分析实际问题的分析与归纳得到的,因此,递推关系式往往是归纳的结果。对于数值型的递推算法必须要注意数值计算的稳定性问题。

(4) 递归法

人们在解决一些复杂问题时,为了降低问题的复杂程度(如问题的规模等),一般总是将问题逐层分解,最后归结为一些最简单的问题。这种将问题逐层分解的过程,实际上并没有对问题进行求解,而只是当解决了最后那些最简单的问题后,再沿着原来分解的逆过程逐步进行综合,这就是递归法的基本思想。

递归法分为直接递归法与间接递归法两种。



(5) 减半递推法

实际问题的复杂程度往往与问题的规模有着密切的联系。因此,利用分治法解决这类实际问题是有效的。工程上常用的分治法是减半递推技术。

所谓“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”的过程。

(6) 回溯法

在工程上,有些实际问题很难归纳出一组简单的递推公式或直观的求解步骤,并且也不能进行无限的列举。对于这类问题,一种有效的方法是“试”。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,若试探成功,就得到问题的解;若试探失败,就逐步回退,换别的路线再逐步试探。

4. 算法设计的要求

通常一个好的算法应达到如下目标。

(1) 正确性

正确性(correctness)大体可以分为以下4个层次:

- ① 程序不含语法错误;
- ② 程序对于几组输入数据能够得出满足规格说明要求的结果;
- ③ 程序对于精心选择的典型、苛刻而带有刁难性的几组输入数据能够得出满足规格说明要求的结果;
- ④ 程序对于一切合法的输入数据都能产生满足规格说明要求的结果。

(2) 可读性

算法主要是为了方便人们的阅读与交流,其次才是执行。可读性(readability)好有助于用户对算法的理解;晦涩难懂的程序易于隐藏较多的错误,难以调试和修改。

(3) 健壮性

所谓健壮性(robustness)当输入数据非法时,算法也能适当地做出反应或进行处理,而不会产生莫名其妙的输出结果。

(4) 效率与低存储量需求

效率是指程序执行时,对于同一个问题如果有多个算法可以解决,则执行时间短的算法效率高;存储量需求是指算法执行过程中所需要的最大存储空间。

1.1.2 算法的复杂度

1. 算法的时间复杂度

算法的时间复杂度是指执行算法所需要的计算工作量。同一个算法用不同的语言实现,或者用不同的编译程序进行编译,或者在不同的计算机上运行,效率均不同。这表明使用绝对的时间单位衡量算法的效率是不合适的。撇开这些与计算机硬件、软件有关的因素,可以认为一个特定算法“运行工作量”的大小,只依赖于问题的规模(通常用整数 n 表示),它是问题的规模函数,即

算法的工作量 $=f(n)$

例如,在 $N \times N$ 矩阵相乘的算法中,整个算法的执行时间与该基本操作(乘法)重复执行的次数 n 的立方即 n^3 成正比,也就是时间复杂度为 n^3 ,即

$$f(n)=O(n^3)$$

在某些情况下,算法中的基本操作重复执行的次数还随问题的输入数据集不同而不同。例如在起泡排序的算法中,当要排序的数组 a 初始序列为自小至大有序时,基本操作的执行次数为 $n(n-1)/2$ 。对这类算法的分析,可以采用以下两种方法来分析。

(1) 平均性态

所谓平均性态(Average Behavior)是指用各种特定输入下的基本运算次数的加权平均值来度量算法的工作量。

设 x 是所有可能输入中的某个特定输入, $p(x)$ 是 x 出现的概率(即输入为 x 的概率), $t(x)$ 是算法在输入为 x 时所执行的基本运算次数,则算法的平均性态定义为

$$A(n) = \sum_{x \in D_n} p(x)t(x)$$

其中 D_n 表示当规模为 n 时,算法执行的所有可能输入的集合。

(2) 最坏情况复杂性

所谓最坏情况复杂性(Worst-case Complexity)分析是指在规模为 n 时,算法所执行的基本运算的最大次数。

2. 算法的空间复杂度

算法的空间复杂度是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的空间、输入的初始数据所占的存储空间,以及算法执行中所需要的额外空间。其中额外空间包括算法程序执行过程中的工作单元以及某种数据结构所需要的附加存储空间。如果额外空间量相对于问题规模来说是常数,则称该算法是原地(inplace)工作的。在许多实际问题中,为了减少算法所占的存储空间,通常采用压缩存储技术,以便尽量减少不必要的额外空间。

1.2 数据结构

1.2.1 数据结构的定义

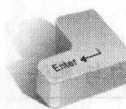
数据结构(Data Structure)是指相互之间存在一种或多种特定关系的数据元素的集合,即数据的组织形式。

数据结构作为计算机的一门学科,主要研究和讨论以下三个方面:

- (1) 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;
- (2) 在对数据元素进行处理时,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。

讨论以上问题的目的是为了提高数据处理的效率,所谓提高数据处理的效率有以下两个方面:

- (1) 提高数据处理的速率;
- (2) 尽量节省在数据处理过程中所占用的计算机存储空间。



在数据结构中,要了解以下3个概念。

(1) 数据(Data): 数据是对客观事物的符号表示,在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

(2) 数据元素(Data Element): 数据元素是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。

(3) 数据对象(Data Object): 数据对象是性质相同的数据元素的集合,它是数据的一个子集。

一般情况下,在具有相同特征的数据元素集合中,各个数据元素之间存在着某种关系(即连续),这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中,通常把数据元素之间这种固有的关系简单地用前后件关系(或直接前驱与直接后继关系)来描述。

前后件关系是数据元素之间的一个基本关系,但前后件关系所表示的实际意义随具体对象的不同而不同。一般来说,数据元素之间的任何关系都可以用前后件关系来描述。

1. 数据的逻辑结构

数据结构是指反映数据元素之间的关系的的数据元素集合的表示。更通俗地说,数据结构是指带有结构的数据元素的集合。所谓结构实际上就是指数据元素之间的前后件关系。

一个数据结构应包含以下两方面信息:

- (1) 数据元素的信息;
- (2) 各数据元素之间的前后件关系。

2. 数据的存储结构

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(也称为数据的物理结构)。

1.2.2 数据结构的图形表示

数据结构除了用二元关系表示外,还可以直观地用图形表示。

在数据结构的图形表示中,对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示,一般称之为数据结点,并简称为结点;为了进一步表示各数据元素之间的前后件关系,对于关系 R 中的每一个二元组,用一条有向线段从前件结点指向后件结点。

在数据结构中,没有前件的结点称为根结点;没有后件的结点称为终端结点(也称为叶子结点)。

一个数据结构中的结点可能是在动态变化的。根据需要或在处理过程中,可以在一个数据结构中增加一个新结点(称为插入运算),也可以删除数据结构中的某个结点(称为删除运算)。插入与删除是对数据结构的两种基本运算。除此之外,对数据结构的运算还有查找、分类、合并、分解、复制和修改等。

1.2.3 线性结构与非线性结构

如果在一个数据结构中没有任何数据元素,则称该数据结构为空的的数据结构。在一个数据结构中有一个或者多个数据元素,则称该数据结构为非空数据结构。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。

非空数据结构满足:

- (1) 有且只有一个根结点;