



教育部高等学校电子信息类专业教学指导委员会规划教材
高等学校电子信息类专业系列教材

适用于电类非计算机专业

Fundamentals of Computer Software

计算机软件基础

汪友生 张新峰 张小玲 郭民 刘芳 王众 编著

Wang Yousheng

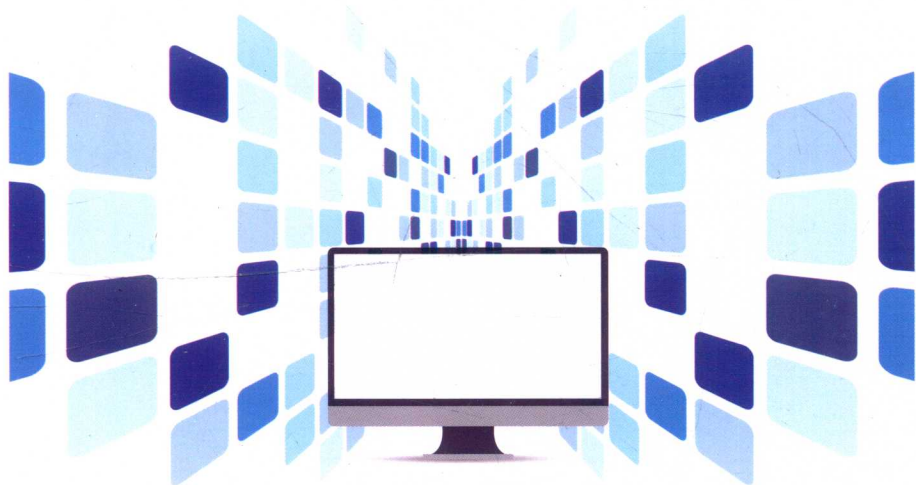
Zhang Xinfeng

Zhang Xiaoling

Guo Min

Liu Fang

Wang Zhong



清华大学出版社





教育部高等学校电子信息类专业教学指导委员会规划教材
高等学校电子信息类专业系列教材

Fundamentals of Computer Software

计算机软件基础

汪友生 张新峰 张小玲 郭民 刘芳 王众 编著

Wang Yousheng

Zhang Xinfeng

Zhang Xiaoling

Guo Min

Liu Fang

Wang Zhong

清华大学出版社
北京

内 容 简 介

本书是根据高等学校电子信息类专业对计算机软件技术课程的基本要求,结合作者多年来的教学改革和教学实践,来组织编写的高等学校计算机软件技术基础课程。

本书内容主要包括计算机软件技术绪论、线性数据结构、非线性数据结构、排序和查找、资源管理、软件开发和数据库设计。每章都配有较多的习题,书后附有部分习题答案。

本书内容丰富、语言简明扼要、实用性强,可作为高等院校本科、专科计算机软件技术基础课程教材,也可作为广大从事计算机应用工作的技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机软件基础/汪友生等编著. —北京:清华大学出版社,2016

(高等学校电子信息类专业系列教材)

ISBN 978-7-302-43660-7

I. ①计… II. ①汪… III. ①软件—高等学校—教材 IV. ①TP31

中国版本图书馆 CIP 数据核字(2016)第 084693 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:梁毅

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.75

字 数:490千字

版 次:2016年12月第1版

印 次:2016年12月第1次印刷

印 数:1~2000

定 价:45.00元

产品编号:061415-01

高等学校电子信息类专业系列教材

顾问委员会

谈振辉	北京交通大学 (教指委高级顾问)	郁道银	天津大学 (教指委高级顾问)
廖延彪	清华大学 (特约高级顾问)	胡广书	清华大学 (特约高级顾问)
华成英	清华大学 (国家级教学名师)	于洪珍	中国矿业大学 (国家级教学名师)
彭启琮	电子科技大学 (国家级教学名师)	孙肖子	西安电子科技大学 (国家级教学名师)
邹逢兴	国防科学技术大学 (国家级教学名师)	严国萍	华中科技大学 (国家级教学名师)

编审委员会

主任	吕志伟	哈尔滨工业大学		
副主任	刘旭	浙江大学	王志军	北京大学
	隆克平	北京科技大学	葛宝臻	天津大学
	秦石乔	国防科学技术大学	何伟明	哈尔滨工业大学
	刘向东	浙江大学		
委员	王志华	清华大学	宋梅	北京邮电大学
	韩焱	中北大学	张雪英	太原理工大学
	殷福亮	大连理工大学	赵晓晖	吉林大学
	张朝柱	哈尔滨工程大学	刘兴钊	上海交通大学
	洪伟	东南大学	陈鹤鸣	南京邮电大学
	杨明武	合肥工业大学	袁东风	山东大学
	王忠勇	郑州大学	程文青	华中科技大学
	曾云	湖南大学	李思敏	桂林电子科技大学
	陈前斌	重庆邮电大学	张怀武	电子科技大学
	谢泉	贵州大学	卞树檀	第二炮兵工程大学
	吴瑛	解放军信息工程大学	刘纯亮	西安交通大学
	金伟其	北京理工大学	毕卫红	燕山大学
	胡秀珍	内蒙古工业大学	付跃刚	长春理工大学
	贾宏志	上海理工大学	顾济华	苏州大学
	李振华	南京理工大学	韩正甫	中国科学技术大学
	李晖	福建师范大学	何兴道	南昌航空大学
	何平安	武汉大学	张新亮	华中科技大学
	郭永彩	重庆大学	曹益平	四川大学
	刘缠牢	西安工业大学	李儒新	中科院上海光学精密机械研究所
	赵尚弘	空军工程大学	董友梅	京东方科技集团
	蒋晓瑜	装甲兵工程学院	蔡毅	中国兵器科学研究院
	仲顺安	北京理工大学	冯其波	北京交通大学
	黄翊东	清华大学	张有光	北京航空航天大学
	李勇朝	西安电子科技大学	江毅	北京理工大学
	章毓晋	清华大学	张伟刚	南开大学
	刘铁根	天津大学	宋峰	南开大学
	王艳芬	中国矿业大学	靳伟	香港理工大学
	苑立波	哈尔滨工程大学		
丛书责任编辑	盛东亮	清华大学出版社		

前言

PREFACE

随着计算机应用领域的扩大和深入,非计算机专业的工程技术人员掌握必要的计算机技术基础知识是提高计算机应用水平、利用计算机技术解决本专业工作中的具体问题的重要途径。非计算机类专业本科生既熟悉自己所从事的专业,又掌握计算机的应用知识是一个优势。事实上,许多应用软件都是由非计算机专业出身的计算机应用人员研发的。

计算机软件基础是高等学校电子信息类专业的一门学科基础必修课,被一些高校列为电子类各专业的重点课程或核心课程。通过多年来对本课程的教学研究和教学改革,我们在教学内容、教学方法和考核方式上已基本形成一套比较完整的体系,以切实提高学生的程序设计能力。好的教材源于教学改革和教学实践,能体现出良好的成果。在多年教学经验的基础上,通过对已有教材的分析研究,结合自己的教研工作,编写此教材。本教材的特点是强调实用性,以应用为目的,含有丰富的实例;可读性强,深入浅出,通俗易懂,力求概念准确,表述清楚,简明扼要;所有算法采用C语言描述;适合非计算机专业的学生阅读。

全书内容以数据结构为主,同时包含操作系统、软件工程和数据库三部分内容。教学时可根据具体情况对讲授内容进行适当取舍。

本书由汪友生编写线性表、栈和队列、串和数组、树等部分;张新峰编写绪论、排序和软件开发等部分;王众编写查找部分;张小玲编写资源管理部分;刘芳编写数据库设计部分;郭民编写图部分。全书由汪友生统稿。

本书在编写时,参考了大量的文献资料,对相关作者表示真诚的感谢!由于编者水平有限,书中难免存在疏漏和不妥之处,恳请各位读者批评指正。

编者

2016年6月

目录

CONTENTS

第 1 章 绪论	1
1.1 计算机软件	2
1.1.1 计算机软件的概念	2
1.1.2 计算机语言	3
1.1.3 计算机软件分类	4
1.1.4 计算机软件的发展	4
1.2 数据结构概述	5
1.2.1 数据基本概念	5
1.2.2 数据结构	6
1.2.3 数据类型	7
1.3 算法及算法分析	8
1.3.1 算法	8
1.3.2 算法的性能分析	9
1.4 小结	11
1.5 习题	11
第 2 章 线性数据结构	14
2.1 线性表的定义	14
2.2 线性表的顺序存储及其运算	16
2.2.1 顺序表	16
2.2.2 顺序表的基本运算	17
2.2.3 插入和删除的时间复杂度	20
2.2.4 线性表顺序存储结构的优缺点	20
2.3 线性表的链式存储及其运算	20
2.3.1 单链表	21
2.3.2 单循环链表	26
2.3.3 双向链表	30
2.4 线性表的应用	34
2.4.1 有序表	34
2.4.2 多项式的表示与运算	36
2.5 栈	39
2.5.1 栈的基本概念	39
2.5.2 栈的运算	40
2.5.3 栈的应用	45

2.6	队列	47
2.6.1	队列的基本概念	47
2.6.2	顺序(循环)队列及其运算	48
2.6.3	链式队列及其运算	51
2.6.4	队列的应用	54
2.7	串	58
2.7.1	串的定义	58
2.7.2	串的运算	59
2.7.3	串的存储方式	60
2.7.4	串的模式匹配	66
2.8	数组	69
2.8.1	数组的定义	69
2.8.2	数组的顺序存储	70
2.8.3	矩阵的压缩存储	73
2.9	小结	80
2.10	习题	81
第3章	非线性数据结构	89
3.1	树的概念	89
3.2	二叉树	91
3.2.1	二叉树的定义	92
3.2.2	二叉树的主要性质	92
3.2.3	二叉树的存储结构	94
3.3	二叉树的遍历	99
3.3.1	遍历的概念	99
3.3.2	二叉树遍历算法	99
3.3.3	二叉树遍历算法的应用	104
3.4	树和森林	106
3.4.1	树和森林的存储结构	106
3.4.2	树和森林与二叉树之间的转换	110
3.4.3	树和森林的遍历	112
3.5	二叉树的应用	113
3.5.1	哈夫曼树及其应用	113
3.5.2	二叉排序树	117
3.6	图	119
3.6.1	图的基本概念	119
3.6.2	图的存储方法	121
3.6.3	图的遍历	125
3.6.4	图的应用	131
3.7	小结	147
3.8	习题	147
第4章	排序和查找	155
4.1	排序的基本概念	155
4.2	插入排序	157

4.2.1	直接插入排序	157
4.2.2	折半插入排序	158
4.2.3	希尔排序	159
4.3	交换排序	160
4.3.1	冒泡排序	160
4.3.2	快速排序	162
4.4	选择排序	164
4.4.1	简单选择排序	165
4.4.2	堆排序	166
4.5	其他排序	171
4.5.1	归并排序	171
4.5.2	基数排序	173
4.6	各种排序方法的比较和选择	175
4.7	查找的基本概念	176
4.8	静态查找表与算法	177
4.8.1	顺序查找	177
4.8.2	折半查找	178
4.8.3	分块查找	180
4.9	动态查找表	181
4.9.1	二叉搜索树	181
4.9.2	平衡二叉树	183
4.10	哈希表及其查找	186
4.10.1	哈希表的概念	186
4.10.2	几种哈希函数	186
4.10.3	处理冲突的方法	188
4.10.4	哈希表的算法	189
4.10.5	哈希表的应用	192
4.11	小结	192
4.12	习题	193
第5章	资源管理	198
5.1	操作系统的概念	198
5.1.1	操作系统的定义	198
5.1.2	操作系统的分类	200
5.1.3	操作系统的特征	202
5.1.4	操作系统的功能	202
5.2	多道程序设计	203
5.2.1	并发程序设计	203
5.2.2	进程	205
5.2.3	进程之间的通信	206
5.2.4	多道程序的组织	208
5.3	存储空间的管理	210
5.3.1	内存管理器的管理	210

5.3.2	外存储器中文件的组织结构	218
5.4	小结	221
5.5	习题	221
第6章	软件开发	224
6.1	软件工程概述	224
6.1.1	软件工程的	225
6.1.2	软件生命周期	226
6.2	软件的需求分析	227
6.2.1	需求分析概述	227
6.2.2	结构化分析方法	228
6.2.3	数据流图	228
6.2.4	数据字典	231
6.3	软件的设计	233
6.3.1	软件设计概述	233
6.3.2	结构化设计方法	236
6.3.3	详细设计方法	239
6.3.4	面向对象的程序设计方法	241
6.4	软件的编程	244
6.5	软件的测试	246
6.5.1	软件测试概述	246
6.5.2	软件测试的过程	247
6.5.3	测试用例的设计	248
6.6	软件的调试	251
6.6.1	软件调试的方法	251
6.6.2	常用的调试策略	252
6.7	软件维护	253
6.8	小结	254
6.9	习题	254
第7章	数据库设计	257
7.1	数据库基本概念	257
7.1.1	数据库技术与数据库系统	257
7.1.2	数据模型	261
7.1.3	数据库系统的结构	264
7.2	关系数据库语言 SQL	266
7.2.1	SQL 语言概述	266
7.2.2	数据定义功能	268
7.2.3	数据查询功能	270
7.2.4	数据更新功能	274
7.3	数据库设计	276
7.3.1	数据库设计概述	276
7.3.2	需求分析	280
7.3.3	概念设计	282
7.3.4	逻辑设计	290

7.3.5 物理设计	292
7.3.6 数据库的实施	295
7.3.7 数据库的运行和维护	296
7.4 小结	297
7.5 习题	297
附录 部分习题参考答案	299
参考文献	304

计算机的出现在人类历史上具有划时代的意义。当前社会已经进入信息时代,这个时代的显著特征是计算机已应用到了各行各业,已进入千家万户。目前,没有计算机的生活是难以想象的,运用计算机成为人们的一项基本技能。计算机的出现为信息处理技术开辟了新纪元,因此电子信息类专业更离不开计算机,计算机软件基础的内容是电子信息类及其相关专业学生所必备的基本知识。这门课程主要包括数据库、数据结构、操作系统和软件工程 4 个部分。学好这门课程,将为电子类本科生相关专业课程的学习和实践奠定良好的基础。

完整的计算机系统包括软件系统和硬件系统两部分,简称硬件和软件。其中,硬件是指组成计算机的各种物理设备的总称,包括各种电子元件和电子线路。所谓软件是指计算机上运行的各种程序、维护这些程序所需的文档及运行这些程序所需的数据的总称。可以简单地说,软件=程序+文档+数据。

迄今为止,各种计算机的硬件结构基本相同,都是冯·诺依曼(John Von-Neumann)体系结构。20 世纪 40 年代中期,美国科学家冯·诺依曼提出了计算机组成结构、程序存储和程序控制等思想。所谓“程序存储”是指人们将解题步骤编写成程序,然后将程序存放在计算机的存储器中。所谓“程序控制”是指计算机的控制器根据程序控制整机执行,从而完成相应的任务。存储程序和程序设计是冯·诺依曼体系结构计算机设计的主要思想。根据这一设计思想,计算机的硬件系统由运算器、控制器、存储器、输入设备和输出设备 5 大功能部件组成。图 1-1 显示了计算机硬件系统的基本组成及各功能部件关系,其中粗实线表示数据传输线路,细实线表示控制信号传输线路。

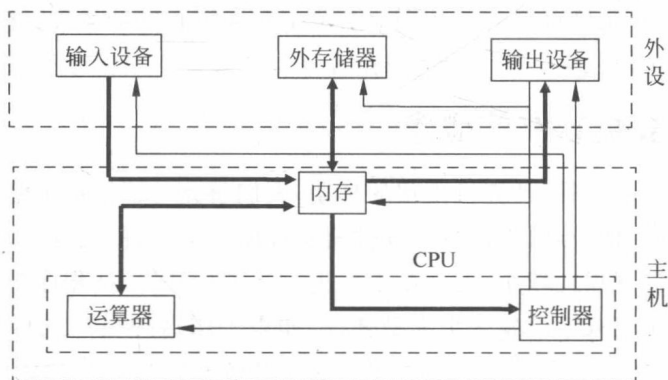


图 1-1 计算机硬件系统的基本组成

1. 控制器

顾名思义,控制器是计算机的控制部件,主要包括程序计数器、指令存储器、指令译码器、时序电路及操作控制器等。控制器从存储器读取指令,经过分析译码,产生操作命令发向各个部件,控制这些部件完成指令所需要的操作,从而保证计算机能够连续、自动、有序地工作。

2. 运算器

运算器是对数据进行运算的部件,可以对数据进行算术运算和逻辑运算。当需要进行某种运算时,通过控制器发出指令,将存放在存储器中的数据送到运算器,然后由控制器发出运算的指令,运算结果送回存储器保存,以便下一次使用或输出。运算器与控制器合称为中央处理机(Central Processing Unit,CPU)。

3. 存储器

存储器用来存放程序和数据,程序是计算机操作的依据,数据是其操作对象。存储器又可分为内存储器和外存储器。内存储器直接与中央处理器(CPU)相连,可由CPU直接读写信息,是CPU根据地址直接寻址的存储空间。一般用来存放正在执行的程序或者正在处理的数据。内存的速度直接影响到计算机的性能。外存储器不能与CPU直接交换信息,存放在外存的程序必须调入内存后才能运行,一般用来存放暂时不用,但又需要长期保存的数据或程序。CPU和内存储器通常称为主机。

4. 输入/输出设备

输入/输出设备简称为I/O设备,是用来输入、输出程序和数据的部件。例如,键盘、鼠标、扫描仪等是常用的输入设备;打印机、显示器、绘图仪是常用的输出设备。输入、输出设备和外存储器统称为外设,外设通过接口电路与主机相连。

1.1 计算机软件

习惯上,人们往往认为软件就是程序。其实,软件是一个随时代而发展的概念。时代不同,软件的内涵也不相同。在计算机出现的早期,可以说软件即是程序。随着时代的发展,计算机处理的问题多样化,程序的规模越来越大,这样在程序的设计、维护、升级等方面产生了一系列新的问题,针对这些问题对软件的概念进行了更新。目前看来,程序只是软件的一个重要组成部分,但不是软件概念的全部。本节主要介绍软件的概念,软件分类及其发展历程。

1.1.1 计算机软件的概念

在20世纪50~60年代,计算机出现的早期,人们研究计算机把主要精力放在提高硬件的可靠性、稳定性等方面,当时人们认为软件就是程序。例如,汇编程序、解释程序及各种管理程序等。那时软件规模比较小,完全基于手工方式进行开发,从程序的设计、编程到调试,一般都是由一个人来完成。开发一个大型软件,如果采用这种模式进行设计,不但效率低,开发周期长,同时模块间的联系和接口很难协调,出错率极高,因而软件维护的成本很高。20世纪60年代出现了软件危机,人们认识到要有一定的规范文档来保证软件设计、调试、

运行的成功。从20世纪70年代开始,人们认为计算机软件不仅仅是程序,还应该包括开发、使用、维护这些程序的一切文档,和运行该程序所需要的数据。这一观点强调了文档在软件中的重要性。1983年,IEEE组织明确定义了软件:软件是计算机程序、方法和规则相关的文档,以及在计算机上运行它所必需的数据。

1.1.2 计算机语言

人们要畅通无阻地与计算机打交道,必须解决“语言”沟通的问题。根据计算机语言的发展阶段,可以将其分为4代,分别为机器语言、汇编语言、高级语言和非过程化语言。

1. 机器语言

计算机并不能理解人们的自然语言,只能接受二进制指令。计算机能够识别和执行的这种指令,称为机器指令。不同类型的计算机规定了不同的若干指令,这些指令的集合就是机器语言。机器语言是由一系列的0和1组成的指令,这些指令难学、难懂、难记、难以修改,使用起来很不方便。此外,这种语言和机器类型相关,在一种类型的机器上能够运行的程序,在另一类型的机器上一般不能运行,这给使用计算机带来很大障碍。

2. 汇编语言

由于机器语言具有难学、难懂、难记忆等特点,人们自然会想到用一些助记符来代替二进制指令。根据这一想法,人们通常用具有指令功能的英文单词缩写来代替二进制指令,例如加法指令用助记符ADD表示,这就是汇编语言。显然,同机器语言相比,汇编语言易于理解和记忆。计算机并不能识别和直接运行汇编语言程序,必须经过一种翻译程序将汇编语言翻译为机器语言,之后计算机才能识别和运行,这种翻译程序称为汇编程序。由此可知,汇编语言就是在机器语言的基础上采用了助记符。由于机器语言和机器类型有关,因此,汇编语言也与机器类型有关。因此,机器语言和汇编语言统称为面向机器的语言,也称为低级语言。

3. 高级语言

低级语言和计算机的类型有关,仅适合于熟悉计算机硬件的人员使用,对计算机的普及使用是一大障碍。为了克服这一缺点,出现了高级语言。高级语言是一种人工设计的语言,它接近于数学语言或者人类的自然语言,具有易于理解和掌握的特点,同时它与具体的机器类型无关。用高级语言编写的程序在任一种计算机上都可以运行,这为非计算机专业人员使用计算机编程提供了很大的方便。这种语言又称为面向过程的语言,只要根据求解问题的算法,写出处理的过程即可。高级语言的通用性和可移植性好。运用高级语言编写的程序显然不能直接在计算机上运行,必须翻译为目标程序(机器语言程序)才能运行。将完成翻译功能的程序称为编译程序。

4. 非过程化语言

采用过程化语言编写程序时,人们须知算法的具体执行步骤,即必须写出算法执行的每一步,才能解决问题。人们希望只要指出做什么,而不必知道如何做,由计算机自己去解决如何做的问题,这就是非过程化语言。非过程化语言广泛应用于关系型数据库中。

1.1.3 计算机软件的分类

计算机软件种类繁多,作用各异,从不同的角度可以有多种分类方法。本节介绍两种较为常用的分类方法。

(1) 按软件与计算机硬件、用户的关系划分,软件一般可以分为系统软件、支撑软件和应用软件。

① 系统软件:是指为整个计算机系统配置的、不依赖具体应用的软件。它与计算机硬件紧密结合,以便计算机的各个部分协调、高效地工作。例如,操作系统、与计算机相连接的各种设备驱动程序、语言处理程序、数据库管理系统等都属于系统软件。

② 支撑软件:用于协助用户开发与维护软件系统的一些工具性软件。例如,C程序编译器、错误检测程序等。

③ 应用软件:为解决各种应用领域中的问题而开发的软件。例如,教学管理软件、工程制图软件、图像处理软件等。按照主要用途可以分为科学计算类、数据处理类、过程控制类、辅助设计类、人工智能软件类等。

(2) 按软件规模划分,可以分为微型软件、小型软件、中型软件和大型软件。

① 微型软件:一般认为源程序在 500 行语句以内的软件系统。

② 小型软件:一般认为源程序在 2000 行语句以内的软件系统。

③ 中型软件:一般认为源程序在 5 万行语句以内的软件系统。

④ 大型软件:一般认为源程序在 5 万行语句以上的软件系统。

1.1.4 计算机软件的发展

计算机软件作为一种新生事物,像其他事物一样,也具有一定的发展历史。计算机软件的发展也经历了一个从小到大、从简单到复杂的历程。总的来说,计算机软件的发展大致可以分为以下三个阶段:

1. 第一阶段程序设计时代(20世纪40年代到50年代)

电子计算机出现的早期,计算机中不安装任何软件。当时电子管是计算机的主要元器件,其价格昂贵,稳定性、可靠性比较差,人们研究计算机把精力主要放在硬件性能的改进和技术指标的提高上,软件处于次要地位。计算机应用主要针对科学计算,例如求解复杂的方程、大型的矩阵求逆计算,等等。这一阶段程序设计者的编程语言,主要采用机器语言和汇编语言,程序规模小,编程方式主要是封闭式个体手工开发。

此时,计算机内存容量小、运行速度慢、运行可靠性差。尽管程序要完成的任务在现在看来可能非常简单,由于受硬件条件的限制,程序设计者需要通过一些编程技巧来提高程序运行效率。在这一阶段,程序设计者往往也是程序使用者,程序还未形成为产品,程序大多是为某个具体应用而编写的,功能单一,仅限在专门的计算机上执行,可移植性较差。

2. 第二阶段软件时代(20世纪60年代到20世纪70年代)

这一阶段,计算机硬件技术获得了很大的发展,计算机的主要元器件是晶体管和集成电路,其体积缩小、稳定性提高、价格降低。软件的应用领域不再局限于科学计算,而是拓宽至商业、办公等多个领域。社会对软件的需求迅速增加,软件的地位和作用不断提升。这一阶段的主要特点是:开发工具为高级程序设计语言,产生了结构化编程的思想和方法。随着

软件规模的增大,软件开发中遇到了一系列的问题,例如软件开发进度难以预测、成本难以控制等,这就产生了“软件危机”。人们认识到正像不能用盖小茅屋的方法盖高楼大厦一样,不能用作坊式的生产方式来生产软件产品。应借助于现代工程的概念和原理,沿用工业化较成熟的管理经验和工程的技术与方法进行计算机程序的开发及文档资料的编写。其目的是要提高软件开发的生产率和软件产品的质量,这就是软件工程的思想。这一时期,人们提出用软件工程的方法来解决软件危机的问题。

3. 第三阶段软件工程时代(20世纪70年代至今)

这一阶段,硬件方面出现了大规模集成电路和超大规模集成电路,计算机的运算速度、数据处理能力进一步提高。尤其是微处理器的诞生,开创了大众化使用计算机的新时代。软件开发方面,结构化程序设计方法日益成熟,并得到了广泛应用。自20世纪90年代起,面向对象程序设计方法为人们提供了新的软件设计思路,很快应用至软件开发中。目前,以软件为主要特征的智能产品不断涌现,尤其是网络通信技术、数据库技术与多媒体技术的结合,彻底改变了软件系统的体系结构。使得计算机的潜能得到了更大程度的发挥。

软件工程自产生以来,人们就寄希望于通过它去解决软件危机。但至今为止,软件危机还没有得到彻底解决。在软件工程实践中,一些老问题解决了,但可能又产生许多新问题,于是人们不得不去研究新的解决方法。然而,正是由于这些问题的不断出现,才推动着软件工程学科的发展。

1.2 数据结构概述

早期的计算机主要用于解决大型的科学计算问题,涉及的数据类型比较简单,例如整型、浮点型数据等。随着计算机的发展,人们发现计算机不仅能够解决数值计算问题,而且能够解决大量的非数值计算问题。例如,利用计算机进行北京工业大学学生的学籍管理、城市交通调度管理、图书馆书目的管理等。像这样的问题,数据类型不再是数值型。此外,在其他一些问题,例如智能交通中,将涉及地图、图像、文本数据等,那么如何用计算机来解决有关非数值类型的问题呢?这里首先涉及数据结构。在介绍数据结构之前,先介绍3个基本概念:数据、数据结构和数据类型。

1.2.1 数据基本概念

数据是对客观事物的符号表示,计算机科学中指所有能输入到计算机并能够被计算机程序处理的所有符号的总称。数据是计算机加工的“原材料”。数据的概念随着计算机的发展而不断扩展。早期计算机主要解决数值计算问题,当时的数据指的就是数值。随着计算机应用范围的扩大,文字、表格、声音、图像都可以作为计算机处理的数据。例如,对表1-1的学生学籍管理表进行学生信息的插入或删除操作时,可以将每一个学生的个人信息作为一个基本的数据单位,类似这样的基本单位就称为一个数据元素。同时可以看出,每个数据元素的内容都包括学号、姓名、性别、籍贯、民族、年龄共6项,其中每一项都称为一个数据项。数据元素是数据的基本单位。数据对象是指性质相同的数据元素的集合,是数据的一个子集。表1-1可以看作一个数据对象。

表 1-1 学生学籍管理表

学号	姓名	性别	籍贯	民族	年龄
14024101	张三	男	河北	回	18
14024102	李四	男	上海	汉	18
14024103	王五	女	新疆	维吾尔	19
14024104	马六	女	北京	汉	17
...

1.2.2 数据结构

实际问题中,数据元素和数据元素之间不是完全孤立的,而是彼此之间存在一定的关系。如表 1-1 所示,学生学籍表中,数据元素之间依次相邻(从位置上看,数据元素之间是一一对一的关系),数据元素之间这种关系就是线性关系。因此,表 1-1 就属于一种线性数据结构。所谓数据结构,指的是数据元素之间的相互关系,即数据的组织形式。一般认为,数据结构包括以下 3 个方面的内容。

(1) 数据的逻辑结构:指的是数据元素之间的逻辑关系。它与数据的存储方式没有关系,是独立于计算机的。根据数据元素之间的不同特性,通常有以下 4 种基本的逻辑结构(参见图 1-2)。

- ① 集合:数据结构中的数据元素具有同一属性,无其他关系。这种结构较少应用。
- ② 线性结构:数据元素之间是一一对一的关系。
- ③ 树形结构:数据元素之间存在一对多的关系。
- ④ 图状结构或网状结构:数据元素之间存在多对多的关系。

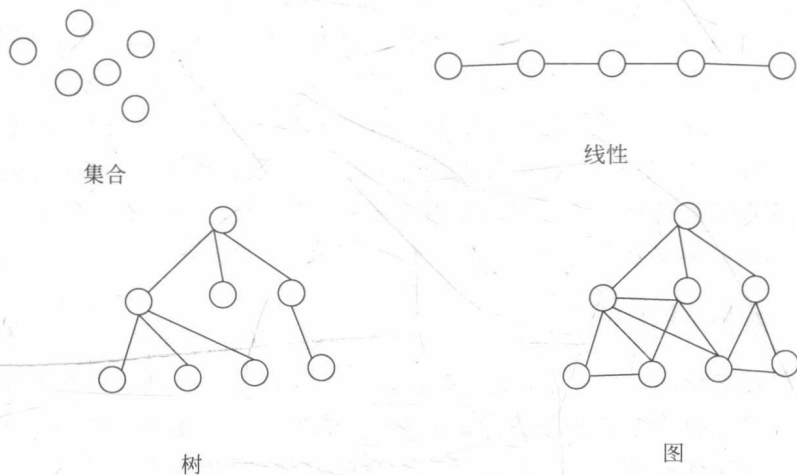


图 1-2 4 种基本的结构关系

(2) 数据的存储结构:讨论数据结构的目的是在计算机上实现对数据的操作,如何通过计算机表示出数据之间的逻辑关系是一个非常重要的问题。数据之间的逻辑关系通过在计算机中存储方式体现出来。数据在计算机中存储的方式就是数据的存储结构。计算机的内存器是由许多存储单元组成,每个存储单元都有一个唯一的地址。数据的存储结构就

是讨论数据在计算机中的存储映像方法。数据在内存中通常有以下4种存储结构。

① 顺序存储结构：主要用于存储线性结构。即把数据元素按照某种顺序存放在一块地址连续的内存单元中。其特点是逻辑上相邻的数据元素在物理存储位置上也相邻，数据元素之间的关系可以通过存储单元的地址来体现。通常借助程序设计语言中的数组来实现。

② 链式存储结构：借助指向数据元素存储地址的指针来表示数据元素之间的逻辑关系，一般将逻辑上相邻的数据元素存放在物理位置上不相邻的存储单元中。即可以用任意的一组存储单元来存放数据元素。这些存储单元的地址可以连续，也可以不连续。通常借助程序设计语言中的指针类型来描述这种存储方式。

③ 索引存储结构：在存储数据元素的同时，建立一个索引表。索引表中的每一项称为索引项，索引项一般包括关键字和地址两部分。关键字指能够唯一标识一个数据元素的数据项，地址表示数据元素的存储地址。

④ 散列存储结构：指在数据元素的关键字与其存储地址之间建立一个映射关系 F 。根据关系 F ，已知某数据元素的关键字 K 就可以得到它的存储地址。令 D 表示数据元素 E 的存储地址，则 $D=F(K)$ ，即数据元素的存储地址是其关键字的函数。所以这种存储结构中函数 F 的设计非常关键。

其中，在实现线性表的存储中，以顺序存储结构和链式存储结构最为常用。

(3) 数据的运算：定义在数据逻辑结构上的操作。每一种结构都有其运算的集合。例如，线性表常用的运算有添加、修改、删除等。数据运算的具体实现依赖于其存储结构。

数据结构的正式化定义为数据结构是一个二元组 $D_S = (D, S)$ 。其中， D 指数据元素的有限集， S 是 D 上关系的有限集。

例如，“复数”这种数据结构可以用如下形式化定义表示。

$$\text{Complex} = (C, R)$$

其中， C 包括两个实数的集合 $\{C_1, C_2\}$ ， R 是定义在集合 C 上的一种关系 $\langle C_1, C_2 \rangle$ ，序偶 $\langle C_1, C_2 \rangle$ 表示 C_1 是复数的实部， C_2 是复数的虚部。

1.2.3 数据类型

高级程序设计语言中，经常遇到数据类型的概念。数据类型指一个值的集合和定义在这个值集合上的一组操作的总称。实际上，数据类型可以看作是数据结构的一个特例。例如，C语言中的整型变量，其值集为某个有限长区间上的整数，定义在其上的操作有加、减、乘、除等算术运算。

按“值”的不同特性，可以将高级程序设计语言中的数据类型分为两大类。一类是非结构的原子类型，这种类型的数据不可分解。例如，C语言中的基本类型（如整型、实型、字符型等）。另一类是结构类型，其值由若干成分按照某种结构组成，因此可以分解。它的成分可以是结构的，也可以是非结构的。C语言中的结构体就是其中一个典型实例。

实际上，计算机中数据类型的概念并非仅限于高级语言。计算机硬件系统、操作系统、高级语言、数据库中都有相应的数据类型。例如，计算机硬件系统通常含有“位”、“字节”、“字”等原子类型数据，它们的操作通过计算机设计的一套指令系统直接由电路系统完成。而高级语言提供的数据类型，其操作通过编译器或解释器转化为汇编语言或机器语言的数